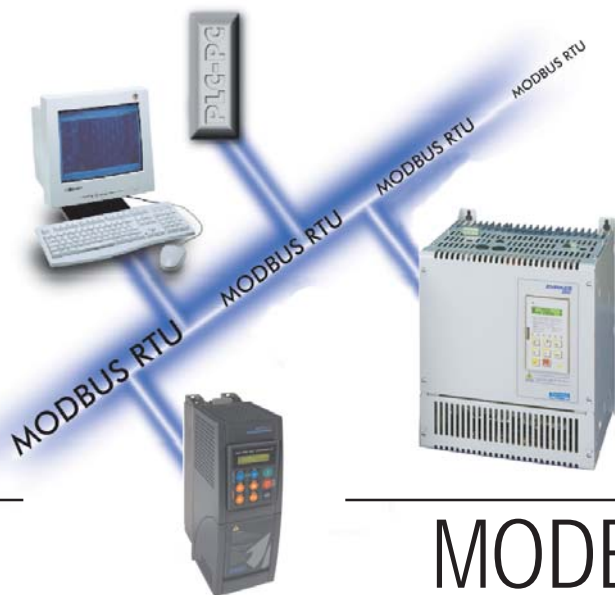


Modbus



MODBUS RTU

Protocol

■ ■ ■ ■ Manuale d'istruzione

Instruction Manual

Manuel d'instruction

Handbuch

Manual de Instruccion



Vi ringraziamo per avere scelto questo prodotto Gefran-Siei.

Saremo lieti di ricevere all'indirizzo e-mail: techdoc@siei.it qualsiasi informazione che possa aiutarci a migliorare questo manuale. Durante il suo periodo di funzionamento conservate il manuale in un luogo sicuro e a disposizione del personale tecnico.

Gefran spa si riserva la facoltà di apportare modifiche e varianti a prodotti, dati, dimensioni, in qualsiasi momento senza obbligo di preavviso.

I dati indicati servono unicamente alla descrizione del prodotto e non devono essere intesi come proprietà assicurate nel senso legale.

Tutti i diritti riservati.

Thank you for choosing this Gefran-Siei product.

We will be glad to receive any possible information which could help us improving this manual. The e-mail address is the following: techdoc@siei.it.

Keep the manual in a safe place and available to engineering and installation personnel during the product functioning period.

Gefran spa has the right to modify products, data and dimensions without notice.

The data can only be used for the product description and they can not be understood as legally stated properties.

All rights reserved

Nous vous remercions pour avoir choisi un produit Gefran-Siei.

Nous serons heureux de recevoir à l'adresse e-mail techdoc@siei.it toute information qui pourrait nous aider à améliorer ce catalogue.

Pendant sa période de fonctionnement conserver la notice dans un endroit sûr et à disposition du personnel technique.

Gefran spa se réserve le droit d'apporter des modifications et des variations aux produits, données et dimensions, à tout moment et sans préavis.

Les informations fournies servent uniquement à la description des produits et ne peuvent en aucun cas revêtir un aspect contractuel. Tous droits réservés.

Danke, dass Sie sich für dieses Gefran-Siei-Produkt entschieden haben.

Wir freuen uns über alle Anregungen an unsere E-Mail Adresse techdoc@siei.it, die uns bei der Verbesserung dieses Handbuchs nützlich sein können.

Bitte bewahren Sie das Handbuch während der gesamten Lebensdauer des Produkts an einem sicheren Ort auf, wo es dem technischen Personal stets zur Verfügung steht.

Gefran spa behält sich das Recht vor, jederzeit und ohne Verpflichtung zur Vorankündigung Änderungen und Abwandlungen von Produkten, Daten und Abmessungen vorzunehmen.

Die angeführten Daten dienen lediglich der Produktbeschreibung und dürfen nicht als versichertes Eigentum im rechtlichen Sinn verstanden werden.

Alle Rechte vorbehalten.

Le agradecemos la compra de este producto Gefran-Siei.

Estaremos encantados de recibirlos en la dirección de e-mail techdoc@sieispa.it para cualquier información que pueda contribuir a mejorar este manual.

Gefran spa se reserva el derecho de realizar modificaciones y variaciones sobre los productos, datos o medidas, en cualquier momento y sin previo aviso.

Los datos indicados están destinados únicamente a la descripción de los productos y no deben ser contemplados como propiedad asegurada en el sentido legal.

Todos los derechos reservados.

Table of Contents

ITALIANO	7
Introduzione	7
1. Il protocollo MODBUS	7
2. Formato dei messaggi	8
2.1. L'indirizzo	8
2.2. Il codice funzione	8
2.3. Il CRC16	9
2.4. Sincronizzazione dei messaggi	10
2.5. Impostazione linea seriale	10
3. Le funzioni MODBUS per Drive	11
3.1. Read Output Status (01)	11
3.2. Read Input Status (02)	12
3.3. Read Output Registers (03)	12
3.4. Read Input Registers (04)	13
3.5. Force Single Coil (05)	13
3.6. Preset Single Register (06)	13
3.7. Read Status (07)	14
3.8. Force Multiple Coils (15)	15
3.9. Preset Multiple Registers (16)	16
4. La gestione degli errori	17
4.1. Codici d'eccezione	17
5. Configurazione del sistema	18
ENGLISH	19
Introduction	19
1. The MODBUS Protocol	19
2. Message format	20
2.1. The address	20
2.2. The function code	20
2.3. CRC16	21
2.4. Message synchronization	22
2.5. Serial line setting	22
3. MODBUS functions for the drive	23
3.1. Read Output Status (01)	23
3.2. Read Input Status (02)	24

3.3. Read Output Registers (03)	24
3.4. Read Input Registers (04)	25
3.5. Force Single Coil (05)	25
3.6. Preset Single Register (06)	26
3.7. Read Status (07)	26
3.8. Force Multiple Coils (15)	27
3.9. Preset Multiple Registers (16)	28
4. Error management	29
4.1. Exception codes	29
5. System configuration	30

FRANÇAISE 31

Introduction	31
1. Le protocole MODBUS	31
2. Format des messages	32
2.1. L'adresse	32
2.2. Le code fonction	32
2.3. Il CRC16	33
2.4. Synchronisation des messages	34
2.5. Programmation ligne série	34
3. Les fonctions MODBUS pour Drive	35
3.1. Read Output Status (01)	35
3.2. Read Input Status (02)	36
3.3. Read Output Registers (03)	36
3.4. Read Input Registers (04)	37
3.5. Force Single Coil (05)	37
3.6. Preset Single Register (06)	38
3.7. Read Status (07)	38
3.8. Force Multiple Coils (15)	39
3.9. Preset Multiple Registers (16)	40
4. La gestion des erreurs	41
4.1. Codes d'exception	41
5. Configuration du système	42

DEUTSCHE	43
Einleitung	43
1. Das MODBUS Protokoll	43
2. Nachrichtenformat	44
2.1. Adresse	44
2.2. Funktionscode	44
2.3. CRC16	45
2.4. Nachrichtensynchronisation	46
2.5. Einstellung serielle Leitung	46
3. MODBUS-Funktionen für Drives	47
3.1. Read Output Status (01)	47
3.2. Read Input Status (02)	48
3.3. Read Output Registers (03)	48
3.4. Read Input Registers (04)	49
3.5. Force Single Coil (05)	49
3.6. Preset Single Register (06)	50
3.7. Read Status (07)	50
3.8. Force Multiple Coils (15)	51
3.9. Preset Multiple Registers (16)	52
4. Fehlerverwaltung	53
4.1. Ausnahmecodes	53
5. Systemkonfiguration	54
 ESPAÑOL	 55
Introducción	55
1. El protocolo MODBUS	55
2. Formato de los mensajes	56
2.1. La dirección	56
2.2. El código de funciones	56
2.3. El CRC16	57
2.4. Sincronización de los mensajes	58
2.5. Configuración de la línea serie	58
3. Las funciones MODBUS para drive	59
3.1. Read Output Status (01)	59
3.2. Read Input Status (02)	60
3.3. Read Output Registers (03)	60

3.4. Read Input Registers (04)	61
3.5. Force Single Coil (05)	61
3.6. Preset Single Register (06)	62
3.7. Read Status (07)	62
3.8. Force Multiple Coils (15)	63
3.9. Preset Multiple Registers (16)	64
4. La gestión de los errores	65
4.1. Códigos de excepción	65
5. Configuración del sistema	66

Appendix - REGISTER AND COIL MODBUS TABLES	67
AVy : Register-Parameter Table (Functions 03, 04, 06, 16).	68
AVy : Register list according to a Parameter progressive order	76
AVy : Coil Table (Functions 01, 02, 05, 15)	84
AVy : Coil list according to a Parameter progressive order	86
TPD32 : Register-Parameter Table (Functions 03, 04, 06, 16).	88
TPD32 : Register list according to a Parameter progressive order	96
TPD32 : Coil Table (Functions 01, 02, 05, 15)	104
TPD32 : Coil list according to a Parameter progressive order	107

ITALIANO

Introduzione

I parametri Drive vengono riferiti nel capitolo come registri Modbus di 16 bit; un parametro Drive di 32 bit occupa quindi 2 registri Modbus.

I parametri Drive che hanno valore digitale 0,1 come comandi, stati, ecc., vengono riferiti come Coils Modbus; un Coil corrisponde quindi ad un parametro discreto del Drive, es. Enable Drive, Start-Stop.

Vedere l'appendice per le corrispondenze *indice parametro - registro Modbus* e *indice parametro - Coil*.

1. Il protocollo MODBUS

Il protocollo MODBUS definisce il formato e la modalità di comunicazione tra un "master" che gestisce il sistema e uno o più "slave" che rispondono alle interrogazioni del master. Esso definisce come il master e gli slave stabiliscono e interrompono la comunicazione, come vengono scambiati i messaggi e come gli errori sono rilevati.

Si possono avere un master e fino a 247 slave su una linea comune; occorre notare che questo è un limite logico del protocollo, l'interfaccia fisica può peraltro limitare ulteriormente il numero di dispositivi; nell'implementazione attuale si prevede un massimo di 64 slave connessi alla linea.

Solo il master può iniziare una transazione. Una transazione può avere il formato domanda/risposta diretta ad un singolo slave o broadcast in cui il messaggio viene inviato a tutti gli slave sulla linea che non danno risposta. Una transazione è composta da una struttura (frame) singola domanda/singola risposta o una struttura singolo messaggio broadcast/nessuna risposta.

Alcune caratteristiche del protocollo non sono definite. Queste sono: standard di interfaccia, baud rate, parità, numero di stop bits. Il protocollo consente inoltre di scegliere tra due "modi" di comunicazione, ASCII e RTU (Remote Terminal Unit). Nel Drive viene implementato solo il modo RTU, in quanto più efficiente.

Il protocollo JBUS è funzionalmente identico al MODBUS e se ne differenzia per la diversa numerazione degli indirizzi: nel MODBUS questi partono da zero (0000 = 1° indirizzo) mentre nel JBUS partono da uno (0001 = 1° indirizzo) mantenendo questo scostamento per tutta la numerazione. Nel seguito, se non esplicitamente menzionato, pur facendo riferimento al MODBUS la descrizione si considera valida per entrambi i protocolli.

2. Formato dei messaggi

Per poter comunicare tra due dispositivi, il messaggio deve essere contenuto in un "involucro". L'involucro lascia il trasmettitore attraverso una "porta" ed è "portato" lungo la linea fino ad una analoga "porta" sul ricevitore. MODBUS stabilisce il formato di questo involucro che, tanto per il master che per lo slave, comprende:

- L'indirizzo dello slave con cui il master ha stabilito la transazione (l'indirizzo 0 corrisponde ad un messaggio broadcast inviato a tutti i dispositivi slave).
- Il codice della funzione che deve essere o è stata eseguita.
- I dati che devono essere scambiati.
- Il controllo d'errore composto secondo l'algoritmo CRC16.

Se uno slave individua un errore nel messaggio ricevuto (di formato, di parità o nel CRC16) il messaggio viene considerato non valido e scartato, uno slave che rilevi un errore nel messaggio quindi non esegue l'azione e non risponde alla domanda, così come se l'indirizzo non corrisponde ad uno slave in linea.

2.1. L'indirizzo

Come sopra menzionato, le transazioni MODBUS coinvolgono sempre il master, che gestisce la linea, ed uno slave per volta (tranne nel caso di messaggi broadcast). Per identificare il destinatario del messaggio viene trasmesso come primo carattere un byte che contiene l'indirizzo numerico dello slave selezionato. Ciascuno degli slave ha quindi assegnato un diverso numero di indirizzo che lo identifica univocamente. Gli indirizzi legali sono quelli da 1 a 247, mentre l'indirizzo 0, che non può essere assegnato ad uno slave, posto in testa al messaggio trasmesso dal master indica che questo è "broadcast", cioè diretto a tutti gli slave contemporaneamente. Possono essere trasmessi come broadcast solo messaggi che non richiedono risposta per espletare la loro funzione, quindi solo le assegnazioni.

2.2. Il codice funzione

Il secondo carattere del messaggio identifica la funzione che deve essere eseguita nel messaggio trasmesso dal master, cui lo slave risponde a sua volta con lo stesso codice ad indicare che la funzione è stata eseguita. È implementato un sottoinsieme delle funzioni MODBUS che comprende:

- 01 Read Coil Status
- 02 Read Input Status
- 03 Read Holding Registers
- 04 Read Input registers
- 05 Force Single Coil

- 06 Preset Single register
- 07 Read Status
- 15 Force multiple Coils
- 16 Preset Multiple Registers

Le funzioni 01 e 02 sono operativamente identiche e intercambiabili, così come le funzioni 03 e 04. Per una descrizione completa e dettagliata delle funzioni si rimanda al capitolo 3.

2.3. II CRC16

Gli ultimi due caratteri del messaggio contengono il codice di ridondanza ciclica (Cyclic Redundancy Check) calcolato secondo l'algoritmo CRC16. Per il calcolo di questi due caratteri il messaggio (indirizzo, codice funzione e dati scartando i bit di start, stop e l'eventuale parità) viene considerato come un unico numero binario continuo di cui il bit più significativo (MSB) viene trasmesso prima. Il messaggio viene innanzitutto moltiplicato per x^{16} (spostato a sinistra di 16 bit) e poi diviso per $x^{16}+x^{15}+x^2+1$ espresso come numero binario (1100000000000101). Il quoziente intero viene poi scartato e il resto a 16 bit (inizializzato a FFFFh all'inizio per evitare il caso di un messaggio di soli zeri) viene aggiunto di seguito al messaggio trasmesso. Il messaggio risultante, quando diviso dallo slave ricevente per lo stesso polinomio ($x^{16}+x^{15}+x^2+1$) deve dare zero come resto se non sono intervenuti errori (lo slave ricalcola il CRC).

Di fatto, dato che il dispositivo che serializza i dati da trasmettere (UART) trasmette prima il bit meno significativo (LSB) anziché il MSB come dovrebbe essere per il calcolo del CRC, questo viene effettuato invertendo il polinomio. Inoltre, dato che il MSB del polinomio influenza solo il quoziente e non il resto, questo viene eliminato rendendolo quindi 1010000000000001.

La procedura passo-passo per il calcolo del CRC16 è la seguente:

- 1) Caricare un registro a 16 bit con FFFFh (tutti i bit a 1).
- 2) Fare l'OR esclusivo del primo carattere con il byte superiore del registro, porre il risultato nel registro.
- 3) Spostare il registro a destra di un bit.
- 4) Se il bit uscito a destra dal registro (flag) è un 1, fare l'OR esclusivo del polinomio generatore 1010000000000001 con il registro.
- 5) Ripetere per 8 volte i passi 3 e 4.
- 6) Fare l'OR esclusivo del carattere successivo con il byte superiore del registro, porre il risultato nel registro.
- 7) Ripetere i passi da 3 a 6 per tutti i caratteri del messaggio.
- 8) Il contenuto del registro a 16 bit è il codice di ridondanza CRC che deve essere aggiunto al messaggio.

2.4. Sincronizzazione dei messaggi

La sincronizzazione del messaggio tra trasmettitore e ricevitore viene ottenuta interponendo una pausa tra i messaggi pari ad almeno 3.5 volte il tempo di un carattere. Se il ricevitore non riceve per un tempo di 4 caratteri, ritiene completato il messaggio precedente e considera che il successivo byte ricevuto sarà il primo di un nuovo messaggio e quindi un indirizzo.

2.5. Impostazione linea seriale

La comunicazione prevede le seguenti impostazioni :

- 1 bit di start
- 8 bits di dati (RTU protocol)
- 1 bit di stop
- no parity

Il baudrate è selezionabile tra i seguenti valori:

Baudrate	Timeout byte-byte (ms)
1200	33
2400	16
4800	8
9600	4
19200	2

3. Le funzioni MODBUS per Drive

Viene riportata di seguito la descrizione dettagliata delle funzioni MODBUS implementate per i Drive. Tutti i valori riportati nelle tabelle sono in esadecimale.

3.1. Read Output Status (01)

Questa funzione permette di richiedere lo stato ON o OFF di parametri Drive discreti (Coil). Il modo broadcast non è permesso.

Richiesta

Oltre all'indirizzo del Drive e al codice funzione (01) il messaggio contiene il numero del Coil di partenza (starting Address) espresso su due bytes e il numero di Coil da leggere anch'esso su due bytes. La numerazione dei Coil parte da zero (bit1 = 0) per il MODBUS, da uno (bit1 = 1) per il JBUS. Esempio:

- Drive address 17 (11_{hex})
- Coil dal 0004 al 0015 (12 Coil).

ADDR	FUNC	DATA	DATA	DATA	DATA	CRC	
		start	start	bit#	bit#	HI	LO
		Addr HI	Addr LO	HI	LO		
11	01	00	03	00	0C	CE	9F

Risposta

Oltre all'indirizzo del Drive e al codice funzione (01) il messaggio comprende un carattere che contiene il numero di dati e i caratteri contenenti i dati. I dati sono impaccati, così che un byte contiene lo stato di 8 Coil, il bit meno significativo del primo byte contiene il Coil corrispondente allo starting Address e così via. Se il numero di Coil da leggere non è multiplo di 8, l'ultimo carattere è completato con zeri nei bit più significativi.

Esempio: Risposta alla richiesta sopra riportata.

ADDR	FUNC	DATA	DATA	DATA	CRC	
		Byte	bit	bit	HI	LO
		Count	04..11	12..15		
11	01	02	CD	0B	6D	68

NOTA:

nel caso si selezioni un range di Coil che include dei coil/output riservati o mancanti, il valore di tali coil verrà posto a 0.

3.2. Read Input Status (02)

Questa funzione è operativamente identica alla precedente.

3.3. Read Output Registers (03)

Questa funzione permette di richiedere il valore di registri a 16 bit (word) contenenti parametri Drive. Il modo broadcast non è permesso.

Richiesta

Oltre all'indirizzo del Drive e al codice funzione (03) il messaggio contiene l'indirizzo di partenza dei registri (starting Address) espresso su due bytes e il numero dei registri da leggere anch'esso su due bytes. **Il numero massimo di registri che possono essere letti è 125.** La numerazione dei registri parte da zero (word1 = 0) per il MODBUS, da uno (word1 = 1) per il JBUS.

Esempio:

- Drive address 25 (19_{hex})
- Registri dal 0069 al 0071 (3 registri).

ADDR	FUNC	DATA start	DATA start	DATA word#	DATA word#	CRC	
		Addr HI	Addr LO	HI	LO	HI	LO
19	03	00	44	00	03	46	06

Risposta

Oltre all'indirizzo del Drive e al codice funzione (03), il messaggio comprende un carattere che contiene il numero di bytes di dati e i caratteri contenenti i dati. I registri richiedono due bytes, il primo dei quali contiene la parte più significativa.

Esempio: Risposta alla richiesta sopra riportata.

ADDR	FUNC	DATA	DATA	DATA	DATA	DATA	DATA	DATA	CRC	CRC	
		Byte Count	word 69 HI	word 69 LO	word 70 HI	word 70 LO	word 71 HI	word 71 LO			
19	03	06	02	2B	00	00	00	00	64	AF	7A

Nota:

nel caso si selezioni un range di registri che include dei registri riservati o mancanti, il valore di tali registri verra' posto a 0.

3.4. Read Input Registers (04)

Questa funzione è operativamente identica alla precedente.

3.5. Force Single Coil (05)

Questa funzione permette di forzare lo stato di un parametro discreto Drive binaria ON o OFF. Il modo broadcast è permesso.

Richiesta

Oltre all'indirizzo del Drive e al codice funzione (05) il messaggio contiene l'indirizzo del Coil da forzare su due bytes e due caratteri di cui il primo è posto a FF_{hex} (255) per forzare lo stato ON o 00_{hex} per forzare OFF, il secondo è posto a zero in ogni caso. La numerazione dei Coil parte da zero (bit1 = 0) per il MODBUS, da uno (bit1 = 1) per il JBUS.

Esempio:

- Drive address 46 (2F_{hex})
- Numero Coil: 4.

ADDR	FUNC	DATA	DATA	DATA	DATA	CRC	
		bit# HI	bit# LO	ON/OFF	(zero)	HI	LO
2F	05	00	03	FF	00	7A	74

Risposta

La risposta consiste nel ritrasmettere il messaggio ricevuto dopo che il Coil è stato modificato.

Esempio: Risposta alla richiesta sopra riportata.

ADDR	FUNC	DATA	DATA	DATA	DATA	CRC	
		bit# HI	bit# LO	ON/OFF	(zero)	HI	LO
2F	05	00	03	FF	00	7A	74

3.6. Preset Single Register (06)

Questa funzione permette di impostare il valore di un singolo registro a 16 bit. Il modo broadcast è permesso.

Richiesta

Oltre all'indirizzo del Drive e al codice funzione (06) il messaggio contiene l'indirizzo del registro (parametro) espresso su due bytes e il valore che deve essere assegnato. La numerazione degli indirizzi dei registri parte da zero

(word1 = 0) per il MODBUS, da uno (word1 = 1) per il JBUS.

Esempio:

- Drive address 38 (26_{hex})
- Registro 26
- Valore 962

ADDR	FUNC	DATA	DATA	DATA	DATA	CRC	
		bit# HI	bit# LO	WORD HI	WORD LO	HI	LO
26	06	00	19	03	9E	DF	82

Risposta

La risposta consiste nel ritrasmettere il messaggio ricevuto dopo che il registro è stato modificato.

Esempio: Risposta alla richiesta sopra riportata.

ADDR	FUNC	DATA	DATA	DATA	DATA	CRC	
		bit# HI	bit# LO	WORD HI	WORD LO	HI	LO
26	06	00	19	03	9E	DF	82

3.7. Read Status (07)

Questa funzione permette di leggere lo stato di otto bit predeterminati con un messaggio compatto. Il modo broadcast non è permesso.

Richiesta

Il messaggio comprende solo l'indirizzo del Drive e il codice funzione (07).

Esempio:

- Drive address 25 (19_{hex})

ADDR	FUNC	CRC	CRC
		HI	LO
19	07	4B	E2

Risposta

Oltre all'indirizzo del Drive e al codice funzione (07) il messaggio comprende un carattere che contiene i bit di stato.

Esempio: Risposta alla richiesta sopra riportata.

ADDR	FUNC	DATA	CRC	
		status byte	HI	LO
19	07	6D	63	DA

Il significato del bit è il seguente:

Bit#	Parameter index	Parameter name
0	314	Enable
1	315	Start / Stop
2	316	Fast stop
3	380	Drive ready
4	372	Speed limited
5	349	Curr limit state
6	394	Set speed
7	395	Spd zero Thr

3.8. Force Multiple Coils (15)

Questa funzione permette di forzare lo stato di ciascun Coil in un blocco consecutivo. Il modo broadcast è permesso.

Richiesta

Oltre all'indirizzo del Drive e al codice funzione (15), il messaggio contiene l'indirizzo di partenza dei Coil (starting Address) espresso su due bytes, il numero di Coil da forzare, il numero di byte che contengono i dati e i caratteri di dati. I dati sono impaccati, così che un byte contiene lo stato di 8 Coil, il bit meno significativo del primo byte deve contenere il Coil corrispondente allo starting Address e così via. Se il numero di Coil da forzare non è multiplo di 8, l'ultimo byte va completato con zeri nei bit più significativi. La numerazione degli indirizzi dei Coil parte da zero (bit1 = 0) per il MODBUS, da uno (bit1 = 1) per il JBUS.

Esempio:

- Drive address 12 (0C_{hex})
- Indirizzo Coil di partenza: 1
- Numero Coil da forzare: 4
- Coil 1 e 4 forzati a "1"; Coil 2 e 3 forzati a "0"

ADDR	FUNC	DATA start		DATA bit#	DATA bit#	DATA	DATA	CRC HI	CRC LO
		Addr HI	Addr LO	HI	LO	Byte Count	bit 1..4		
0C	0F	00	00	00	04	01	09	3F	09

Risposta

Oltre all'indirizzo del Drive e al codice funzione (15) il messaggio comprende l'indirizzo di partenza dei Coil (starting Address) e il numero di Coil forzati.

Esempio: Risposta alla richiesta sopra riportata.

ADDR	FUNC	DATA start		DATA bit#	DATA bit#	CRC HI	CRC LO
		Addr HI	Addr LO	HI	LO		
0C	0F	00	00	00	04	55	15

3.9. Preset Multiple Registers (16)

Questa funzione permette di impostare il valore di un blocco consecutivo di registri a 16 bit. Il modo broadcast è permesso.

Richiesta

Oltre all'indirizzo del Drive e al codice funzione (15) il messaggio contiene l'indirizzo di partenza dei registri da scrivere (starting Address), il numero di registri da scrivere, il numero di byte che contengono i dati e i caratteri di dati. La numerazione dei registri parte da zero (word1 = 0) per il MODBUS, da uno (word1 = 1) per il JBUS.

Esempio:

- Drive address 17 (11_{hex})
- Registro di partenza 35
- Numero registri da scrivere 1
- Valore 268

ADDR	FUNC	DATA start Addr HI	DATA start Addr LO	DATA word# HI	DATA word# LO	DATA Byte Count	DATA word 35 HI	DATA word 35 LO	CRC HI	CRC LO
11	10	00	22	00	01	02	01	0C	6C	87

Risposta

Oltre all'indirizzo del Drive e al codice funzione (16) il messaggio comprende l'indirizzo di partenza (starting Address) e il numero di registri scritti.

Esempio: Risposta alla richiesta sopra riportata.

ADDR	FUNC	DATA start Addr HI	DATA start Addr LO	DATA word# HI	DATA word# LO	CRC HI	CRC LO
11	10	00	22	00	01	A3	53

4. La gestione degli errori

Nel MODBUS esistono due tipi di errori, gestiti in modo diverso: errori di trasmissione ed errori operativi. Gli errori di trasmissione sono errori che alterano il messaggio, nel suo formato, nella parità (se è usata), o nel CRC16. Il Drive che rileva errori di questo tipo nel messaggio lo considera non valido e non dà risposta. Qualora invece il messaggio sia corretto nella sua forma ma la funzione richiesta, per qualsiasi motivo, non sia eseguibile, si ha un errore operativo. A questo errore il Drive risponde con un messaggio di eccezione. Questo messaggio è composto dall'indirizzo del Drive, dal codice della funzione richiesta, da un codice d'errore e dal CRC. Per indicare che la risposta è la notifica di un errore il codice funzione viene ritornato con il bit più significativo a "1".

Esempio:

- Drive address 10 (0A_{hex})
- Coil 1185

ADDR	FUNC	DATA start Addr HI	DATA start Addr LO	DATA bit# HI	DATA bit# LO	CRC HI	CRC LO
0A	01	04	A1	00	01	AC	63

Risposta

La richiesta chiede il contenuto del Coil 1185, che non esiste nel Drive slave. Questo risponde con il codice d'errore "02" (ILLEGAL DATA ADDRESS) e ritorna il codice funzione 81_{hex} (129). Esempio: Eccezione alla richiesta sopra riportata.

ADDR	FUNC	DATA Except. Code	CRC HI	CRC LO
0A	81	02	B0	53

4.1. Codici d'eccezione

L'implementazione attuale del protocollo prevede solo quattro codici d'eccezione:

Codice	Nome	Significato
01	ILLEGAL FUNCTION	Il codice di funzione ricevuto non corrisponde ad una funzione permessa sullo slave indirizzato.
02	ILLEGAL DATA ADDRESS	Il numero indirizzo cui fa riferimento il campo dati non è un registro permesso sullo slave indirizzato.
03	ILLEGAL DATA VALUE	Il valore da assegnare cui fa riferimento il campo dati non è permesso per questo registro.
07	NAK - NEGATIVE ACKNOWLEDGEMENT	La funzione non può essere eseguita nelle attuali condizioni operative o si è tentato di scrivere in un parametro a sola lettura.

5. Configurazione del sistema

Al fine di poter selezionare la configurazione della linea seriale, nei Drive della famiglia AVy e TPD32 è stato introdotto nel menù principale CONFIGURATION un sottomenù denominato "Set serial comm"; alcuni dei parametri sono comuni per i vari tipi di protocollo implementato (SLINK3, Modbus); nel menù sono contenuti i seguenti parametri:

Parameter	Allowed values	Description
Device address (P319)	0..255	Device address (all protocols)
Ser answer delay (P408)	0..900	Delay between end of reception and start of transmission (all protocols)
Ser protocol sel (P323)	0 = SLINK3 1 = MODBUS RTU 2 = JBUS	Serial protocol selection
Ser baudrate sel (P326)	0 = 19200 1 = 9600 2 = 4800 3 = 2400 4 = 1200	Baudrate selection (Except SLINK3)

IMPORTANTE!

Da notare che le impostazioni di Ser protocol sel (IPA 323) e Ser baudrate sel (IPA 326) sono rese attive allo start-up del drive, per cui occorre memorizzarle e spegnere l'azionamento per attivarle.

ENGLISH

Introduction

In the chapter the Drive parameters are referred to as 16-bit Modbus registers; a 32-bit Drive parameter covers therefore two Modbus registers.

Those Drive parameters with a 0.1 digital value, such as commands, states, etc., are referred to as Modbus Coils; a Coil corresponds therefore to a Drive discrete parameter, ex. Enable Drive, Start-Stop.

See the appendix for the following correspondences: *parameter index - Modbus register* and *parameter index - Coil*.

1. The MODBUS Protocol

The MODBUS protocol defines the format and the communication modes between a system controlling “master” and one or more “slaves” aimed at answering to the master requests. The protocol states how the master and the slaves start and stop their communication, how the messages can be exchanged and how the errors can be detected. A common line can host one master and 247 slaves; this is a protocol logic limit, the device number can be further limited by the physical interface; the present implementation foresees a maximum number of 64 slaves to be line-connected.

A transaction can be started exclusively by the master. A transaction can have a direct demand/response format or a broadcast format. The former is addressed to a single slave, the latter to all the line slaves, which, on their turn, give no response. A transaction can have a single demand/single response frame or a single broadcast message/no response frame.

Some protocol features have not been defined. They are: interface standard, baud rate, parity, stop bit number. The protocol allows also to choose between two communication “modes”: ASCII and RTU (Remote Terminal Unit). The RTU mode, which is the most efficient, is implemented in the Drives.

The JBUS protocol is similar to the MODBUS protocol; the only difference is given by the address numbering system: in MODBUS the numbering system starts from zero (0000 = 1st address) while in JBUS it starts from one (0001 = 1st address); this variance is maintained throughout the whole system. The following descriptions, if not otherwise stated, refer to both protocols.

2. Message format

In order to communicate between the two devices, the message has to be contained into a "casing". The casing leaves the transmitter via a "port" and it is "brought" along the line to a similar "port" on the receiver. MODBUS states the format of the casing, which, both for the master and for the slave, contains:

- The slave address for the master stated transaction (the address 0 corresponds to a broadcast message sent to all the slaves).
- The code of the function (already performed or to be performed).
- The data to be exchanged.
- The error control according to the CRC16 algorithm.

If a slave detects an error in the received message (a format, parity or CRC16 error), the message is invalid and therefore rejected; when a slave detects an error in the message, it does not perform the required action and does not answer to the demand as if the address does not correspond to an on-line slave.

2.1. The address

As stated above, the MODBUS transactions always involve the master (which controls the line) and one slave at the time (with the exception of broadcast messages). In order to detect the message receiver, the first sent character is a byte containing the numeric address of the selected slave. Each slave owns therefore a different address number for its identification. The legal addresses go from 1 to 247, while a master message starting with the address 0 means that this is a "broadcast" message simultaneously addressed to all the slaves (the address 0 can not be allocated to a slave). Broadcast messages are those messages which do not need a response to perform their function, i.e. the allocations.

2.2. The function code

The second character of the message states the function to be performed by the master message; the slave response contains the same code, thus stating that the function has been performed.

An implemented subset of the MODBUS functions contains:

- 01 Read Coil Status
- 02 Read Input Status
- 03 Read Holding Registers
- 04 Read Input registers
- 05 Force Single Coil
- 06 Preset Single register
- 07 Read Status
- 15 Force multiple Coils

- 16 Preset Multiple Registers

The 01 and 02 functions, so as the 03 and 04 functions, are similar and interchangeable. See chapter 3 for a complete and detailed description of the functions.

2.3. CRC16

The last two characters of the message contain the cyclic redundancy code (Cyclic Redundancy Check) calculated according to the CRC16 algorithm. As for the calculation of these two characters, the message (address, function code and data thus rejecting the parity and the start and stop bits) is considered as a single and continuous binary number whose most significant bit (MSB) is transmitted as first. The message is multiplied by x^{16} (it undergoes a 16-bit shift on the left) and then it is divided by $x^{16}+x^{15}+x^2+1$; it is stated as a binary number (1100000000000101). The integer quotient is rejected and the 16-bit remainder (it is initialized with FFFFh in order to avoid a zero made message) is added to the sent message. The obtained message, when the receiver slave has divided it by the same polynomial ($x^{16}+x^{15}+x^2+1$), must have a zero remainder if no error occurred (if not the slave calculates the CRC again).

Considering that the data serializing device (UART) transmits first the less significant bit (LSB) instead of the MSB as required by the CRC calculation, such calculation is performed by inverting the polynomial. Furthermore, as the MSB polynomial influences only the quotient and not the remainder, the remainder is deleted by making it equal to 1010000000000001.

The step by step procedure for the CRC16 calculation is the following:

- 1) Load a 16-bit register with FFFFh (the bit value is 1).
- 2) Perform the exclusive OR of the first character with the highest byte in the register; place the result in the register.
- 3) Perform a one-bit shift of the register on the right.
- 4) If the bit outcoming the register right side (flag) is 1, perform the exclusive OR between the 1010000000000001 generating polynomial and the register.
- 5) Repeat the steps 3 and 4 for eight times.
- 6) Perform the exclusive OR of the following character with the highest byte in the register; place the result in the register.
- 7) Repeat the steps from 3 to 6 for all the message characters.
- 8) The content of the 16-bit register is the CRC redundancy code to be added to the message.

2.4. Message synchronization

The message synchronization between the transmitter and the receiver is obtained by interposing a pause between the messages, such pause being equal to 3.5 times the character period. If the receiver does not receive for a period equal to 4 characters, the message is considered to be over; as a consequence the following received byte is treated as the first byte of a new message: an address.

2.5. Serial line setting

The communication foresees the following settings:

- 1 start bit
- 8 data bits (RTU protocol)
- 1 stop bit
- no parity

The baud rate can be selected among the following values:

Baudrate	Timeout byte-byte (ms)
1200	33
2400	16
4800	8
9600	4
19200	2

3. MODBUS functions for the drive

Here following is a detailed description of the MODBUS functions implemented for the Drive. All the values listed in the tables are hexadecimal.

3.1. Read Output Status (01)

This function allows to require the ON or OFF condition of the Drive discrete parameters (Coil). The broadcast mode is not allowed.

Request

Together with the Drive address and the function code (01), the message contains the number of the starting Coil (starting address) and the number of the Coil to be read; they are both stated on two bytes. The Coil numbering system starts from zero (bit1 = 0) for the MODBUS and from one (bit1 = 1) for the JBUS.

Example:

- Drive address 17 (11_{hex})
- Coil from 0004 to 0015 (12 Coil).

ADDR	FUNC	DATA	DATA	DATA	DATA	CRC	
		start	start	bit#	bit#	HI	LO
		Addr HI	Addr LO	HI	LO		
11	01	00	03	00	0C	CE	9F

Response

Together with the Drive address and the function code (01), the message includes a character containing the data byte number and some other characters containing the data. The data are packed so that a byte can contain the status corresponding to 8 Coils, the less significative bit of the first byte contains the Coil corresponding to the starting Address and so on. If the number of the Coils to be read is not a multiple of 8, the last character is completed with some zeros placed in the most significative bits.

Example: Response to the above mentioned request.

ADDR	FUNC	DATA	DATA	DATA	CRC	
		Byte	bit	bit	HI	LO
		Count	04..11	12..15		
11	01	02	CD	0B	6D	68

NOTE:

in case the Coil selected range includes some reserved or missing coils/outputs, the value of these coils is set with 0.

3.2. Read Input Status (02)

This function is similar to the previous one.

3.3. Read Output Registers (03)

This function allows to require the value of 16-bit (word) registers containing Drive parameters. The broadcast mode is not allowed.

Request

Together with the Drive address and the function code (03), the message contains the register starting address (starting Address) and the number of the registers to be read; they are both stated on two bytes. **The maximum number of registers which can be read is 125.** The register numbering system starts from zero (word1 = 0) for the MODBUS and from one (word1 = 1) for the JBUS.

Example:

- Drive address 25 (19_{hex})
- Registri from 0069 to 0071 (3 registers).

ADDR	FUNC	DATA start Addr HI	DATA start Addr LO	DATA word# HI	DATA word# LO	CRC HI	CRC LO
19	03	00	44	00	03	46	06

Response

Together with the Drive address and the function code (03), the message includes a character containing the data byte number and some other characters containing the data. The registers require two bytes where the first one contains the most significative section.

Example: Response to the above mentioned request.

ADDR	FUNC	DATA Byte Count	DATA word 69 HI	DATA word 69 LO	DATA word 70 HI	DATA word 70 LO	DATA word 71 HI	DATA word 71 LO	CRC HI	CRC LO
19	03	06	02	2B	00	00	00	64	AF	7A

NOTE:

in case the register selected range includes some reserved or missing registers, the value of these registers is set with 0.

3.4. Read Input Registers (04)

This function is similar to the previous one.

3.5. Force Single Coil (05)

This function allows to force the ON or OFF status of a Drive binary discrete parameter. The broadcast mode is allowed.

Request

Together with the Drive address and the function code (05), the message contains the address of the Coil to be forced (on two bytes) and two characters where the first one is set with FF_{hex} (255) to force the ON status or with 00_{hex} to force the OFF status, while the second one is always set with zero. The Coil numbering system starts from zero (bit1 = 0) for the MODBUS and from one (bit1 = 1) for the JBUS.

Example:

- Drive address 46 (2F_{hex})
- Coil number: 4.

ADDR	FUNC	DATA bit# HI	DATA bit# LO	DATA ON/OFF	DATA (zero)	CRC HI	CRC LO
2F	05	00	03	FF	00	7A	74

Response

The response is given by transmitting again the received message after the Coil has been modified.

Example: Response to the above mentioned request.

ADDR	FUNC	DATA bit# HI	DATA bit# LO	DATA ON/OFF	DATA (zero)	CRC HI	CRC LO
2F	05	00	03	FF	00	7A	74

3.6. Preset Single Register (06)

This function allows to set the value of a single 16-bit register. The broadcast mode is allowed.

Request

Together with the Drive address and the function code (06), the message contains the register address (parameter) on two bytes and the value to be allocated. The numbering system of the register addresses starts from zero (word1 = 0) for the MODBUS and from one (word1 = 1) for the JBUS.

Example:

- Drive address 38 (26_{hex})
- Register 26
- Value 962

ADDR	FUNC	DATA	DATA	DATA	DATA	CRC	
		bit#	bit#	WORD	WORD	HI	LO
		HI	LO	HI	LO		
26	06	00	19	03	9E	DF	82

Response

The response is given by transmitting again the received message after the register has been modified.

Example: Response to the above mentioned request.

ADDR	FUNC	DATA	DATA	DATA	DATA	CRC	
		bit#	bit#	WORD	WORD	HI	LO
		HI	LO	HI	LO		
26	06	00	19	03	9E	DF	82

3.7. Read Status (07)

This function allows to read the status of eight predefined bits with a compact message. The broadcast mode is not allowed.

Request

The message contains only the Drive address and the function code (07).

Example:

- Drive address 25 (19_{hex})

ADDR	FUNC	CRC	
		HI	LO
19	07	4B	E2

Response

Together with the Drive address and the function code (07), the message includes a character containing the status bits.

Example: Response to the above mentioned request.

ADDR	FUNC	DATA	CRC	
		status byte	HI	LO
19	07	6D	63	DA

The bit meaning is the following:

Bit#	Parameter index	Parameter name
0	314	Enable
1	315	Start / Stop
2	316	Fast stop
3	380	Drive ready
4	372	Speed limited
5	349	Curr limit state
6	394	Set speed
7	395	Spd zero Thr

3.8. Force Multiple Coils (15)

This function allows to force the status of each Coil into a consecutive block. The broadcast mode is allowed.

Request

Together with the Drive address and the function code (15), the message includes the Coil starting address (starting Address) on two bytes, the number of Coils to be forced, the number of bytes containing the data and the data characters. The data are packed, so that a byte contains the status corresponding to 8 Coils, the less significative bit of the first byte contains the Coil corresponding to the starting address and so on. If the number of the Coils to be forced is not a multiple of 8, the last byte is completed with some zeros placed in the most significative bits. The numbering system of the Coil address starts from zero (bit1 = 0) for the MODBUS and from one (bit1 = 1) for the JBUS.

Example:

- Drive address 12 (0C_{hex})
- Address of the starting Coil: 1
- Numero of Coils to be forced: 4
- Coils 1 and 4 forced at "1"; Coils 2 and 3 forced at "0"

ADDR	FUNC	DATA	DATA	DATA	DATA	DATA	DATA	CRC	CRC
		start Addr HI	start Addr LO	bit# HI	bit# LO	Byte Count	bit 1..4		
0C	0F	00	00	00	04	01	09	3F	09

Response

Together with the Drive address and the function code (15), the message contains the Coil starting address (starting Address) and the number of forced Coils.

Example: Response to the above mentioned request.

ADDR	FUNC	DATA start Addr HI	DATA start Addr LO	DATA bit# HI	DATA bit# LO	CRC HI	CRC LO
0C	0F	00	00	00	04	55	15

3.9. Preset Multiple Registers (16)

This function allows to set the value of a consecutive block made of 16-bit registers. The broadcast mode is allowed.

Request

Together with the Drive address and the function code (15), the message contains the starting address of the registers to be written (starting Address), the number of registers to be written, the number of bytes containing the data and the data characters. The register numbering system starts from zero (word1 = 0) for the MODBUS and from one (word1 = 1) for the JBUS.

Example:

- Drive address 17 (11_{hex})
- Starting Register 35
- Number of registers to be written 1
- Value 268

ADDR	FUNC	DATA start Addr HI	DATA start Addr LO	DATA word# HI	DATA word# LO	DATA Byte Count	DATA word 35 HI	DATA word 35 LO	CRC HI	CRC LO
11	10	00	22	00	01	02	01	0C	6C	87

Response

Together with the Drive address and the function code (16), the message contains the starting address (starting Address) and the number of written registers.

Example: Response to the above mentioned request.

ADDR	FUNC	DATA start Addr HI	DATA start Addr LO	DATA word# HI	DATA word# LO	CRC HI	CRC LO
11	10	00	22	00	01	A3	53

4. Error management

In MODBUS there are two kinds of errors which are managed in different ways: transmission errors and operating errors. The transmission errors change the format, the parity (if used) or the CRC16 of the message. When the Drive detects such errors, it considers the message invalid and gives no response. If the message format is the right one but its function can not be performed, the error is an operating one. The Drive answers to this error with a particular message. This message contains the Drive address, the code of the required function, an error code and the CRC. In order to underline that the response is aimed at stating the presence of an error, the function code is returned with the most significative bit set with "1".

Example:

- Drive address 10 (0A_{hex})
- Coil 1185

ADDR	FUNC	DATA start	DATA start	DATA bit#	DATA bit#	CRC	CRC
		Addr HI	Addr LO	HI	LO	HI	LO
0A	01	04	A1	00	01	AC	63

Response

The request refers to the content of the Coil 1185 which does not exist in the Drive slave. The slave answers with the error code "02" (ILLEGAL DATA ADDRESS) and goes back to the function code 81h (129).

Example: Exception to the above mentioned request.

ADDR	FUNC	DATA	CRC	
		Except. Code	HI	LO
0A	81	02	B0	53

4.1. Exception codes

This protocol implementation foresees only four exception codes:

Code	Name	Meaning
01	ILLEGAL FUNCTION	The received function code does not correspond to a function allowed on the addressed slave.
02	ILLEGAL DATA ADDRESS	The address number, which the data field refers to, is not a register allowed on the addressed slave.
03	ILLEGAL DATA VALUE	The value to be allocated, which the data field refers to, is not allowed for this register.
07	NAK - NEGATIVE ACKNOWLEDGEMENT	The function can not be performed with the present operating conditions or attempt to write an only-reading parameter.

5. System configuration

In order to select the configuration of the serial line, the Drives of the AVy and TPD32 series are supplied in the main CONFIGURATION menu with a submenu called "Ser com settings"; some parameters are common to the different kinds of implemented protocols (SLINK3, Modbus); the menu contains the following parameters:

Parameter	Allowed values	Description
Device address (P319)	0..255	Device address (all protocols)
Ser answer delay (P408)	0..900	Delay between end of reception and start of transmission (all protocols)
Ser protocol sel (P323)	0 = SLINK3 1 = MODBUS RTU 2 = JBUS	Serial protocol selection
Ser baudrate sel (P326)	0 = 19200 1 = 9600 2 = 4800 3 = 2400 4 = 1200	Baudrate selection (Except SLINK3)

IMPORTANT!

The settings of Ser protocol sel (IPA323) and Ser baud rate sel (IPA326) are enabled with the drive start-up; it is therefore required to store them and to switch the drive off.

FRANÇAISE

Introduction

Les paramètres Drive sont indiqués dans le chapitre comme des registres Modbus de 16 bits; un paramètre Drive de 32 bits occupe donc 2 registres Modbus.

Les paramètres Drive qui ont une valeur numérique 0,1 comme commandes, états, etc., sont indiqués comme Coils Modbus; un Coil correspond donc à un paramètre discret du Drive, ex. Enable Drive, Start-Stop.

Voir l'appendice pour les équivalences *index paramètre - registre Modbus* et *index paramètre - Coil*.

1. Le protocole MODBUS

Le protocole MODBUS définit le format et le mode de communication entre un «master» qui gère le système et un ou plusieurs «slave» qui répondent aux interrogations du master. Il définit comment le master et les slaves établissent et interrompent la communication, comment sont échangés les messages et comment les erreurs sont détectées.

On peut avoir un master et jusqu'à 247 slaves sur une ligne commune; il faut savoir que c'est une limite logique du protocole, l'interface physique peut d'autre part limiter encore le nombre des dispositifs; dans l'implémentation actuelle on prévoit un maximum de 64 slaves connectés à la ligne.

Seul le master peut débiter une transaction. Une transaction peut avoir le format demande/réponse directe à un seul slave ou broadcast où le message est envoyé à tous les slaves sur la ligne qui ne répondent pas. Une transaction comprend une structure (frame) simple demande/simple réponse ou une structure simple message broadcast/aucune réponse.

Certaines caractéristiques du protocole ne sont pas définies. Ce sont: standard d'interface, baud rate, parité, nombre de stop bits. En outre, le protocole permet de choisir entre deux «modes» de communication, ASCII et RTU (Remote Terminal Unit). Sur les Drive on implémente seulement le mode RTU car il est plus efficace.

Le protocole JBUS fonctionne exactement comme le MODBUS et n'en diffère que par la numération des adresses: dans le MODBUS elles partent de zéro (0000 = 1ère adresse) alors que dans le JBUS elles partent de 1 (0001 = 1ère adresse) en maintenant cet écart pour toute la numération. Par la suite, si ce n'est pas explicitement mentionné, même en faisant référence au MODBUS la description est considérée comme valable pour les deux protocoles.

2. Format des messages

Pour pouvoir communiquer entre deux dispositifs, le message doit être contenu dans une «enveloppe». L'enveloppe laisse l'émetteur, traverse une «porte» et est «amenée» sur la ligne jusqu'à une «porte» analogue sur le récepteur. MODBUS établit le format de cette enveloppe qui, tant pour le master que pour le slave, comprend:

- L'adresse du slave avec lequel le master a établi la transaction (l'adresse 0 correspond à un message broadcast envoyé à tous les dispositifs slaves).
- Le code de la fonction qui doit être ou qui a été exécutée.
- Les données qui doivent être échangées.
- Le contrôle d'erreur composé selon l'algorithme CRC16.

Si un slave trouve une erreur dans le message reçu (de format, de parité ou dans le CRC16) le message n'est pas valable et est éliminé, un slave qui détecte une erreur dans le message n'exécute donc pas l'action et ne répond pas à la demande et c'est la même chose si l'adresse ne correspond pas à un slave en ligne.

2.1. L'adresse

Comme indiqué ci-dessus, les transactions MODBUS impliquent toujours le master, qui gère la ligne, et un slave à la fois (sauf dans le cas de messages broadcast). Pour identifier le destinataire du message un byte est transmis comme premier caractère et il contient l'adresse numérique du slave sélectionné. Chaque slave a donc un numéro d'adresse différent qui lui est assigné et qui l'identifie clairement. Les adresses légales sont celles de 1 à 247, alors que l'adresse 0, qui ne peut être attribuée à un slave, se trouvant au début du message transmis par le master indique que c'est un «broadcast», c'est-à-dire adressé à tous les slaves en même temps. Seuls les messages exigeant une réponse pour accomplir leur fonction peuvent être transmis comme broadcast et par conséquent uniquement les attributions.

2.2. Le code fonction

Il secondo carattere del messaggio identifica la funzione che deve essere eseguita nel messaggio trasmesso dal master, cui lo slave risponde a sua volta con lo stesso codice ad indicare che la funzione è stata eseguita. Un sous-ensemble des fonctions MODBUS est implémenté et il comprend:

- 01 Read Coil Status
- 02 Read Input Status
- 03 Read Holding Registers

- 04 Read Input registers
- 05 Force Single Coil
- 06 Preset Single register
- 07 Read Status
- 15 Force multiple Coils
- 16 Preset Multiple Registers

Les fonctions 01 et 02 sont opérativement identiques et interchangeables, ainsi que les fonctions 03 et 04. Pour la description complète et détaillée des fonctions voir le chapitre 3.

2.3. II CRC16

Les deux derniers caractères du message contiennent le code de redondance cyclique (Cyclic Redundancy Check) calculé selon l'algorithme CRC16. Pour le calcul de ces deux caractères le message (adresse, code fonction et données en éliminant les bits de start, stop et l'éventuelle parité) est considéré comme un seul numéro binaire continu dont le bit plus significatif (MSB) est transmis en premier. Le message est tout d'abord multiplié par x^{16} (déplacé à gauche de 16 bits), puis divisé par $x^{16}+x^{15}+x^2+1$ exprimé comme numéro binaire (110000000000101). Le quotient entier est ensuite éliminé et le reste à 16 bits (initialisé à FFFFh au début pour éviter le cas d'un message uniquement de zéros) est ajouté à la suite du message transmis. Le message qui résulte, lorsqu'il est divisé par le slave récepteur pour le même polynôme ($x^{16}+x^{15}+x^2+1$), doit donner zéro de reste s'il n'y a eu aucune erreur (le slave recalcule le CRC).

En effet, comme le dispositif qui sérialise les données à transmettre (UART) transmet d'abord le bit le moins significatif (LSB) plutôt que le MSB comme cela devrait être pour le calcul du CRC, il est effectué en inversant le polynôme. En outre, comme le MSB du polynôme n'influence que le quotient et pas le reste, il est éliminé en le faisant donc devenir 101000000000001.

La procédure pas à pas pour le calcul du CRC16 est la suivante:

- 1) Charger un registre à 16 bits avec FFFFh (tous les bits à 1).
- 2) Faire l'OR exclusif du premier caractère avec le byte supérieur du registre, placer le résultat dans le registre.
- 3) Déplacer le registre à droite d'un bit.
- 4) Si le bit sorti à droite du registre (flag) est un 1, faire l'OR exclusif du polynôme générateur 101000000000001 avec le registre.
- 5) Répéter 8 fois de suite les pas 3 et 4.
- 6) Faire l'OR exclusif du caractère suivant avec le byte supérieur du registre, placer le résultat dans le registre.

- 7) Répéter les pas de 3 à 6 pour tous les caractères du message.
- 8) Le contenu du registre à 16 bits est le code de redondance CRC qui doit être ajouté au message.

2.4. Synchronisation des messages

La synchronisation du message entre l'émetteur et le récepteur est obtenue en plaçant une pause entre les messages qui est égale à au moins 3,5 fois le temps d'un caractère. Si le récepteur ne reçoit pas un temps de 4 caractères, il considère que le message précédent est terminé et que le byte suivant reçu sera le premier d'un nouveau message et donc une adresse.

2.5. Programmation ligne série

La communication prévoit les programmations suivantes:

- 1 bit de start
- 8 bits de données (protocole RTU)
- 1 bit de stop
- no parity

Le baudrate peut être sélectionné parmi les valeurs suivantes:

Baudrate	Timeout byte-byte (ms)
1200	33
2400	16
4800	8
9600	4
19200	2

3. Les fonctions MODBUS pour Drive

Vous trouverez ci-après la description détaillée des fonctions MODBUS implémentées pour les Drive. Toutes les valeurs figurant dans les tableaux sont hexadécimales.

3.1. Read Output Status (01)

Cette fonction permet de demander la situation ON ou OFF des paramètres Drive décrits (Coil). Le mode broadcast n'est pas permis.

Demande

En plus de l'adresse du Drive et du code fonction (01) le message contient le numéro du Coil de départ (starting Address) exprimé sur deux bytes et le numéro de Coil à lire également sur deux bytes. La numérotation des Coils part de zéro (bit1 = 0) pour le MODBUS, de un (bit1 = 1) pour le JBUS.

Exemple:

- Drive address 17 (11_{hex})
- Coil de 0004 à 0015 (12 Coil).

ADDR	FUNC	DATA	DATA	DATA	DATA	CRC	
		start	start	bit#	bit#	HI	LO
		Addr HI	Addr LO	HI	LO		
11	01	00	03	00	0C	CE	9F

Réponse

En plus de l'adresse du Drive et du code fonction (01) le message comprend un caractère qui contient le nombre de bytes des données et les caractères contenant les données. Les données sont enveloppées, de manière à ce qu'un byte contienne l'état de 8 Coils, le bit moins significatif du premier byte contient le Coil correspondant au starting Address et ainsi de suite. Si le numéro du Coil à lire n'est pas un multiple de 8, le dernier caractère est complété par des zéros dans les bits les plus significatifs. Exemple: Réponse à la demande figurant ci-dessus.

ADDR	FUNC	DATA	DATA	DATA	CRC	
		Byte	bit	bit	HI	LO
		Count	04..11	12..15		
11	01	02	CD	0B	6D	68

NOTE:

si l'on sélectionne une plage de Coils qui comprend des coils/ outputs réservés ou manquants, la valeur de ces coils sera sur 0.

3.2. Read Input Status (02)

Cette fonction est opérativement identique à la précédente.

3.3. Read Output Registers (03)

Cette fonction permet de demander la valeur des registres à 16 bits (word) contenant les paramètres Drive. Le mode broadcast n'est pas permis.

Demande

En plus de l'adresse du Drive et du code fonction (03) le message contient l'adresse de départ des registres (starting Address) exprimée sur deux bytes et le nombre des registres à lire est également sur deux bytes. **le nombre maximum de registres pouvant être lu est 125**. La numérotation des registres part de zéro (word1 = 0) pour le MODBUS, de un (word1 = 1) pour le JBUS.

Exemple:

- Drive address 25 (19_{hex})
- Registres de 0069 à 0071 (3 registres).

ADDR	FUNC	DATA start Addr HI	DATA start Addr LO	DATA word# HI	DATA word# LO	CRC HI	CRC LO
19	03	00	44	00	03	46	06

Réponse

En plus de l'adresse du Drive et du code fonction (03) le message comprend un caractère qui contient le nombre de bytes des données et les caractères contenant les données. Les registres demandent deux bytes, dont le premier contient la partie la plus significative.

Exemple: Réponse à la demande figurant ci-dessus.

ADDR	FUNC	DATA Byte Count	DATA word 69 HI	DATA word 69 LO	DATA word 70 HI	DATA word 70 LO	DATA word 71 HI	DATA word 71 LO	CRC HI	CRC LO
19	03	06	02	2B	00	00	00	64	AF	7A

NOTE:

si l'on sélectionne une plage de registres qui comprend des registres réservés ou manquants, la valeur de ces registres sera sur 0.

3.4. Read Input Registers (04)

Cette fonction est opérativement identique à la précédente.

3.5. Force Single Coil (05)

Cette fonction permet de forcer l'état d'un paramètre discret Drive binaire ON ou OFF. Le mode broadcast est permis.

Demande

En plus de l'adresse du Drive et du code fonction (05) le message contient l'adresse du Coil à forcer sur deux bytes et sur deux caractères dont le premier est sur FF_{hex} (255) pour forcer l'état ON ou 00_{hex} pour forcer OFF, de toute manière le deuxième est sur zéro. La numérotation des Coils part de zéro (bit1 = 0) pour le MODBUS, de un (bit1 = 1) pour le JBUS.

Exemple:

- Drive address 46 (2F_{hex})
- Numéro Coil: 4.

ADDR	FUNC	DATA bit# HI	DATA bit# LO	DATA ON/OFF	DATA (zero)	CRC HI	CRC LO
2F	05	00	03	FF	00	7A	74

Réponse

La réponse consiste à transmettre le message reçu après la modification du Coil.

Exemple: Réponse à la demande figurant ci-dessus.

ADDR	FUNC	DATA bit# HI	DATA bit# LO	DATA ON/OFF	DATA (zero)	CRC HI	CRC LO
2F	05	00	03	FF	00	7A	74

3.6. Preset Single Register (06)

Cette fonction permet de programmer la valeur d'un seul registre à 16 bits. Le mode broadcast est permis.

Demande

En plus de l'adresse du Drive et du code fonction (06) le message contient l'adresse du registre (paramètre) exprimé sur deux bytes et la valeur qui doit être attribuée. La numération des adresses des registres part de zéro (word1 = 0) pour le MODBUS, de un (word1 = 1) pour le JBUS.

Exemple:

- Drive address 38 (26_{hex})
- Registre 26
- Valeur 962

ADDR	FUNC	DATA	DATA	DATA	DATA	CRC	
		bit#	bit#	WORD	WORD	HI	LO
		HI	LO	HI	LO		
26	06	00	19	03	9E	DF	82

Réponse

La réponse consiste à retransmettre le message reçu après la modification du registre.

Exemple: Réponse à la demande figurant ci-dessus.

ADDR	FUNC	DATA	DATA	DATA	DATA	CRC	
		bit#	bit#	WORD	WORD	HI	LO
		HI	LO	HI	LO		
26	06	00	19	03	9E	DF	82

3.7. Read Status (07)

Cette fonction permet de lire l'état de huit bits prédéterminés avec un message compact. Le mode broadcast n'est pas permis.

Demande

Le message comprend seulement l'adresse du Drive et le code fonction (07).

Exemple:

- Drive address 25 (19_{hex})

ADDR	FUNC	CRC	CRC
		HI	LO
19	07	4B	E2

Réponse

En plus de l'adresse du Drive et du code fonction (07) le message comprend un caractère qui contient les bits d'état.

Exemple: Réponse à la demande figurant ci-dessus.

ADDR	FUNC	DATA status byte	CRC	
			HI	LO
19	07	6D	63	DA

La signification du bit est la suivante:

Bit#	Parameter index	Parameter name
0	314	Enable
1	315	Start / Stop
2	316	Fast stop
3	380	Drive ready
4	372	Speed limited
5	349	Curr limit state
6	394	Set speed
7	395	Spd zero Thr

3.8. Force Multiple Coils (15)

Cette fonction permet de forcer l'état de chaque Coil en un bloc consécutif. Le mode broadcast est permis.

Demande

En plus de l'adresse du Drive et du code fonction (15), le message contient l'adresse de départ des Coils (starting Address) exprimée sur deux bytes, le nombre de Coils à forcer, le nombre de bytes que contiennent les données et les caractères des données. Les données sont enveloppées, de manière à ce qu'un byte contienne l'état de 8 Coils, le bit moins significatif du premier byte contient le Coil correspondant au starting Address et ainsi de suite. Si le nombre de Coils à forcer n'est pas un multiple de 8, le dernier byte doit être complété par des zéros dans les bits plus significatifs. La numérotation des adresses des Coils part de zéro (bit1 = 0) pour le MODBUS, de un (bit1 = 1) pour le JBUS.

Exemple:

- Drive address 12 (0C_{hex})
- Adresse Coil de départ: 1
- Nombre de Coils à forcer: 4
- Coil 1 et 4 forcés à "1"; Coil 2 et 3 forcés à "0".

ADDR	FUNC	DATA start Addr HI	DATA start Addr LO	DATA bit#	DATA bit#	DATA Byte Count	DATA bit 1..4	CRC	
				HI	LO			HI	LO
0C	0F	00	00	00	04	01	09	3F	09

Réponse

En plus de l'adresse du Drive et du code fonction (15) le message comprend l'adresse de départ des Coils (starting Address) et le nombre de Coils forcés.

Exemple: Réponse à la demande figurant ci-dessus.

ADDR	FUNC	DATA start Addr HI	DATA start Addr LO	DATA bit# HI	DATA bit# LO	CRC HI	CRC LO
0C	0F	00	00	00	04	55	15

3.9. Preset Multiple Registers (16)

Cette fonction permet de programmer la valeur d'un bloc consécutif de registres à 16 bits. Le mode broadcast est permis.

Demande

En plus de l'adresse du Drive et du code fonction (15) le message contient l'adresse de départ des registres à écrire (starting Address), le nombre de registres à écrire, le nombre de bytes que contiennent les données et les caractéristiques des données. La numération des registres part de zéro (word1 = 0) pour le MODBUS, de un (word1 = 1) pour le JBUS.

Exemple:

- Drive address 17 (11_{hex})
- Registre de départ 35
- Nombre de registres à écrire 1
- Valeur 268

ADDR	FUNC	DATA start Addr HI	DATA start Addr LO	DATA word# HI	DATA word# LO	DATA Byte Count	DATA word 35 HI	DATA word 35 LO	CRC HI	CRC LO
11	10	00	22	00	01	02	01	0C	6C	87

Réponse

En plus de l'adresse du Drive et du code fonction (16) le message comprend l'adresse de départ (starting Address) et le nombre de registres écrits.

Exemple: Réponse à la demande figurant ci-dessus.

ADDR	FUNC	DATA start Addr HI	DATA start Addr LO	DATA word# HI	DATA word# LO	CRC HI	CRC LO
11	10	00	22	00	01	A3	53

4. La gestion des erreurs

Dans le MODBUS il existe deux types d'erreurs, gérés de manière différente: erreurs de transmission et erreur opératives. Les erreurs de transmission sont des erreurs qui altèrent le message, dans son format, dans sa parité (si elle est utilisée), ou dans le CRC16. Le Drive qui détecte des erreurs de ce type dans le message considère qu'il n'est pas valable et ne donne pas de réponse. Par contre si le message est correct dans sa forme mais la fonction demandée, pour n'importe quelle raison, ne peut être exécutée, on a une erreur opérative. Le Drive répond par un message d'exception à cette erreur. Ce message comprend l'adresse du Drive, du code de la fonction demandée, à un code d'erreur et du CRC. Pour indiquer que la réponse est la notification d'une erreur le code fonction est retourné avec le bit plus significatif à «1».

Exemple:

- Drive address 10 (0A_{hex})
- Coil 1185

ADDR	FUNC	DATA	DATA	DATA	DATA	CRC	CRC
		start	start	bit#	bit#		
		Addr HI	Addr LO	HI	LO	HI	LO
0A	01	04	A1	00	01	AC	63

Réponse

La demande demande le contenu du Coil 1185, qui n'existe pas dans le Drive slave. Ce dernier répond avec le code d'erreur «02» (ILLEGAL DATA ADDRESS) et revient au code fonction 81h (129).

Exemple: Exception pour la demande indiquée ci-dessus.

ADDR	FUNC	DATA	CRC	CRC
		Except. Code	HI	LO
0A	81	02	B0	53

4.1. Codes d'exception

L'implémentation actuelle du protocole prévoit uniquement quatre codes d'exception:

Code	Nom	Signification
01	ILLEGAL FUNCTION	Le code de fonction reçu ne correspond pas une fonction permise sur le slave adressé.
02	ILLEGAL DATA ADDRESS	Le numéro adresse auquel se réfère le champ des données n'est pas un registre permis sur le slave adressé.
03	ILLEGAL DATA VALUE	La valeur à attribuer à laquelle se réfère le champ données n'est pas permise pour ce registre.
07	NAK - NEGATIVE ACKNOWLEDGEMENT	La fonction ne peut être exécutée dans les conditions actuelles de travail ou on a essayé d'écrire dans un paramètre à lecture seule.

5. Configuration du système

Afin de pouvoir sélectionner la configuration de la ligne série, dans les Drive de la famille AVy et TPD32 il a été introduit dans le menu principal CONFIGURATION un sous-menu appelé «Ser com settings»; certains de ces paramètres sont communs pour varier les types de protocole implémenté (SLINK3, Modbus); le menu contient les paramètres suivants:

Parameter	Allowed values	Description
Device address (P319)	0..255	Device address (all protocols)
Ser answer delay (P408)	0..900	Delay between end of reception and start of transmission (all protocols)
Ser protocol sel (P323)	0 = SLINK3 1 = MODBUS RTU 2 = JBUS	Serial protocol selection
Ser baudrate sel (P326)	0 = 19200 1 = 9600 2 = 4800 3 = 2400 4 = 1200	Baudrate selection (Except SLINK3)

IMPORTANT!

Il faut noter que les programmations de Ser protocol sel (IPA323) et Ser baudrate sel (IPA326) sont activées par le start-up du drive, il faut donc les mémoriser et arrêter l'actionnement pour les activer.

DEUTSCHE

Einleitung

Die Driveparameter werden im Kapitel als Modbus Register zu 16 Bit verstanden; ein Driveparameter zu 32 Bit nimmt daher 2 Modbusregister ein.

Die Driveparameter, die einen digitalen Wert von 0,1 als Befehle, Statusse etc. haben, werden als Modbus Coils verstanden; eine Coil entspricht daher einem diskreten Driveparameter, z. Bsp. Enable Drive, Start-Stop.

Für die Entsprechungen *Parameterindex – Modbus Register* und *Parameterindex – Coil* siehe Anhang.

1. Das MODBUS Protokoll

Das MODBUS Protokoll definiert Format und Modus der Kommunikation zwischen einem "Master", der das System verwaltet, und einem oder mehreren "Slaves", die auf die Abfragen des Master antworten. Es legt fest, wie der Master und die Slaves die Kommunikation herstellen und unterbrechen, wie die Nachrichten ausgetauscht und wie die Fehler erhoben werden.

Auf einer gemeinsamen Leitung können sich ein Master und bis zu 247 Slaves befinden; beachten Sie, dass es sich hierbei um eine logische Grenze des Protokolls handelt, die physikalische Schnittstelle kann überdies die Anzahl der Einrichtungen weiter einschränken; in der momentanen Implementierung ist eine Höchstanzahl von 64 an die Leitung angeschlossenen Slaves vorgesehen. Nur der Master kann eine Transaktion beginnen. Eine Transaktion kann ein Anfrage/Antwort-Format haben, das an einen einzelnen Slave gerichtet ist oder ein Broadcast, in dem die Nachricht an alle Slaves auf der Leitung gesandt wird, von denen keine Antwort kommt. Eine Transaktion besteht aus einer Struktur (Frame) einzelne Anfrage/individuelle Antwort oder einer Struktur einzelne Broadcast-Nachricht/keine Antwort.

Einige Merkmale des Protokolls sind nicht definiert. Hierzu gehören: Schnittstellenstandard, Baudrate, Parität, Anzahl der Stop Bits. Das Protokoll ermöglicht es ausserdem, zwischen zwei „Kommunikationsmodi“, ASCII und RTU (Remote Terminal Unit), zu wählen. In den Drives wird nur der effizientere RTU Modus implementiert.

Das Protokoll JBUS hat dieselben Funktionen wie MODBUS und unterscheidet sich von diesem durch die unterschiedliche Adressnummerierung: im MODBUS beginnen die Adressen bei null (0000 = 1. Adresse), während sie im JBUS bei eins beginnen (0001 = 1. Adresse) und diese Verschiebung für die gesamte Nummerierung beibehalten. Falls nicht ausdrücklich erwähnt, wird im Folgenden trotz Bezugnahme auf MODBUS die Beschreibung als für beide Protokolle gültig angesehen.

2. Nachrichtenformat

Um die Kommunikation zwischen zwei Einrichtungen zu ermöglichen, muss die Nachricht in einer "Hülle" enthalten sein. Die Hülle verlässt den Sender durch ein "Tor" und wird entlang der Leitung bis zu einem analogen "Tor" auf dem Empfänger "getragen". MODBUS legt das Format dieser Hülle fest, die sowohl für den Master als auch für den Slave Folgendes umfasst:

- Die Adresse des Slave, mit dem der Master die Transaktion eingegangen ist (die Adresse 0 entspricht einer Broadcast-Nachricht, die an alle Slave-Einrichtungen gesandt wurde).
- Den Code der Funktion, die ausgeführt werden muss oder ausgeführt wurde.
- Die Daten, die ausgetauscht werden sollen.
- Die Fehlerkontrolle, die sich gemäß Algorithmus CRC16 zusammensetzt.

Wenn ein Slave in der erhaltenen Nachricht einen Fehler feststellt (Formatfehler, Paritätsfehler oder im CRC16), wird die Nachricht als ungültig angesehen und ausgesondert; daher führt ein Slave, der einen Fehler in der Nachricht feststellt, die Tätigkeit nicht aus und antwortet nicht auf die Anfrage, so wie in jenen Fällen, in denen die Adresse keinem in der Leitung befindlichen Slave entspricht.

2.1. Adresse

Wie bereits oberhalb erwähnt, nehmen an den MODBUS Transaktionen immer der Master, der die Leitung verwaltet, und jeweils ein Slave teil (ausser im Fall der Broadcast-Nachrichten). Zur Identifizierung des Nachrichtenempfängers wird als erstes Zeichen ein Byte übermittelt, das die numerische Adresse des gewählten Slave enthält. Jedem Slave ist also eine andere Adressnummer zugewiesen, die ihn eindeutig identifiziert. Die rechtmäßigen Adressen sind jene von 1 bis 247, während die Adresse 0, die keinem Slave zugewiesen werden kann und an den Anfang der vom Master übertragenen Nachricht gestellt wird, angibt, dass es sich um ein „Broadcast“ handelt, das heisst, dass die Nachricht an alle Slaves gleichzeitig gerichtet ist. Im Broadcastmodus können nur Nachrichten übermittelt werden, die keine Antwort für die Ausführung ihrer Funktion benötigen, das heisst nur die Zuordnungen.

2.2. Funktionscode

Das zweite Zeichen der Nachricht identifiziert die Funktion, die in der vom Master übertragenen Nachricht ausgeführt werden soll, auf die der Slave seinerseits mit demselben Code antwortet, um anzugeben, dass die Funktion ausgeführt wurde.

Implementiert ist eine Untermenge von MODBUS Funktionen, die Folgendes umfasst:

- 01 Read Coil Status

- 02 Read Input Status
- 03 Read Holding Registers
- 04 Read Input registers
- 05 Force Single Coil
- 06 Preset Single register
- 07 Read Status
- 15 Force multiple Coils
- 16 Preset Multiple Registers

Die Funktionen 01 und 02 sind betriebsmäßig identisch und austauschbar, dasselbe gilt für die Funktionen 03 und 04. Für eine komplette und detaillierte Beschreibung der Funktionen wird auf Kapitel 3 verwiesen.

2.3. CRC16

Die letzten beiden Zeichen der Nachricht enthalten den Code der zyklischen Redundanz (Cyclic Redundancy Check), berechnet nach dem Algorithmus CRC16. Für die Berechnung dieser beiden Zeichen wird die Nachricht (Adresse, Funktionscode und Daten, unter Aussonderung der Start- und Stopbits und der eventuellen Parität) als eine einzige fortlaufende binäre Zahl angesehen, deren höchstwertiges Bit (MSB) zuerst übertragen wird. Die Nachricht wird als erstes mit x^{16} multipliziert (um 16 Bits nach links versetzt), dann durch $x^{16}+x^{15}+x^2+1$ dividiert und als Binärzahl ausgedrückt (1100000000000101). Der ganzzahlige Quotient wird danach ausgesondert und der Rest zu 16 Bits (am Anfang bei FFFFh initialisiert, um eine Nachricht mit nur Nullen zu vermeiden) wird in der Folge der übertragenen Nachricht hinzugefügt. In der hieraus resultierenden Nachricht muss sich bei der Division durch den Empfängerslave durch dasselbe Polynom ($x^{16}+x^{15}+x^2+1$) der Rest Null ergeben, wenn keine Fehler aufgetreten sind (der Slave berechnet den CRC neu).

Da die Einrichtung, die die zu übertragenden Daten serialisiert (UART), zuerst das niedrigstwertige Bit (LSB) überträgt und nicht das MSB, wie dies für die Berechnung des CRC der Fall sein müsste, geschieht dies durch Umkehrung des Polynoms. Da ausserdem das MSB des Polynoms lediglich den Quotienten und nicht den Rest beeinflusst, wird dieser aufgehoben, indem er zu 1010000000000001 wird.

Folgende Schritte müssen für die Berechnung des CRC16 befolgt werden:

- 1) Ein Register zu 16 Bit mit FFFFh laden (alle Bits zu 1).
- 2) Das ausschließliche OR des ersten Zeichens mit dem obersten Byte des Registers durchführen, das Resultat in das Register geben.
- 3) Das Register um ein Bit nach rechts verschieben.
- 4) Wenn das rechts aus dem Register ausgegangene Bit (flag) 1 ist, das ausschließliche OR des Erzeugerpolynoms 1010000000000001 mit dem Register durchführen.
- 5) Die Schritte 3 und 4 8-Mal wiederholen.

- 6) Das ausschließliche OR des folgenden Zeichens mit dem obersten Byte des Registers durchführen, das Resultat in das Register geben.
- 7) Die Schritte von 3 bis 6 für alle Zeichen der Nachricht wiederholen.
- 8) Der Inhalt des Registers zu 16 Bit ist der Redundanzcode CRC, der der Nachricht hinzugefügt werden muss.

2.4. Nachrichtensynchronisation

Die Synchronisation der Nachricht zwischen Sender und Empfänger erhält man durch Einschaltung einer Pause zwischen den Nachrichten, deren Länge mindestens 3.5 Mal der Zeit eines Zeichens entspricht. Wenn der Empfänger für eine Zeit von 4 Zeichen nichts empfängt, wird die vorhergehende Nachricht als vollständig angesehen und das in der Folge empfangene Byte für das erste einer neuen Nachricht und somit für eine Adresse gehalten.

2.5. Einstellung serielle Leitung

Die Kommunikation sieht die folgenden Einstellungen vor:

- 1 Startbit
- 8 Datenbits (RTU protocol)
- 1 Stopbit
- no parity

Die Baudrate kann aus den folgenden Werten ausgewählt werden:

Baudrate	Timeout byte-byte (ms)
1200	33
2400	16
4800	8
9600	4
19200	2

3. MODBUS-Funktionen für Drives

Es folgt eine detaillierte Beschreibung der für die Drives implementierten MODBUS-Funktionen. Alle in den Tabellen angeführten Werte sind in Hexadezimalzahlen ausgedrückt.

3.1. Read Output Status (01)

Diese Funktion ermöglicht die Anforderung des ON- und OFF-Status von diskreten Driveparametern (Coils). Der Broadcastmodus ist nicht zulässig.

Anforderung

Ausser der Adresse des Drive und dem Funktionscode (01) enthält die Nachricht die Nummer der Startcoil (starting Address), ausgedrückt auf zwei Bytes, sowie die Nummer der zu lesenden Coil, die ebenfalls auf zwei Bytes ausgedrückt ist. Die Nummerierung der Coils beginnt für MODBUS bei null (bit1 = 0), für JBUS bei eins (bit1 = 1).

Beispiel:

- Drive address 17 (11_{hex})
- Coils von 0004 bis 0015 (12 Coils).

ADDR	FUNC	DATA start Addr HI	DATA start Addr LO	DATA bit# HI	DATA bit# LO	CRC HI	CRC LO
11	01	00	03	00	0C	CE	9F

Antwort

Ausser der Adresse des Drive und dem Funktionscode (01) beinhaltet die Nachricht ein Zeichen, das die Anzahl der Datenbytes und der die Daten enthaltenden Zeichen enthält. Die Daten sind gepackt, so dass ein Byte den Status von 8 Coils enthält, das niedrigstwertige Bit des ersten Byte die der Coil entsprechende Starting Address enthält und so weiter. Wenn die Anzahl der zu lesenden Coils kein Vielfaches von 8 ist, wird das letzte Zeichen in den höchstwertigsten Bits mit Nullen ergänzt.

Beispiel: Antwort auf die oben angeführte Anforderung.

ADDR	FUNC	DATA Byte Count	DATA bit 04..11	DATA bit 12..15	CRC HI	CRC LO
11	01	02	CD	0B	6D	68

HINWEIS:

falls ein Coillbereich gewählt wird, der reservierte oder fehlende Coils/Outputs miteinschließt, wird der Wert dieser Coils auf 0 gesetzt.

3.2. Read Input Status (02)

Diese Funktion ist betriebsmäßig identisch mit der vorhergehenden.

3.3. Read Output Registers (03)

Diese Funktion ermöglicht die Anforderung des Werts von Registern zu 16 Bits (Word), die Driveparameter enthalten. Der Broadcastmodus ist nicht zulässig.

Anforderung

Ausser der Adresse des Drive und dem Funktionscode (03) enthält die Nachricht die Startadresse der Register (starting Address), ausgedrückt auf zwei Bytes, sowie die Anzahl der zu lesenden Register, die ebenfalls auf zwei Bytes ausgedrückt sind. **Es können höchstens 125 Register gelesen werden.** Die Nummerierung der Register beginnt für MODBUS bei null (word1 = 0), für JBUS bei eins (word1 = 1).

Beispiel:

- Drive address 25 (19_{hex})
- Register von 0069 bis 0071 (3 register).

ADDR	FUNC	DATA start Addr HI	DATA start Addr LO	DATA word# HI	DATA word# LO	CRC HI	CRC LO
19	03	00	44	00	03	46	06

Antwort

Ausser der Adresse des Drive und dem Funktionscode (01) beinhaltet die Nachricht ein Zeichen, das die Anzahl der Datenbytes und der die Daten enthaltenden Zeichen enthält. Die Register fordern zwei Bytes an, von denen das erste den höchstwertigen Teil enthält.

Beispiel: Antwort auf die oben angeführte Anforderung.

ADDR	FUNC	DATA Byte Count	DATA word 69 HI	DATA word 69 LO	DATA word 70 HI	DATA word 70 LO	DATA word 71 HI	DATA word 71 LO	CRC HI	CRC LO
19	03	06	02	2B	00	00	00	64	AF	7A

HINWIES:

falls ein Registerbereich gewählt wird, der reservierte oder fehlende Register miteinschließt, wird der Wert dieser Register auf 0 gesetzt.

3.4. Read Input Registers (04)

Diese Funktion ist betriebsmäßig identisch mit der vorhergehenden.

3.5. Force Single Coil (05)

Diese Funktion ermöglicht es, den Status eines diskreten Driveparameters binär ON oder OFF zu modifizieren. Der Broadcastmodus ist zulässig.

Anforderung

Ausser der Adresse des Drive und dem Funktionscode (05) enthält die Nachricht die Adresse der zu modifizierenden Coil auf zwei Bytes, sowie zwei Zeichen, von denen das erste auf FF_{hex} (255) gesetzt ist, um den ON-Status, oder auf 00_{hex} um den OFF-Status zu modifizieren, wobei letzterer in jedem Fall auf null gestellt ist. Die Nummerierung der Coils beginnt für MODBUS bei null (bit1 = 0), für JBUS bei eins (bit1 = 1).

Beispiel:

- Drive address 46 (2F_{hex})
- Coil Nummer: 4.

ADDR	FUNC	DATA bit# HI	DATA bit# LO	DATA ON/OFF	DATA (zero)	CRC HI	CRC LO
2F	05	00	03	FF	00	7A	74

Antwort

Die Antwort besteht in einer neuerlichen Übertragung der empfangenen Nachricht, nachdem die Coil abgeändert wurde.

Beispiel: Antwort auf die oben angeführte Anforderung.

ADDR	FUNC	DATA bit# HI	DATA bit# LO	DATA ON/OFF	DATA (zero)	CRC HI	CRC LO
2F	05	00	03	FF	00	7A	74

3.6. Preset Single Register (06)

Diese Funktion ermöglicht die Einstellung des Wertes eines einzelnen Registers zu 16 Bit. Der Broadcastmodus ist zulässig.

Anforderung

Ausser der Adresse des Drive und dem Funktionscode (06) enthält die Nachricht die Adresse des Registers (Parameters), das auf zwei Bytes ausgedrückt ist und den zuzuordnenden Wert. Die Nummerierung der Registeradressen beginnt für MODBUS bei null (word1 = 0), für JBUS bei eins (word1 = 1).

Beispiel:

- Drive address 38 (26_{hex})
- Register 26
- Wert 962

ADDR	FUNC	DATA	DATA	DATA	DATA	CRC	CRC
		bit#	bit#	WORD	WORD	HI	LO
		HI	LO	HI	LO	HI	LO
26	06	00	19	03	9E	DF	82

Antwort

Die Antwort besteht in einer neuerlichen Übertragung der empfangenen Nachricht, nachdem das Register abgeändert wurde.

Beispiel: Antwort auf die oben angeführte Anforderung.

ADDR	FUNC	DATA	DATA	DATA	DATA	CRC	CRC
		bit#	bit#	WORD	WORD	HI	LO
		HI	LO	HI	LO	HI	LO
26	06	00	19	03	9E	DF	82

3.7. Read Status (07)

Diese Funktion ermöglicht das Lesen des Status von acht vorbestimmten Bits mit einer kompakten Nachricht. Der Broadcastmodus ist nicht zulässig.

Anforderung

Die Nachricht enthält nur die Adresse des Drive und den Funktionscode (07).

Beispiel:

- Drive address 25 (19_{hex})

ADDR	FUNC	CRC	CRC
		HI	LO
19	07	4B	E2

Antwort

Ausser der Adresse des Drive und dem Funktionscode (07) beinhaltet die Nachricht ein Zeichen, das die Statusbits enthält.

Beispiel: Antwort auf die oben angeführte Anforderung.

ADDR	FUNC	DATA status byte	CRC HI	CRC LO
19	07	6D	63	DA

Das Bit hat folgende Bedeutung:

Bit#	Parameter index	Parameter name
0	314	Enable
1	315	Start / Stop
2	316	Fast stop
3	380	Drive ready
4	372	Speed limited
5	349	Curr limit state
6	394	Set speed
7	395	Spd zero Thr

3.8. Force Multiple Coils (15)

Diese Funktion ermöglicht es, den Status jeder Coil in einen fortlaufenden Block zu modifizieren. Der Broadcastmodus ist zulässig.

Anforderung

Ausser der Adresse des Drive und dem Funktionscode (15) enthält die Nachricht die Startadresse der Coils (starting Address), die auf zwei Bytes ausgedrückt ist, die Anzahl der zu modifizierenden Coils, die Anzahl der die Daten enthaltenden Bytes und die Datenzeichen. Die Daten sind gepackt, so dass ein Byte den Status von 8 Coils enthält; das niedrigstwertige Bit des ersten Byte muss die der Starting Address entsprechende Coil enthalten und so weiter. Wenn die Anzahl der zu modifizierenden Coils kein Vielfaches von 8 ist, muss des letzte Byte in den höchstwertigen Bits mit Nullen ergänzt werden. Die Nummerierung der Adressen der Coils beginnt für MODBUS bei null (bit1 = 0), für JBUS bei eins (bit1 = 1).

Beispiel:

- Drive address 12 (0C_{hex})
- Adresse Startcoil: 1
- Anzahl der zu modifizierenden Coils
- Coils 1 und 4 auf "1" modifiziert; Coils 2 und 3 auf "0" modifiziert.

ADDR	FUNC	DATA start Addr HI	DATA start Addr LO	DATA bit# HI	DATA bit# LO	DATA Byte Count	DATA bit 1..4	CRC HI	CRC LO
0C	0F	00	00	00	04	01	09	3F	09

Antwort

Ausser der Adresse des Drive und dem Funktionscode (15) enthält die Nachricht die Startadresse der Coils (starting Address) und die Anzahl der modifizierten Coils.

Beispiel: Antwort auf die oben angeführte Anforderung.

ADDR	FUNC	DATA start Addr HI	DATA start Addr LO	DATA bit# HI	DATA bit# LO	CRC HI	CRC LO
0C	0F	00	00	00	04	55	15

3.9. Preset Multiple Registers (16)

Diese Funktion ermöglicht die Einstellung des Wertes eines fortlaufenden Blocks von Registern zu 16 Bit. Der Broadcastmodus ist zulässig.

Anfrage

Ausser der Adresse des Drive und dem Funktionscode (15) enthält die Nachricht die Startadresse der zu schreibenden Register (starting Address), die Anzahl der zu schreibenden Register, die Anzahl der die Daten enthaltenden Bytes und die Datenzeichen. Die Nummerierung der Register beginnt für MODBUS bei null (word1 = 0), für JBUS bei eins (word1 = 1).

Beispiel:

- Drive address 17 (11_{hex})
- Startregister 35
- Anzahl der zu schreibenden Register 1
- Wert 268

ADDR	FUNC	DATA start Addr HI	DATA start Addr LO	DATA word# HI	DATA word# LO	DATA Byte Count	DATA word 35 HI	DATA word 35 LO	CRC HI	CRC LO
11	10	00	22	00	01	02	01	0C	6C	87

Antwort

Ausser der Adresse des Drive und dem Funktionscode (16) enthält die Nachricht die Startadresse (starting Address) und die Anzahl der geschriebenen Register.

Beispiel: Antwort auf die oben angeführte Anforderung.

ADDR	FUNC	DATA start Addr HI	DATA start Addr LO	DATA word# HI	DATA word# LO	CRC HI	CRC LO
11	10	00	22	00	01	A3	53

4. Fehlerverwaltung

Im MODBUS existieren zwei Arten von Fehlern, die auf unterschiedliche Weise verwaltet werden: Übertragungsfehler und Betriebsfehler. Übertragungsfehler sind Fehler, die die Nachricht in ihrem Format, in der Parität (falls verwendet) oder im CRC16 verändern. Der Drive, der in der Nachricht diese Art von Fehlern feststellt, sieht die Nachricht als ungültig an und gibt keine Antwort. Wenn hingegen die Nachricht in ihrer Form korrekt ist, die angeforderte Funktion jedoch aus irgendwelchen Gründen nicht ausführbar ist, dann liegt ein Betriebsfehler vor. Auf diesen Fehler antwortet der Drive mit einer Ausnahmenachricht. Diese Nachricht besteht aus der Adresse des Drive, dem Code der angeforderten Funktion, einem Fehlercode und dem CRC. Um anzugeben, dass die Antwort eine Fehlermitteilung ist, wird der Funktionscode mit dem höchstwertigen Bit auf „1“ zurückgesandt.

Beispiel:

- Drive address 10 (0A_{hex})
- Coil 1185

ADDR	FUNC	DATA	DATA	DATA	DATA	CRC	CRC
		start	start	bit#	bit#		
		Addr HI	Addr LO	HI	LO	HI	LO
0A	01	04	A1	00	01	AC	63

Antwort

Die Anforderung fragt nach dem Inhalt von Coil 1185, der im Slave-Drive nicht existiert. Dieser antwortet mit Fehlercode "02" (ILLEGAL DATA ADDRESS) und sendet den Funktionscode 81h (129) zurück.

Beispiel: Ausnahme auf die oben angeführte Anforderung.

ADDR	FUNC	DATA	CRC	CRC
		Except. Code		
		HI	LO	LO
0A	81	02	B0	53

4.1. Ausnahmecodes

Die derzeitige Implementierung des Protokolls sieht lediglich vier Ausnahmecodes vor:

Code	Name	Bedeutung
01	ILLEGAL FUNCTION	Der empfangene Funktionscode entspricht keiner auf dem adressierten Slave erlaubten Funktion.
02	ILLEGAL DATA ADDRESS	Die Adressnummer, auf die sich das Datenfeld bezieht, ist kein auf dem adressierten Slave erlaubtes Register.
03	ILLEGAL DATA VALUE	Der zuzuordnende Wert, auf den sich das Datenfeld bezieht, ist für dieses Register nicht erlaubt.
07	NAK - NEGATIVE ACKNOWLEDGEMENT	Die Funktion kann unter den aktuellen Betriebsbedingungen nicht ausgeführt werden oder es wurde versucht, in einen reinen Leseparameter zu schreiben.

5. Systemkonfiguration

Für die Wahl der seriellen Leitungskonfiguration wurde für die Drives der Familien AVy und TPD32 im Hauptmenü CONFIGURATION ein Untermenü mit dem Namen "Ser com settings" eingeführt; einige der Parameter sind den verschiedenen implementierten Protokolltypen gemein (SLINK3, Modbus); im Menü sind die folgenden Parameter enthalten:

Parameter	Allowed values	Description
Device address (P319)	0..255	Device address (all protocols)
Ser answer delay (P408)	0..900	Delay between end of reception and start of transmission (all protocols)
Ser protocol sel (P323)	0 = SLINK3 1 = MODBUS RTU 2 = JBUS	Serial protocol selection
Ser baudrate sel (P326)	0 = 19200 1 = 9600 2 = 4800 3 = 2400 4 = 1200	Baudrate selection (Except SLINK3)

WICHTIG!

Beachten Sie, dass die Einstellungen von Ser protocol sel (IPA323) und Ser baudrate sel (IPA326) beim Start-Up des Drive aktiv werden, sie müssen daher gespeichert, und für ihre Aktivierung muss der Antrieb ausgeschaltet werden.

ESPAÑOL

Introducción

Se hace referencia a los parámetros Drive en el capítulo como registros Modbus de 16 bits; así un parámetro Drive de 32 bits ocupa 2 registros Modbus.

Los parámetros Drive que tienen valor digital 0,1 como mandos, estados, etc., se clasifican como Coils Modbus; así un Coil corresponde a un parámetro discreto del Drive, ej. Enable Drive, Start-Stop.

Ver apéndice para las correspondencias índice *parámetro - registro Modbus* e *índice parámetro - Coil*.

1. El protocolo MODBUS

El protocolo MODBUS define el formato y la modalidad de comunicación entre un «master» que administra el sistema y uno o más «slave» que responden las preguntas del master. Esto define como el maestro y los esclavos establecen e interrumpen la comunicación, como se cambian mensajes y como se encuentran los errores.

Se puede tener un maestro y hasta 247 esclavos en una línea común; debe tenerse en cuenta que este es un límite lógico del protocolo, la interface física puede por otro lado limitar ulteriormente el número de dispositivos; en la instalación actual se prevé un máximo de 64 esclavos conectados a la línea.

Solamente el maestro puede iniciar una transacción. Una transacción puede tener el formato pregunta/respuesta directa a un solo slave o broadcast en el que el mensaje se envía a todos los slave de la línea que no dan respuesta. Una transacción se compone de una estructura (frame) una pregunta/una respuesta o una estructura de un mensaje broadcast/ ninguna respuesta.

Algunas de las características del protocolo no están definidas. Son las siguientes: estándar de interface, baud rate, paridad, número de stop bits. El protocolo consiente además de escoger entre dos «modos» de comunicación, ASCII y RTU (Remote Terminal Unit). En los Drive se encuentra instalado solamente el modo RTU ya que es más eficiente.

El protocolo JBUS es funcionalmente idéntico al MODBUS y se diferencia de éste por la distinta numeración de las direcciones: en el MODBUS se parte de cero (0000= 1ª dirección) mientras que en JBUS se parte de uno (0001 = 1ª dirección) manteniendo esta separación en toda la numeración. Para continuar, si no se menciona específicamente, solamente haciendo referencia al MODBUS la descripción se considera válida para ambos protocolos.

2. Formato de los mensajes

Para poder comunicarse entre dos dispositivos, se tiene que tener el mensaje en una «envoltura». La envoltura deja el transmisor a través de una «puerta» y se «transporta» a lo largo de la línea hasta una «puerta» análoga en el receptor. MODBUS establece el formato de esta envoltura que, tanto para el maestro como para el esclavo, contiene:

- La dirección del esclavo con la cual el maestro ha establecido la transacción (la dirección 0 corresponde a un mensaje broadcast enviado a todos los dispositivos esclavos).
- El código de la función que tiene que ejecutarse o que se ha ejecutado.
- Los datos que tienen que cambiarse.
- El control de errores compuesto según el algoritmo CRC16.

Si un esclavo individualiza un error en el mensaje recibido (de formato, de paridad o en el CRC16) el mensaje se considera no válido y se descarta, del mismo modo que se descarta cuando un esclavo localiza un error en el mensaje y no ejecuta la acción sin responder a la pregunta, y también se descarta el mensaje si la dirección no corresponde a un esclavo en línea.

2.1. La dirección

Como se ha mencionado anteriormente, las transacciones MODBUS implican siempre al maestro, que administra la línea, y un esclavo cada vez (excepto en el caso de mensajes broadcast). Para identificar el destinatario del mensaje se transmite como primer carácter un byte que contiene la dirección numérica del esclavo seleccionado. Cada uno de los esclavos tiene pues asignado un número de dirección distinto que lo identifica unívocamente. Las direcciones legales son aquellas que se sitúan entre 1 y 247, mientras que la dirección 0, que no se puede asignar a un esclavo, puesta en cabeza del mensaje transmitido por el maestro indica que este es “broadcast”, o sea directo a todos los esclavos al mismo tiempo. Solamente se pueden transmitir como broadcast mensajes que no requieran respuesta para acabar su función, o sea solamente las asignaciones.

2.2. El código de funciones

El segundo carácter del mensaje identifica la función que debe ser ejecutada en el mensaje transmitido por el maestro, que el esclavo responde a su vez con el mismo código para indicar que la función ya ha sido ejecutada.

Se implementa un subíndice de las funciones MODBUS que comprende:

- 01 Read Coil Status
- 02 Read Input Status

- 03 Read Holding Registers
- 04 Read Input registers
- 05 Force Single Coil
- 06 Preset Single register
- 07 Read Status
- 15 Force multiple Coils
- 16 Preset Multiple Registers

Las funciones 01 y 02 son operativamente idénticas e intercambiables, así como las funciones 03 y 04. Para una descripción completa y detallada de las funciones se remite al capítulo 3.

2.3. El CRC16

Los últimos dos caracteres del mensaje contienen el código de redundancia cíclica (Cyclic Redundancy Check) calculado según el algoritmo CRC16. Para el cálculo de estos dos caracteres el mensaje (dirección, código función y datos descartando los bits de start, stop y la paridad eventual) se considera como un único número binario continuo del cual el bit más significativo (MSB) se transmite primero. En primer lugar se multiplica el mensaje por x^{16} (trasladado a la izquierda de 16 bits) y luego dividido por $x^{16}+x^{15}+x^2+1$ expresado como número binario (1100000000000101). El cociente entero se descarta y el resto para 16 bits (inicializado a FFFFh al inicio para evitar el caso de un mensaje solamente de ceros) se adjunta a continuación del mensaje transmitido. El mensaje resultante, cuando el esclavo receptor lo ha dividido por el mismo polinomio ($x^{16}+x^{15}+x^2+1$) debe dar cero como resultado si no ha habido errores (el esclavo recalcula el CRC).

De hecho, dado que el dispositivo que transmite en serie los datos (UART) transmite primero el bit menos significativo (LSB) y no el MSB como debería ser para el cálculo del CRC, esto se efectúa invirtiendo el polinomio. Además, dado que el MSB del polinomio influencia solamente el cociente y no el resto, éste se elimina quedando 1010000000000001.

El procedimiento paso a paso para el cálculo del CRC16 es el siguiente:

- 1) Cargar un registro a 16 bits con FFFFh (todos los bits a 1).
- 2) Hacer el OR exclusivo del primer carácter con el byte superior del registro, poner el resultado en el registro.
- 3) Trasladar el registro a derecha de un bit.
- 4) Si el bit trasladado a la derecha del registro (flag) es un 1, hacer el OR exclusivo del polinomio generador 1010000000000001 con el registro.
- 5) Repetir 8 veces los pasos 3 y 4.

- 6) Hacer el OR exclusivo del carácter sucesivo con el byte superior del registro, poner el resultado en el registro.
- 7) Repetir los pasos del 3 al 6 para todos los caracteres del mensaje.
- 8) El contenido del registro a 16 bits y el código de redundancia CRC que tiene que adjuntarse al mensaje.

2.4. Sincronización de los mensajes

La sincronización del mensaje entre transmisor y receptor se obtiene interponiendo una pausa entre los mensajes de un mínimo de 3.5 veces el tiempo de un carácter. Si el receptor no recibe durante un tiempo de 4 caracteres, retiene el mensaje precedente completado y considera que el sucesivo byte recibido será el primero de un nuevo mensaje y consecuentemente de una dirección.

2.5. Configuración de la línea serie

Configuración de la línea serie prevé las siguientes configuraciones:

- 1 bit de start
- 8 bits de datos (protocolo RTU)
- 1 bit de stop
- no paridad

El baudrate se puede seleccionar entre los siguientes valores:

Baudrate	Timeout byte-byte (ms)
1200	33
2400	16
4800	8
9600	4
19200	2

3. Las funciones MODBUS para drive

Se obtiene a continuación la descripción detallada de las funciones MODBUS instaladas para los Drive. Todos los valores obtenidos en las tablas son en hexadecimales.

3.1. Read Output Status (01)

Esta función permite pedir el estado ON o OFF de parámetros Drive discretos (Coil). El modo broadcast no está permitido.

Pregunta

Además de la dirección del Drive y el código función (01) el mensaje contiene el número del Coil de partida (dirección de inicio) expresado en dos bytes y el número de Coil a leer también en dos bytes. La numeración de los Coil parte de cero (bit1 = 0) para el MODBUS, de uno (bit1 = 1) para el JBUS.

Ejemplo:

- Drive address 17 (11_{hex})
- Coil del 0004 al 0015 (12 Coil).

ADDR	FUNC	DATA start	DATA start	DATA bit#	DATA bit#	CRC	CRC
		Addr HI	Addr LO	HI	LO	HI	LO
11	01	00	03	00	0C	CE	9F

Respuesta

Además de la dirección del Drive y el código función (01) el mensaje comprende un carácter que contiene el número de bytes de datos y los caracteres que contienen los datos. Los datos son empaquetados, así un byte contiene el estado de 8 Coil, el bit menos significativo del primer byte contiene el Coil correspondiente a la starting Address y así sucesivamente. Si el número de Coil a leer no es múltiplo de 8, el último carácter se completa con ceros en los bits más significativos.

Ejemplo: Respuesta a la pregunta anteriormente planteada.

ADDR	FUNC	DATA	DATA	DATA	CRC	CRC
		Byte Count	bit 04..11	bit 12..15	HI	LO
11	01	02	CD	0B	6D	68

NOTA:

en el caso que se seleccione un rango de registros que incluya los registros reservados o faltantes, el valor de tales registros se pondrá a 0.

3.2. Read Input Status (02)

Esta función es operativamente idéntica a la precedente.

3.3. Read Output Registers (03)

Esta función permite pedir el valor de registros a 16 bits (palabra) que contienen parámetros Drive. El modo broadcast no está permitido.

Pregunta

Además de la dirección del Drive y el código de función (03) el mensaje contiene la dirección de partida de los registros (starting Address) expresado en dos bytes y el número de los registros a leer también en dos bytes. El **número máximo de registros que se puede leer es 125**. La numeración de los registros parte de cero (word1 = 0) para el MODBUS, de uno (word1 = 0) para el JBUS.

Ejemplo:

- Drive address 25 (19_{hex})
- Registros del 0069 al 0071 (3 registros).

ADDR	FUNC	DATA start Addr HI	DATA start Addr LO	DATA word# HI	DATA word# LO	CRC HI	CRC LO
19	03	00	44	00	03	46	06

Respuesta

Además de la dirección del Drive y el código función (01) el mensaje comprende un carácter que contiene el número de bytes de datos y los caracteres que contienen los datos. Los registros necesitan dos bytes, el primero de los cuales contiene la parte más significativa.

Ejemplo: Respuesta a la pregunta anteriormente planteada.

ADDR	FUNC	DATA Byte Count	DATA word 69 HI	DATA word 69 LO	DATA word 70 HI	DATA word 70 LO	DATA word 71 HI	DATA word 71 LO	CRC HI	CRC LO
19	03	06	02	2B	00	00	00	64	AF	7A

Nota:

en el caso que se seleccione un rango de registros que incluya los registros reservados o faltantes, el valor de tales registros se pondrá a 0.

3.4. Read Input Registers (04)

Esta función es operativamente idéntica a la precedente.

3.5. Force Single Coil (05)

Esta función permite forzar el estado de un parámetro discreto Drive binaria ON o OFF. El modo broadcast está permitido.

Pregunta

Además de la dirección del Drive y el código función (05) el mensaje contiene la dirección del Coil a forzar en dos bytes y dos caracteres de los cuales el primero se encuentra a FF_{hex} (255) para forzar el estado ON o 00hex para forzar OFF, el segundo se encuentra a cero en cada caso. La numeración de los Coil parte de cero (bit1 = 0) para el MODBUS, de uno (bit1 = 1) para el JBUS.

Ejemplo:

- Drive address 46 (2F_{hex})
- Número Coil: 4.

ADDR	FUNC	DATA bit# HI	DATA bit# LO	DATA ON/OFF	DATA (zero)	CRC HI	CRC LO
2F	05	00	03	FF	00	7A	74

Respuesta

La respuesta consiste en transmitir el mensaje recibido después de que el Coil haya sido modificado.

Ejemplo: Respuesta a la pregunta anteriormente planteada.

ADDR	FUNC	DATA bit# HI	DATA bit# LO	DATA ON/OFF	DATA (zero)	CRC HI	CRC LO
2F	05	00	03	FF	00	7A	74

3.6. Preset Single Register (06)

Esta función permite configurar el valor de un solo registro a 16 bits. El modo broadcast está permitido.

Pregunta

Además de la dirección del Drive y el código función (06) el mensaje contiene la dirección del registro (parámetro) expresado en dos bytes y el valor que se debe asignar. La numeración de las direcciones de los registros parte de cero (word1 = 0) para el MODBUS, de uno (word1 = 1) para el JBUS.

Ejemplo:

- Drive address 38 (26_{hex})
- Registro 26
- Valor 962

ADDR	FUNC	DATA	DATA	DATA	DATA	CRC	CRC
		bit#	bit#	WORD	WORD	HI	LO
		HI	LO	HI	LO		
26	06	00	19	03	9E	DF	82

Respuesta

La respuesta consiste en transmitir el mensaje recibido después de que el registro haya sido modificado.

Ejemplo: Respuesta a la pregunta anteriormente planteada.

ADDR	FUNC	DATA	DATA	DATA	DATA	CRC	CRC
		bit#	bit#	WORD	WORD	HI	LO
		HI	LO	HI	LO		
26	06	00	19	03	9E	DF	82

3.7. Read Status (07)

Esta función permite leer el estado de ocho bits predeterminados con un mensaje compacto. El modo broadcast no está permitido.

Pregunta

El mensaje comprende solamente la dirección del Drive y el código función (07)

- Drive address 25 (19_{hex})

ADDR	FUNC	CRC	CRC
		HI	LO
19	07	4B	E2

Respuesta

Además de la dirección del Drive y el código función (07) el mensaje comprende un carácter que contiene los bits de estado.

Ejemplo: Respuesta a la pregunta anteriormente planteada.

ADDR	FUNC	DATA status byte	CRC	CRC
			HI	LO
19	07	6D	63	DA

El significado del bit es el siguiente:

Bit#	Parameter index	Parameter name
0	314	Enable
1	315	Start / Stop
2	316	Fast stop
3	380	Drive ready
4	372	Speed limited
5	349	Curr limit state
6	394	Set speed
7	395	Spd zero Thr

3.8. Force Multiple Coils (15)

Esta función permite forzar el estado de cada uno de los Coil en un bloque consecutivo. El modo broadcast está permitido.

Pregunta

Además de la dirección del Drive y el código función (15), el mensaje contiene la dirección de partida de los Coil (starting Address) expresado en dos bytes, el número de Coil a forzar, el número de bytes que contienen los datos y los caracteres de datos. Los datos son empacados, así un byte contiene el estado de 8 Coil, el bit menos significativo del primer byte contiene el Coil correspondiente a la starting Address y así sucesivamente. Si el número de Coil a forzar no es múltiplo de 8, el último byte se completa con ceros en los bits más significativos. La numeración de las direcciones de los Coil parte de cero (bit1 = 0) para el MODBUS, de uno (bit1 = 1) para el JBUS.

Ejemplo:

- Drive address 12 (0C_{hex})
- Dirección Coil de partida: 1
- Número Coil a forzar: 4
- Bobinas 1 y 4 forzadas a "1", Bobinas 2 y 3 forzadas a 0.

ADDR	FUNC	DATA start Addr HI	DATA start Addr LO	DATA bit#	DATA bit#	DATA Byte Count	DATA bit 1..4	CRC	CRC
				HI	LO			HI	LO
0C	0F	00	00	00	04	01	09	3F	09

Respuesta

Además de la dirección del Drive y el código función (15) el mensaje comprende la dirección de partida de los Coil (Starting Address) y el número de Coil forzados.

Ejemplo: Respuesta a la pregunta anteriormente planteada.

ADDR	FUNC	DATA start Addr HI	DATA start Addr LO	DATA bit# HI	DATA bit# LO	CRC HI	CRC LO
0C	0F	00	00	00	04	55	15

3.9. Preset Multiple Registers (16)

Esta función permite configurar el valor de un bloque consecutivo de registros a 16 bits. El modo broadcast está permitido.

Pregunta

Además de la dirección del Drive y el código función (15), el mensaje contiene la dirección de partida de los registros a recibir (starting Address), el número de registros a escribir, el número de bytes que contienen los datos y los caracteres de datos. La numeración de los registros parte de cero (word1 = 0) para el MODBUS, de uno (word1 = 1) para el JBUS.

Ejemplo:

- Drive address 17 (11_{hex})
- Registro de partida 35
- Número de registros a escribir 1
- Valor 268

ADDR	FUNC	DATA start Addr HI	DATA start Addr LO	DATA word# HI	DATA word# LO	DATA Byte Count	DATA word 35 HI	DATA word 35 LO	CRC HI	CRC LO
11	10	00	22	00	01	02	01	0C	6C	87

Respuesta

Además de la dirección del Drive y el código función (16) el mensaje comprende la dirección de partida (Starting Address) y el número de registros escritos.

Ejemplo: Respuesta a la pregunta anteriormente planteada.

ADDR	FUNC	DATA start Addr HI	DATA start Addr LO	DATA word# HI	DATA word# LO	CRC HI	CRC LO
11	10	00	22	00	01	A3	53

4. La gestión de los errores

En el MODBUS existen dos tipos de errores, tratados de distinto modo: errores de transmisión y errores operativos. Los errores de transmisión son errores que alteran el mensaje, en su formato, en la paridad (si se usa), o en CRC16. El Drive que detecta los errores de este tipo en el mensaje lo considera no válido y no da respuesta. Pero en el caso de que el mensaje sea correcto en su forma, pero la función que se pide, por el motivo que sea, no sea ejecutable, nos encontramos delante de un error operativo. A este error el Drive responde con un mensaje de excepción. Este mensaje se compone de la dirección del Drive, del código de la función que se pide, de un código de error y del CRC. Para indicar que la respuesta es la notificación de un error el código función se devuelve con el bit más significativo a "1".

Ejemplo:

- Drive address 10 (0A_{hex})
- Coil 1185

ADDR	FUNC	DATA start Addr HI	DATA start Addr LO	DATA bit# HI	DATA bit# LO	CRC HI	CRC LO
0A	01	04	A1	00	01	AC	63

Respuesta

La pregunta pide el contenido del Coil 1185, que no existe en el Drive esclavo. Estos responden con el código de error "02"(ILLEGAL DATA ADDRESS) y devuelve el código función 81h (129). Ejemplo: Excepción a la pregunta anteriormente planteada.

ADDR	FUNC	DATA Except. Code	CRC HI	CRC LO
0A	81	02	B0	53

4.1. Códigos de excepción

La implementación actual del protocolo prevé solamente cuatro códigos de excepción:

Codigo	Nombre	Significado
01	ILLEGAL FUNCTION	El código de funciones recibido no corresponde a una función permitida en el esclavo al que nos hemos dirigido
02	ILLEGAL DATA ADDRESS	El número dirección que hace referencia al campo de datos no es un registro permitido en el esclavo al que nos hemos dirigido.
03	ILLEGAL DATA VALUE	El valor a asignar que hace referencia al campo datos no está permitido en este registro.
07	NAK - NEGATIVE ACKNOWLEDGEMENT	La función no puede ser ejecutada en las actuales condiciones operativas o se ha intentado escribir en un parámetro de sólo lectura.

5. Configuración del sistema

Con el fin de poder seleccionar la configuración de la línea serie, en los Drives de la familia AVy y TPD32 se ha introducido en el menú principal CONFIGURATION un submenú denominado "Ser com settings"; algunos de los parámetros son comunes para los diferentes tipos de protocolo instalados (SLINK3, Modbus); en el menú se encuentran los siguientes parámetros:

Parameter	Allowed values	Description
Device address (P319)	0..255	Device address (all protocols)
Ser answer delay (P408)	0..900	Delay between end of reception and start of transmission (all protocols)
Ser protocol sel (P323)	0 = SLINK3 1 = MODBUS RTU 2 = JBUS	Serial protocol selection
Ser baudrate sel (P326)	0 = 19200 1 = 9600 2 = 4800 3 = 2400 4 = 1200	Baudrate selection (Except SLINK3)

IMPORTANTE!

Debe tenerse en cuenta que las configuraciones de Ser protocol sel (IPA323) y Ser baudrate sel (IPA326) están activadas en el start-up del drive, por lo cual se deben memorizar y cancelar el accionamiento para activarlas.

Appendix - REGISTER AND COIL MODBUS TABLES

Legenda / *Caption* / Légende / *Legende* / Leyenda:

RES = registro riservato per futuri impieghi
register reserved for a future use
registre réservé pour de futures utilisations
für zukünftige Einsätze reserviertes Register
registro reservado para futuras utilizaciones

HIP = parte più alta del registro precedente
(float o double word parameter)
highest part of the previous register
(float or double word parameter)
part plus haut que le registre précédent
(float ou double word parameter)
oberster Teil des vorhergehenden Registers
(float oder double word parameter)
parte más alta del registro precedente
(float o double word parameter)

AVy : Register-Parameter Table (Functions 03, 04, 06, 16).

IPA - AVy	Register
RES	0
RES	1
RES	2
RES	3
109	4
112	5
115	6
119	7
925	8
923	9
HIP	10
110	11
113	12
118	13
122	14
427	15
420	16
197	17
198	18
204	19
HIP	20
206	21
HIP	22
924	23
RES	24
RES	25
111	26
HIP	27
114	28
HIP	29
117	30
HIP	31
121	32
HIP	33

IPA - AVy	Register
RES	34
RES	35
RES	36
227	37
229	38
HIP	39
233	40
HIP	41
324	42
HIP	43
231	44
HIP	45
230	46
HIP	47
41	48
927	49
928	50
926	51
HIP	52
234	53
HIP	54
881	55
1147	56
914	57
564	58
581	59
582	60
583	61
RES	62
RES	63
RES	64
RES	65
1071	66
1072	67

IPA - AVy	Register
1073	68
RES	69
RES	70
RES	71
RES	72
161	73
HIP	74
162	75
HIP	76
163	77
HIP	78
164	79
HIP	80
371	81
HIP	82
167	83
HIP	84
168	85
HIP	86
165	87
HIP	88
726	89
HIP	90
166	91
HIP	92
436	93
HIP	94
437	95
HIP	96
251	97
RES	98
RES	99
RES	100
705	101

IPA - AVy	Register
RES	102
RES	103
683	104
HIP	105
644	106
HIP	107
685	108
HIP	109
645	110
HIP	111
686	112
HIP	113
RES	114
RES	115
684	116
HIP	117
89	118
HIP	119
687	120
HIP	121
RES	122
RES	123
682	124
HIP	125
90	126
HIP	127
688	128
HIP	129
176	130
HIP	131
689	132
HIP	133
692	134
690	135

IPA - AVy	Register
691	136
HIP	137
907	138
HIP	139
908	140
HIP	141
1024	142
HIP	143
909	144
HIP	145
1029	146
RES	147
1048	147
RES	148
1030	148
HIP	149
1031	150
HIP	151
1032	152
HIP	153
1033	154
HIP	155
RES	156
RES	157
RES	158
RES	159
646	160
643	161
HIP	162
647	163
HIP	164
712	165
709	166
710	167
711	168
722	169
723	170
724	171
725	172
HIP	173
727	174
HIP	175
893	176

IPA - AVy	Register
HIP	177
894	178
HIP	179
895	180
896	181
897	182
900	183
901	184
RES	185
RES	186
RES	187
RES	188
RES	189
44	190
48	191
47	192
HIP	193
49	194
HIP	195
42	196
43	197
378	198
HIP	199
379	200
HIP	201
39	202
40	203
RES	204
RES	205
RES	206
RES	207
RES	208
1	209
HIP	210
2	211
HIP	212
5	213
HIP	214
3	215
HIP	216
6	217

IPA - AVy	Register
HIP	218
4	219
HIP	220
7	221
8	222
9	223
10	224
11	225
13	226
467	227
889	228
HIP	229
RES	230
RES	231
RES	232
RES	233
RES	234
21	235
HIP	236
22	237
29	238
HIP	239
30	240
37	241
HIP	242
38	243
18	244
19	245
HIP	246
663	247
HIP	248
664	249
HIP	250
20	251
673	252
RES	253
RES	254
RES	255
RES	256
RES	257
1062	258

IPA - AVy	Register
HIP	259
411	260
HIP	261
1055	262
HIP	263
1059	264
HIP	265
1056	266
HIP	267
1057	268
HIP	269
1060	270
HIP	271
1058	272
HIP	273
1061	274
HIP	275
1065	276
HIP	277
RES	278
1063	279
HIP	280
1064	281
RES	282
RES	283
RES	284
RES	285
RES	286
236	287
1016	288
444	289
733	290
HIP	291
126	292
HIP	293
106	294
892	295
1078	296
1070	297
HIP	298
1069	299

AVy : Register-Parameter Table (Functions 03, 04, 06, 16).

IPA - AVy	Register
1079	300
RES	301
RES	302
445	303
HIP	304
446	305
HIP	306
447	307
696	308
HIP	309
697	310
698	311
700	312
1014	313
HIP	314
1015	315
HIP	316
1013	317
HIP	318
1012	319
RES	320
RES	321
RES	322
RES	323
RES	324
267	325
HIP	326
350	327
HIP	328
351	329
HIP	330
352	331
HIP	332
RES	333
RES	334
RES	335
469	336
500	337
HIP	338
234	339
HIP	340

IPA - AVy	Register
921	341
HIP	342
RES	343
RES	344
RES	345
RES	346
87	347
HIP	348
88	349
HIP	350
89	351
HIP	352
90	353
HIP	354
644	355
HIP	356
645	357
HIP	358
91	359
HIP	360
92	361
HIP	362
1022	363
HIP	364
902	365
HIP	366
93	367
HIP	368
94	369
HIP	370
95	371
HIP	372
96	373
HIP	374
97	375
HIP	376
98	377
HIP	378
1023	379
HIP	380
903	381

IPA - AVy	Register
HIP	382
99	383
HIP	384
100	385
HIP	386
RES	387
RES	388
RES	389
RES	390
RES	391
252	392
253	393
45	394
HIP	395
321	396
179	397
HIP	398
303	399
675	400
HIP	401
413	402
412	403
240	404
713	405
414	406
415	407
416	408
HIP	409
1146	410
178	411
169	412
HIP	413
1054	414
912	415
913	416
RES	417
RES	418
RES	419
RES	420
333	421
332	422

IPA - AVy	Register
334	423
802	424
HIP	425
331	426
300	427
RES	428
RES	429
RES	430
RES	431
50	432
HIP	433
51	434
HIP	435
52	436
HIP	437
54	438
53	439
359	440
360	441
368	442
1140	443
1148	444
365	445
354	446
639	447
634	448
636	449
637	450
386	451
728	452
737	453
RES	454
RES	455
RES	456
RES	457
319	458
408	459
85	460
HIP	461
RES	462
RES	463

IPA - AVy	Register
RES	464
RES	465
RES	466
66	467
62	468
HIP	469
67	470
63	471
HIP	472
68	473
64	474
HIP	475
69	476
65	477
HIP	478
70	479
71	480
72	481
HIP	482
73	483
HIP	484
792	485
1042	486
1043	487
1044	488
74	489
75	490
76	491
77	492
HIP	493
78	494
HIP	495
79	496
80	497
81	498
82	499
HIP	500
83	501
HIP	502
84	503
RES	504

IPA - AVy	Register
RES	505
RES	506
RES	507
145	508
146	509
147	510
148	511
149	512
150	513
151	514
152	515
629	516
RES	517
RES	518
RES	519
137	520
138	521
139	522
140	523
141	524
142	525
143	526
144	527
1143	528
1144	529
1145	530
RES	531
RES	532
RES	533
1020	534
1021	535
RES	536
RES	537
RES	538
RES	539
182	540
183	541
184	542
HIP	543
185	544
HIP	545

IPA - AVy	Register
186	546
HIP	547
187	548
HIP	549
188	550
HIP	551
189	552
HIP	553
190	554
HIP	555
191	556
HIP	557
192	558
HIP	559
193	560
HIP	561
RES	562
RES	563
1066	564
RES	565
RES	566
101	567
102	568
103	569
104	570
105	571
RES	572
RES	573
107	574
108	575
RES	576
RES	577
375	578
266	579
RES	580
RES	581
208	582
154	583
155	584
156	585
157	586

IPA - AVy	Register
158	587
159	588
160	589
RES	590
RES	591
RES	592
202	593
659	594
HIP	595
660	596
665	597
HIP	598
661	599
HIP	600
662	601
666	602
HIP	603
23	604
HIP	605
24	606
667	607
HIP	608
31	609
HIP	610
32	611
668	612
HIP	613
25	614
HIP	615
26	616
669	617
HIP	618
33	619
HIP	620
34	621
670	622
HIP	623
27	624
HIP	625
28	626
671	627

AVy : Register-Parameter Table (Functions 03, 04, 06, 16).

IPA - AVy	Register
HIP	628
35	629
HIP	630
36	631
672	632
HIP	633
RES	634
RES	635
RES	636
RES	637
RES	638
626	639
627	640
628	641
RES	642
1074	643
RES	644
RES	645
1017	646
1018	647
1019	648
HIP	649
943	650
941	651
942	652
RES	653
656	654
657	655
RES	656
655	657
RES	658
740	659
741	660
801	661
RES	662
1083	663
1082	664
1080	665
1081	666
1084	667
1087	668

IPA - AVy	Register
1085	669
HIP	670
1086	671
HIP	672
1089	673
1091	674
RES	675
RES	676
RES	677
58	678
59	679
HIP	680
60	681
HIP	682
61	683
HIP	684
RES	685
RES	686
RES	687
RES	688
RES	689
RES	690
RES	691
235	692
HIP	693
327	694
HIP	695
328	696
329	697
417	698
330	699
RES	700
RES	701
905	702
717	703
716	704
RES	705
484	706
485	707
486	708
HIP	709

IPA - AVy	Register
487	710
HIP	711
488	712
HIP	713
489	714
HIP	715
490	716
HIP	717
491	718
HIP	719
RES	720
RES	721
553	722
554	723
555	724
HIP	725
556	726
HIP	727
557	728
HIP	729
558	730
HIP	731
559	732
HIP	733
560	734
HIP	735
RES	736
RES	737
503	738
504	739
505	740
506	741
507	742
508	743
509	744
510	745
511	746
512	747
513	748
514	749
515	750

IPA - AVy	Register
516	751
517	752
518	753
RES	754
RES	755
RES	756
RES	757
519	758
536	759
RES	760
RES	761
1095	762
1096	763
1097	764
1098	765
1099	766
1100	767
1101	768
1102	769
1103	770
1104	771
1105	772
1106	773
RES	774
RES	775
RES	776
RES	777
RES	778
786	779
787	780
HIP	781
758	782
759	783
763	784
762	785
760	786
761	787
1046	788
HIP	789
1047	790
HIP	791

IPA - AVy	Register
757	792
765	793
HIP	794
764	795
HIP	796
695	797
731	798
793	799
HIP	800
734	801
HIP	802
779	803
776	804
HIP	805
777	806
HIP	807
778	808
HIP	809
784	810
HIP	811
785	812
HIP	813
771	814
418	815
768	816
HIP	817
766	818
HIP	819
788	820
HIP	821
789	822
HIP	823
790	824
HIP	825
791	826
HIP	827
767	828
HIP	829
421	830
774	831
782	832

IPA - AVy	Register
773	833
HIP	834
RES	835
RES	836
RES	837
795	838
796	839
797	840
HIP	841
798	842
799	843
RES	844
RES	845
RES	846
RES	847
RES	848
57	849
55	850
56	851
44	852
115	853
119	854
1	855
HIP	856
2	857
HIP	858
5	859
HIP	860
3	861
HIP	862
6	863
HIP	864
4	865
HIP	866
21	867
HIP	868
22	869
29	870
HIP	871
30	872
713	873

IPA - AVy	Register
37	874
HIP	875
38	876
54	877
53	878
50	879
HIP	880
51	881
HIP	882
52	883
HIP	884
45	885
46	886
116	887
120	888
RES	889
RES	890
RES	891
RES	892
RES	893
RES	894
589	895
590	896
592	897
593	898
HIP	899
594	900
HIP	901
595	902
HIP	903
596	904
598	905
599	906
600	907
601	908
603	909
604	910
HIP	911
605	912
HIP	913
606	914

IPA - AVy	Register
HIP	915
607	916
608	917
HIP	918
610	919
HIP	920
611	921
612	922
HIP	923
613	924
HIP	925
614	926
615	927
616	928
HIP	929
617	930
618	931
620	932
621	933
623	934
625	935
299	936
HIP	937
298	938
292	939
HIP	940
693	941
HIP	942
RES	943
RES	944
RES	945
RES	946
RES	947
RES	948
1107	949
1108	950
1109	951
1110	952
1111	953
1112	954
1113	955

AVy : Register-Parameter Table (Functions 03, 04, 06, 16).

IPA - AVy	Register
1114	956
1115	957
1116	958
1117	959
1118	960
1119	961
1120	962
1121	963
1122	964
1123	965
1124	966
1125	967
1126	968
1127	969
1128	970
1129	971
1130	972
1131	973
1132	974
1133	975
1134	976
1135	977
1136	978
1137	979
1138	980
RES	981
RES	982
RES	983
RES	984
RES	985
RES	986
RES	987
RES	988
RES	989
RES	990
RES	991
RES	992
RES	993
RES	994
RES	995
RES	996

IPA - AVy	Register
RES	997
RES	998
RES	999
RES	1000
RES	1001
RES	1002
RES	1003
RES	1004
RES	1005
RES	1006
RES	1007
RES	1008
RES	1009
RES	1010
RES	1011
RES	1012
RES	1013
RES	1014
RES	1015
RES	1016
RES	1017
RES	1018
RES	1019
RES	1020
RES	1021
RES	1022
RES	1023
RES	1024
RES	1025
RES	1026
RES	1027
RES	1028
RES	1029
RES	1030
RES	1031
RES	1032
RES	1033
RES	1034
RES	1035
RES	1036
RES	1037

IPA - AVy	Register
RES	1038
RES	1039
RES	1040
RES	1041
RES	1042
RES	1043
RES	1044
RES	1045
RES	1046
RES	1047
RES	1048
RES	1049
RES	1050
RES	1051
RES	1052
RES	1053
RES	1054
RES	1055
RES	1056
RES	1057
RES	1058
RES	1059
RES	1060
RES	1061
RES	1062
RES	1063
RES	1064
RES	1065
RES	1066
RES	1067
RES	1068
RES	1069
RES	1070
RES	1071
RES	1072
RES	1073
RES	1074
RES	1075
RES	1076
RES	1077
RES	1078

IPA - AVy	Register
RES	1079
RES	1080
RES	1081
RES	1082
RES	1083
RES	1084
RES	1085
RES	1086
RES	1087
RES	1088
RES	1089
RES	1090
RES	1091
RES	1092
RES	1093
RES	1094
RES	1095
RES	1096
RES	1097
RES	1098
RES	1099
RES	1100
RES	1101
RES	1102
RES	1103
RES	1104
RES	1105
RES	1106
RES	1107
RES	1108
RES	1109
RES	1110
RES	1111
RES	1112
RES	1113
RES	1114
RES	1115
RES	1116
RES	1117
RES	1118
RES	1119

IPA - AVy	Register
RES_	1120
RES_	1121
RES_	1122
RES_	1123
RES_	1124
RES_	1125
RES_	1126
RES_	1127
RES_	1128
RES_	1129
RES_	1130
RES_	1131
RES_	1132
RES_	1133
RES_	1134
RES_	1135
RES_	1136
RES_	1137
RES_	1138
RES_	1139
RES_	1140
RES_	1141
RES_	1142
RES_	1143
RES_	1144
RES_	1145
RES_	1146
RES_	1147
RES_	1148
RES_	1149
RES_	1150
RES_	1151
RES_	1152
RES_	1153
RES_	1154
RES_	1155
RES_	1156
RES_	1157
RES_	1158
RES_	1159
RES_	1160

IPA - AVy	Register
RES_	1161
RES_	1162
RES_	1163
RES_	1164
RES_	1165
RES_	1166
RES_	1167
RES_	1168
RES_	1169
RES_	1170
RES_	1171
RES_	1172
RES_	1173
RES_	1174
RES_	1175
RES_	1176
RES_	1177
RES_	1178
RES_	1179
RES_	1180
RES_	1181
RES_	1182
RES_	1183
RES_	1184
RES_	1185
RES_	1186
RES_	1187
RES_	1188
RES_	1189
RES_	1190
RES_	1191
RES_	1192
RES_	1193
RES_	1194
RES_	1195
RES_	1196
RES_	1197
RES_	1198
RES_	1199
RES_	1200
1289	1201

IPA - AVy	Register
HIP_	1202
1290	1203
HIP_	1204
1291	1205
1294	1206

IPA - AVy	Register

AVy : Register-Parameter Table (Functions 03, 04, 06, 16).

AVy : Register list according to a Parameter progressive order

AVy - Register list according to a Parameter progressive order

IPA - AVy	Register
1	209
1	855
2	211
2	857
3	215
3	861
4	219
4	865
5	213
5	859
6	217
6	863
7	221
8	222
9	223
10	224
11	225
13	226
18	244
19	245
20	251
21	235
21	867
22	237
22	869
23	604
24	606
25	614
26	616
27	624
28	626
29	238
29	870
30	240

IPA - AVy	Register
30	872
31	609
32	611
33	619
34	621
35	629
36	631
37	241
37	874
38	243
38	876
39	202
40	203
41	48
42	196
43	197
44	190
44	852
45	394
45	885
46	886
47	192
48	191
49	194
50	432
50	879
51	434
51	881
52	436
52	883
53	439
53	878
54	438
54	877

IPA - AVy	Register
55	850
56	851
57	849
58	678
59	679
60	681
61	683
62	468
63	471
64	474
65	477
66	467
67	470
68	473
69	476
70	479
71	480
72	481
73	483
74	489
75	490
76	491
77	492
78	494
79	496
80	497
81	498
82	499
83	501
84	503
85	460
87	347
88	349
89	118

IPA - AVy	Register
89	351
90	126
90	353
91	359
92	361
93	367
94	369
95	371
96	373
97	375
98	377
99	383
100	385
101	567
102	568
103	569
104	570
105	571
106	294
107	574
108	575
109	4
110	11
111	26
112	5
113	12
114	28
115	6
115	853
116	887
117	30
118	13
119	7
119	854

AVy - Register list according to a Parameter progressive order

IPA - AVy	Register
120	888
121	32
122	14
126	292
137	520
138	521
139	522
140	523
141	524
142	525
143	526
144	527
145	508
146	509
147	510
148	511
149	512
150	513
151	514
152	515
154	583
155	584
156	585
157	586
158	587
159	588
160	589
161	73
162	75
163	77
164	79
165	87
166	91
167	83
168	85
169	412
176	130
178	411
179	397
182	540
183	541

IPA - AVy	Register
184	542
185	544
186	546
187	548
188	550
189	552
190	554
191	556
192	558
193	560
197	17
198	18
202	593
204	19
206	21
208	582
227	37
229	38
230	46
231	44
233	40
234	53
234	339
235	692
236	287
240	404
251	97
252	392
253	393
266	579
267	325
292	939
298	938
299	936
300	427
303	399
319	458
321	396
324	42
327	694
328	696

IPA - AVy	Register
329	697
330	699
331	426
332	422
333	421
334	423
350	327
351	329
352	331
354	446
359	440
360	441
365	445
368	442
371	81
375	578
378	198
379	200
386	451
408	459
411	260
412	403
413	402
414	406
415	407
416	408
417	698
418	815
420	16
421	830
427	15
436	93
437	95
444	289
445	303
446	305
447	307
467	227
469	336
484	706
485	707

IPA - AVy	Register
486	708
487	710
488	712
489	714
490	716
491	718
500	337
503	738
504	739
505	740
506	741
507	742
508	743
509	744
510	745
511	746
512	747
513	748
514	749
515	750
516	751
517	752
518	753
519	758
536	759
553	722
554	723
555	724
556	726
557	728
558	730
559	732
560	734
564	58
581	59
582	60
583	61
589	895
590	896
592	897
593	898

IPA - AVy	Register
594	900
595	902
596	904
598	905
599	906
600	907
601	908
603	909
604	910
605	912
606	914
607	916
608	917
610	919
611	921
612	922
613	924
614	926
615	927
616	928
617	930
618	931
620	932
621	933
623	934
625	935
626	639
627	640
628	641
629	516
634	448
636	449
637	450
639	447
643	161
644	106
644	355
645	110
645	357
646	160
647	163

IPA - AVy	Register
655	657
656	654
657	655
659	594
660	596
661	599
662	601
663	247
664	249
665	597
666	602
667	607
668	612
669	617
670	622
671	627
672	632
673	252
675	400
682	124
683	104
684	116
685	108
686	112
687	120
688	128
689	132
690	135
691	136
692	134
693	941
695	797
696	308
697	310
698	311
700	312
705	101
709	166
710	167
711	168
712	165

IPA - AVy	Register
713	405
713	873
716	704
717	703
722	169
723	170
724	171
725	172
726	89
727	174
728	452
731	798
733	290
734	801
737	453
740	659
741	660
757	792
758	782
759	783
760	786
761	787
762	785
763	784
764	795
765	793
766	818
767	828
768	816
771	814
773	833
774	831
776	804
777	806
778	808
779	803
782	832
784	810
785	812
786	779
787	780

IPA - AVy	Register
788	820
789	822
790	824
791	826
792	485
793	799
795	838
796	839
797	840
798	842
799	843
801	661
802	424
881	55
889	228
892	295
893	176
894	178
895	180
896	181
897	182
900	183
901	184
902	365
903	381
905	702
907	138
908	140
909	144
912	415
913	416
914	57
921	341
923	9
924	23
925	8
926	51
927	49
928	50
941	651
942	652

AVy - Register list according to a Parameter progressive order

AVy - Register list according to a Parameter progressive order

IPA - AVy	Register
943	650
1012	319
1013	317
1014	313
1015	315
1016	288
1017	646
1018	647
1019	648
1020	534
1021	535
1022	363
1023	379
1024	142
1029	146
1030	148
1031	150
1032	152
1033	154
1042	486
1043	487
1044	488
1046	788
1047	790
1048	147
1054	414
1055	262
1056	266
1057	268
1058	272
1059	264
1060	270
1061	274
1062	258
1063	279
1064	281
1065	276
1066	564
1069	299
1070	297
1071	66

IPA - AVy	Register
1072	67
1073	68
1074	643
1078	296
1079	300
1080	665
1081	666
1082	664
1083	663
1084	667
1085	669
1086	671
1087	668
1089	673
1091	674
1095	762
1096	763
1097	764
1098	765
1099	766
1100	767
1101	768
1102	769
1103	770
1104	771
1105	772
1106	773
1107	949
1108	950
1109	951
1110	952
1111	953
1112	954
1113	955
1114	956
1115	957
1116	958
1117	959
1118	960
1119	961
1120	962

IPA - AVy	Register
1121	963
1122	964
1123	965
1124	966
1125	967
1126	968
1127	969
1128	970
1129	971
1130	972
1131	973
1132	974
1133	975
1134	976
1135	977
1136	978
1137	979
1138	980
1140	443
1143	528
1144	529
1145	530
1146	410
1147	56
1148	444
1289	1201
1290	1203
1291	1205
1294	1206
HIP	10
HIP	20
HIP	22
HIP	27
HIP	29
HIP	31
HIP	33
HIP	39
HIP	41
HIP	43
HIP	45
HIP	47

IPA - AVy	Register
HIP	52
HIP	54
HIP	74
HIP	76
HIP	78
HIP	80
HIP	82
HIP	84
HIP	86
HIP	88
HIP	90
HIP	92
HIP	94
HIP	96
HIP	105
HIP	107
HIP	109
HIP	111
HIP	113
HIP	117
HIP	119
HIP	121
HIP	125
HIP	127
HIP	129
HIP	131
HIP	133
HIP	137
HIP	139
HIP	141
HIP	143
HIP	145
HIP	149
HIP	151
HIP	153
HIP	155
HIP	162
HIP	164
HIP	173
HIP	175
HIP	177

IPA - AVy	Register
HIP_	179
HIP_	193
HIP_	195
HIP_	199
HIP_	201
HIP_	210
HIP_	212
HIP_	214
HIP_	216
HIP_	218
HIP_	220
HIP_	229
HIP_	236
HIP_	239
HIP_	242
HIP_	246
HIP_	248
HIP_	250
HIP_	259
HIP_	261
HIP_	263
HIP_	265
HIP_	267
HIP_	269
HIP_	271
HIP_	273
HIP_	275
HIP_	277
HIP_	280
HIP_	291
HIP_	293
HIP_	298
HIP_	304
HIP_	306
HIP_	309
HIP_	314
HIP_	316
HIP_	318
HIP_	326
HIP_	328
HIP_	330

IPA - AVy	Register
HIP_	332
HIP_	338
HIP_	340
HIP_	342
HIP_	348
HIP_	350
HIP_	352
HIP_	354
HIP_	356
HIP_	358
HIP_	360
HIP_	362
HIP_	364
HIP_	366
HIP_	368
HIP_	370
HIP_	372
HIP_	374
HIP_	376
HIP_	378
HIP_	380
HIP_	382
HIP_	384
HIP_	386
HIP_	395
HIP_	398
HIP_	401
HIP_	409
HIP_	413
HIP_	425
HIP_	433
HIP_	435
HIP_	437
HIP_	461
HIP_	469
HIP_	472
HIP_	475
HIP_	478
HIP_	482
HIP_	484
HIP_	493

IPA - AVy	Register
HIP_	495
HIP_	500
HIP_	502
HIP_	543
HIP_	545
HIP_	547
HIP_	549
HIP_	551
HIP_	553
HIP_	555
HIP_	557
HIP_	559
HIP_	561
HIP_	595
HIP_	598
HIP_	600
HIP_	603
HIP_	605
HIP_	608
HIP_	610
HIP_	613
HIP_	615
HIP_	618
HIP_	620
HIP_	623
HIP_	625
HIP_	628
HIP_	630
HIP_	633
HIP_	649
HIP_	670
HIP_	672
HIP_	680
HIP_	682
HIP_	684
HIP_	693
HIP_	695
HIP_	709
HIP_	711
HIP_	713
HIP_	715

IPA - AVy	Register
HIP_	717
HIP_	719
HIP_	725
HIP_	727
HIP_	729
HIP_	731
HIP_	733
HIP_	735
HIP_	781
HIP_	789
HIP_	791
HIP_	794
HIP_	796
HIP_	800
HIP_	802
HIP_	805
HIP_	807
HIP_	809
HIP_	811
HIP_	813
HIP_	817
HIP_	819
HIP_	821
HIP_	823
HIP_	825
HIP_	827
HIP_	829
HIP_	834
HIP_	841
HIP_	856
HIP_	858
HIP_	860
HIP_	862
HIP_	864
HIP_	866
HIP_	868
HIP_	871
HIP_	875
HIP_	880
HIP_	882
HIP_	884

AVy - Register list according to a Parameter progressive order

AVy - Register list according to a Parameter progressive order

IPA - AVy	Register
HIP	899
HIP	901
HIP	903
HIP	911
HIP	913
HIP	915
HIP	918
HIP	920
HIP	923
HIP	925
HIP	929
HIP	937
HIP	940
HIP	942
HIP	1202
HIP	1204
RES	0
RES	1
RES	2
RES	3
RES	24
RES	25
RES	34
RES	35
RES	36
RES	62
RES	63
RES	64
RES	65
RES	69
RES	70
RES	71
RES	72
RES	98
RES	99
RES	100
RES	102
RES	103
RES	114
RES	115
RES	122

IPA - AVy	Register
RES	123
RES	156
RES	157
RES	158
RES	159
RES	185
RES	186
RES	187
RES	188
RES	189
RES	204
RES	205
RES	206
RES	207
RES	208
RES	230
RES	231
RES	232
RES	233
RES	234
RES	253
RES	254
RES	255
RES	256
RES	257
RES	278
RES	282
RES	283
RES	284
RES	285
RES	286
RES	301
RES	302
RES	320
RES	321
RES	322
RES	323
RES	324
RES	333
RES	334
RES	335

IPA - AVy	Register
RES	343
RES	344
RES	345
RES	346
RES	387
RES	388
RES	389
RES	390
RES	391
RES	417
RES	418
RES	419
RES	420
RES	428
RES	429
RES	430
RES	431
RES	454
RES	455
RES	456
RES	457
RES	462
RES	463
RES	464
RES	465
RES	466
RES	504
RES	505
RES	506
RES	507
RES	517
RES	518
RES	519
RES	531
RES	532
RES	533
RES	536
RES	537
RES	538
RES	539
RES	562

IPA - AVy	Register
RES	563
RES	565
RES	566
RES	572
RES	573
RES	576
RES	577
RES	580
RES	581
RES	590
RES	591
RES	592
RES	634
RES	635
RES	636
RES	637
RES	638
RES	642
RES	644
RES	645
RES	653
RES	656
RES	658
RES	662
RES	675
RES	676
RES	677
RES	685
RES	686
RES	687
RES	688
RES	689
RES	690
RES	691
RES	700
RES	701
RES	705
RES	720
RES	721
RES	736
RES	737

IPA - AVy	Register
RES	754
RES	755
RES	756
RES	757
RES	760
RES	761
RES	774
RES	775
RES	776
RES	777
RES	778
RES	835
RES	836
RES	837
RES	844
RES	845
RES	846
RES	847
RES	848
RES	889
RES	890
RES	891
RES	892
RES	893
RES	894
RES	943
RES	944
RES	945
RES	946
RES	947
RES	948
RES	981
RES	982
RES	983
RES	984
RES	985
RES	986
RES	987
RES	988
RES	989
RES	990

IPA - AVy	Register
RES	991
RES	992
RES	993
RES	994
RES	995
RES	996
RES	997
RES	998
RES	999
RES	1000
RES	1001
RES	1002
RES	1003
RES	1004
RES	1005
RES	1006
RES	1007
RES	1008
RES	1009
RES	1010
RES	1011
RES	1012
RES	1013
RES	1014
RES	1015
RES	1016
RES	1017
RES	1018
RES	1019
RES	1020
RES	1021
RES	1022
RES	1023
RES	1024
RES	1025
RES	1026
RES	1027
RES	1028
RES	1029
RES	1030
RES	1031

IPA - AVy	Register
RES	1032
RES	1033
RES	1034
RES	1035
RES	1036
RES	1037
RES	1038
RES	1039
RES	1040
RES	1041
RES	1042
RES	1043
RES	1044
RES	1045
RES	1046
RES	1047
RES	1048
RES	1049
RES	1050
RES	1051
RES	1052
RES	1053
RES	1054
RES	1055
RES	1056
RES	1057
RES	1058
RES	1059
RES	1060
RES	1061
RES	1062
RES	1063
RES	1064
RES	1065
RES	1066
RES	1067
RES	1068
RES	1069
RES	1070
RES	1071
RES	1072

IPA - AVy	Register
RES	1073
RES	1074
RES	1075
RES	1076
RES	1077
RES	1078
RES	1079
RES	1080
RES	1081
RES	1082
RES	1083
RES	1084
RES	1085
RES	1086
RES	1087
RES	1088
RES	1089
RES	1090
RES	1091
RES	1092
RES	1093
RES	1094
RES	1095
RES	1096
RES	1097
RES	1098
RES	1099
RES	1100
RES	1101
RES	1102
RES	1103
RES	1104
RES	1105
RES	1106
RES	1107
RES	1108
RES	1109
RES	1110
RES	1111
RES	1112
RES	1113

AVy - Register list according to a Parameter progressive order

AVy - Register list according to a Parameter progressive order

IPA - AVy	Register
RES	1114
RES	1115
RES	1116
RES	1117
RES	1118
RES	1119
RES	1120
RES	1121
RES	1122
RES	1123
RES	1124
RES	1125
RES	1126
RES	1127
RES	1128
RES	1129
RES	1130
RES	1131
RES	1132
RES	1133
RES	1134
RES	1135
RES	1136
RES	1137
RES	1138
RES	1139
RES	1140
RES	1141
RES	1142
RES	1143
RES	1144
RES	1145
RES	1146
RES	1147
RES	1148
RES	1149
RES	1150
RES	1151
RES	1152
RES	1153
RES	1154

IPA - AVy	Register
RES	1155
RES	1156
RES	1157
RES	1158
RES	1159
RES	1160
RES	1161
RES	1162
RES	1163
RES	1164
RES	1165
RES	1166
RES	1167
RES	1168
RES	1169
RES	1170
RES	1171
RES	1172
RES	1173
RES	1174
RES	1175
RES	1176
RES	1177
RES	1178
RES	1179
RES	1180
RES	1181
RES	1182
RES	1183
RES	1184
RES	1185
RES	1186
RES	1187
RES	1188
RES	1189
RES	1190
RES	1191
RES	1192
RES	1193
RES	1194
RES	1195

IPA - AVy	Register
RES	1196
RES	1197
RES	1198
RES	1199
RES	1200

IPA - AVy	Register

AVy : Coil Table (Functions 01, 02, 05, 15)

IPA - AVy	Register	IPA - AVy	Register	IPA - AVy	Register	IPA - AVy	Register
RES	0	679	34	123	68	1149	102
314	1	680	35	124	69	1150	103
315	2	681	36	125	70	1151	104
316	3	_RES_	37	422	71	366	105
380	4	1027	38	890	72	367	106
343	5	1028	39	891	73	355	107
RES	6	_RES_	40	1068	74	356	108
RES	7	_RES_	41	699	75	363	109
RES	8	898	42	_RES_	76	364	110
RES	9	899	43	_RES_	77	210	111
565	10	_RES_	44	353	78	211	112
566	11	_RES_	45	_RES_	79	640	113
567	12	372	46	_RES_	80	633	114
568	13	715	47	_RES_	81	635	115
569	14	349	48	_RES_	82	387	116
570	15	342	49	_RES_	83	729	117
571	16	_RES_	50	648	84	730	118
572	17	_RES_	51	651	85	738	119
573	18	_RES_	52	649	86	749	120
574	19	245	53	652	87	_RES_	121
575	20	293	54	911	88	_RES_	122
576	21	294	55	_RES_	89	_RES_	123
577	22	344	56	_RES_	90	_RES_	124
578	23	345	57	_RES_	91	_RES_	125
579	24	373	58	_RES_	92	295	126
580	25	346	59	357	93	389	127
RES	26	347	60	358	94	259	128
RES	27	_RES_	61	361	95	1045	129
RES	28	_RES_	62	362	96	296	130
RES	29	_RES_	63	369	97	390	131
694	30	242	64	370	98	260	132
676	31	322	65	1152	99	297	133
677	32	348	66	1141	100	391	134
678	33	1067	67	1142	101	261	135

AVy : Coil list according to a Parameter progressive order

IPA - AVy	Register	IPA - AVy	Register	IPA - AVy	Register	IPA - AVy	Register
123	68	346	59	402	162	546	223
124	69	347	60	403	165	547	224
125	70	348	66	404	166	548	225
153	159	349	48	406	174	549	226
181	143	353	78	407	176	550	227
210	111	355	107	422	71	551	228
211	112	356	108	425	236	552	229
242	64	357	93	435	186	561	195
243	164	358	94	492	194	565	10
244	155	361	95	520	198	566	11
245	53	362	96	521	199	567	12
246	150	363	109	522	200	568	13
248	151	364	110	523	201	569	14
249	154	366	105	524	202	570	15
256	187	367	106	525	203	571	16
258	188	369	97	526	204	572	17
259	128	370	98	527	205	573	18
260	132	372	46	528	206	574	19
261	135	373	58	529	207	575	20
262	189	380	4	530	208	576	21
263	190	387	116	531	209	577	22
293	54	388	142	532	210	578	23
294	55	389	127	533	211	579	24
295	126	390	131	534	212	580	25
296	130	391	134	535	213	591	252
297	133	393	145	537	214	602	253
314	1	394	146	538	215	609	254
315	2	395	147	539	216	619	255
316	3	396	152	540	217	622	256
322	65	397	153	541	218	624	257
342	49	398	156	542	219	630	168
343	5	399	157	543	220	633	114
344	56	400	160	544	221	635	115
345	57	401	161	545	222	640	113

TPD32 : Register-Parameter Table (Functions 03, 04, 06, 16).

IPA - TPD32	Register	IPA - TPD32	Register	IPA - TPD32	Register	IPA - TPD32	Register
RES	0	1052	34	1073	68	_RES_	102
RES	1	_RES_	35	_RES_	69	_RES_	103
RES	2	199	36	_RES_	70	683	104
RES	3	227	37	_RES_	71	_RES_	105
109	4	229	38	_RES_	72	644	106
112	5	_RES_	39	161	73	_RES_	107
115	6	233	40	_RES_	74	685	108
119	7	_RES_	41	162	75	_RES_	109
925	8	324	42	_RES_	76	645	110
923	9	_RES_	43	163	77	_RES_	111
RES	10	231	44	_RES_	78	686	112
110	11	_RES_	45	164	79	_RES_	113
113	12	230	46	_RES_	80	_RES_	114
118	13	_RES_	47	371	81	_RES_	115
122	14	41	48	_RES_	82	684	116
427	15	927	49	167	83	_RES_	117
420	16	928	50	_RES_	84	89	118
197	17	926	51	168	85	_RES_	119
198	18	_RES_	52	_RES_	86	687	120
204	19	234	53	165	87	_RES_	121
RES	20	_RES_	54	_RES_	88	_RES_	122
206	21	881	55	726	89	_RES_	123
RES	22	1147	56	_RES_	90	682	124
924	23	914	57	166	91	_RES_	125
RES	24	564	58	_RES_	92	90	126
466	25	581	59	436	93	_RES_	127
111	26	582	60	_RES_	94	688	128
RES	27	583	61	437	95	_RES_	129
114	28	588	62	_RES_	96	176	130
RES	29	_RES_	63	251	97	_RES_	131
117	30	_RES_	64	_RES_	98	689	132
RES	31	_RES_	65	_RES_	99	_RES_	133
121	32	1071	66	_RES_	100	692	134
RES	33	1072	67	705	101	690	135

TPD32 : Register-Parameter Table (Functions 03, 04, 06, 16).

IPA - TPD32	Register
691	136
RES	137
907	138
RES	139
908	140
RES	141
1024	142
RES	143
909	144
RES	145
1029	146
1048	147
1030	148
RES	149
1031	150
RES	151
1032	152
RES	153
1033	154
RES	155
RES	156
RES	157
RES	158
RES	159
646	160
643	161
RES	162
647	163
RES	164
712	165
709	166
710	167
711	168
722	169
723	170
724	171
725	172
RES	173
727	174
RES	175
893	176

IPA - TPD32	Register
RES	177
894	178
RES	179
895	180
896	181
897	182
900	183
901	184
RES	185
RES	186
RES	187
RES	188
RES	189
44	190
48	191
47	192
RES	193
49	194
RES	195
42	196
43	197
378	198
RES	199
379	200
RES	201
39	202
40	203
RES	204
RES	205
RES	206
RES	207
RES	208
1	209
RES	210
2	211
RES	212
5	213
RES	214
3	215
RES	216
6	217

IPA - TPD32	Register
RES	218
4	219
RES	220
7	221
8	222
9	223
10	224
11	225
13	226
467	227
889	228
RES	229
468	230
RES	231
RES	232
RES	233
RES	234
21	235
RES	236
22	237
29	238
RES	239
30	240
37	241
RES	242
38	243
18	244
19	245
RES	246
663	247
RES	248
664	249
RES	250
20	251
673	252
RES	253
RES	254
RES	255
RES	256
RES	257
1062	258

IPA - TPD32	Register
RES	259
411	260
RES	261
1055	262
RES	263
1059	264
RES	265
1056	266
RES	267
1057	268
RES	269
1060	270
RES	271
1058	272
RES	273
1061	274
RES	275
1065	276
RES	277
RES	278
1063	279
RES	280
1064	281
RES	282
RES	283
RES	284
RES	285
RES	286
236	287
1016	288
444	289
733	290
RES	291
126	292
RES	293
106	294
892	295
1078	296
1070	297
RES	298
1069	299

IPA - TPD32	Register
1079	300
RES	301
RES	302
445	303
RES	304
446	305
RES	306
447	307
696	308
RES	309
697	310
698	311
700	312
1014	313
RES	314
1015	315
RES	316
1013	317
RES	318
1012	319
RES	320
RES	321
RES	322
RES	323
RES	324
267	325
RES	326
350	327
RES	328
351	329
RES	330
352	331
RES	332
RES	333
RES	334
RES	335
469	336
500	337
RES	338
234	339
RES	340

IPA - TPD32	Register
921	341
RES	342
175	343
RES	344
RES	345
RES	346
87	347
RES	348
88	349
RES	350
89	351
RES	352
90	353
RES	354
644	355
RES	356
645	357
RES	358
91	359
RES	360
92	361
RES	362
493	363
RES	364
494	365
RES	366
93	367
RES	368
94	369
RES	370
95	371
RES	372
96	373
RES	374
97	375
RES	376
98	377
RES	378
495	379
RES	380
496	381

IPA - TPD32	Register
RES	382
99	383
RES	384
100	385
RES	386
459	387
RES	388
460	389
RES	390
RES	391
252	392
253	393
45	394
RES	395
321	396
179	397
RES	398
303	399
675	400
RES	401
413	402
412	403
240	404
713	405
414	406
415	407
416	408
RES	409
1146	410
178	411
169	412
RES	413
1054	414
912	415
913	416
RES	417
RES	418
RES	419
RES	420
333	421
332	422

IPA - TPD32	Register
334	423
802	424
RES	425
331	426
300	427
RES	428
RES	429
RES	430
RES	431
50	432
RES	433
51	434
RES	435
52	436
RES	437
54	438
53	439
359	440
360	441
368	442
1140	443
1148	444
365	445
354	446
639	447
634	448
636	449
637	450
386	451
728	452
737	453
RES	454
RES	455
RES	456
RES	457
319	458
408	459
85	460
RES	461
RES	462
RES	463

TPD32 : Register-Parameter Table (Functions 03, 04, 06, 16).

TPD32 : Register-Parameter Table (Functions 03, 04, 06, 16).

IPA - TPD32	Register
RES	464
RES	465
RES	466
66	467
62	468
RES	469
67	470
63	471
RES	472
68	473
64	474
RES	475
69	476
65	477
RES	478
70	479
71	480
72	481
RES	482
73	483
RES	484
792	485
1042	486
1043	487
1044	488
74	489
75	490
76	491
77	492
RES	493
78	494
RES	495
79	496
80	497
81	498
82	499
RES	500
83	501
RES	502
84	503
RES	504

IPA - TPD32	Register
RES	505
RES	506
RES	507
145	508
146	509
147	510
148	511
149	512
150	513
151	514
152	515
629	516
RES	517
RES	518
RES	519
137	520
138	521
139	522
140	523
141	524
142	525
143	526
144	527
1143	528
1144	529
1145	530
RES	531
RES	532
RES	533
1020	534
1021	535
RES	536
RES	537
RES	538
RES	539
182	540
183	541
184	542
RES	543
185	544
RES	545

IPA - TPD32	Register
186	546
RES	547
187	548
RES	549
188	550
RES	551
189	552
RES	553
190	554
RES	555
191	556
RES	557
192	558
RES	559
193	560
RES	561
RES	562
RES	563
1066	564
RES	565
RES	566
101	567
102	568
103	569
104	570
105	571
RES	572
RES	573
107	574
108	575
RES	576
RES	577
375	578
266	579
RES	580
RES	581
208	582
154	583
155	584
156	585
157	586

IPA - TPD32	Register
158	587
159	588
160	589
RES	590
RES	591
RES	592
202	593
659	594
RES	595
660	596
665	597
RES	598
661	599
RES	600
662	601
666	602
RES	603
23	604
RES	605
24	606
667	607
RES	608
31	609
RES	610
32	611
668	612
RES	613
25	614
RES	615
26	616
669	617
RES	618
33	619
RES	620
34	621
670	622
RES	623
27	624
RES	625
28	626
671	627

IPA - TPD32	Register
RES	628
35	629
RES	630
36	631
672	632
RES	633
RES	634
RES	635
RES	636
RES	637
RES	638
626	639
627	640
628	641
RES	642
1074	643
RES	644
RES	645
1017	646
1018	647
1019	648
RES	649
943	650
941	651
942	652
RES	653
656	654
657	655
RES	656
655	657
RES	658
740	659
741	660
801	661
RES	662
1083	663
1082	664
1080	665
1081	666
1084	667
1087	668

IPA - TPD32	Register
1085	669
RES	670
1086	671
RES	672
1089	673
1091	674
RES	675
RES	676
RES	677
58	678
59	679
RES	680
60	681
RES	682
61	683
RES	684
RES	685
RES	686
RES	687
RES	688
RES	689
RES	690
RES	691
235	692
RES	693
327	694
RES	695
328	696
329	697
417	698
330	699
RES	700
RES	701
905	702
717	703
716	704
RES	705
484	706
485	707
486	708
RES	709

IPA - TPD32	Register
487	710
RES	711
488	712
RES	713
489	714
RES	715
490	716
RES	717
491	718
RES	719
RES	720
RES	721
553	722
554	723
555	724
RES	725
556	726
RES	727
557	728
RES	729
558	730
RES	731
559	732
RES	733
560	734
RES	735
RES	736
RES	737
503	738
504	739
505	740
506	741
507	742
508	743
509	744
510	745
511	746
512	747
513	748
514	749
515	750

IPA - TPD32	Register
516	751
517	752
518	753
RES	754
RES	755
RES	756
RES	757
519	758
536	759
RES	760
RES	761
1095	762
1096	763
1097	764
1098	765
1099	766
1100	767
1101	768
1102	769
1103	770
1104	771
1105	772
1106	773
RES	774
RES	775
RES	776
RES	777
RES	778
786	779
787	780
RES	781
758	782
759	783
763	784
762	785
760	786
761	787
1046	788
RES	789
1047	790
RES	791

TPD32 : Register-Parameter Table (Functions 03, 04, 06, 16).

TPD32 : Register-Parameter Table (Functions 03, 04, 06, 16).

IPA - TPD32	Register
757	792
765	793
RES	794
764	795
RES	796
695	797
731	798
793	799
RES	800
734	801
RES	802
779	803
776	804
RES	805
777	806
RES	807
778	808
RES	809
784	810
RES	811
785	812
RES	813
771	814
418	815
768	816
RES	817
766	818
RES	819
788	820
RES	821
789	822
RES	823
790	824
RES	825
791	826
RES	827
767	828
RES	829
421	830
774	831
782	832

IPA - TPD32	Register
773	833
RES	834
RES	835
RES	836
RES	837
795	838
796	839
797	840
RES	841
798	842
799	843
RES	844
RES	845
RES	846
RES	847
RES	848
57	849
55	850
56	851
44	852
115	853
119	854
1	855
RES	856
2	857
RES	858
5	859
RES	860
3	861
RES	862
6	863
RES	864
4	865
RES	866
21	867
RES	868
22	869
29	870
RES	871
30	872
713	873

IPA - TPD32	Register
37	874
RES	875
38	876
54	877
53	878
50	879
RES	880
51	881
RES	882
52	883
RES	884
45	885
46	886
116	887
120	888
RES	889
RES	890
RES	891
RES	892
RES	893
RES	894
589	895
590	896
592	897
593	898
RES	899
594	900
RES	901
595	902
RES	903
596	904
598	905
599	906
600	907
601	908
603	909
604	910
RES	911
605	912
RES	913
606	914

IPA - TPD32	Register
RES	915
607	916
608	917
RES	918
610	919
RES	920
611	921
612	922
RES	923
613	924
RES	925
614	926
615	927
616	928
RES	929
617	930
618	931
620	932
621	933
623	934
625	935
299	936
RES	937
298	938
292	939
RES	940
693	941
RES	942
RES	943
RES	944
RES	945
RES	946
RES	947
RES	948
1107	949
1108	950
1109	951
1110	952
1111	953
1112	954
1113	955

IPA - TPD32	Register
1114	956
1115	957
1116	958
1117	959
1118	960
1119	961
1120	962
1121	963
1122	964
1123	965
1124	966
1125	967
1126	968
1127	969
1128	970
1129	971
1130	972
1131	973
1132	974
1133	975
1134	976
1135	977
1136	978
1137	979
1138	980
RES	981
RES	982
453	983
HIP	984
454	985
HIP	986
587	987
916	988
HIP	989
917	990
HIP	991
918	992
HIP	993
374	994
HIP	995
280	996

IPA - TPD32	Register
HIP	997
562	998
HIP	999
563	1000
HIP	1001
456	1002
455	1003
RES	1004
RES	1005
RES	1006
RES	1007
465	1008
502	1009
501	1010
584	1011
212	1012
586	1013
585	1014
473	1015
475	1016
474	1017
478	1018
480	1019
481	1020
359	1021
203	1022
482	1023
483	1024
RES	1025
RES	1026
RES	1027
RES	1028
309	1029
318	1030
312	1031
313	1032
310	1033
311	1034
RES	1035
RES	1036
751	1037

IPA - TPD32	Register
752	1038
753	1039
754	1040
755	1041
756	1042
1218	1043
1219	1044
1220	1045
HIP	1046
1221	1047
HIP	1048
1222	1049
HIP	1050
1223	1051
HIP	1052
1224	1053
HIP	1054
1225	1055
HIP	1056
1227	1057
1228	1058
1229	1059
HIP	1060
1230	1061
HIP	1062
1231	1063
HIP	1064
1232	1065
HIP	1066
1233	1067
HIP	1068
1234	1069
HIP	1070
1236	1071
1237	1072
1238	1073
HIP	1074
1239	1075
HIP	1076
1240	1077
HIP	1078

IPA - TPD32	Register
1241	1079
HIP	1080
1242	1081
HIP	1082
1243	1083
HIP	1084
1245	1085
1246	1086
1247	1087
HIP	1088
1248	1089
HIP	1090
1249	1091
HIP	1092
1250	1093
HIP	1094
1251	1095
HIP	1096
1252	1097
HIP	1098
RES	1099
RES	1100
RES	1101
RES	1102
RES	1103
RES	1104
1254	1105
HIP	1106
1154	1107
HIP	1108
1160	1109
HIP	1110
1153	1111
HIP	1112
1204	1113
1156	1114
1163	1115
1155	1116
1162	1117
1206	1118
1207	1119

TPD32 : Register-Parameter Table (Functions 03, 04, 06, 16).

TPD32 : Register-Parameter Table (Functions 03, 04, 06, 16).

IPA - TPD32	Register
1158	1120
1168	1121
1164	1122
1165	1123
1166	1124
1167	1125
RES	1126
RES	1127
1180	1128
HIP	1129
1181	1130
1194	1131
HIP	1132
1193	1133
HIP	1134
RES	1135
RES	1136
1182	1137
HIP	1138
1212	1139
1184	1140
HIP	1141
1185	1142
HIP	1143
1186	1144
HIP	1145
1171	1146
HIP	1147
1172	1148
HIP	1149
1192	1150
HIP	1151
1191	1152
HIP	1153
1173	1154
HIP	1155
1174	1156
HIP	1157
1175	1158
HIP	1159
1213	1160

IPA - TPD32	Register
1208	1161
1177	1162
HIP	1163
1178	1164
HIP	1165
1180	1166
HIP	1167
1179	1168
HIP	1169
1202	1170
1200	1171
1196	1172
HIP	1173
1197	1174
HIP	1175
1216	1176
1199	1177
1198	1178
HIP	1179
1210	1180
1217	1181
1255	1182
1262	1183
1263	1184
1266	1185
RES	1186
RES	1187
RES	1188
RES	1189
1284	1190
1285	1191
1286	1192
HIP	1193
RES	1194
RES	1195
RES	1196
RES	1197
RES	1198
RES	1199
RES	1200
RES	1201

IPA - TPD32	Register
RES	1202
RES	1203
RES	1204
RES	1205
RES	1206

IPA - TPD32	Register
-------------	----------

TPD32 : Register list according to a Parameter progressive order

TPD32 : Register list according to a Parameter progressive order

IPA - TPD32	Register
1	209
1	855
2	211
2	857
3	215
3	861
4	219
4	865
5	213
5	859
6	217
6	863
7	221
8	222
9	223
10	224
11	225
13	226
18	244
19	245
20	251
21	235
21	867
22	237
22	869
23	604
24	606
25	614
26	616
27	624
28	626
29	238
29	870
30	240

IPA - TPD32	Register
30	872
31	609
32	611
33	619
34	621
35	629
36	631
37	241
37	874
38	243
38	876
39	202
40	203
41	48
42	196
43	197
44	190
44	852
45	394
45	885
46	886
47	192
48	191
49	194
50	432
50	879
51	434
51	881
52	436
52	883
53	439
53	878
54	438
54	877

IPA - TPD32	Register
55	850
56	851
57	849
58	678
59	679
60	681
61	683
62	468
63	471
64	474
65	477
66	467
67	470
68	473
69	476
70	479
71	480
72	481
73	483
74	489
75	490
76	491
77	492
78	494
79	496
80	497
81	498
82	499
83	501
84	503
85	460
87	347
88	349
89	118

IPA - TPD32	Register
89	351
90	126
90	353
91	359
92	361
93	367
94	369
95	371
96	373
97	375
98	377
99	383
100	385
101	567
102	568
103	569
104	570
105	571
106	294
107	574
108	575
109	4
110	11
111	26
112	5
113	12
114	28
115	6
115	853
116	887
117	30
118	13
119	7
119	854

TPD32 : Register list according to a Parameter progressive order

IPA - TPD32	Register
120	888
121	32
122	14
126	292
137	520
138	521
139	522
140	523
141	524
142	525
143	526
144	527
145	508
146	509
147	510
148	511
149	512
150	513
151	514
152	515
154	583
155	584
156	585
157	586
158	587
159	588
160	589
161	73
162	75
163	77
164	79
165	87
166	91
167	83
168	85
169	412
175	343
176	130
178	411
179	397
182	540

IPA - TPD32	Register
183	541
184	542
185	544
186	546
187	548
188	550
189	552
190	554
191	556
192	558
193	560
197	17
198	18
199	36
202	593
203	1022
204	19
206	21
208	582
212	1012
227	37
229	38
230	46
231	44
233	40
234	53
234	339
235	692
236	287
240	404
251	97
252	392
253	393
266	579
267	325
280	996
292	939
298	938
299	936
300	427
303	399

IPA - TPD32	Register
309	1029
310	1033
311	1034
312	1031
313	1032
318	1030
319	458
321	396
324	42
327	694
328	696
329	697
330	699
331	426
332	422
333	421
334	423
350	327
351	329
352	331
354	446
359	440
359	1021
360	441
365	445
368	442
371	81
374	994
375	578
378	198
379	200
386	451
408	459
411	260
412	403
413	402
414	406
415	407
416	408
417	698
418	815

IPA - TPD32	Register
420	16
421	830
427	15
436	93
437	95
444	289
445	303
446	305
447	307
453	983
454	985
455	1003
456	1002
459	387
460	389
465	1008
466	25
467	227
468	230
469	336
473	1015
474	1017
475	1016
478	1018
480	1019
481	1020
482	1023
483	1024
484	706
485	707
486	708
487	710
488	712
489	714
490	716
491	718
493	363
494	365
495	379
496	381
500	337

IPA - TPD32	Register
501	1010
502	1009
503	738
504	739
505	740
506	741
507	742
508	743
509	744
510	745
511	746
512	747
513	748
514	749
515	750
516	751
517	752
518	753
519	758
536	759
553	722
554	723
555	724
556	726
557	728
558	730
559	732
560	734
562	998
563	1000
564	58
581	59
582	60
583	61
584	1011
585	1014
586	1013
587	987
588	62
589	895
590	896

IPA - TPD32	Register
592	897
593	898
594	900
595	902
596	904
598	905
599	906
600	907
601	908
603	909
604	910
605	912
606	914
607	916
608	917
610	919
611	921
612	922
613	924
614	926
615	927
616	928
617	930
618	931
620	932
621	933
623	934
625	935
626	639
627	640
628	641
629	516
634	448
636	449
637	450
639	447
643	161
644	106
644	355
645	110
645	357

IPA - TPD32	Register
646	160
647	163
655	657
656	654
657	655
659	594
660	596
661	599
662	601
663	247
664	249
665	597
666	602
667	607
668	612
669	617
670	622
671	627
672	632
673	252
675	400
682	124
683	104
684	116
685	108
686	112
687	120
688	128
689	132
690	135
691	136
692	134
693	941
695	797
696	308
697	310
698	311
700	312
705	101
709	166
710	167

IPA - TPD32	Register
711	168
712	165
713	405
713	873
716	704
717	703
722	169
723	170
724	171
725	172
726	89
727	174
728	452
731	798
733	290
734	801
737	453
740	659
741	660
751	1037
752	1038
753	1039
754	1040
755	1041
756	1042
757	792
758	782
759	783
760	786
761	787
762	785
763	784
764	795
765	793
766	818
767	828
768	816
771	814
773	833
774	831
776	804

TPD32 : Register list according to a Parameter progressive order

TPD32 : Register list according to a Parameter progressive order

IPA - TPD32	Register
777	806
778	808
779	803
782	832
784	810
785	812
786	779
787	780
788	820
789	822
790	824
791	826
792	485
793	799
795	838
796	839
797	840
798	842
799	843
801	661
802	424
881	55
889	228
892	295
893	176
894	178
895	180
896	181
897	182
900	183
901	184
905	702
907	138
908	140
909	144
912	415
913	416
914	57
916	988
917	990
918	992

IPA - TPD32	Register
921	341
923	9
924	23
925	8
926	51
927	49
928	50
941	651
942	652
943	650
1012	319
1013	317
1014	313
1015	315
1016	288
1017	646
1018	647
1019	648
1020	534
1021	535
1024	142
1029	146
1030	148
1031	150
1032	152
1033	154
1042	486
1043	487
1044	488
1046	788
1047	790
1048	147
1052	34
1054	414
1055	262
1056	266
1057	268
1058	272
1059	264
1060	270
1061	274

IPA - TPD32	Register
1062	258
1063	279
1064	281
1065	276
1066	564
1069	299
1070	297
1071	66
1072	67
1073	68
1074	643
1078	296
1079	300
1080	665
1081	666
1082	664
1083	663
1084	667
1085	669
1086	671
1087	668
1089	673
1091	674
1095	762
1096	763
1097	764
1098	765
1099	766
1100	767
1101	768
1102	769
1103	770
1104	771
1105	772
1106	773
1107	949
1108	950
1109	951
1110	952
1111	953
1112	954

IPA - TPD32	Register
1113	955
1114	956
1115	957
1116	958
1117	959
1118	960
1119	961
1120	962
1121	963
1122	964
1123	965
1124	966
1125	967
1126	968
1127	969
1128	970
1129	971
1130	972
1131	973
1132	974
1133	975
1134	976
1135	977
1136	978
1137	979
1138	980
1140	443
1143	528
1144	529
1145	530
1146	410
1147	56
1148	444
1153	1111
1154	1107
1155	1116
1156	1114
1158	1120
1160	1109
1162	1117
1163	1115

IPA - TPD32	Register
1164	1122
1165	1123
1166	1124
1167	1125
1168	1121
1171	1146
1172	1148
1173	1154
1174	1156
1175	1158
1177	1162
1178	1164
1179	1168
1180	1128
1180	1166
1181	1130
1182	1137
1184	1140
1185	1142
1186	1144
1191	1152
1192	1150
1193	1133
1194	1131
1196	1172
1197	1174
1198	1178
1199	1177
1200	1171
1202	1170
1204	1113
1206	1118
1207	1119
1208	1161
1210	1180
1212	1139
1213	1160
1216	1176
1217	1181
1218	1043
1219	1044

IPA - TPD32	Register
1220	1045
1221	1047
1222	1049
1223	1051
1224	1053
1225	1055
1227	1057
1228	1058
1229	1059
1230	1061
1231	1063
1232	1065
1233	1067
1234	1069
1236	1071
1237	1072
1238	1073
1239	1075
1240	1077
1241	1079
1242	1081
1243	1083
1245	1085
1246	1086
1247	1087
1248	1089
1249	1091
1250	1093
1251	1095
1252	1097
1254	1105
1255	1182
1262	1183
1263	1184
1266	1185
1284	1190
1285	1191
1286	1192
HIP	984
HIP	986
HIP	989

IPA - TPD32	Register
HIP	991
HIP	993
HIP	995
HIP	997
HIP	999
HIP	1001
HIP	1046
HIP	1048
HIP	1050
HIP	1052
HIP	1054
HIP	1056
HIP	1060
HIP	1062
HIP	1064
HIP	1066
HIP	1068
HIP	1070
HIP	1074
HIP	1076
HIP	1078
HIP	1080
HIP	1082
HIP	1084
HIP	1088
HIP	1090
HIP	1092
HIP	1094
HIP	1096
HIP	1098
HIP	1106
HIP	1108
HIP	1110
HIP	1112
HIP	1129
HIP	1132
HIP	1134
HIP	1138
HIP	1141
HIP	1143
HIP	1145

IPA - TPD32	Register
HIP	1147
HIP	1149
HIP	1151
HIP	1153
HIP	1155
HIP	1157
HIP	1159
HIP	1163
HIP	1165
HIP	1167
HIP	1169
HIP	1173
HIP	1175
HIP	1179
HIP	1193
RES	0
RES	1
RES	2
RES	3
RES	10
RES	20
RES	22
RES	24
RES	27
RES	29
RES	31
RES	33
RES	35
RES	39
RES	41
RES	43
RES	45
RES	47
RES	52
RES	54
RES	63
RES	64
RES	65
RES	69
RES	70
RES	71

TPD32 : Register list according to a Parameter progressive order

TPD32 : Register list according to a Parameter progressive order

IPA - TPD32	Register
RES	72
RES	74
RES	76
RES	78
RES	80
RES	82
RES	84
RES	86
RES	88
RES	90
RES	92
RES	94
RES	96
RES	98
RES	99
RES	100
RES	102
RES	103
RES	105
RES	107
RES	109
RES	111
RES	113
RES	114
RES	115
RES	117
RES	119
RES	121
RES	122
RES	123
RES	125
RES	127
RES	129
RES	131
RES	133
RES	137
RES	139
RES	141
RES	143
RES	145
RES	149

IPA - TPD32	Register
RES	151
RES	153
RES	155
RES	156
RES	157
RES	158
RES	159
RES	162
RES	164
RES	173
RES	175
RES	177
RES	179
RES	185
RES	186
RES	187
RES	188
RES	189
RES	193
RES	195
RES	199
RES	201
RES	204
RES	205
RES	206
RES	207
RES	208
RES	210
RES	212
RES	214
RES	216
RES	218
RES	220
RES	229
RES	231
RES	232
RES	233
RES	234
RES	236
RES	239
RES	242

IPA - TPD32	Register
RES	246
RES	248
RES	250
RES	253
RES	254
RES	255
RES	256
RES	257
RES	261
RES	263
RES	265
RES	267
RES	269
RES	271
RES	273
RES	275
RES	277
RES	278
RES	280
RES	282
RES	283
RES	284
RES	285
RES	286
RES	291
RES	293
RES	298
RES	301
RES	302
RES	304
RES	306
RES	309
RES	314
RES	316
RES	318
RES	320
RES	321
RES	322
RES	323
RES	324

IPA - TPD32	Register
RES	326
RES	328
RES	330
RES	332
RES	333
RES	334
RES	335
RES	338
RES	340
RES	342
RES	344
RES	345
RES	346
RES	348
RES	350
RES	352
RES	354
RES	356
RES	358
RES	360
RES	362
RES	364
RES	366
RES	368
RES	370
RES	372
RES	374
RES	376
RES	378
RES	380
RES	382
RES	384
RES	386
RES	388
RES	390
RES	391
RES	395
RES	398
RES	401
RES	409
RES	413

IPA - TPD32	Register
RES	417
RES	418
RES	419
RES	420
RES	425
RES	428
RES	429
RES	430
RES	431
RES	433
RES	435
RES	437
RES	454
RES	455
RES	456
RES	457
RES	461
RES	462
RES	463
RES	464
RES	465
RES	466
RES	469
RES	472
RES	475
RES	478
RES	482
RES	484
RES	493
RES	495
RES	500
RES	502
RES	504
RES	505
RES	506
RES	507
RES	517
RES	518
RES	519
RES	531
RES	532

IPA - TPD32	Register
RES	533
RES	536
RES	537
RES	538
RES	539
RES	543
RES	545
RES	547
RES	549
RES	551
RES	553
RES	555
RES	557
RES	559
RES	561
RES	562
RES	563
RES	565
RES	566
RES	572
RES	573
RES	576
RES	577
RES	580
RES	581
RES	590
RES	591
RES	592
RES	595
RES	598
RES	600
RES	603
RES	605
RES	608
RES	610
RES	613
RES	615
RES	618
RES	620
RES	623
RES	625

IPA - TPD32	Register
RES	628
RES	630
RES	633
RES	634
RES	635
RES	636
RES	637
RES	638
RES	642
RES	644
RES	645
RES	649
RES	653
RES	656
RES	658
RES	662
RES	670
RES	672
RES	675
RES	676
RES	677
RES	680
RES	682
RES	684
RES	685
RES	686
RES	687
RES	688
RES	689
RES	690
RES	691
RES	693
RES	695
RES	700
RES	701
RES	705
RES	709
RES	711
RES	713
RES	715
RES	717

IPA - TPD32	Register
RES	719
RES	720
RES	721
RES	725
RES	727
RES	729
RES	731
RES	733
RES	735
RES	736
RES	737
RES	754
RES	755
RES	756
RES	757
RES	760
RES	761
RES	774
RES	775
RES	776
RES	777
RES	778
RES	781
RES	789
RES	791
RES	794
RES	796
RES	800
RES	802
RES	805
RES	807
RES	809
RES	811
RES	813
RES	817
RES	819
RES	821
RES	823
RES	825
RES	827
RES	829

TPD32 : Register list according to a Parameter progressive order

TPD32 : Register list according to a Parameter progressive order

IPA - TPD32	Register
RES	834
RES	835
RES	836
RES	837
RES	841
RES	844
RES	845
RES	846
RES	847
RES	848
RES	856
RES	858
RES	860
RES	862
RES	864
RES	866
RES	868
RES	871
RES	875
RES	880
RES	882
RES	884
RES	889
RES	890
RES	891
RES	892
RES	893
RES	894
RES	899
RES	901
RES	903
RES	911
RES	913
RES	915
RES	918
RES	920
RES	923
RES	925
RES	929
RES	937
RES	940

IPA - TPD32	Register
RES	942
RES	943
RES	944
RES	945
RES	946
RES	947
RES	948
RES	981
RES	982
RES	1004
RES	1005
RES	1006
RES	1007
RES	1025
RES	1026
RES	1027
RES	1028
RES	1035
RES	1036
RES	1099
RES	1100
RES	1101
RES	1102
RES	1103
RES	1104
RES	1126
RES	1127
RES	1135
RES	1136
RES	1186
RES	1187
RES	1188
RES	1189
RES	1194
RES	1195
RES	1196
RES	1197
RES	1198
RES	1199
RES	1200
RES	1201

IPA - TPD32	Register
RES	1202
RES	1203
RES	1204
RES	1205
RES	1206

IPA - TPD32	Register
-------------	----------

TPD32 : Coil Table (Functions 01, 02, 05, 15)

IPA - TPD32	Register	IPA - TPD32	Register	IPA - TPD32	Register	IPA - TPD32	Register
RES	0	679	34	123	68	1149	102
314	1	680	35	124	69	1150	103
315	2	681	36	125	70	1151	104
316	3	452	37	422	71	366	105
380	4	1027	38	890	72	367	106
343	5	1028	39	891	73	355	107
RES	6	919	40	1068	74	356	108
RES	7	920	41	699	75	363	109
RES	8	898	42	_RES_	76	364	110
RES	9	899	43	_RES_	77	210	111
565	10	_RES_	44	353	78	211	112
566	11	_RES_	45	497	79	640	113
567	12	372	46	498	80	633	114
568	13	715	47	499	81	635	115
569	14	349	48	_RES_	82	387	116
570	15	342	49	_RES_	83	729	117
571	16	_RES_	50	648	84	730	118
572	17	_RES_	51	651	85	738	119
573	18	_RES_	52	649	86	749	120
574	19	245	53	652	87	471	121
575	20	293	54	911	88	472	122
576	21	294	55	457	89	477	123
577	22	344	56	458	90	_RES_	124
578	23	345	57	194	91	_RES_	125
579	24	373	58	195	92	295	126
580	25	346	59	357	93	389	127
RES	26	347	60	358	94	259	128
RES	27	1259	61	361	95	1045	129
RES	28	1260	62	362	96	296	130
RES	29	_RES_	63	369	97	390	131
694	30	242	64	370	98	260	132
676	31	322	65	1152	99	297	133
677	32	348	66	1141	100	391	134
678	33	1067	67	1142	101	261	135

TPD32 : Coil Table (Functions 01, 02, 05, 15)

IPA - TPD32	Register
RES	136
RES	137
RES	138
RES	139
RES	140
RES	141
388	142
181	143
RES	144
393	145
394	146
395	147
RES	148
RES	149
246	150
248	151
396	152
397	153
249	154
244	155
398	156
399	157
RES	158
153	159
400	160
401	161
402	162
RES	163
243	164
403	165
404	166
RES	167
630	168
750	169
940	170
944	171
RES	172
658	173
406	174
1139	175
407	176

IPA - TPD32	Register
736	177
RES	178
1088	179
1090	180
1093	181
1094	182
1092	183
RES	184
RES	185
435	186
256	187
258	188
262	189
263	190
904	191
RES	192
RES	193
492	194
561	195
RES	196
RES	197
520	198
521	199
522	200
523	201
524	202
525	203
526	204
527	205
528	206
529	207
530	208
531	209
532	210
533	211
534	212
535	213
537	214
538	215
539	216
540	217

IPA - TPD32	Register
541	218
542	219
543	220
544	221
545	222
546	223
547	224
548	225
549	226
550	227
551	228
552	229
RES	230
RES	231
RES	232
RES	233
RES	234
RES	235
425	236
RES	237
RES	238
769	239
770	240
783	241
772	242
794	243
780	244
781	245
800	246
RES	247
RES	248
RES	249
RES	250
RES	251
591	252
602	253
609	254
619	255
622	256
624	257
RES	258

IPA - TPD32	Register
RES	259
RES	260
RES	261
1075	262
888	263
1076	264
1077	265
RES	266
RES	267
RES	268
RES	269
201	270
464	271
1226	272
1235	273
1244	274
1253	275
RES	276
1258	277
1209	278
1161	279
1205	280
1187	281
1157	282
1159	283
1183	284
1214	285
1188	286
1189	287
1190	288
1176	289
1215	290
1201	291
1195	292
1256	293
1264	294
1265	295
1267	296
1268	297
1269	298
1270	299

TPD32 : Coil list according to a Parameter progressive order

IPA - TPD32	Register	IPA - TPD32	Register	IPA - TPD32	Register	IPA - TPD32	Register
123	68	343	5	399	157	532	210
124	69	344	56	400	160	533	211
125	70	345	57	401	161	534	212
153	159	346	59	402	162	535	213
181	143	347	60	403	165	537	214
194	91	348	66	404	166	538	215
195	92	349	48	406	174	539	216
201	270	353	78	407	176	540	217
210	111	355	107	422	71	541	218
211	112	356	108	425	236	542	219
242	64	357	93	435	186	543	220
243	164	358	94	452	37	544	221
244	155	361	95	457	89	545	222
245	53	362	96	458	90	546	223
246	150	363	109	464	271	547	224
248	151	364	110	471	121	548	225
249	154	366	105	472	122	549	226
256	187	367	106	477	123	550	227
258	188	369	97	492	194	551	228
259	128	370	98	497	79	552	229
260	132	372	46	498	80	561	195
261	135	373	58	499	81	565	10
262	189	380	4	520	198	566	11
263	190	387	116	521	199	567	12
293	54	388	142	522	200	568	13
294	55	389	127	523	201	569	14
295	126	390	131	524	202	570	15
296	130	391	134	525	203	571	16
297	133	393	145	526	204	572	17
314	1	394	146	527	205	573	18
315	2	395	147	528	206	574	19
316	3	396	152	529	207	575	20
322	65	397	153	530	208	576	21
342	49	398	156	531	209	577	22

IPA - TPD32	Register
578	23
579	24
580	25
591	252
602	253
609	254
619	255
622	256
624	257
630	168
633	114
635	115
640	113
648	84
649	86
651	85
652	87
658	173
676	31
677	32
678	33
679	34
680	35
681	36
694	30
699	75
715	47
729	117
730	118
736	177
738	119
749	120
750	169
769	239
770	240
772	242
780	244
781	245
783	241
794	243
800	246

IPA - TPD32	Register
888	263
890	72
891	73
898	42
899	43
904	191
911	88
919	40
920	41
940	170
944	171
1027	38
1028	39
1045	129
1067	67
1068	74
1075	262
1076	264
1077	265
1088	179
1090	180
1092	183
1093	181
1094	182
1139	175
1141	100
1142	101
1149	102
1150	103
1151	104
1152	99
1157	282
1159	283
1161	279
1176	289
1183	284
1187	281
1188	286
1189	287
1190	288
1195	292

IPA - TPD32	Register
1201	291
1205	280
1209	278
1214	285
1215	290
1226	272
1235	273
1244	274
1253	275
1256	293
1258	277
1259	61
1260	62
1264	294
1265	295
1267	296
1268	297
1269	298
1270	299
1271	300
1272	301
1273	302
1274	303
1275	304
1276	305
1277	306
1278	307
1279	308
1280	309
1281	310
1282	311
1283	312
1287	313
1292	314
RES	0
RES	6
RES	7
RES	8
RES	9
RES	26
RES	27

IPA - TPD32	Register
RES	28
RES	29
RES	44
RES	45
RES	50
RES	51
RES	52
RES	63
RES	76
RES	77
RES	82
RES	83
RES	124
RES	125
RES	136
RES	137
RES	138
RES	139
RES	140
RES	141
RES	144
RES	148
RES	149
RES	158
RES	163
RES	167
RES	172
RES	178
RES	184
RES	185
RES	192
RES	193
RES	196
RES	197
RES	230
RES	231
RES	232
RES	233
RES	234
RES	235
RES	237

TPD32 : Coil list according to a Parameter progressive order

GEFRAN SENSORI

via Cave, 11
25050 PROVAGLIO D'ISEO (BS)
ITALY
Ph. +39 030 9291411
Fax. +39 030 9823201
info@gefran.com

GEFRAN BENELUX

Lammerdries, 14A
B-2250 OLEN
Ph. +32 (0) 14248181
Fax. +32 (0) 14248180
info@gefran.be

**GEFRAN BRASIL
ELETOELETRÔNICA**

Avenida Dr. Altino Arantes,
377/379 Vila Clementino
04042-032 SÃO PAULO - SP
Ph. +55 (0) 1155851133
Fax +55 (0) 1155851425
gefran@gefran.com.br

GEFRAN DEUTSCHLAND

Philipp-Reis-Straße 9a
63500 SELIGENSTADT
Ph. +49 (0) 61828090
Fax +49 (0) 6182809222
vertrieb@gefran.de

GEFRAN SUISSE

Rue Fritz Courvoisier, 40
2302 LA CHAUX-DE-FONDS
Ph. +41 (0) 329684955
Fax +41 (0) 329683574
office@acome.ch

GEFRAN SIEI - FRANCE

4, rue Jean Desparmet - BP
8237
69355 LYON Cedex 08
Ph. +33 (0) 478770300
Fax +33 (0) 478770320
commercial@gefran.fr
contact@sieifrance.fr

GEFRAN ISI

8 Lowell Avenue
WINCHESTER - MA 01890
Toll Free 1-888-888-4474
Ph. +1 (781) 7295249
Fax +1 (781) 7291468
info@gefranisi.com

SIEI AREG - GERMANY

Zachersweg, 17
D 74376 - Gemmingen
Ph. +49 7143 9730
Fax +49 7143 97397
info@sieiareg.de

GEFRAN SIEI - UK

7 Pearson Road, Central Park
TELFORD, TF2 9TX
Ph. +44 (0) 8452 604555
Fax +44 (0) 8452 604556
sales@gefran.co.uk
sales@sieiuuk.co.uk

GEFRAN SIEI - ASIA

No.160 Paya Lebar Road
05-07 Orion Industrial Building
409022 Singapore
Ph. +65 6 8418300
Fax +65 6 7428300
info@sieiasia.com.sg

GEFRAN SIEI Electric Pte Ltd

Block B, Gr:Flr, No.155,
Fu Te Xi Yi Road,
Wai Gao Giao Trade Zone
200131 Shanghai
Ph. +86 21 5866 7816
Ph. +86 21 5866 1555
Ph. +86 21 5866 7688
gefransh@online.sh.cn

SIEI DRIVES TECHNOLOGY

No.1265, B1, Hong De Road,
Jia Ding District
201821 Shanghai
Ph. +86 21 69169898
Fax +86 21 69169333
info@sieiasia.com.cn

SIEI AMERICA - USA

14201 D South Lakes Drive
NC 28273 - Charlotte
Ph. +1 704 3290200
Fax +1 704 3290217
salescontact@sieiamerica.com

GEFRAN**GEFRAN S.p.A.**

Via Sebina 74
25050 Provaglio d'Iseo (BS)
ITALY
Ph. +39 030 98881
Ph. +39 030 9839063
info@gefran.com
www.gefran.com

Motion Control

Via Carducci 24
21040 Gerenzano [VA]
ITALY
Ph. +39 02 967601
Ph. +39 02 9682653
info@siei.it

Technical Assistance :

technohelp@siei.it

Customer Service :

customer@siei.it
Ph. +39 02 96760500

AUDIN S.p.A. - avenue de la malle - 51370 Saint-Pierre-Courcelles

Tel : 03.26.04.20.21 - Fax : 03.26.04.28.20 - Web : <http://www.audin.fr> - Email : info@audin.fr

Protocollo Modbus RTU
Rev. 0.1 / 27.12.2007



1S0EG8