

From the introduction to smart motors, we will introduce how to use them and typical programs. For the

introductory edition, we have prepared the following programs and supplementary materials. (Excerpted part = *)

If you need an introductory version, please contact us or our authorized distributor.

sample program

3-1	Input com	mand and completion signal output (I/O(Settings and usage) Z		
3-2	Origin sea	Origin search using phase input		
3-3	Pushin	Pushing origin search + origin search using sensor*		
3-4	error ha	andling		
3-5	Repeated back	s and forth motion		
3-6	Monitor lo	oads during repeated operations		
3-7	Speed ch	ange and control using analog input *		
3-8	Speed o	hange & control via digital input *		
3-9	during o _l	perationI/Ooutput a signal from		
	Sends a si	gnal when the encoder count reaches the specified position.I/OOutput from When the		
3-10	execution	voltage drops below the specified voltage, rotation will stop.		
3-11	Non-vola	tile (EEPROM)How to save memory and variable data		
3-12	Changing	g speed by program setting during reciprocating		
3-13	operation	n Program example for linear actuator*		
	16-point	positioning with 4 input points, soft limit, push origin search		
3-14	Position	teaching and playback control		
3-15	Pulse train co	Pulse train command control		
3-16	Electronic gea	Electronic gear control (following)		
3-17	Closed lo	Closed loop control using external encoder		
3-18	Combitronic	communication (multi-axis control)		
	3-18-1	How to specify a motor using a colon		
	3-18-2	Comparison of positions of each motor		
	3-18-3	encoder modulo		
	3-18-4	2axis control*		
	3-18-5	Linear interpolation (synchronous control)*		
	3-18-6	How to use interrupts in multi-axis control		
	3-18-7	Connection precautions & confirmation		
	3-18-8	Directives and violation examples at Combitronic		
	3-18-9	to slave motorI/Ointerrupt		
3-19	Set the wait b	etween operations using random numbers		
3-20	Origin s	earch using hardware limit (sensor input)		
3-21	Encode	modulo function		



3-22 PLC • Healthy signal program from microcomputer

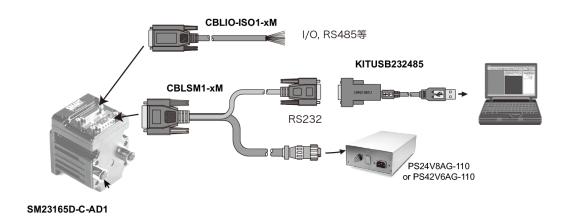
appendix

4-1	I/Ospecification
4-2	I/Otype
4-3	PIDSetting guideline
4-4	PIDChange & save value
4-5	D.E.Option (control power input separation
4-6	option) User-specified interrupt—maximum8
4-7	About torque mode
4-8	power selection
4-9	About shunts
4-10	RS232&RS485Connection method
4-11	RS485Initial settings for communication
4-12	ASCIICommunication by command/character code
4-13	Communication command timing
4-14	Terminal emulator (Tera Term)Usage
4-15	example Serial data analyzer
4-16	Output chart data to Excel
4-17	LEDFeature Description
4-18	Program capacity



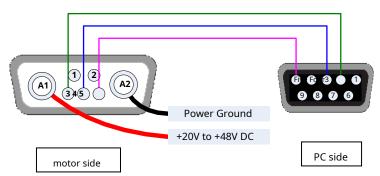
1Chapter Basic settings and servo motor control

Connection example (basic)



Smart Motor Comboprefix

electronic computerRS232prefix



7pin comboDsub connector

A1 +20~+48DVC

A2power supplyGND

- 1 I/OG
- 2 +5Voutput
- 3 RS232send(Tx)

FouRS232Receive (Rx)

Five e signalGND

9pinRS232

- 2 RS232Receive (Rx)
- 3 RS232send (Tx)

FiveRS232 GND



1-2 Establishing a communication link

Launch SMI using a PC

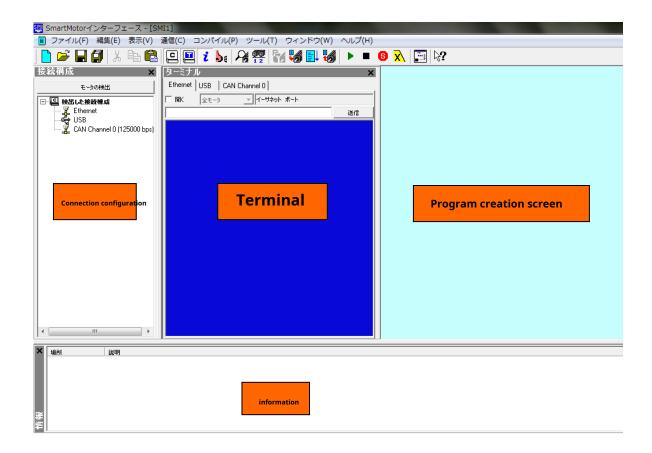


SMIofPCWhen installing, follow the steps below.URLInstrumented from Please roll. (free of charge)

https://www.moog.co.jp/products/motors-ser

vomotors/smartmotor.html

SMI interface basic screen

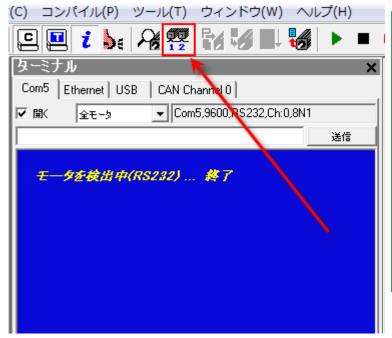


*Serial data analyzer

SMIThis is a convenient function for checking data information between the smart motor and the smart motor. Please refer to the appendix.



Establish communication between PC and integrated servo motor



When using a single axis, specify the address.

Although it is not necessary to specify

For network use,SMI

Automatically assign addresses in order from

If you don't have a terminal window (SMI

(blue screen) provides accurate control instructions.

I can't. (Note: Address each axis in advance.)

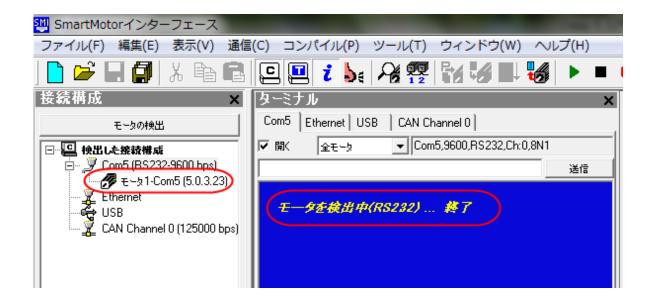
and enter the address

When detecting,

Click

please.)

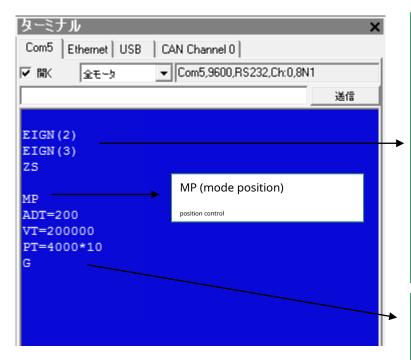
Check the establishment of communication





1-3 in the positive directionTenRotate (position control)

Write the command in the terminal (inside the blue screen) and press G(GO) + Enter to start rotating.



Ports 2&3 have overtravel limits defaults to as input is. Disable the limit or
Limit sensor (normally closed)
It works only when connected and set to Low state.
I don't.
EIGN(2): Sets port 2 as input and resets
Disable mitt input
EIGN(3): Set port 3 as input and restart.
disable mitt
ZS: Clear status bit

Code Absolu PT=a Exam Check current location with RPA and reach

SM23XX: 4000 counts/rotation (enco resolution)

(SM34XX): 8000 counts/rotation (engine

coder resolution)

Absolute target position

PT=absolute target position

Example: PT=4000 *10(rotation speed)=40000

(Regardless of the current position, the absolute value

Move to 40000. 0=>40000,

2500=>40000)

Relative target position

PRT=relative position movement amount

Example: PRT=4000 (current absolute position is

For 40000, 40000+4000=44000

Operate by specifying absolute position

[Achieved speed value]VT

RPA

PT=60000

Encoder resolution4000with the motor of PIDThe settings are PID2in the case of:

40000

VT=rev/sec*32768

example:200,000/32,768=6.10rps

6.10*60=366rpm

Encoder resolution8000with the motor of PIDThe settings are PID2in the case of:

VT=rev/sec*65536

[Acceleration/deceleration value]ADT

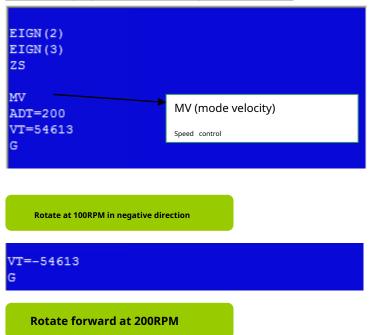
(Acceleration value:AT,Deceleration value:DT)

 $ADT=4.096\ xrevolutions/second^2 (encoder\ resolution 4000 with\ the\ motor\ of PIDThe\ settings\ are PID2 in\ the\ case\ of)$

ADT=8.192 xrevolutions/second2 (Encoder resolution8000with the motor ofPIDThe settings arePID2in the case of)



1-4Setting speed100RPM(speed control)



If the encoder resolution is 4000:

VT=rev/sec*32768

Ex: VT=1.66rps*32768

=54613

Hint: If you change the previously set parameters to

If you want to continue using it, please re-pair it.

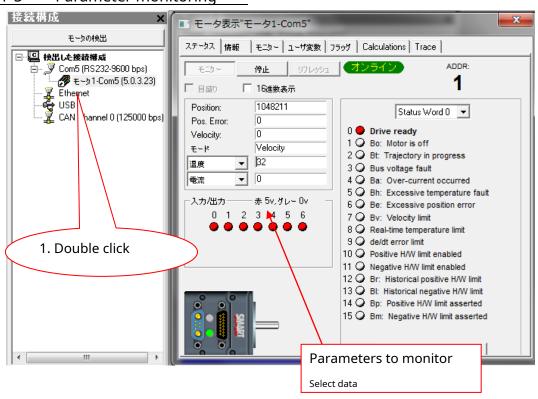
There is no need to write parameters.

VT=109227 G

Rotation stop (deceleration stop)

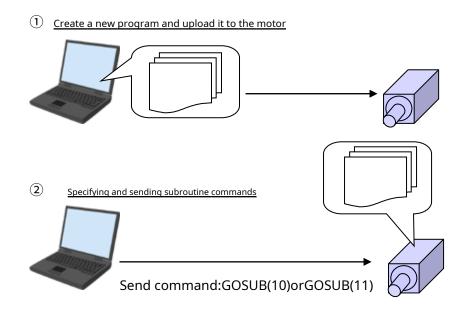
Х

1-5 Parameter monitoring

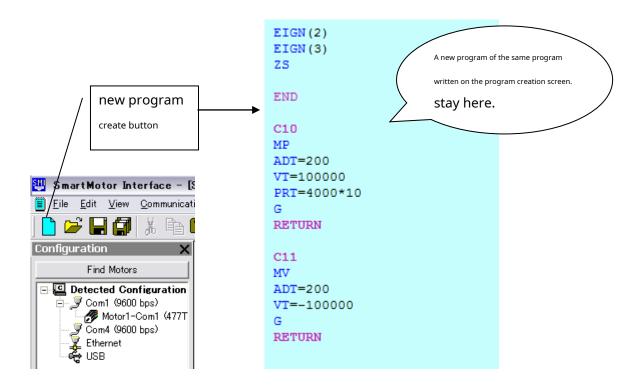




2Chapter Uploading the control program to the motor

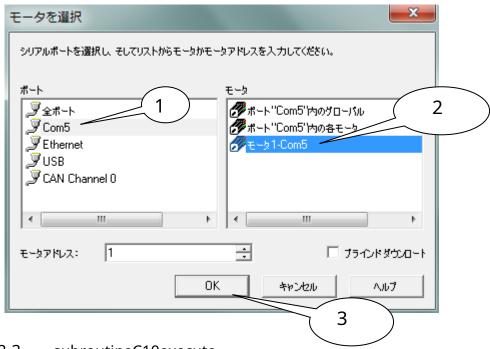


2-1 Creating a new program

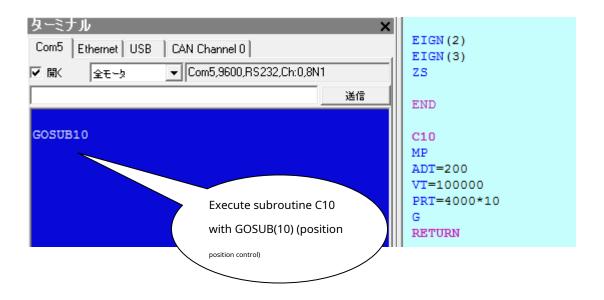


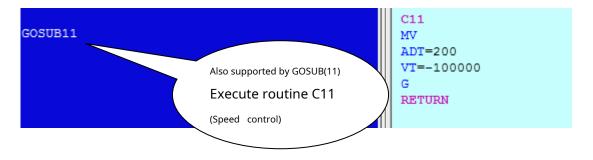


2-2 Download program to motor



2-3 subroutineC10execute







3-8Speed change & control via digital input

Start with position control, digital inputAbutLowThe speed will become faster.

EIGN(W,0) 'Hardware limitI/ODisable

KD=10010 'KDchange the value of

F 'KDValid value of

O=0 'Origin setting

ADT=100 Acceleration settings VT=10000 Speed setting

PT=40000 'Achievement point setting

MP 'Position mode setting

G 'start

WHILEBt 'Loop command during operation

IFIN(0)==0 '0port islowWhen

IFVT==10000 'VTThe value of10000when it becomes

VT=12000 'newVTadd value of

G 'start

ENDIF

ENDIF

LOOP'Loop command in action (WHILE Bt) END



3-7Speed change and control using analog input

This program includes a dead band function so that it does not react sensitively to analog signals. Operator is Ana

The motor speed will not change unless you change the log input.

EIGN(W,0) 'Disable hardware limits

KP=3020 'PIDvalue(Proportional) from default KD=10010 'PIDvalue(Differential) changed from default

F 'PIDSetting buffer value valid

ADT=100 'Acceleration settings MV 'Speed mode

d=10 'analog dead band,5000 =fault scale

o=2500 'Offset for negative movement

m=40 'Speed (multiple)
w=10 'time(10mseconds)

b=0 'speed

C10

a=INA(V1,3)-o '5VAnalog value reading

x=ab 'Set x to change input value

IFx>d 'Check if the dead band is exceeded

VT=b*m 'Set speed value

G 'Start at a new speed

ELSEIFx<-d 'Check if the dead band is exceeded

VT=b*m 'Set speed value

G 'Start at the set speed

ENDIF

b=a 'Anti-hunting update

WAIT=w 'wWait for seconds

GOTO10 'labelTenthe call of

END

Dsub15pin I/Os (5V I/Os)

INA(A, exp)Raw analog read:10Bit
 resolution,0-32736=0-5VDC

- INA(V1, exp)Voltage scale (millibo)
(loaded with root),3456=3.456VDC

Expansion 10 points24DC I/O

- INA(A, exp)Raw analog read:10Bit resolution.0-32736=0-41.25VDC

- The resolution is10bitin32Every increase/decrease,

voltage value0-41.25VDCis inside the board
 Reference value.I/OThe maximum input value is
 24VDC

INA(V, exp)Voltage scale reading
 (24000-0), 15500=15.5V

- INA(V1, exp) Read voltage scale (5000-0), 550=0.55V

11



3-1 Determining the origin from sensor input

EILP 'Forward limit switch enabled (default setting)

EILN 'Negative limit switch enabled (default setting)

ZS

END

CO 'Origin search subroutine

EIGN(3) 'port3Disable the limit switch for the negative direction of

ZS

G 'Start rotating towards the origin (limit sensor)

WHILEIN(3)==0LOOP 'Port 3LowLoop when ->Highexits the loop when

MTB 'Stop using torque brake mode

WAIT=50 '50mWait for seconds

O=-8000 'Current position minus8000and setting

MPPT=0G TWAIT 'Start the motor to position zero

EILN 'port3Set for limit switch (negative direction limit switch enabled)

RETURN



3-3Pushing origin search

What you will understand from this program:

1. How to return to origin by pushing without using sensor input

Caution: When pressing, be careful of the set speed and use the lowest possible deviation limit value. high speed

If the motor stops due to collision or if the position deviation value is set to a large value and the motor is pressed, the motor may be damaged.

Please use the pushing origin search at your own discretion.

'-----

'Return to origin routine (return to origin by pushing)

EIGN(2) EIGN(3)ZS 'Remove travel limit,I/Oset as normal input/output

'parameter settings

rr=-1 'Origin return rotation direction

vv=70000 'Homing speed

aa=1000 'Home return acceleration

ee=300 'Home return deviation limit tt=3000 'Homing torque limit

hh=4000 'Homing offset ZS 'Clear error bit

MV 'Speed mode setting

ADT=aa 'Homing acceleration setting

VT=vv*rr 'Homing speed setting

G 'Start rotating in the pushing direction

WHILEABS(EA)<eeLOOP'Values within the deviation limit are looped

MTB 'Stop using torque brake mode

MT 'When pressing, it bounces, so press again in torque mode.

T=tt*rr 'Preset torque value

G

WAIT=50 '50mWait for seconds

O=hh*rr 'Set the origin offset position as the origin

MPPT=0G TWAIT 'Set motor to position zero

END



3-4error handling

EIGN(W,0,12)'Remove travel limit,I/Oset as normal input/output

'Bit converts all inputs into binary values12and input2and3set as general input

Ru.

ZS 'Clear fault bit (Note: If the travel limit is not grounded at boot-up)

(if applicable)

'Fault interrupt settings

'ITR(interrupt number, status word, status bit, bit status,C

label call)

'Note: interrupt is8configurable,0~7, Oshould be given the highest priority

ITR(0,0,0,0,0)'set interrupts to zero

'status word zero

'bit zero

'Set bit status to zero (drive ready bit)

'Subroutine zero (C0)call

EITR(0) 'Enable zero interrupts

ITRE 'Enable global interrupt scanner

PAUSE 'Pause command (ENDcommand prevents interrupts from being disabled)

END

C0

'Place error code

'This routine is interrupted on travel limit, overtemperature, position deviation error, and overcurrent. IF

Be

PRINT("Position Error",#13)

ENDIF

IFBh

PRINT("Over Temp Error",#13)

ENDIF

IFBa

PRINT("Over Current Error",#13)

ENDIF

RETURNI



3-13Program example for linear actuator

inputFourby point16Point positioning, soft limit, push origin search

Our company sellsSmartBox BCDIt is convenient to use.

Four16 subroutines are called based on the point input, and input 6 (G)It works when received.

'ECHO 'Echo on

EIGN(W,0) 'All ofI/Oset to inputZS 'Reset all error bitsMDS 'Sine wave drive mode

(4)=1 'ESet port to busy output
OUT(5)=0 'FSet port to error output

'Set parameters

rr=-1 'Return to origin vv=100000 'speed aa=100 'acceleration

ee=300 'Current error value tt=4000 'Torque limit

hh=4000 'offset

nn=-300 'Software limit (minus direction)
pp=110000 'Software limit (plus direction)

'Enable software limits

SLN=nn 'Minus direction rotation limit
SLP=pp 'Plus direction rotation limit
SLM(1) 'Set to soft limit mode

(0,0,0,0,100) 'Interrupt on error(4-6reference)

EITR(0) 'Enable zero interrupts

(1,16,6,0,101) 'Interrupt settings(4-6reference)

ITRE 'Enable global interrupt scanner

EITR(1) 'interrupt1enable

WHILE1LOOP

END



C100

OUT(5)=1 'Eporthigh

IFBe 'Position deviation error bit

PRINT("Position Error",#13) ENDIF

IFBh 'Overtemperature error bit

PRINT("Over Temp Error",#13) ENDIF

IFBa 'Overcurrent error bit

PRINT("Over Current Error",#13) ENDIF

RETURNI

'portFWhen there is an input to 101 call

PRINT("GO PRESSED",#13)

x=15-(IN(W,0)&15) GOSUB(x) 'Input port0,1,2,3 (A, B, C, D)confirm

'subroutine0-15call

ITR(1,16,6,0,101) '(4-6reference)

RETURNI

'-----

'Press return to origin

C0

S.L.D.

ZS 'Reset all error bits

OUT(4)=0 'Dport outputLow

VT=vv*rr 'speed ADT=aa 'acceleration

MV 'Speed mode ZS 'Clear error bit

G

WHILEABS(EA)<eeLOOP MTB 'Loop within deviation limit

'Torque brake

'Soft limit disabled

MT 'Torque mode

T=tt*rr 'Torque value setting

G

WAIT=500 '500mWait for seconds
O=hh*rr 'Position settings

MPPT=0G TWAIT 'Position mode/Return to origin



OUT(4)=1	'Dport outputHigh
SLE	'Soft limit enabled
RETURN	
'	====== C1
OUT(4)=0	
MP	
ADT=300	
VT=900000	
P.T.=90000	
G TWAIT	
P.T.=0	
G TWAIT	
P.T.=90000	
G TWAIT	
P.T.=0	
G TWAIT	
OUT(4)=1	
RETURN	
'	====== C2
C3	
C4	
C5	
C6	
C7	
C8	
C9	
C10	
C11	
C12	
C13	
C14	
C15	
PRINT("Called SUBROUTINE ",x,#13) PRINT	
("EXITING SUBROUTINE ",x,#13) RETURN	



<u>CAN&CombitronicConnection example (multi-axis control)</u> 1.

Upper controller (PC/PLC)If you don't use:

all smart motorsCANnetwork connection

Uses Combitronic communication

Multi-axis motor control and each axis using any smart motor as the masterI/Ocontrol.



2. Upper controller (PC/PLC)everything you needCANWhen connecting with:

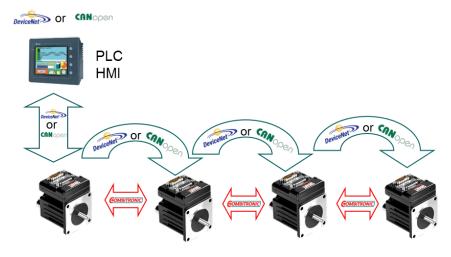
2-1The upper controller is the main master.

CANopenusing the protocol**Smart motor is a slave device**controlled as (CANopen)

2-2The smart motor is the main master.

The upper controller sends and receives minimal data (CANopen)Just dosmart motor starand control (Combitronic).

Control using Combitronic communication (CANopenandCobitronic)





3-18-4 2axis control

Master Motor(Motor 1)

CADDR=1

ADDR=CADDR

EIGN(2) :0

EIGN(3) :0

ZS:

MV0

VT=164424*3

ADT:0=1000

G:1

WAIT=1000*10

VT=264424*5

G:2

WHILEP.A.:1 > P.A2 LOOP

X:0

MP0

PML:0 =4000

PMT:0 =2000

G:0

END

'Declare motor address 1

'CANaddress and RS232 Unify addresses '

all motors: Specify the second port as the general input port

all motors :Specify the third port as the general input port

whole mor Taliset

' all motors :Speed mode

Motor 1 : Speed specification (3rps)

all motors : Acceleration specification

Motor 1:execution

'TenWait for seconds

motor2: Speed specification (5rps)

motor2 :execution

'loop [motor1 position> motor2 Position of]

' all motors :Stop

all motors :Position mode

all motors: Encoder modulo limit specification

all motors: Encoder modulo target specification

all motors: Execution (encoder modulo function)

Slave motor(Motor 2)

CADDR=2

ADDR=CADDR

'Declare motor address 2

'CANaddress andRS232Unify addresses

END



3-18-5 Linear interpolation (synchronous control)

 $host PC/PLC Linear\ interpolation\ control\ is\ possible\ without\ using\ .$

ADTS=100 Set synchronous command acceleration

VTS=100000 Set the synchronous command attained speed

 $PTS(30000;1,40000;2) \\ \hspace*{0.2cm} \text{'Set the destination position of motors 1 and 2}$

 $\label{position2} \mbox{'PTS(position1;Axis number1,position2;Axis number2[,position3;Axis number3])} \\$

G.S. 'Start, start synchronization

Class5 command table (Ver1_10)

command	explanation
A	ехраничи
az	user variable
a=z=	User variable settings
aazz	user variable
aa=zz=	User variable settings
aaazzz	user variable
aaa=zzz=	User variable settings
ab[index]	8 bit variable
ab[index]=	8-bit variable settings
af[index]	float variable
af[index]=	Float variable settings
al[index]	32 bit variables
al[index]=	32 bit variable settings
aw[index]	16 bit variable
aw[index]=	16-bit variable settings
Ai(0)	Get internal encoder Z phase position at rising edge
Ai(1)	Obtain external encoder Z phase position at rising edge
Aij(0)	Get internal encoder Z phase position at rising edge and falling edge
Aij(1)	Obtain external encoder Z phase position at rising edge and falling edge
Aj(0)	Get internal encoder Z phase position at falling edge
Aj(1)	Obtain external encoder Z phase position at falling edge
Aji(0)	Get internal encoder Z phase position at falling edge and rising edge
Aji(1)	Obtain external encoder Z phase position at falling edge and rising edge
ABS()	integer absolute value
A.C.	Command acceleration
ACOS()	Arccosine (inverse of cosine) by angle
ADDR	motor serial address
ADDR=	Motor serial address setting
ADT=	Acceleration/deceleration setting
ADTS=	Synchronous operation acceleration/deceleration setting
AMPS	Maximum PWM limit
AMPS=	Maximum PWM limit setting
ASIN()	arc-sine (inverse function of sine) by angle
AT	acceleration
AT=	Acceleration settings
ATS=	Synchronous operation acceleration setting
ATAN()	Arctangent by angle (inverse tangent)
ATOF()	Get ASCII and convert to float type
В	
B()	status bit
Ва	Overcurrent status bit
BAUD(0)	channel 0 baud rate
BAUD(1)	channel 1 baud rate
BAUD#	Channel 0 baud rate setting
BAUD(0)=	Channel 0 baud rate setting
BAUD(1)=	Channel 1 baud rate setting
Ве	Position deviation error status bit

1

	Overtemperature status bit
Bi(0)	Get internal encoder Z phase status bit on rising edge
Bi(1)	Get external encoder Z phase status bit on rising edge
Bj(0)	Get internal encoder Z phase status bit on falling edge
Bj(1)	Get external encoder Z phase status bit on falling edge
Bk	EEPROM data integrity status bit
Bl	Left/Negative Hardware Overtravel Limit History Status Bit
Bls	Left/Negative Direction Software Overtravel Limit History Status Bit
Bm	Left/Negative Hardware Overtravel Limit Status Bit
Bms	Left/Negative Direction Software Overtravel Limit Status Bit
Во	Motor OFF status bit
Вр	Right/Forward Hardware Overtravel Limit Status Bit
Bps	Right/Forward Software Overtravel Limit Status Bit
Br	Right/Forward Hardware Overtravel Limit History Status Bit
Brs	Right/Forward Software Overtravel Limit History Status Bit
Bs	syntax error status bit
Bt	Operation in progress status bit
Bv	Speed error status bit
Bw	32-bit position counter Wrap around status bit
Bx(0)	Real-time internal index input status bit
Bx(1)	Real-time external index input status bit
BREAK	Program execution flow control
BRKENG	Brake operation
BRKRLS	brake release
BRKSRV	Brake operation while servo is stopped
BRKTRJ	Brake activation while operation is stopped
С	
C#	Program subroutine label
i	<u> </u>
CADDR	CAN address
CADDR CADDR=	CAN address CAN address setting
CADDR=	CAN address setting
CADDR=	CAN address setting CAN error
CADDR= CAN CANCTL()	CAN address setting CAN error Network function control
CADDR= CAN CANCTL() CASE#	CAN address setting CAN error Network function control program flow instructions
CADDR= CAN CANCTL() CASE# CBAUD	CAN address setting CAN error Network function control program flow instructions CAN baud rate
CADDR= CAN CANCTL() CASE# CBAUD CBAUD=	CAN address setting CAN error Network function control program flow instructions CAN baud rate CAN baud rate setting
CADDR= CAN CANCTL() CASE# CBAUD CBAUD= CCHN()	CAN address setting CAN error Network function control program flow instructions CAN baud rate CAN baud rate setting serial channel closed
CADDR= CAN CANCTL() CASE# CBAUD CBAUD= CCHN()	CAN address setting CAN error Network function control program flow instructions CAN baud rate CAN baud rate setting serial channel closed RS232 communication error flag
CADDR= CAN CANCTL() CASE# CBAUD CBAUD= CCHN() CHN(0) CHN(1)	CAN address setting CAN error Network function control program flow instructions CAN baud rate CAN baud rate setting serial channel closed RS232 communication error flag RS485 communication error flag
CADDR= CAN CANCTL() CASE# CBAUD CBAUD= CCHN() CHN(0) CHN(1) CLK	CAN address setting CAN error Network function control program flow instructions CAN baud rate CAN baud rate CAN baud rate setting serial channel closed RS232 communication error flag RS485 communication error flag 1 ms clock variable
CADDR= CAN CANCTL() CASE# CBAUD CBAUD= CCHN() CHN(0) CHN(1) CLK CLK=	CAN address setting CAN error Network function control program flow instructions CAN baud rate CAN baud rate setting serial channel closed RS232 communication error flag RS485 communication error flag 1 ms clock variable 1 ms clock setting
CADDR= CAN CANCTL() CASE# CBAUD CBAUD= CCHN() CHN(0) CHN(1) CLK CLK= COS()	CAN address setting CAN error Network function control program flow instructions CAN baud rate CAN baud rate setting serial channel closed RS232 communication error flag RS485 communication error flag 1 ms clock variable 1 ms clock setting Cosine (angle)
CADDR= CAN CANCTL() CASE# CBAUD CBAUD= CCHN(0) CHN(1) CLK CLK= COS() C.P.	CAN address setting CAN error Network function control program flow instructions CAN baud rate CAN baud rate setting serial channel closed RS232 communication error flag RS485 communication error flag 1 ms clock variable 1 ms clock setting Cosine (angle) cam pointer
CADDR= CAN CANCTL() CASE# CBAUD CBAUD= CCHN() CHN(0) CHN(1) CLK CLK= COS() C.P.	CAN address setting CAN error Network function control program flow instructions CAN baud rate CAN baud rate setting serial channel closed RS232 communication error flag RS485 communication error flag 1 ms clock variable 1 ms clock setting Cosine (angle) cam pointer Add cam table
CADDR= CAN CANCTL() CASE# CBAUD CBAUD= CCHN() CHN(0) CHN(1) CLK CLK= COS() C.P. CTA() CTE()	CAN address setting CAN error Network function control program flow instructions CAN baud rate CAN baud rate setting serial channel closed RS232 communication error flag RS485 communication error flag 1 ms clock variable 1 ms clock setting Cosine (angle) cam pointer Add cam table Delete cam table
CADDR= CAN CANCTL() CASE# CBAUD CBAUD= CCHN(0) CHN(1) CLK CLK= COS() C.P. CTA() CTE() CTR(0)	CAN address setting CAN error Network function control program flow instructions CAN baud rate CAN baud rate CAN baud rate setting serial channel closed RS232 communication error flag RS485 communication error flag 1ms clock variable 1ms clock setting Cosine (angle) cam pointer Add cam table Delete cam table Primary encoder/pulse train & direction counter
CADDR= CAN CANCTL() CASE# CBAUD CBAUD= CCHN(0) CHN(0) CHN(1) CLK CLK= COS() C.P. CTA() CTE() CTR(0) CTR(1)	CAN address setting CAN error Network function control program flow instructions CAN baud rate CAN baud rate CAN baud rate setting serial channel closed RS232 communication error flag RS485 communication error flag 1 ms clock variable 1 ms clock setting Cosine (angle) cam pointer Add cam table Delete cam table Primary encoder/pulse train & direction counter
CADDR= CAN CANCTL() CASE# CBAUD CBAUD= CCHN() CHN(0) CHN(1) CLK CLK= COS() C.P. CTA() CTE() CTR(0) CTR(1)	CAN address setting CAN error Network function control program flow instructions CAN baud rate CAN baud rate CAN baud rate setting serial channel closed R5232 communication error flag R5485 communication error flag 1ms clock variable 1ms clock setting Cosine (angle) cam pointer Add cam table Delete cam table Primary encoder/pulse train & direction counter Second encoder/pulse & direction counter

DEFAULT	Switch-case syntax element
DEL	de/dt fault limit
DEL=	de/dt fault limit setting
DFS()	In 32-bit IEEE format, af[] variable
DITR()	Interrupt stop
DT	Deceleration
DT=	Deceleration setting
DTS=	Synchronous action deceleration setting
E	
EA	Position deviation error
ECHO	Output input data to main channel
ECHO_OFF	Main channel echo stop
ECHO1	Output input data to secondary channel
ECHO_OFF1	Secondary channel echo stop
EIGN()	Set I/O pin to input
EILN	Left/negative side hardware limit switch on
EILP	Right/positive side hardware limit switch on
EIRE	Configuring index signal acquisition pin for external encoder
EIRI	Configure index signal acquisition pin for internal encoder
EISM(6)	Set pin 6 to G command input
EITR()	Interrupt settings
EL	Position deviation error fault limit
EL=	Position deviation error fault limit setting
ELSE	IF syntax element
ELSEIF	ELSE syntax element
ENC0	Internal encoder selection
ENC1	External encoder selection
END	The end of the program
ENDIF	IF statement end
ENDS	switch syntax end
EOBK()	Sends brake signal to I/O output
EPTR	EEPROM pointer
EPTR=	EEPROM pointer setting
ERRC	Most recent command error code
ERRW	Latest command error communication channel
F	
F	Buffered PID settings enabled
FABS()	floating point absolute value error
FSA()	Fault action settings
FSQRT()	floating point square root
FW	Firmware version
G	
G	Motion start (GO)
G()	Specified orbit motion start (GO)
G.S.	Synchronous motion motion start (GO)
GETCHR	Main communication channel string
GETCHR1	Secondary communication channel string
GOSUB()	Subroutine call by number or variable
GOSUB#	subroutine call
GOTO()	Jump to program label by number or variable

GOTO#	Jump to program label
Н	
HEX()	hex string variable
I	
I(0)(capital i)	Encoder Z phase input position variable (rising edge, internal encoder)
I(1)(capital i)	Encoder Z phase input position variable (rising edge, external encoder)
IF	IF syntax element
IN()	I/O input
INA()	analog input
ITR()	User interrupt settings
ITRD	Stop all user interrupts
ITRE	All user interrupts enabled
J	
J(0)	Encoder Z phase position variable acquisition (falling edge, internal encoder)
J(1)	Encoder Z phase position variable acquisition (falling edge, external encoder)
K	
K.A.	KA buffer PID value (acceleration)
KA=	KA buffer PID value setting
K.C.	KC value
KC=	KC value setting
KCS	KCS value
KCS=	KCS value setting
KD	KD buffer PID value (differential)
KD=	KD buffer PID value setting (differentiation)
KG	KG buffer PID value (gravity)
KG=	KG buffer PID value setting (gravity)
K.I.	KI buffer PID value (integral)
KI=	KI buffer PID value setting (integral)
KL	KL buffer PID value (integral limit)
KL=	KL buffer PID value setting (integral limit)
KP	KP buffer PID value (proportional)
KP=	KP buffer PID value setting (proportional)
K.S.	KS buffer PID value (integral filter control)
KS=	KS buffer PID value setting (integral filter control)
KV	KV buffer PID value (velocity feedforward)
KV=	KV buffer OID value setting (velocity feedforward)
L	
LEN	Main communication channel buffer occupancy level (data mode)
LEN1	Secondary communication channel buffer occupancy level (data mode)
LFS()	32-bit IEEE format Float value
LOAD	Download program to motor
LOCKP	Prevent program upload until new program is loaded
LOOP	WHILE syntax element
М	
MC	Cam mode enabled
MC()	Cam mode enabled (additional trajectory)
MCE()	Cam spline enabled
MCW()	Cam start point
MDB	TOB commutation enabled
MDC	Sine wave current commutation mode

MDE	Trapezoidal encoder commutation mode
MDS	Sine wave voltage commutation mode
MDT	Trapezoid Hall sensor commutation mode
MF0	Set CTR(1) to 0 and select external encoder to 4x mode
MFA()	Follow mode, rising
MFD()	Follow mode, falling
MFDIV	Tracking mode divisor
MFDIV=	Tracking mode divisor setting
MFMUL	Follow mode multiplier
MFMUL=	Follow mode multiplier setting
MFR	4 multiplication tracking mode selection
MFR()	4-fold tracking mode selection (additional trajectory)
MFSDC()	Follow mode (stall – pause – continue)
MFSLEW()	Follow mode constant
MINV()	reverse commutation
MODE	action mode
MODE()	Operation mode (specific orbit)
MP	Position mode enabled
MP()	Position mode enabled (additional trajectory)
MS0	Set CTR(1) to 0 and select pulse/direction mode using external encoder
MSR	Select tracking mode in pulse/direction mode
MSR()	Select following mode in pulse/direction mode (additional trajectory)
MT	Torque mode enabled
MTB	mode torque brake
MV	Speed mode enabled
MV()	Speed mode enabled (additional trajectory)
0	
O=	Origin setting
O()=	Setting the origin of a specific trajectory
OC()	Output status (24V I/O)
OCHN()	open communication channel
OF()	Output fault (24V I/O)
OFF	servo off
OR()	Set output to low
OS()	Set output to High
OSH=	Origin shift
OSH()=	Specific origin shift
OUT()=	Set one or more outputs to a specific state
P	
P.A.	actual position
PAUSE	Pause program execution
PC	Command position
PC()	Command position (specific trajectory)
P.I.	Get π value
PID1	16,000Hz PID rate
PID2	8,000 Hz PID rate (default)
PID4	4,000Hz PID rate
PID8	2,000Hz PID rate
PMA	Actual position modulo
PML	position modulo limit
L .	· · · · · · · · · · · · · · · · · · ·

PML=	Position modulo limit setting
PMT	Position modulo target (position motion)
PMT=	Position modulo target setting (position operation)
PRA	Acquisition of actual measurement start position
PRC	Operation start command position
PRINT()	Data output to main communication channel
PRINT1()	Data output to secondary communication channel
PRT	Relative target position
PRT=	Relative target position setting
PRTS=()	Synchronous relative target position setting
PRTSS=()	Additional synchronized relative target position setting
P.T.	Target position
PT=	Target position setting
PTS=()	Synchronous absolute target position setting
PTSS=()	Additional synchronized absolute target position setting
PTSD	Synchronous linear working distance
R	
RaRzzz	Get user variable
RaRzzz	User variable report
RaaaRzzz	User variable report
Rab[index]	8-bit array variable report
Raf[index]	Float type variable report
Ral[index]	32-bit variable reporting
Raw[index]	16-bit variable reporting
RABS()	Integer type absolute value reporting
R.A.C.	Command acceleration value report
RACOS()	Arc-cosine (inverse cosine) value reporting by angle
RADDR	Motor serial address report
RAMPS	Maximum PWM limit reporting
RANDOM	Obtain random number Example: a=RANDOM
RANDOM=	Random number setting
RASIN()	Arc-sin (inverse sine) value reporting by angle
RAT	Target acceleration value report
RATAN()	Arc-tan (inverse tangent) value reporting by angle
RATOF()	Convert ASCII to Float type and report
RB()	status bit report
RBa	Overcurrent status bit reporting
RBAUD(0)	Channel 0 baud rate report
RBAUD(1)	Channel 1 baud rate report
	Position deviation error status bit report
RBe	
RBe RBh	Overtemperature status bit reporting
RBh RBi(0)	Overtemperature status bit reporting Encoder Z phase status bit (internal encoder, rising signal) report
RBh RBi(0) RBi(1)	Overtemperature status bit reporting Encoder Z phase status bit (internal encoder, rising signal) report Encoder Z phase status bit (external encoder, rising signal) report
RBh RBi(0) RBi(1) RBj(0)	Overtemperature status bit reporting Encoder Z phase status bit (internal encoder, rising signal) report Encoder Z phase status bit (external encoder, rising signal) report Encoder Z phase status bit (internal encoder, falling signal) report
RBh RBi(0) RBi(1) RBj(0) RBj(1)	Overtemperature status bit reporting Encoder Z phase status bit (internal encoder, rising signal) report Encoder Z phase status bit (external encoder, rising signal) report Encoder Z phase status bit (internal encoder, falling signal) report Encoder Z phase status bit (external encoder, falling signal) report
RBh RBi(0) RBi(1) RBj(0) RBj(1) RBjk	Overtemperature status bit reporting Encoder Z phase status bit (internal encoder, rising signal) report Encoder Z phase status bit (external encoder, rising signal) report Encoder Z phase status bit (internal encoder, falling signal) report
RBh RBi(0) RBi(1) RBj(0) RBj(1) RBk RBI	Overtemperature status bit reporting Encoder Z phase status bit (internal encoder, rising signal) report Encoder Z phase status bit (external encoder, rising signal) report Encoder Z phase status bit (internal encoder, falling signal) report Encoder Z phase status bit (external encoder, falling signal) report
RBh RBi(0) RBi(1) RBj(0) RBj(1) RBk RBI RBIS	Overtemperature status bit reporting Encoder Z phase status bit (internal encoder, rising signal) report Encoder Z phase status bit (external encoder, rising signal) report Encoder Z phase status bit (internal encoder, falling signal) report Encoder Z phase status bit (external encoder, falling signal) report EEPROM data integrity status bit reporting Hardware left/negative overtravel limit history bit reporting Software left/negative overtravel limit history bit reporting
RBh RBi(0) RBi(1) RBj(0) RBj(1) RBk RBI	Overtemperature status bit reporting Encoder Z phase status bit (internal encoder, rising signal) report Encoder Z phase status bit (external encoder, rising signal) report Encoder Z phase status bit (internal encoder, falling signal) report Encoder Z phase status bit (external encoder, falling signal) report EEPROM data integrity status bit reporting Hardware left/negative overtravel limit history bit reporting

RBp Right/Forward Hardware Overtravel Limit Bit Report RBps Right/Forward Soft Limit Overtravel Limit Bit Report RBps Right/Forward Soft Limit Overtravel limit history bit reporting RBr Hardware right/forward overtravel limit history bit reporting RBrs Software right/forward overtravel limit history bit reporting RBrs Software right/forward overtravel limit history bit reporting RBs Software right/forward overtravel limit history bit reporting RBb Operation in progress status bit report RBv Speed error bit reporting RBw 32bit position counter wrap/around status bit report RBw 32bit position counter wrap/around status bit report RBw. 32bit position counter wrap/around status bit report RCADDR CAN address report RCANN CAN address report RCANN CAN address report RCANN CAN address report RCANN CAN address report RCANNO) R6-22 communication error flag report RCHN(1) R5-85 communication error flag report RCHN(1) R5-85 communication error flag report RCKS Program checksum report RCKS Program checksum report RCKK Intro dock variable reporting RCCKS Program checksum report RCTR(1) Report cosine as an analyse RCP Cam pointer report RCTR(1) Second encoder/public stain & direction counter reporting RCTR Beaution status report RCAN RCTR Beaution status report RCAN Recent command error counter reporting RCTR RCTR Recent command error counter reporting RCTR RCTR Recent command error code report RCTR RCTR RCTR RCTR Recent command error code report RCTR RCTR RCTR RCTR RCTR RCTR RCTR reporting report (right early a report	RBo	Servo off status bit report
RBps Right/Forward Soft Limit Overtravel Limit Bits Report RBr Hardware right/forward overtravel limit history bit reporting RBrs Software right/forward overtravel limit history bit reporting RBs Operation in progress status bit report RBt Operation in progress status bit report RBt Operation in progress status bit report RBv Seed error bit reporting RBw 32bit position counter wrap/around status bit report RBc() Real-time internal index input status bit reporting RCADDR CAN address report RCADDR CAN address report RCAN CAN amore reporting RCBAUD CAN bould rate report RCAN BEALD CAN bould rate report RCHN(0) R5232 communication error flag report RCHN(1) R5435 communication error flag report RCKS Program checksum report RCR(0) Primary encoder/pulse train & direction counter reporting RCR(1) Second encoder/pulse train & direction counter reporting RCTR(1) Second encoder/pulse train & direction counter reporting RCTR(1) Second encoder/pulse & direction counter reporting RCTR(1) Second encoder/pulse & direction counter reporting RCTR(1) Second encoder/pulse & direction counter reporting RCTR(1) Report direction of counter reporting RCTR Second encoder/pulse & direction counter reportin	RBp	
RBF Hardware right/forward overtravel limit history bit reporting RBFS Software right/forward overtravel limit history bit reporting RBFS Software right/forward overtravel limit history bit report RBE Software right/forward overtravel limit history bit report RBE Operation in progress status bit report RBE Operation in progress status bit report RBW 32bit position counter wrap/around status bit report RBW, 32bit position counter wrap/around status bit report RBW, 32bit position counter wrap/around status bit report RBW, 32bit position counter wrap/around status bit report RCAN CAN address report RCAN CAN address report RCAN CAN address report RCAN CAN address report RCHN(0) RS-322 communication error flag report RCHN(1) RS-325 communication error flag report RCKS Program Checksum report RCKS Program Checksum report RCKS Program Checksum report RCRCK Ints clock variable reporting RCRCH Ints clock variable reporting RCRCH RCP CAm pointer report RCTR(0) Primary encoder/pulse train & direction counter reporting RCRCH RCP CAm pointer report RCTR(1) Second encoder/pulse & direction counter reporting RCRCH RCP CAM District of counter reporting RCRCH RCP CAM District of counter report RCRCH RCP CAM District RCP CAM Dis		
RBrs Software right/forward overtravel limit history bit reporting RBs Syntax error status bit report RBt Operation in progress status bit report RBv Speed error bit reporting RBv Status bit report RBv Speed error bit reporting RBw 32bit position counter wrap/around status bit report RBw 32bit position counter wrap/around status bit report RBw.) Real-time internal index input status bit reporting RCANDR CAN address report RCAN AN CAN error reporting RCAND CAN address report RCAN CAN error reporting RCHN(1) RS-S22 communication error flag report RCLK Inschool status status erport RCTR(0) RS-S22 communication error flag report RCTR(0) Primary encoder/pulse train & direction counter reporting RCTT Report of number of can tables in EEPROM RCTR(1) Second encoder/pulse & direction counter reporting RCTT Report of mumber of can tables in EEPROM RDEA Executed the dust limit reporting RDFS in 32-bit EEF format, all yvariable reporting RCT REPORT RE	-	
RBs Syntax error status bit report RBt Operation in progress saus bit report RBV Speed error bit reporting RBW 32bit position counter wrap/around status bit report RBALDDR CAN address report RCADDR CAN address report RCAN CAN error reporting RCAND CAN baud rate report RCHN(0) RES232 communication error flag report RCHN(1) RES432 communication error flag report RCHN(1) RES432 communication error flag report RCKS Program checksum report RCKS Program checksum report RCKS Program checksum report RCKS Program checksum report RCRCK Inscience and angle RCADD RCHN(1) Report copies as an angle RCOS) Report copies as an angle RCP Camp pointer report RCTR(0) Primary encoder/pulse & direction counter reporting RCTR(1) Second encoder/pulse & direction counter reporting RCTR(1) Second encoder/pulse & direction counter reporting RCTR(1) Second encoder/pulse & direction counter reporting RCTR ROBA Processe deviation report RCTR REPORT of number of cam tables in EEPROM RDEA Processe deviation report RDEA Position deviation error RCAD REPORT of substances and report RCAD REPORT of number of cam tables in EEPROM REPORT of number of ca	RBrs	
RBW 32bit position counter wrap/around status bit report RBW, 32bit position counter wrap/around status bit report RBW, 32bit position counter wrap/around status bit report REALD RCADDR CAN address report RCAN CAN error reporting RCHAID CAN baud rate report RCHN(0) R5232 communication error fing report RCHN(1) R6465 communication error fing report RCKS Program checksum report RCLK Tims clock variable reporting RCSC() Report cosine as an angle RCP Cam pointer report RCTR(0) Primary encoder/pulse train & direction counter reporting RCTR(1) Second encoder/pulse & direction counter reporting RCTR(1) Second encoder/pulse & direction counter reporting RCTR(1) REPORT RCTT Report of number of an tables in RFROM RCTT REPORT RE	RBs	
RBW 32bit position counter wrap/around status bit report RBX	RBt	
REALDR RCADDR CAN address report RCAN CAN error reporting RCADDR CAN address report RCAN CAN error reporting RCHN(I) RS-232 communication error flag report RCHN(I) RS-445 communication error flag report RCHN(I) RCHN(I) RS-445 communication error flag report RCHN(I) RCKS Program checksum report RCLK Instruction as an angle RCCS RCCS RCCS REPORT Cosine as an angle RCCP Cam pointer report RCTR(I) RCTR(I) Report or insuber train & direction counter reporting RCTR(I) RCTR(I) Second encoder/public & direction counter reporting RCTT Report of number of cam tables in EEPROM RCTR(I) REPORT RE	RBv	Speed error bit reporting
RCADDR RCAN CAN address report RCAN CAN baud rate report RCHN(0) RS-522 communication error flag report RCHN(1) RCHN(1) RS-685 communication error flag report RCKS RCLK Tims clock variable reporting RCOS() Report cosine as an angle RCP Cam pointer report RCTR(0) RTM-10 Second encoder/pulse & direction counter reporting RCTR(1) RCTR RCTT Report of number of cam bable in EEPROM RCTR(1) RDEA Encoder an bable in EEPROM RCTR RDEL Setting de/dt fault limit reporting RDFS in 32-bit IEEE format, afjl variable reporting RCTR REL Position deviation error report REL Position deviation error report RERA Pasition-deviation error report RERA RERC Recent command error code report RERRW RERCOM Pointer data report RESRW RERCOM Recent command error code report RESRW RERCOM Recent command error code report RES RESUME RESUME RESUME RESUME Resume program secution after passing RETURN RE	RBw	32bit position counter wrap/around status bit report
RCADDR RCAN CAN address report RCAN CAN baud rate report RCHN(0) RS-522 communication error flag report RCHN(1) RCHN(1) RS-685 communication error flag report RCKS RCLK Tims clock variable reporting RCOS() Report cosine as an angle RCP Cam pointer report RCTR(0) RTM-10 Second encoder/pulse & direction counter reporting RCTR(1) RCTR RCTT Report of number of cam bable in EEPROM RCTR(1) RDEA Encoder an bable in EEPROM RCTR RDEL Setting de/dt fault limit reporting RDFS in 32-bit IEEE format, afjl variable reporting RCTR REL Position deviation error report REL Position deviation error report RERA Pasition-deviation error report RERA RERC Recent command error code report RERRW RERCOM Pointer data report RESRW RERCOM Recent command error code report RESRW RERCOM Recent command error code report RES RESUME RESUME RESUME RESUME Resume program secution after passing RETURN RE	RBx()	Real-time internal index input status bit reporting
RCBAUD CAN baud rate report RCHN(0) RS-222 communication error flag report RCHN(1) RS-323 communication error flag report RCHN(1) RS-323 communication error flag report RCKS Program checksum report RCKS Program checksum report RCCK RCDS() Report cosine as an angle RCP Cam pointer report RCTR(0) Primary encoder/pulse & rain & direction counter reporting RCTR(1) RCTR(1) Report of number of cam tables in EPPROM RCTR(1) ROPA RETERION REPORT	RCADDR	· · · · · · · · · · · · · · · · · · ·
RCHN(0) R5-232 communication error flag report RCHN(1) RCHN(1) RS-8485 communication error flag report RCKS Program checksum report RCLK Ins clock variable reporting RCOS() Report cosine as an angle RCP Cam pointer report RCTR(0) RTR(1) RS-8400 error report RCTR(1) RECTT REPORT REPORT REPORT REPORT REL Setting der/dt fault limit reporting RCTR(1) REPORT REPORT REPORT REL Setting der/dt fault limit reporting RDFS In 32-bit IEEE format, af[] variable report(ng REFA Position deviation error fault limit report REL POSITION REPORT REPO	RCAN	CAN error reporting
RCHN(1) RCKS Program checksum report RCLK I'ms clock variable reporting RCOS() Report cosine as an angle RCP Cam pointer report RCTR(0) Primary encoder/pulse train & direction counter reporting RCTR(1) Second encoder/pulse train & direction counter reporting RCTR(1) REPORT of number of cam tables in EPPROM RDEA REDEA Second encoder/pulse & direction counter reporting RCTR REPROM RE	RCBAUD	CAN baud rate report
RCKS Program checksum report RCLK 1ms clock variable reporting RCOS() Report cosine as an angle RCP Cam pointer report RCTR(1) Primary encoder/pulse at all direction counter reporting RCTR(1) Second encoder/pulse & direction counter reporting RCTR(1) Second encoder/pulse & direction counter reporting RCTT Report of number of cam tables in EEPROM RDEA Description of cam tables in EEPROM RDEA Setting de/fut fault limit reporting RDFS in 32-bit IEEE format, aff] variable reporting RDFS in 32-bit IEEE format, aff] variable reporting RDFS Position deviation error raport REA Position deviation error fault limit report REA Position deviation error fault limit report REFA Position deviation error fault limit report REFA REPORD FOR POSITION PROPERTY REPORT RECORD PROPERTY RECORD PROPERTY REPORT	RCHN(0)	RS-232 communication error flag report
RCKS RCSCA RCSCA REDATE RCDA RCDA RCDA RCTR(1) REPOT Cam pointer report RCTR(1) RECTR(1) RECTR(1) RESCOND Primary encoder/pulse & direction counter reporting RCTR(1) RECTR RESPOND RCTR REPOT FOR A direction counter reporting RCTR REPOT FOR A direction counter reporting RCTT RESPOND RCTR REPOT FOR A direction counter reporting RCTT RESPOND RCTT RCTT RCTT RCTT RCTT RCTT RCTT RCT	RCHN(1)	RS485 communication error flag report
RCOS() Report cosine as an angle RCP Cam pointer report RCTR(0) Primary encoder/pulse train & direction counter reporting RCTR(1) Second encoder/pulse & direction counter reporting RCTR(1) REPORT of number of cam tables in EEPROM RDEA Encoder define the value report RDEL Setting de/dt fault limit reporting RDFS RDFS In 32-bit IEEE format, af[] variable reporting RDFS RDFA Position deviation error report REA Position deviation error report REL Position deviation error report REPORD REPRORD REPR	RCKS	Program checksum report
RCOS() Report cosine as an angle RCP Cam pointer report RCTR(0) Primary encoder/pulse train & direction counter reporting RCTR(1) Second encoder/pulse & direction counter reporting RCTR(1) Report of number of cam tables in EEPROM RDEA Escause delét value report RDEL Setting de/dt fault limit reporting RDFS RDFS In 32-bit IEEE format, af[] variable reporting RDFS RDFA REA Position deviation error report REL Position deviation error report REL Position deviation error fault limit report RERRC Recent command error code report RERRC RERRC Recent command error code report RESS RESUME RESS RESUME RESUME Resume program execution after pausing RETURN RETURN RETURN RETURN RETURN RETURN RETURN RETURN RESUME RESUME REsume program execution after pausing RESUME RESUME RESUME REsume program execution after poutine RESORT() RESPORT() Resume program execution after poutine REFURN RESPORT() Resume program communication channel REFURN RESPORT() Resume program communication channel REFURN RESPORT() Resume program communication channel REFURN RESPORT() Responsable sa he a calumnel REFURN RESPORT() Responsable report (rising edge, internal encoder) RI(1) Recoder Z phase input position variable report (rising edge, external encoder) RI(1) Recoder Z phase input position variable report (rising edge, external encoder) RI(1) Recoder Z phase input position variable report (rising edge, external encoder) RI(1) Recoder Z phase input position variable report (rising edge, external encoder) RI(1) Recoder Z phase input position variable report (rising edge, external encoder) RI(A REA RESPONDED RESPONDED RETURN RESPONDED RESPONDED RESPONDED RESPONDED RESPONDED RESPONDED RESPONDED R	RCLK	
RCTR(0) Primary encoder/pulse train & direction counter reporting RCTR(1) Second encoder/pulse & direction counter reporting RCTT Report of number of cam tables in EEPROM RDEA Execution dedit value report RDEL Setting de/dt fault limit reporting RDFS in 32-bit IEEE format, af[] variable reporting RDT Setting de/dt fault limit reporting RDT Setting developerative value report REA Position deviation error report REA Position deviation error report REA Position deviation error fault limit report REFTR EEPROM pointer data report RERRC Recent command error code report RERRW Recent command error code report RES Encoder resolution report example: a-RES RESUME Resume program execution after pausing RETURN Return from subroutine RETURNI Return from interrupt routine REFABS() Routing point absolute value error reports REFORT() Routing point absolute value error report REFORT() Routing point appoint square root report REGETCHR String reporting from main communication channel RGETCHR1 String report from secondary communication channel RHEX() Report variables as hes columns RI(0) Encoder 2 phase input position variable report (rising edge, internal encoder) RIN() L/O input report RIN() L/O input report RINA() Analog input position variable report (risling edge, external encoder) RINA() Encoder 2 phase input position variable report (risling edge, internal encoder) RINA KA buffer PID value (acceleration feedforward) report	RCOS()	Report cosine as an angle
RCTR(0) Primary encoder/pulse train & direction counter reporting RCTR(1) Second encoder/pulse & direction counter reporting RCTT Report of number of cam tables in EEPROM RDEA Esecution debet value report RDEA Setting devid fault interporting RDFS In 32-bit IEEE format, af[] variable reporting RDFS In 32-bit IEEE format, af[] variable reporting RDT Setting deviation error report REA Position deviation error report REA Position deviation error fault limit report REFIR EEPROM pointer data report REFIR EEPROM pointer data report RERRC Recent command error code report RERRW Recent command error code report RES Encoder resolution report example: a-RES RESUME Resume program execution after pausing RETURN RETURN Return from subroutine RETURNI Return from interrupt routine REFABS() Ifloating point absolute value error reporting REFSQRT() Ifloating point absolute value error reporting REFORM String reporting from main communication channel RGETCHR String reporting from main communication channel RGETCHR1 String report from secondary communication channel RHEX() Report variables as her columns RI(0) Encoder Z phase input position variable report (rising edge, external encoder) RIN() L/O input report RIN() L/O input report RIN() L/O input report RIN() Encoder Z phase input position variable report (risling edge, external encoder) RIN() Encoder Z phase input position variable report (risling edge, external encoder) RINA KA buffer PID value (acceleration feedforward) report RKA	RCP	· · · · · · · · · · · · · · · · · · ·
RCTR(1) Report of number of cam tables in EEPROM RDEA Section devict value report RDEL Setting devict fault limit reporting RDFS in 32-bit IEEE format, affly variable report REPOR RDFS RDFS RDFS RDFS RDFS RDFS RDFS RDF	RCTR(0)	
RCTT Report of number of cam tables in EEPROM RDEA RDEL Setting devidt value report RDFS In 32-bit IEEE format, af[] variable reporting RDF RDF RDF RDF RDF RDF RDF RDF RDF RD		
RDEL Setting de/dt fault limit reporting RDFS in 32-bit IEEE format, af[] variable reporting RDT setting decide after value report REA Position deviation error report REA Position deviation error report REL Position deviation error fault limit report REPTR EEPROM pointer data report RERRC Recent command error code report RERRW Recent command error code report RESS Encoder resolution report example: a=RES RESUME Resume program execution after pausing RETURN RETURN Return from subroutine RETURNI Return from interrupt routine REFASS() Floating point absolute value error reporting RFSQRT() floating point appoint supaire root report REFECT String reporting from main communication channel REFURNI String reporting from main communication channel REFECTHRI String reporting from main communication channel REFECTHRI String reporting from main communication channel REGETCHRI String reporting from main communication channel REFECTHRI String report from secondary communication channel RICO Encoder Z phase input position variable report (fising edge, external encoder) RICO Encoder Z phase input position variable report (fising edge, external encoder) RICO Encoder Z phase input position variable report (falling edge, internal encoder) RICO Encoder Z phase input position variable report (falling edge, external encoder) RICO Encoder Z phase input position variable report (falling edge, external encoder) RICO Encoder Z phase input position variable report (falling edge, external encoder) RICO Encoder Z phase input position variable report (falling edge, external encoder) RICO Encoder Z phase input position variable report (falling edge, external encoder) RICO Encoder Z phase input position variable report (falling edge, external encoder) RICO Encoder Z phase input position variable report (falling edge, external encoder)		· · · · · · · · · · · · · · · · · · ·
RDEL Setting de/dt fault limit reporting RDFS in 32-bit IEEE format, af[] variable reporting RDT Setting deviation where report REA Position deviation error report REA Position deviation error report REL Position deviation error fault limit report REPTR EEPROM pointer data report RERRC Recent command error commender ror code report RERRW Recent commander or communication channel report RESS Encoder resolution report example: a=RES RESUME RESUME Resume program execution after pausing RETURN RETURNI Return from subroutine RETURNI Return from interrupt routine RFABS() Ribating point absolute value error reporting RFSQRT() Ribating point absolute value error reporting RFSQRT() Ribating point spoint eventual error reporting RETURN String reporting from main communication channel RETURNI Return from secondary communication channel RETURNI Report variables as hex columns RICHER String report from secondary communication channel RECECHRI Report variables as hex columns RICHI Encoder Z phase input position variable report (rising edge, external encoder) RIVAL) Analog input report RIVAL) Analog input report RIVAL) Analog input report RIVAL) Recoder Z phase input position variable report (falling edge, internal encoder) RIVAL) Recoder Z phase input position variable report (falling edge, external encoder) RIVAL REVALUE RECODER SECONDARY (falling edge, external encoder) RIVAL REVALUE REPORT (falling edge, external encoder) RIVAL RECODER SECONDARY (falling edge, external encoder) RIVAL RECODER SECONDARY (falling edge, external encoder) RIVAL RECODER SECONDARY (falling edge, external encoder)	RDEA	Execution de/dt value report
RDFS In 32-bit IEEE format, af[] variable reporting RDT Setting decideration value report REA Position deviation error report REL Position deviation error report REPTR REPTR REPROM pointer data report RERRC Recent command error code report RERRW Recent command error code report RES Encoder resolution report example: a=RES RESUME RESUME RESUME RESUME RESUME RESUME RETURN RETURN RETURN RETURN RETURN RETURN RETURN RETORN RESONAL PROVING PROVING REFABS() Floating point absolute value error reporting RFSQRT() REFORT() REFORT Firmware version report RGETCHR REGETCHR String reporting from main communication channel RGETCHR1 REFEX) REPROVING Proving from secondary communication channel RHEX() Report variables as hex columns RI(0) Encoder Z phase input position variable report (rising edge, external encoder) RI(1) Encoder Z phase input position variable report (falling edge, internal encoder) RI(0) Encoder Z phase input position variable report (falling edge, external encoder) RI(1) Encoder Z phase input position variable report (falling edge, external encoder) RI(1) Encoder Z phase input position variable report (falling edge, external encoder) RI(1) Encoder Z phase input position variable report (falling edge, external encoder) RI(1) Encoder Z phase input position variable report (falling edge, external encoder) RI(1) Encoder Z phase input position variable report (falling edge, external encoder) RI(1) Encoder Z phase input position variable report (falling edge, external encoder) RI(1) Encoder Z phase input position variable report (falling edge, external encoder) RI(2) Encoder Z phase input position variable report (falling edge, external encoder) RI(3) Encoder Z phase input position variable report (falling edge, external encoder) RI(4) Encoder Z phase input position variable report (falling edge, external encoder)	RDEL	
RDT REA Position deviation error report REL Position deviation error fault limit report REPTR REPTR REPROM pointer data report RERRC Recent command error code report RERRW Recent command error communication channel report RES Encoder resolution report example: a=RES RESUME RESUME RESUME RESUME RESUME RESUME RESUME RESUME RETURN RETURNI RETURNI RETURNI RETURNI RETURNI RETURNI RETURNI RETURNI RESORT() floating point absolute value error reporting RFSQRT() RFSQRT() RFSQRT() REFURNI RETURNI RETURNI RETURNI RETURNI RETURNI RESORTE String reporting from main communication channel REFURNI REFURNI RETURNI RETURNI RETURNI RETURNI RESORTE String reporting from main communication channel REW Firmware version report RGETCHR String report from secondary communication channel REETCHR1 REPORT variables as hex columns RI(0) Encoder Z phase input position variable report (rising edge, internal encoder) RI(1) Encoder Z phase input position variable report (rising edge, external encoder) RI(0) Encoder Z phase input position variable report (falling edge, internal encoder) RI(0) Encoder Z phase input position variable report (falling edge, external encoder) RI(1) Encoder Z phase input position variable report (falling edge, external encoder) RKA KA buffer PID value (acceleration feedforward) report	RDFS	
REL Position deviation error fault limit report REPTR EEPROM pointer data report RERRC Recent command error code report RERRW Recent command error code report RES Encoder resolution report example: a=RES RESUME Resume program execution after pausing RETURN Return from subroutine RETURN Return from interrupt routine REFABS() Floating point absolute value error reporting RFSQRT() floating point square root report REFURN String reporting from main communication channel REFURN String report from secondary communication channel REFICHR String report variables as hex columns RI(0) Encoder Z phase input position variable report (rising edge, external encoder) RI(1) Encoder Z phase input position variable report (rising edge, internal encoder) RI(1) Encoder Z phase input position variable report (falling edge, internal encoder) RI(1) Encoder Z phase input position variable report (falling edge, external encoder) RI(1) Encoder Z phase input position variable report (falling edge, external encoder) RI(1) Encoder Z phase input position variable report (falling edge, external encoder) RI(1) Encoder Z phase input position variable report (falling edge, external encoder) RI(1) Encoder Z phase input position variable report (falling edge, external encoder) RI(1) Encoder Z phase input position variable report (falling edge, external encoder) RI(2) Encoder Z phase input position variable report (falling edge, external encoder) RI(3) Encoder Z phase input position variable report (falling edge, external encoder) RI(4) Encoder Z phase input position variable report (falling edge, external encoder)	RDT	
REPTR REPRC Recent command error code report RERRW Recent command error code report RES RESUME RESUM	REA	Position deviation error report
RERRC Recent command error code report RES RESUME RESUME RESUME RETURN RETURN RETURNI RETURNI RESORT() RESORT() RESORT() RESORT() RESORT() REFORM REFORM RETURN RETURN RETURNI RESORT() RESORT() RESORT() RESORT() RESORT () RESORT () REFORM RESOR	REL	Position deviation error fault limit report
RERRW RESUME RESORT() RESORT() RESORT () RESUME RESUM RESUME RESUME RESUME RESUME RESUME RESUME RESUME RESUME RESUM RESUME RESUM RESUME RESUME RESUME RESUME RESUME RESUME RESUME RESUME RESUM RESUME R	REPTR	EEPROM pointer data report
RES RESUME RESUME Resume program execution after pausing RETURN RETURN RETURNI RETURNI RETURNI RESUME RESUME RESUME RESUME RESUME RESUME RETURNI RETURNI RETURNI RETURNI RETORNI RESUME RESUME RESUME RESUME RESUME REFABS() Floating point absolute value error reporting RESQRT() RESQRT() RESQRT() RESQRT() REFUM REFUM RESTORTE String reporting from main communication channel RESCHTHR String report from secondary communication channel RHEX() Report variables as hex columns RI(0) Encoder Z phase input position variable report (rising edge, internal encoder) RI(1) Encoder Z phase input position variable report (rising edge, external encoder) RINA() Analog input report RINA() Analog input report RI(0) Encoder Z phase input position variable report (falling edge, internal encoder) RI(1) Encoder Z phase input position variable report (falling edge, external encoder) RI(1) Encoder Z phase input position variable report (falling edge, external encoder) RI(1) Encoder Z phase input position variable report (falling edge, external encoder) RI(1) Encoder Z phase input position variable report (falling edge, external encoder) RI(1) Encoder Z phase input position variable report (falling edge, external encoder) RI(1) Encoder Z phase input position variable report (falling edge, external encoder) RI(1) Encoder Z phase input position variable report (falling edge, external encoder) RI(2) Encoder Z phase input position variable report (falling edge, external encoder) RI(3) Encoder Z phase input position variable report (falling edge, external encoder) RI(2) Encoder Z phase input position variable report (falling edge, external encoder) RI(3) Encoder Z phase input position variable report (falling edge, external encoder)	RERRC	Recent command error code report
RESUME Resume program execution after pausing RETURN Return from subroutine RETURNI Return from interrupt routine RFABS() Return from interrupt routine RFSQRT() Floating point absolute value error reporting RFSQRT() floating point square root report RFW Firmware version report RGETCHR String reporting from main communication channel RGETCHR1 String report from secondary communication channel RHEX() Report variables as hex columns RI(0) Encoder Z phase input position variable report (rising edge, internal encoder) RI(1) Encoder Z phase input position variable report (rising edge, external encoder) RIN() I/O input report RINA() Analog input report RI(0) Encoder Z phase input position variable report (falling edge, internal encoder) RI(1) Encoder Z phase input position variable report (falling edge, external encoder) RI(1) Encoder Z phase input position variable report (falling edge, external encoder) RI(1) Encoder Z phase input position variable report (falling edge, external encoder) RI(1) Encoder Z phase input position variable report (falling edge, external encoder) RI(1) Encoder Z phase input position variable report (falling edge, external encoder) RI(A) Encoder Z phase input position variable report (falling edge, external encoder) RI(A) Encoder Z phase input position variable report (falling edge, external encoder) RI(A) Encoder Z phase input position variable report (falling edge, external encoder)	RERRW	Recent command error communication channel report
RETURN Return from subroutine RETURNI Return from interrupt routine RFABS() Floating point absolute value error reporting RFSQRT() floating point square root report RFW Firmware version report RGETCHR String reporting from main communication channel RGETCHR1 String report from secondary communication channel RHEX() Report variables as hex columns RI(0) Encoder Z phase input position variable report (rising edge, internal encoder) RI(1) Encoder Z phase input position variable report (rising edge, external encoder) RIN() I/O input report RINA() Analog input report RI(0) Encoder Z phase input position variable report (falling edge, internal encoder) RI(1) Encoder Z phase input position variable report (falling edge, external encoder) RINA() Fincoder Z phase input position variable report (falling edge, external encoder) RI(1) Encoder Z phase input position variable report (falling edge, external encoder) RI(1) Encoder Z phase input position variable report (falling edge, external encoder) RI(2) Encoder Z phase input position variable report (falling edge, external encoder) RI(3) Encoder Z phase input position variable report (falling edge, external encoder) RI(4) Encoder Z phase input position variable report (falling edge, external encoder) RI(5) Encoder Z phase input position variable report (falling edge, external encoder) RI(6) Encoder Z phase input position variable report (falling edge, external encoder)	RES	Encoder resolution report example: a=RES
RETURNI Return from interrupt routine RFABS() Floating point absolute value error reporting RFSQRT() floating point square root report RFW Firmware version report RGETCHR String reporting from main communication channel RGETCHR1 String report from secondary communication channel RHEX() Report variables as hex columns RI(0) Encoder Z phase input position variable report (rising edge, internal encoder) RI(1) Encoder Z phase input position variable report (rising edge, external encoder) RIN() I/O input report RINA() Analog input report RJ(0) Encoder Z phase input position variable report (falling edge, internal encoder) RJ(1) Encoder Z phase input position variable report (falling edge, external encoder) RJ(1) Encoder Z phase input position variable report (falling edge, external encoder) RJ(1) Encoder Z phase input position variable report (falling edge, external encoder) RJ(1) Encoder Z phase input position variable report (falling edge, external encoder) RKA KA buffer PID value (acceleration feedforward) report RKC	RESUME	Resume program execution after pausing
RFABS() RFAGRT() RFSQRT() RFSQRT() RFSQRT() REFU REFU REFU REFU REFU REFU REFU REFU REFU String report from secondary communication channel REFU REFU REPORT variables as hex columns RI(0) Encoder Z phase input position variable report (rising edge, internal encoder) RIN() RIN() RINA() Report variables report (rising edge, external encoder) RIN() RINA() Report variables report (rising edge, external encoder) RIN() RINA() Report variable report (rising edge, external encoder) RIN() RINA() Report variable report (rising edge, internal encoder) RIN() RINA() Report variable report (rising edge, external encoder) RINA() RINA() Report variable report (rising edge, external encoder) RINA() RINA() Report variable report (falling edge, internal encoder) RINA() RINA() RINA() Report variable report (falling edge, external encoder) RINA()	RETURN	Return from subroutine
RFSQRT() RFW Firmware version report RGETCHR String report from secondary communication channel RGETCHR1 String report from secondary communication channel RHEX() Report variables as hex columns RI(0) Encoder Z phase input position variable report (rising edge, internal encoder) RI(1) Encoder Z phase input position variable report (rising edge, external encoder) RIN() I/O input report RINA() Analog input report RINA() Analog input report (falling edge, internal encoder) RJ(1) Encoder Z phase input position variable report (falling edge, external encoder) RINA() KA buffer PID value (acceleration feedforward) report RKC KC value report	RETURNI	Return from interrupt routine
RFW Firmware version report RGETCHR String reporting from main communication channel RGETCHR1 String report from secondary communication channel RHEX() Report variables as hex columns RI(0) Encoder Z phase input position variable report (rising edge, internal encoder) RI(1) Encoder Z phase input position variable report (rising edge, external encoder) RIN() I/O input report RINA() Analog input report RINA() Analog input report RJ(0) Encoder Z phase input position variable report (falling edge, internal encoder) RJ(1) Encoder Z phase input position variable report (falling edge, external encoder) RJ(1) Encoder Z phase input position variable report (falling edge, external encoder) RKA KA buffer PID value (acceleration feedforward) report RKC (value report	RFABS()	Floating point absolute value error reporting
RGETCHR RGETCHR1 String report from secondary communication channel RHEX() Report variables as hex columns RI(0) Encoder Z phase input position variable report (rising edge, internal encoder) RI(1) Encoder Z phase input position variable report (rising edge, external encoder) RIN() I/O input report RINA() Analog input report RJ(0) Encoder Z phase input position variable report (falling edge, internal encoder) RJ(1) Encoder Z phase input position variable report (falling edge, internal encoder) RJ(1) Encoder Z phase input position variable report (falling edge, external encoder) RKA KA buffer PID value (acceleration feedforward) report RKC	RFSQRT()	floating point square root report
RGETCHR1 RHEX() Report variables as hex columns RI(0) Encoder Z phase input position variable report (rising edge, internal encoder) RI(1) Encoder Z phase input position variable report (rising edge, external encoder) RIN() I/O input report RINA() Analog input report RINA() Analog input report (falling edge, internal encoder) RJ(1) Encoder Z phase input position variable report (falling edge, internal encoder) RJ(1) Encoder Z phase input position variable report (falling edge, internal encoder) RJ(1) Encoder Z phase input position variable report (falling edge, external encoder) RKA KA buffer PID value (acceleration feedforward) report RKC KC value report	RFW	Firmware version report
RHEX() Report variables as hex columns RI(0) Encoder Z phase input position variable report (rising edge, internal encoder) RI(1) Encoder Z phase input position variable report (rising edge, external encoder) RIN() I/O input report RINA() Analog input report RJ(0) Encoder Z phase input position variable report (falling edge, internal encoder) RJ(1) Encoder Z phase input position variable report (falling edge, external encoder) RKA KA buffer PID value (acceleration feedforward) report RKC KC value report	RGETCHR	String reporting from main communication channel
RI(0) Encoder Z phase input position variable report (rising edge, internal encoder) RI(1) Encoder Z phase input position variable report (rising edge, external encoder) RIN() I/O input report RINA() Analog input report RJ(0) Encoder Z phase input position variable report (falling edge, internal encoder) RJ(1) Encoder Z phase input position variable report (falling edge, external encoder) RKA KA buffer PID value (acceleration feedforward) report RKC KC value report	RGETCHR1	String report from secondary communication channel
RI(1) Encoder Z phase input position variable report (rising edge, external encoder) RIN() I/O input report RINA() Analog input report RJ(0) Encoder Z phase input position variable report (falling edge, internal encoder) RJ(1) Encoder Z phase input position variable report (falling edge, external encoder) RKA KA buffer PID value (acceleration feedforward) report RKC KC value report	RHEX()	Report variables as hex columns
RIN() RINA() Analog input report RJ(0) Encoder Z phase input position variable report (falling edge, internal encoder) RJ(1) Encoder Z phase input position variable report (falling edge, external encoder) RKA KA buffer PID value (acceleration feedforward) report RKC KC value report	RI(0)	Encoder Z phase input position variable report (rising edge, internal encoder)
RINA() Analog input report RJ(0) Encoder Z phase input position variable report (falling edge, internal encoder) RJ(1) Encoder Z phase input position variable report (falling edge, external encoder) RKA KA buffer PID value (acceleration feedforward) report RKC KC value report	RI(1)	Encoder Z phase input position variable report (rising edge, external encoder)
RJ(0) Encoder Z phase input position variable report (falling edge, internal encoder) RJ(1) Encoder Z phase input position variable report (falling edge, external encoder) RKA KA buffer PID value (acceleration feedforward) report RKC KC value report	RIN()	I/O input report
RJ(1) Encoder Z phase input position variable report (falling edge, external encoder) RKA KA buffer PID value (acceleration feedforward) report RKC KC value report	RINA()	Analog input report
RJ(1) Encoder Z phase input position variable report (falling edge, external encoder) RKA KA buffer PID value (acceleration feedforward) report RKC KC value report	RJ(0)	Encoder Z phase input position variable report (falling edge, internal encoder)
RKC KC value report		Encoder Z phase input position variable report (falling edge, external encoder)
	RKA	KA buffer PID value (acceleration feedforward) report
RKCS KCS value report	RKC	KC value report
	RKCS	KCS value report

RKD	KD buffer PID value (differential) report
RKG	KG buffer PID value (gravity) report
RKI	KI buffer PID value (integral) report
RKL	KL buffer PID value (integral limit) report
RKP	KP buffer PID value (proportional) report
RKS	KS buffer PID value (integral filter control) report
RKV	KV buffer PID value (velocity feedforward) reporting
RLEN	Main communication channel buffer occupancy level (data mode) report
RLEN1	Secondary communication channel buffer occupancy level (data mode) report
RLFS()	32-bit IEEE format Float value reporting
RMFDIV	Tracking mode divisor report
RMFMUL	Follow mode multiplier report
RMODE	Operating mode report
RMODE()	Operation mode (specific trajectory) report
ROC()	Output status (24V I/O) report
ROF()	Output fault (24V I/O) reporting
RPA	Execution position report
RPC	
RPC()	Command motor position report
RPI	Command motor position (specific trajectory) report
RPMA	Report pi numbers
RPML	Execution position modulo report
RPMT	Position modulo limit report
	Position modulo target (position motion) reporting
RPRA	Operation start actual position report
RPRC	Operation start command position report
RPRT	Relative target position report
RPT	Setting target position report
RPTSD	Synchronous linear motion distance report
RPTST	Synchronous operation time report (ms)
RRANDOM	Random number report
RRES	Encoder resolution report
RSAMP	Sample rate (Hz) report
RSIN()	Report sine function in degrees
RSLM	Soft limit mode report
RSLN	Soft limit right/forward setting report
R.S.L.P.	Soft limit left/negative direction setting report
RSP	Sample rate, firmware number reporting
RSP1	Firmware compilation date and time report
RSP2	Bootloader revision report
RSQRT()	Integer square root value reporting
RT	Request torque value report
RTAN()	Report tangent function in degrees
RTEMP	Internal temperature report
RTH	Temperature limit setpoint report
RTHD	Current limit timer setting value report
RTMR()	User timer value report
RTRQ	Real-time torque value reporting
RTS	Torque slope setting value report
RUIA	Current value report
RUJA	Voltage value report

RUN	Program execution
RUN?	The program will stop unless a RUN command is sent after the power is turned on.
RVA	Execution speed report (filter value)
R.V.C.	Command speed value report
RVL	Speed limit report
RVT	Specified speed report
RW()	Specific status word reporting
S	
S	sudden stop
S()	Sudden stop (orbit specification)
SADDR#	Motor addressing
SAMP	Sampling rate (Hz)
SILENT	Ignore commands received on port 1
SILENT1	Ignore commands received on port 2
SIN()	Get sine function in degrees
S.L.D.	Disable software limits
SLE	Enable software limits
SLEEP	Start sleep mode on port 1
SLEEP1	Start sleep mode on port 2
SLM	software limit
SLM()	Set software limits
SLN	Left direction software limit
SLN=	Set left direction software limit
SLP	Right direction software limit
SLP=	Right direction software limit setting
SQRT()	integer square root
SRC()	Tracking and/or cam encoder source settings
STACK	Nesting reset
STDOUT=	Set the output destination of the report command
SWITCH	Program flow command
Т	
Т	Setting torque value
T=	Torque value setting
TALK	Enable PRINT message output to port 1
TALK1	Enable PRINT message output to port 2
TAN()	Get tangent function in degrees
TEMP	temperature
T.H.	temperature limit
TH=	Temperature limit setting
THD	Obtains a timer until the drive power is turned off when the temperature exceeds (Bh)
THD=	Timer setting until drive power is turned off when temperature exceeds (Bh)
TMR()	Specific timer setting time, e.g. a=TMR(0)
TMR() (as cmd)	Timer settings, e.g. TMR(0,1000) = set timer 0 to 1 second
T.S.	Torque slope setting value
TS=	Torque ramp setting
TSWAIT	Synchronous operation in progress Standby
TWAIT	Operating Standby
TWAIT()	Operating Standby (orbit specification)
U	
UIA	Motor current value

UJA	bus voltage
UO()=	Set one or more status bits to a specific value
UP	Upload user EEPROM program
UPLOAD	Upload user EEPROM readable program
UR()	Set one or more status bits to 0
US()	Set one or more status bits to 1
V	
V.A.	execution speed
VAC()	Speed filter settings
V.C.	Specified speed
VL	Specified speed limit
VL=	Speed limit setting
VLD()	Reading from non-volatile memory
VST()	Writing to non-volatile memory
VT	target speed
VT=	Set target speed
VTS=	Set target speed for synchronous operation
W	
W()	Report status word
WAIT=	Wait for specified time (specified in msec)
WAKE	Exit port 1 sleep mode
WAKE1	Exit port 2 sleep mode
WHILE	infinite loop command
X	
X	deceleration stop
X()	Deceleration and stop (specify trajectory)
Z	
Z	Reset all errors
Z()	Reset specific errors
Za	Current error reset
Ze	Deviation error reset
Zh	Temperature error reset
ZI	Hardware (sensor input) limit history Left direction status bit release
Zls	Software limit history left direction status bit release
Zr	Hardware (sensor input) limit history Right direction status bit release
Zrs	Software limit history Right direction status bit release
Zs	System tech error bit release
ZS	Clear all errors
Zv	Speed error release
Zw	Encoder value wraparound latch bit reset
	•