DENSO
RAC communication specifications

Version 1.1.12

February 9, 2012

【 Remarks 】

【 Revision history 】

Date	Number of versions	Content
2012-02-09	1.0.0	First edition

Contents

1. Introduction

This book is a user's guide of DENSO RAC mounted on RC8.

RAC(Robot Action Command) is the one developed aiming at a united robot language to move the robot.

The RAC message is transmitted to the robot by using the socket communication (TCP/IP) from the client application. In DENSO RAC, the RAC message is transmitted to RC8 by using the socket communication. RC8 analyze the RAC message that has been transmitted to, and acquires and sets the variable corresponding to the content of the message.

2. RAC specification

RAC rules two types of character strings. "Request character string" is to issue a request from the client to the server. "Response character string" is to response to a request from the server to the client.

This chapter explains the specification of these character strings.

2.1. RAC character string

2.1.1. Common specification

RAC character string consists of the request character string and the response character string. The following shows the common specification of these strings.

- Data shall be a text
- One command (Terminator (CR) is included) shall be within 256 bytes.
- Space and the Tab character (indent etc.) that attaches to the head of line shall be disregarded.

2.1.2. Request character string

The request character string of RAC is described as follows. Words written in square blankets is omittable.

<top-level command> :[<part name>]:[<part number>]:[<subcommand>]:[<parameter>]<terminator>

< top-level command >	:	DENSO RAC server supports “PUT" and “GET" only. “PUT"command sets variables. “GET" commands acquires variables.
< part name >	:	Controller name (Fixed as “RC8")
< part number >	:	Variable index number
< subcommand >	:	Variable type The following variable types can be specified. “I",“F",“D",“S",“V",“P",“J",“T",“IO"
< parameter >	:	Data Specify the value that is set to the variable when specifying “PUT" command. This string is not used when “GET" command is specified. Details are shown in 2.2 with regards to data discription.
< terminator >	:	Terminator Fixed as CR(0x0D)

2.1.3. Response character string

The response character string of RAC is described as follows.Words written in square blankets is omittable.

< processing result > [< response data >]< terminator >

< processing result >	:	Show the execution result of the RAC command by the integral

value of the HRESULT type.

< response data > : Acquire the value of the variable specified by“GET" command.
This data is not used in “PUT" command..
Details are shown in 2.2 with regards to data description

< terminator > : Terminator
Fixed as CR(0x0D)

2.2. Description method of data

Both the parameter of the request character string and the response data of the response character string are followed the method of expressing the VARIANT type of ORiN2 by the character string. The format expresses the data type and the data string, delimited by commas as follows.

< data type >,< data string >

The integral value written by the VARTYPE type is shown in < data type >. Table 2-1 shows available data types and values.

Table2-1 Available data types

Data type	Value	Explanation
VT_EMPTY	0	Empty data
VT_NULL	1	NULL value
VT_ERROR	10	Error code
VT_UI1	17	Byte type
VT_I2	2	Short integer
VT_UI2	18	Unsigned short integer
VT_I4	3	Length integer
VT_UI4	19	Unsigned length integer
VT_R4	4	Single-precision floating point
VT_R8	5	Double-precision floating point
VT_CY	6	Currency type
VT_DATE	7	Date type
VT_BOOL	11	Boolean type
VT_BSTR	8	Character string type
VT_VARIANT	12	Variant type The same one as the argument part enters data. Available only at VT_ARRAY.
VT_ARRAY	0x2000	Array

		Specify bitwise OR with other data types. In the data string, write the data by the character string. To express the array data, use”,” (comma) to delimit.

e.g. 1)	2,100	Type:	VT_I2	Value:	100
e.g. 2)	8,Sample	Type:	VT_BSTR	Value:	Sample
e.g. 3)	12,(8,Sample)	Type:	VT_VARIANT	Value:	“Sample" (character string type)
e.g. 4)	8194,100,200,300	Type:	VT_I2|VTARRAY	Value:	100,200,300
e.g. 5)	8200,Sample,Test	Type:	VT_BSTR|VTARRAY	Value:	“Sample",“Test"
e.g. 6)	8,Sample,Test	Type:	VT_BSTR	Value:	“Sample,Test"
e.g. 7)	8204,(8,Sample),(2,100)	Type:	VT_VARIANT|VTARRAY	Value:	“Sample",100

2.3. Return code

The return code is assigned in b-CAP as follows.

Table2-2 Assignment of return code

Return code	Explanation
0x00000000～0x8000FFFF 0x80010000～0x800101FF 0x80070000～0x8007FFFF	Default return code, reservation area
0x80040200～0x8004FFFF	User definition error

When creating return codes other than the following "default return code", an arbitrary code can be assigned within the range of the value of "User definition error".

Table2-3 Default return code list

Return code	Error	Explanation
0x00000000	S_OK	Normal termination
0x80004001	E_NOTIMPL	Unimplemented function
0x80004004	E_ABORT	The function was aborted.
0x80004005	E_FAIL	The function failed.
0x8000FFFF	E_UNEXPECTED	A fatal error occurred.
0x80010001	E_INVALIDRCVPACKET	n packet that the robot controller received is invalid. When this error occurs, the robot

		controller automatically cuts the connection. Please confirm the packet transmitted to the robot controller.
0x80010002	E_INVALIDSNDPACKET	The transmission packet is invalid.
0x80010003	E_INVALIDARGTYPE	The argument type in the received packet is invalid.
0x80010004	E_ROBOTISBUSY	A new operation instruction was received while the robot was working. It is not possible to operate.
0x80010005	E_INVALIDCOMMAND	Invalid character string was received as a command. It is not possible to execute.
0x80010011	E_PACKETSIZEOVER	The size of the received packet is invalid. (>16Mbytes)
0x80010012	E_ARGSIZEOVER	The argument size of the reception packet is invalid. (>16Mbytes)
0x80070005	E_ACCESSDENIED	It is not possible to access it.
0x80070006	E_HANDLE	The handle is invalid.
0x8007000E	E_OUTOFMEMORY	The memory is insufficient.
0x80070057	E_INVALIDARG	The argument is invalid.

3. Communication procedure

3.1. Communication sequence

The sequence of RAC starts by transmitting the request packet from the client. The server side executes the function of the request packet, and returns the client the response packet.

In one session, after the request message is transmitted, it is necessary to wait for the reception of the response message to synchronize. When two or more request messages are used, it is recommended to do by two or more sessions.

On the server side, there is no particular latency-time regulation from the time of request packet reception to sending a response. Therefore, be aware that the time-out is occured on the client side when the processing time of the server side takes long time meanwhile the time-out detection time of the client is short.

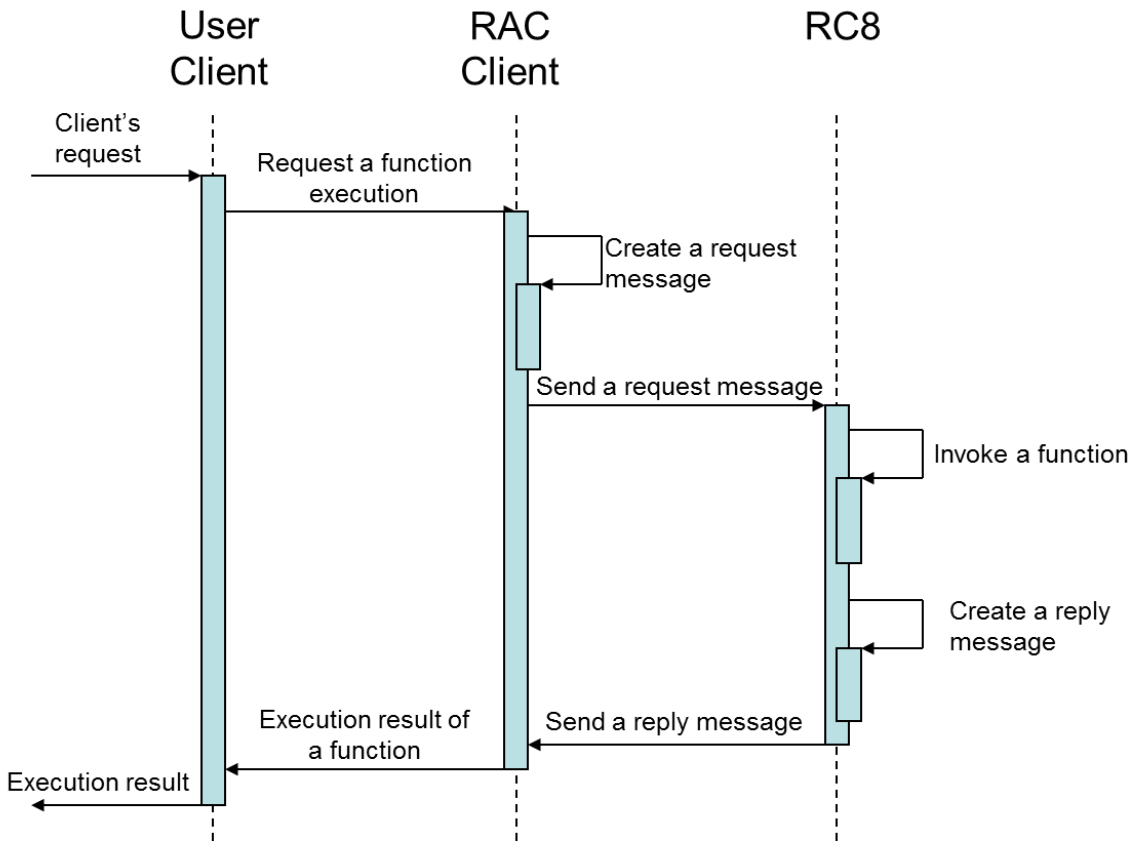

Figure3-1Communication sequence

3.2. Communication procedure of client

The outline of the communication procedure of the client is shown as follows.

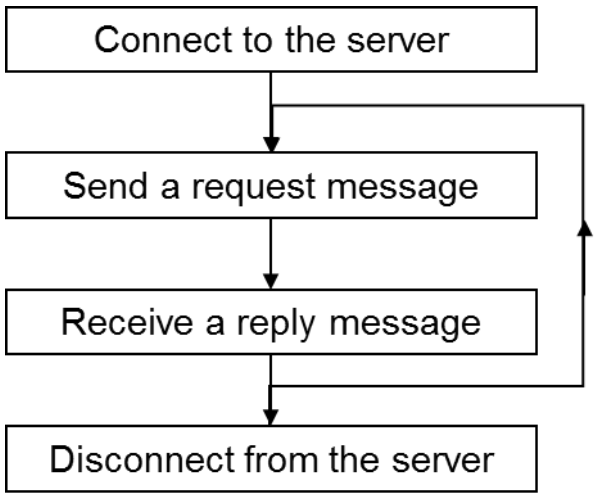

Figure3-2Communication procedure of client

The client connects to the #5006 port of the TCP of the server in order to establishe the session first..

After the connection, send the request message of the excuted function and wait for the execution result from the server.

The client should process the time-out when there is no response from the server. However, when setting the time-out time, please keep in mind that the response time of the server is different depends on the content of proceeding.

4. RAC command

4.1. GET command

・ Request character string

GET:RC8:<Index number>:<Variable type>:<CR>

・ Response character string

<Processing result>,<Response data type>,<Response data><CR>

Communication sample 1		
Acquire the value of I type variable (The index number: 10)		
Transmission Command	Client → server: GET:RC8:10:I:<CR>	
	Parameter	Value
	Index number	10
	Variable type	I
Reception Command	Server → client: 0,3,123<CR>	
	Argument	Value
	Processing result	0 (success)
	Response data type	VT_I4
	Response data	123

Communication sample 2		
Acquire the value of F type variable (The index number: 10) .		
Transmission Command	Client → server: GET:RC8:10:F:<CR>	
	Parameter	Value
	Index number	10
	Variable type	F
Reception Command	Server → client: 0,4,123.01<CR>	
	Argument	Value
	Processing result	0 (success)
	Response data type	VT_R4

	Response data	123.01

Communication sample 3		
Acquire the value of D type variable (The index number: 10).		
Transmission Command	Client → server: GET:RC8:10:D:<CR>	
	Parameter	Value
	Index number	10
	Variable type	D
Reception Command	Server → client: 0,5,123.01<CR>	
	Argument	Value
	Processing result	0 (success)
	Response data type	VT_R8
	Response data	123.01

Communication sample 4		
Acquire the value of S type variable (The index number: 10).		
Transmission Command	Client → server: GET:RC8:10:S:	
	Parameter	Value
	Index number	10
	Variable type	S
Reception Command	Server → client: 0,8,Test<CR>	
	Argument	Value
	Processing result	0 (success)
	Response data type	VT_BSTR
	Response data	“Test"

Communication sample 5		
Acquire the value of V type variable (The index number: 10).		
Transmission Command	Client → server: GET:RC8:10:V:<CR>	
	Parameter	Value

	Index number	10
	Variable type	V
Reception Command	Server → client: 0,8196,1,2,3<CR>	
	Argument	Value
	Processing result	0 (success)
	Response data type	VT_ARRAY | VT_R4
	Response data	(1, 2, 3)

Communication sample 6		
Acquire the value of P type variable (The index number: 10).		
Transmission Command	Client → server: GET:RC8:10:P:<CR>	
	Parameter	Value
	Index number	10
	Variable type	P
Reception Command	Server → client: 0,8196,1,2,3,4,5,6,-1<CR>	
	Argument	Value
	Processing result	0 (success)
	Response data type	VT_ARRAY | VT_R4
	Response data	(1, 2, 3, 4, 5, 6, -1)

Communication sample 7		
Acquire the value of J type variable (The index number: 10).		
Transmission Command	Client → server: GET:RC8:10:J:<CR>	
	Parameter	Value
	Index number	10
	Variable type	J
Reception Command	Server → client: 0,8196,1,2,3,4,5,6,7,8<CR>	
	Argument	Value
	Processing result	0 (success)
	Response data type	VT_ARRAY | VT_R4

	Response data	(1, 2, 3, 4, 5, 6, 7, 8)

Communication sample 8		
Acquire the value of T type variable (The index number: 10).		
Transmission Command	Client → server: GET:RC8:10:T:<CR>	
	Parameter	Value
	Index number	10
	Variable type	J
Reception Command	Server → client: 0,8196,1,2,3,4,5,6,7,8,9,-1<CR>	
	Argument	Value
	Processing result	0 (success)
	Response data type	VT_ARRAY | VT_R4
	Response data	(1, 2, 3, 4, 5, 6, 7, 8, 9, -1)

Communication sample 9		
Acquire the value of the IO type variable (The index number: 10).		
Transmission Command	Client → server: GET:RC8:10:IO:<CR>	
	Parameter	Value
	Index number	10
	Variable type	J
Reception Command	Server → client: 0,11,0<CR>	
	Argument	Value
	Processing result	0 (success)
	Response data type	VT_BOOL
	Response data	FALSE

4.2. PUT command

・ Request character string

PUT:RC8:<Index number>:<Variable type>:<Transmission data type>,<Transmission data><CR>

・ Response character string

< processing result >< CR >

Communication sample 1		
Set the value of I type variable (The index number: 10).		
Transmission Command	Client → server: PUT:RC8:10:I:3,123<CR>	
	Parameter	Value
	Index number	10
	Variable type	I
	Transmission data type	VT_I4
	Transmission data	123
Reception Command	Server → client: 0<CR>	
	Argument	Value
	Processing result	0 (success)

Communication sample 2		
Set the value of F type variable (The index number: 10).		
Transmission Command	Client → server: PUT:RC8:10:F:4,123.01<CR>	
	Parameter	Value
	Index number	10
	Variable type	F
	Transmission data type	VT_R4
	Transmission data	123.01
Reception Command	Server → client: 0<CR>	
	Argument	Value
	Processing result	0 (success)

Communication sample 3
Set the value of D type variable (The index number: 10).

Transmission Command	Client → server: PUT:RC8:10:D:5,123.01<CR>	
	Parameter	Value
	Index number	10
	Variable type	D
	Transmission data type	VT_R8
	Transmission data	123.01
Reception Command	Server → client: 0<CR>	
	Argument	Value
	Processing result	0 (success)

Communication sample 4		
Set the value of S type variable (The index number: 10).		
Transmission Command	Client → server: PUT:RC8:10:S:8,Test<CR>	
	Parameter	Value
	Index number	10
	Variable type	S
	Transmission data type	VT_BSTR
	Transmission data	“Test"
Reception Command	Server → client: 0<CR>	
	Argument	Value
	Processing result	0 (success)

Communication sample 5		
Set the value of V type variable (The index number: 10).		
Transmission Command	Client → server: PUT:RC8:10:V: 8196,1,2,3<CR>	
	Parameter	Value
	Index number	10
	Variable type	V

	Transmission data type	VT_ARRAY | VT_R4
	Transmission data	(1, 2, 3)
Reception Command	Server → client: 0<CR>	
	Argument	Value
	Processing result	0 (success)

Communication sample 6		
Set the value of P type variable (The index number: 10).		
Transmission Command	Client → server: PUT:RC8:10:P:8196,1,2,3,4,5,6,-1<CR>	
	Parameter	Value
	Index number	10
	Variable type	P
	Transmission data type	VT_ARRAY | VT_R4
	Transmission data	(1, 2, 3, 4, 5, 6, -1)
Reception Command	Server → client: 0<CR>	
	Argument	Value
	Processing result	0 (success)

Communication sample 7		
Set the value of J type variable (The index number: 10).		
Transmission Command	Client → server: PUT:RC8:10:J:8196,1,2,3,4,5,6,7,8<CR>	
	Parameter	Value
	Index number	10
	Variable type	J
	Transmission data type	VT_ARRAY | VT_R4
	Transmission data	(1, 2, 3, 4, 5, 6, 7, 8)
Reception Command	Server → client: 0<CR>	

	Argument	Value
	Processing result	0 (success)

Communication sample 8		
Set the value of T type variable (The index number: 10).		
Transmission Command	Client → server: PUT:RC8:10:T:8196,1,2,3,4,5,6,7,8,9,-1<CR>	
	Parameter	Value
	Index number	10
	Variable type	J
	Transmission data type	VT_ARRAY | VT_R4
	Transmission data	(1, 2, 3, 4, 5, 6, 7, 8, 9, -1)
Reception Command	Server → client: 0<CR>	
	Argument	Value
	Processing result	0 (success)

Communication sample 9		
Set the value of the IO type variable (The index number: 10).		
Transmission Command	Client → server: PUT:RC8:10:IO:11,0<CR>	
	Parameter	Value
	Index number	10
	Variable type	J
	Transmission data type	VT_BOOL
	Transmission data	FALSE
Reception Command	Server → client: 0<CR>	
	Argument	Value
	Processing result	0 (success)