Robot Information Protocol (rip)

Date: **8 April 2008**

Version **1.6**

Author: **Daniel Gavelle**

Customer: **Airbus Spain**

Projects: **Brutus (Kuka)**
Low pressure wheel probe (Crawler)

Distribution: **Alvaro Espada (Airbus Spain)**
Federico Moreno-Farinas (Airbus Spain)
Natalia SUAREZ-FERNANDEZ (Airbus Spain)
Jose-ignacio SANZ-CARRERA (Airbus Spain)
Tom Marshall (NDT Solutions)
Daniel Gavelle (NDT Solutions)
Fernando Matia (Universidad Politécnica de Madrid (UPM) Crawler)
Gustavo Munoz (UPM Crawler)
Manuel Ferre (UPM Brutus – Kuka)
Matias Collar Narocki (UPM Brutus – Kuka)
Dion Jansen (Focal Point NDE – Inspection Ware)

NDT Solutions Ltd
Dunston Innovation Centre
Dunston Road
Chesterfield, U.K. S41 8NG

Tel: +44 (0) 1246 267550
Fax: +44 (0) 1246 269381

ndts@ndtsolutions.com
www.ndtsolutions.com

Contents

Revision History

Version	Date	Author	Comment
1.0	25 Apr. 07	Daniel Gavelle	Initial version
1.1	08 May 07	Daniel Gavelle	Updated with suggestions in email from Fernando Matia
1.2	30 May 07	Daniel Gavelle	Updated with suggestions in email from Matias Collar Narocki
1.3	29 Jun 07	Daniel Gavelle	Added RTQ and RTI commands for IW to get route info from robot.
1.4	3 Oct 07	Daniel Gavelle	Added block diagram. Corrections and minor changes.
1.5	23 Nov 07	Daniel Gavelle	Clarifications from the meeting at Airbus. Addition of ENC message. Addition of Overlap section
1.6	02 April 07	Daniel Gavelle	PAU is valid following INI as well as RUN. Added HOM command. INI can be sent at any time to reset robot controller

Introduction

The Robot information Protocol (rip) defines a method of communication between the Inspection Ware ultrasonic inspection application and a robot or crawler. Rip will be used on both the Brutus project to control a Kuka robot and the low pressure wheel probe project to control the UPM crawler. The document begins by explaining terms used in the rip specification. It then discusses the protocol. Finally, configuration is discussed.

Specification of terms

The following terms are defined as part of rip:-

Inspection Ware (IW)

Inspection Ware (IW) is the application that will be used to control the ultrasonic inspection. IW will also provide the Graphical User Interface (GUI) used by the operator controlling the system. IW will use the Rapid Scan 2 Automated (RS2A) .dll to access the ultrasonic hardware of the Rapid Scan 2 computer.

Rapid Scan 2 (RS2)

The Rapid Scan 2 software is an alternative application that will be used to control the ultrasonic inspection for the crawler project. It provides a GUI for the operator and sends RIP commands to the robot control computer to control the inspection.

Robot

The term “robot” in the remainder of this document refers to either the Kuka robot and the crawler. Robot does not refer to the physical robot but rather the computer that is participating in rip that communicates with the robot or robot controller in a robot specific way.

Coordinate

A coordinate is used to represent a point and orientation in 3D space. It is expressed as 6 values:-
x, y, z, a, b, c
Coordinates are expressed in SI units (Metres and Radians). When coordinates are transferred, they are represented by 8 bit ASCII text. Each number consists of an optional +/- sign, up to three digits before the decimal point, a decimal point and up to 10 digits after the decimal point (a buffer of at least 15 bytes per number). Since the distance is measured in Metres, this gives a range of 999 Metres and an accuracy of 0.0000000001 Metres or 0.0000001 mm. The number of digits is the maximum number and the following are all valid numbers 2, 20.6, -5, -102.0123456789, 0.1.

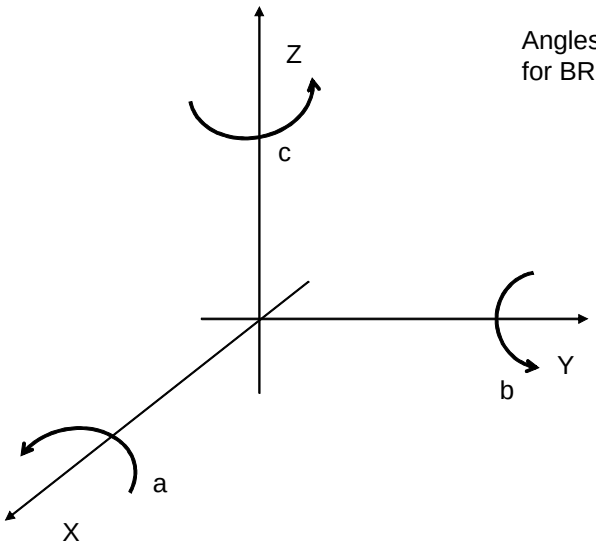
Coordinates are exchanged with a , and no spaces between each number. Numbers must NOT be written in exponent form (1.002E2). The absolute minimum buffer size for a coordinate is (15*6 + 6 = 96 bytes.

x, y, z are the coordinates of the point. In Brutus, these coordinates are in the coordinate system of the CAD data. On the crawler, x=0, y=0 represents the bottom left of the part and the z is zero.

a,b,c are the angles of rotation. a is the angle of rotation about the X axis, b the angle of rotation about the y and c the angle of rotation about the z.

In the crawler, a and b will be zero with c being the angle of rotation of the crawler. Zero radians of the c angle is defined as pointing in the direction of increasing X. Looking down on the crawler, the angle c increases as the crawler is rotated anticlockwise. The angle decreases as the crawler rotates clockwise. The angle is always a positive number between 0 and 2*PI radians..

Brutus will use all six values, as illustrated below.



Angles of rotation (a, b, c)
for BRUTUS project

May 14, 2007

Route

A route is defined by two coordinates. The wheel probe is moved along this route in as straight a line as possible over the part. A straight line will ensure that the wheel probe rolls forwards or backwards correctly and that errors due to interpolation of the route are minimised. It is particularly important to make sure that the encoder wheel is rotating forwards or backwards all the time and is not forced to skid sideways.

Inspection

An inspection consists of a sequence of straight line routes that covers the part. Once a route has been completed, the robot can choose how to move the probe from the end of one route to the start of the second route.

In Brutus, this may mean lifting the probe away from the part, moving the probe and putting it down on the part at the start of the next route. If the following route begins in an appropriate position relative to the end of the current route, there will be no need to lift off the part.
In the crawler, the movement between the two routes must ensure that the crawler is at the correct angle when it arrives at the start of the next route.

Trajectory

When the robot is commanded to follow a route, it will not necessarily follow an exact straight line from the start point to the end point. The Brutus robot will track the surface of the part in 3D space. A robot may also have to start or finish in a slightly different position due to the physical part not exactly matching the data used to specify the route. The robot will record the trajectory that is actually followed when completing the route. This trajectory is a list of coordinates. The first coordinate is the exact actual starting point of the route. The last coordinate of a trajectory is the actual end point. These may differ slightly from the start and end coordinates specified in the route.

There is only a single linear encoder that accurately records the position of each ultrasonic sample[1]. The position information from the robot is transmitted at a lower rate and is not tied accurately to the ultrasonic data. For example at 100mm/sec there will be 100 linear encoder positions a second and maybe 5 position information messages from the robot. Therefore there will be 20 times as many encoder values as 3D position points.

During the initial development of the project, IW will use the encoder data as a straight line interpolation between the 3D position points from the robot. IW will not use the CAD data for this interpolation but will take the Euclidian distance[2] between the two points. The Brutus robot should be programmed to follow the geodesic distance[3] between the start and end of the route as accurately as possible. In order to minimise these interpolation errors, the robot should be programmed to move more slowly around areas where the Euclidian distance differs significantly from the geodesic distance, for example areas of steep curvature. Errors can be further reduced by making a section consisting of several routes around problem areas.

A trajectory is sent in rip as a series of coordinates, using POS messages. It was originally proposed to store the trajectory in a file, however this is likely to lead to file locking complications. An example trajectory from route 1 above would be:-
0,0,0,0,0,0,
0.001,0.1,0,0,0,0.00232
0,0.2,0,0,0,6.242

---

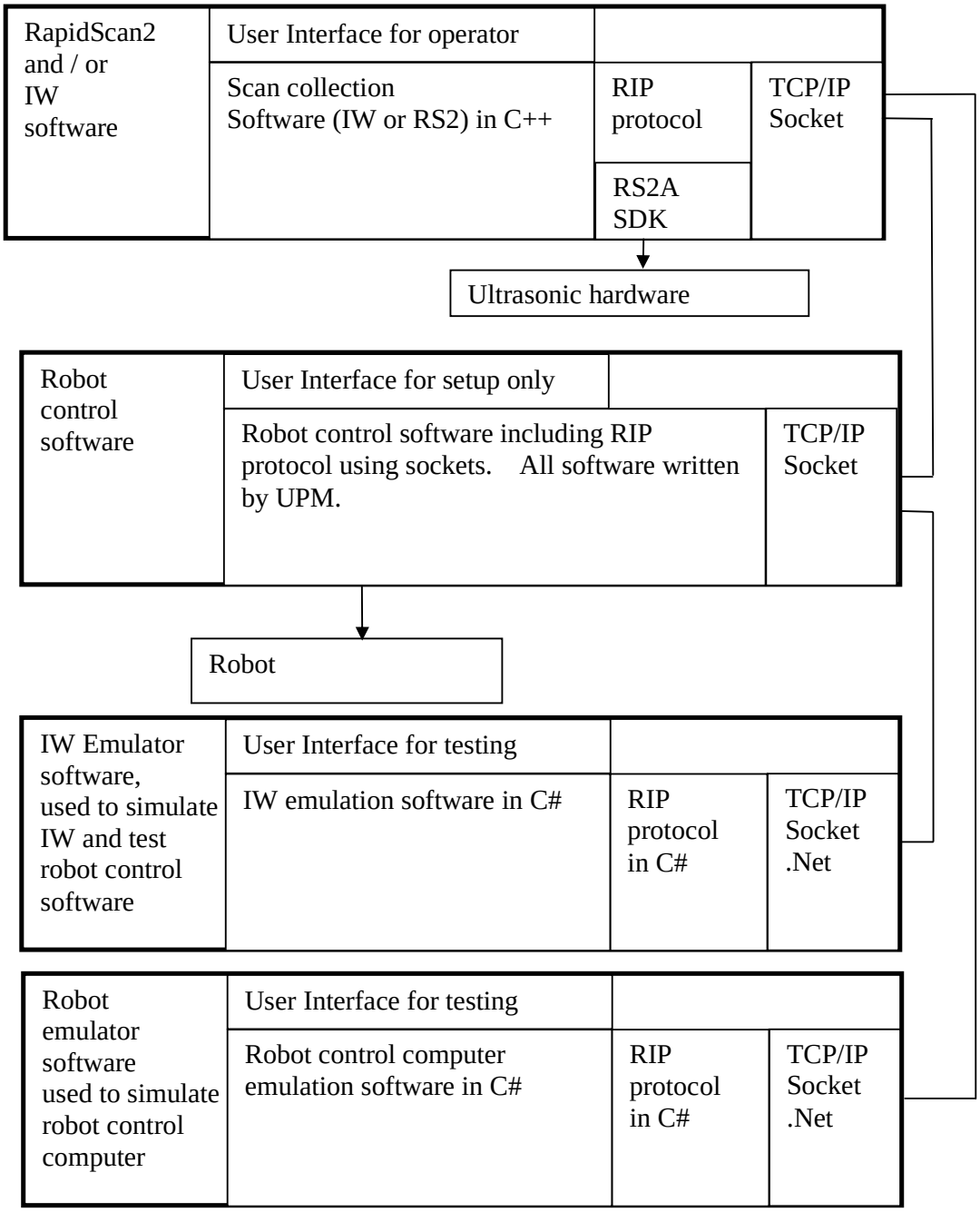
[1] The encoder actually records one position for each “B Scan”. A B Scan contains one sample for each group of elements that is fired from the array. For example, a 64 element array with an aperture of 8 will have 57 samples in a B Scan with one encoder position. IW uses the rotation information to extrapolate the positions of the 57 samples from the single encoder position.

[2] The Euclidian distance is a straight line between two points in 3D space, ignoring the physical constraints of the part being scanned. E.g. the distance “as the crow flies”.

[3] The Geodesic distance is the shortest distance between two points following the surface of the part. E.g. as if a string were stretched as tightly as possible between the two points on the surface of the part.

0,0.3,0,0,0,0
0,0.45,0,0,0,0
0,0.67,0,0,0,0
0,0.88,0,0,0
0,1,0,0,0,0

Block Diagram

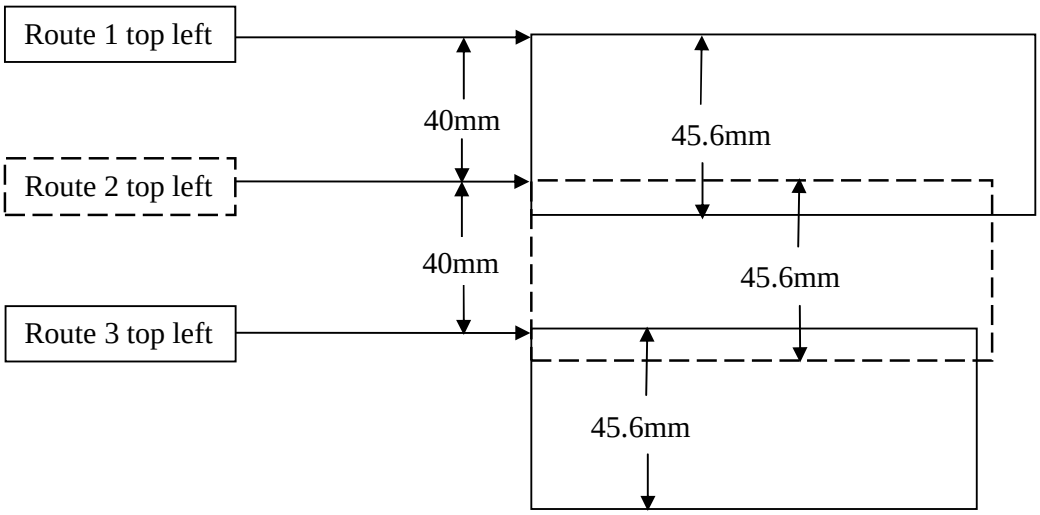


The diagram shows the major software components of the system. Each computer runs software that communicates via a TCP / IP socket. The software on each computer is entirely written by the team responsible for that computer. There are no libraries written by NDTS on the robot control computer. The robot control computer interfaces directly to the socket using for example C++ code making calls to the Windows Socket API. Other languages / socket interfaces can be used on the robot control computer, the robot control computer does not even have to be running a Windows operating system.
The GUI for the operator of the system is provided by IW / RS2. Additional GUIs may be run on other machines, such as robot control computers, to aid commissioning and fault finding. However, it should be possible to operate a commissioned system solely by the GUI in IW / RS2.
The two boxes at the bottom of the diagram are the emulator software written by Daniel Gavelle that is available on source forge. The emulator software is only used for initial commissioning and testing of the software. The IW emulator should be used for testing the robot controller. The Robot emulator is used for testing the RS2 / IW software.

Overlap

Routes must normally overlap by 10% to ensure that variations in the trajectory do not result in areas that are not scanned. The ultrasonic software will use a “Max Peak” algorithm to choose the best ultrasonic data in overlapping areas. For each overlapping sample, the ultrasonic software will determine which of the candidate samples has the maximum amplitude. A single max peak gate will be selected before the scan begins and this gate will be used for all the max peak amplitude measurements.

The standard wheel probe array used on both the crawler and Brutus projects is a 64 element array with a 0.8mm element pitch and an aperture of 8 elements. This means that there will be 57 points of sample data from the array. 57 * 0.8 = 45.6mm. In order to have a 10|% overlap, linear stripes should therefore be spaced 40mm apart.



Sockets protocol specification

Communications between IW and the robot controller will be via TCP sockets. The IW computer will be the client and the robot controller the server. There should only ever be one active connection to the server. Messages are contained within { }. This means that message synchronisation can always be gained by searching for the next {. The end of a message can always be found when a } is read. All messages consist entirely of printable 8 bit ASCII characters in the range 32 to 126 decimal inclusive. Messages are all of one of the types below:-

- Control messages are used to control the inspection process. They may be sent by either IW or the robot.
- Acknowledgements are used to confirm that a control message has been received.
- POS messages are used to send position information. They are not acknowledged.
- TRM messages are sent as a connection is closed.

Control messages

Each control message from IW to the robot consists of three capital letters, a space and the route number. E.g.
{INI 3}
Messages from the robot to IW have additional status information, described later. E.g.
{FIN 1 OK 0 OK}

Each control message must be acknowledged within one second or the sender of the message will assume an error has occurred. Both IW and the robot can send control messages. However, the robot should only send a control message following completion of action requested by IW.

Message	Source	Description
INI	IW	The robot moves into position ready to follow the requested route.
RDY	Robot	The robot has reached the position ready to start the route or has failed to move to the start position.
RUN	IW	The robot follows the route. This message is only valid following a successful INI and RDY of the same route.
FIN	Robot	The robot has completed the route. If the robot cannot complete the route, e.g. because of an obstruction, it issues the FIN message when it stops to allow the data collected so far to be used.
PAU	IW	Stop the movement immediately and wait for either CNT or INI. Can be sent when the robot is moving to the start following either an INI or running a route following a RUN.
CNT	IW	Continue moving on the current route or the current INI. Only valid following a PAU. CNT means continue from the current point in the route and not return to the start of the route. It is mainly for the crawler that will be moving much more slowly than the Brutus robot. If IW does want the robot to return to the start of the route and re run the same route, it can send the commands:- {PAU 1} {INI 1} {RUN 1}
CAL	IW	Recalibrate. This is only used for the crawler. An operator can request this message if for example a post is moved. A dummy 0 is passed as the route number.
RTQ	IW	Route Query. IW asks the robot for a line of route information. This should only be sent when the robot is idle.
HOM	IW	Move to home. The robot moves to its home position. When the robot reaches the home position, it replies with an RDY. The route number 0 is used in the messages to keep the format consistent with the RUN / ACK / RDY messages.

Acknowledgements

Each control message must be acknowledged within one second of receipt (five seconds for the crawler). If the message is OK, the receiver replies with ACK, a space and the route number of the sent message e.g.

{ACK 1}

If there is a problem, the receiver replies with ERR and at least two parameters, the route number and the error code. An optional string can follow this. IW can display this string to the operator. e.g.

{ERR 1 1006 Motor 1 has failed}

The string must not contain either { or } characters.

Error codes are allocated as follows:-

1 to 999 – common error codes.
1000-1999 – Brutus specific error codes.
2000-2999 – Crawler specific error codes.

Common error codes

Code	Source	Reason
1	Robot	The route index is out of range. This can be used to find out how many routes there are.
2	Robot or IW	Requested control message is not for the expected route, e.g. {INI 1} {RDY 2} or {INI 1} {RDY 1} {RUN 2}.
3	IW	Acknowledgement of a response to a robot command that has status of ER.
4	IW	The inspection has been completed and the operator has closed the application or workspace. Only used in a TRM command.
5	Robot	A new connection request has been received by the listening socket. Only used in a TRM command.

Acknowledgements are only ever sent in a response to a control message. The ERR message cannot be sent unilaterally.

Error information from the robot to IW

When the robot sends a message to IW, this message contains additional status information. This information can be passed to the operator if operator action is required to solve a problem.

The first part of the error information is a two character status code:-

- OK - The movement was completed within an acceptable tolerance.
- WN - Warning - the route deviated significantly from the intended trajectory.
- ER - Error - It is not currently possible to run this route at all.

This is followed by the error code and an error text string. Error code 0 means OK. Example messages are:-

{FIN 1 WN 1001 Route was not completed due to obstruction}

{FIN 1 ER 1002 Not possible to run this route}

{FIN 1 OK 0 OK}

The RDY command will also use these responses, e.g.

{RDY 1 WN 1005 Obstruction near start, route will start mid way}

{RDY 1 ER 1002 Not possible to run this route}

{RDY 1 OK 0 OK}

The Brutus system will decide whether to send OK, WN or ERR based on the difference between the actual location and theoretical location of the robot at the start of the route. It is possible that the warnings will appear as Windows message boxes in IW and so should not occur too frequently. Alternatively, the messages may be stored in a log and so may occur more frequently. The threshold in mm between OK / WN and WN / ERR will be configurable on the robot control computer during the commissioning process.

Position messages

Position messages are used by the robot to inform IW of the actual trajectory of a route. They are only issued between a RUN and FIN message. They do not require acknowledgement. The first POS following the run must be the coordinate of the actual position of the robot at the start of the route. The final POS before sending a FIN must be the position where the robot has stopped at the end of the route.

The rip specification does not currently contain any ping or keep alive messages. The TCP socket should provide a reliable connection making such messages unnecessary to safeguard the computers participating in rip. Non terminal problems with the robot can be indicated to IW following the next motion request. Total failure of the robot can be indicated by a TRM message. However if a ping type of message is required, it could be based on the robot sending POS messages at all times, rather than just when a route is being executed. The Crawler does not generate POS messages but follows a straight line between the start and end point and runs on a flat surface.

Encoder messages

Once IW has completed a scan and receives a FIN, it sends an ENC message to the robot. The ENC message contains the distance as measured by the encoder from the RUN to the FIN. ENC messages do not require an ACK. At the moment, ENC messages are only sent after a FIN command. However, it would be possible to send ENC messages more often if this is useful for debugging.

IW	Robot	Description
{RUN 1}		IW tells robot to start the route (previous INI etc. messages omitted for clarity)
	{ACK 1}	Robot acknowledges the run request
	{POS 0,0,0,0,0,0}	Robot sends the current position and then starts moving
	{POS 0.001,0.1,0,0,0,0.00232}	Robot sends intermediate positions
	{POS 0,0.2,0,0,0,6.242}	
	{POS 0,0.88,0,0,0}	
	{POS 0,1,0,0,0,0}	Robot reaches the end of the route and stops. It sends the real final point of the route
	{FIN 1 OK 0 OK}	Robot tells IW the route is complete
{ACK 1}		IW acknowledges the route is complete
{ENC 1.53}		IW tells the robot that the encoder measurement was 1.53 metres between the RUN and the FIN.

On the Brutus system, IW will collect the C Scan and POS messages as a route is run. After the route, it will use linear interpolation to match the POS messages to scan data. The scan data will then be pasted on to the 3-D model of the part as a rectangle between each POS point. Changing velocity will not cause a problem with the interpolation as both the encoder and the POS data are distance measurements and time is not measured. However, a slower velocity will result in more POS messages per metre and so more accurate pinning of the data on to the part.

Route messages

IW may need to know all the routes before they are run, for example to allocate space for the scans. IW sends a RTQ – route query command to the robot. If the route number is valid the robot replies with an RTI Route Information message. The RTI is sent immediately after the ACK command for the route. The RTI command contains the route number then the route as 12 comma separated floating point numbers.

IW	Robot	Description
{RTQ 1}		IW asks the robot for the coordinates of route 1
	{ACK 1}	Robot acknowledges the RQT
	{RTI 1 0,0,0,0,0,0,1,1,1,0,0,0}	Robot sends the start and end coordinates for route 1
{RTQ 10}		IW asks the robot for the coordinates of route 10
	{ERR 10 1 Invalid route no.}	There are less than 10 routes so the robot indicates this is an error

An RTI message is always the theoretical position of a route, not the actual measured position. POS messages always refer to the real position of the robot. An RTI can be sent at any time, e.g. RTI's for all routes may be sent before starting the inspection. This will allow IW to draw all the proposed routes on the CAD data before the inspection is started.

Connect and disconnect

The robot should be ready to receive an INI control message as soon as it receives a connection to its listening socket. If there is any initial calibration required, this is best performed as soon as the system starts. It is not an error for IW to connect and send an INI before calibration is complete. The robot should acknowledge the INI with ACK but delay sending RDY until calibration is complete and the robot has moved to the start of the route.

IW should only every make one connection to the server at a time. If the server already has an active connection and a new one is requested, it should drop the existing connection.

A TRM command should always be sent before the socket is disconnected. When the socket is disconnected, the robot can assume that the inspection has ended. It can return to a home position, give control back to the Kuka pendant etc. The socket should be disconnected before setting up the robot to inspect a different part or before editing the inspection table. If a socket is disconnected by the robot because of a new connection request to the listening server socket the command below is sent:-

{TRM 5 A new connection request has been received by the listening socket}

Termination of connection

If there is a fatal error or the application closes, the socket connection must be dropped. The other end of the connection is informed of the reason the socket has been closed by the TRM command. The TRM command can be sent asynchronously by either the robot or IW just before the connection is closed. It has the same format as the ERR command e.g.

{TRM 0 1099 Motor has failed - maintenance required}

{TRM 0 4 IW has closed}

Synchronising State Machines

It is possible that IW and the robot state machines may get out of sync. In order to resynchronise the state machines, an INI can be sent from IW at any time. If the robot receives an INI it must immediately stop its current motion, ACK the INI and move to the position specified by the INI. This means that if any computer gets out of synch with another, it will be resynchronised the next time an INI message is sent.

Configuration

Any device using rip to communicate must support the configuration of the following parameters:-

- The well known port that is used to connect to the robot.
- IW must be able to configure the IP address or host name of the server.
- A path to a file to which all the communications and errors are logged. If this is blank, logging is disabled.