RESULTATS

8. RESULTATS

Seguidament es mostra dos exemples amb els resultats de l’aplicació. En aquestes podrem observar la captura resultant, així com la seva representació en format VRML.

8.1. Exemple mode-3

En aquest mode movem el sensor en vertical segons l’eix Z d’aquest. Es aconsellable per objectes reduïts o quan vulguem una resolució em més gran possible, tant per objectes que resten prop del sensor com lluny.

![Imatge general del robot i l'element a escanejar](image)

Com podem veure a la figura Fig.27 el sensor làser està subjecte al canell del robot. Preparat per escanejar un objecte (en aquest cas es tracta de la tassa)

A continuació podem veure la captura de la imatge amb una representació tridimensional per punts. Una vegada finalitzada la captura ens mostra els paràmetres de l’operació a la finestra de diàleg.

Finalment el programa ens permet gravar el resultat en un fitxer en format VRML per a poder ser visualitzat posteriorment, o per traspassar-ne les dades a algun programa de gestió d’imatges tridimensional que reconegui aquest format.

PFC.: Fusió de dades volumètriques per obtenir models 3D
Fig. 28 Captura de la imatge 3D de l’objecte

Fig. 29 Representació de l’objecte en format VRML

8.2. Exemple Mode-2

Aquest mode, on l’escanejat es realitza rotant el sensor làser sobre el seu eix X, ens permet d’obtenir la representació tridimensional d’escenes d’un volum més gran que en
el mode-3. En aquesta captura s’ha optat per capturar la representació 3D d’una figura humana.

Fig.30 Imatge general del robot i la figura a escanejar

Cal advertir que el sensor làser pot arribar a ser perjudicial per a la vista, i que per realitzat aquesta captura s’ha utilitzat un protecció visual adequada per tal d’evitar possibles danys.

Podem apreciar que en el quadre de diàleg ens mostra les dades referents a aquest mode de funcionament. Informant-nos dels paràmetres mes rellevants de l’obtenció de l’escena.
Fig. 31 Captura de la imatge 3D de l’objecte

Fig. 32 Representació de la figura en format VRML
9. ESTUDI ECONÒMIC

En aquest apartat es realitzarà una valoració en termes de cost de desenvolupament del projecte. Valorant tots els recursos que s'han requerit en el desenvolupament d'aquest, ja sigui recursos de maquinari, recursos de programari o recursos humans.

9.1. Cost maquinari

En aquest cas el cost del maquinari es limita a un sol ordinador. A l'hora de fer l'estudi econòmic sols s'ha tingut en compte el relació de costos del desenvolupament del programari. Per tant no s'hi han inclòs el cost del làser i els robot.

<table>
<thead>
<tr>
<th>Equip</th>
<th>Especificacions</th>
<th>cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ordinador personal</td>
<td>Pentium 4 3.0 GHz HT 1024 Mb RAM 250 Gb HDD ATI Radeon 9250 128 Mb 1 Monitor TFT 19&quot; 1 Monitor TFT 19&quot; Gravadora DVD Lector DVD</td>
<td>1100 €</td>
</tr>
</tbody>
</table>

Taula.1 Relació de maquinari per al càlcul de costos

En aquest cost no s’hi contempla el valor íntegre del maquinari, ja que la vida útil d’aquest equip no es limita a aquests 7 mesos de treball. Per tant l’import estimable depèn del termini d'amortització (en aquest cas he contat un termini de 42 mesos)
<table>
<thead>
<tr>
<th>Concepte</th>
<th>Valor</th>
<th>Termini d'amortització</th>
<th>% imputable</th>
<th>Cost imputable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ordinador</td>
<td>1100 €</td>
<td>42 mesos</td>
<td>(7/42) x 100 = 17%</td>
<td>187.00 €</td>
</tr>
<tr>
<td>Láser</td>
<td>4000 €</td>
<td>120 mesos</td>
<td>(4/120)x100 = 3.33%</td>
<td>133.33 €</td>
</tr>
<tr>
<td>Robot</td>
<td>35000 €</td>
<td>240 mesos</td>
<td>(2/240)x100 = 0.83%</td>
<td>291.67 €</td>
</tr>
<tr>
<td>TOTAL</td>
<td></td>
<td></td>
<td></td>
<td>612.00 €</td>
</tr>
</tbody>
</table>

Taula.2 Relació costos de maquinari

9.2. Cost programari

El projecte s’ha desenvolupat en Visual C++ sobre una plataforma Windows, també s’han requerit de l’utilització de programari per a la gestió del projecte així com programari d’ofimàtica.

<table>
<thead>
<tr>
<th>Eines pel desenvolupament</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microsoft Windows XP Pro</td>
<td>164.62 €</td>
</tr>
<tr>
<td>Microsoft office</td>
<td>288.71 €</td>
</tr>
<tr>
<td>Microsoft Visió 2003</td>
<td>164.62 €</td>
</tr>
<tr>
<td>Microsoft Visual estudio 2005</td>
<td>679.00 €</td>
</tr>
<tr>
<td>TOTAL</td>
<td>1296.95 €</td>
</tr>
</tbody>
</table>

Taula.3 Relació de programari per al càlcul de costos
ESTUDI ECONÒMIC

Com s'explica a l'apartat anterior estimarem el cost de les eines de desenvolupament segons el termini d'amortització d'aquest. Al igual que en el cas del maquinari s'ha considerat un termini d'amortització de 42 mesos.

<table>
<thead>
<tr>
<th>Valor</th>
<th>Termini d'amortització</th>
<th>% imputable</th>
<th>Cost imputable</th>
</tr>
</thead>
<tbody>
<tr>
<td>1296.95 €</td>
<td>42 mesos</td>
<td>(7/42) x 100 = 17%</td>
<td>220.48 €</td>
</tr>
<tr>
<td>TOTAL</td>
<td></td>
<td></td>
<td>220.48 €</td>
</tr>
</tbody>
</table>

**Taula 4 Relació costos de maquinari**

9.3. **Cost recursos humans**

En aquest últim cas, tot i que el projecte ha estat desenvolupat per una sola persona, es desglossarà depenent del rol que ha adoptat durant el transcurs del projecte. A cada rol se li imputa un cost diferent.

<table>
<thead>
<tr>
<th>Rol</th>
<th>Tasques que desenvolupa</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analista</td>
<td>• Anàlisi</td>
</tr>
<tr>
<td></td>
<td>• Especificació</td>
</tr>
<tr>
<td></td>
<td>• Disseny</td>
</tr>
<tr>
<td></td>
<td>• Implantació</td>
</tr>
<tr>
<td></td>
<td>• Documentació</td>
</tr>
<tr>
<td>Programador</td>
<td>• Implementació</td>
</tr>
<tr>
<td></td>
<td>• Test</td>
</tr>
</tbody>
</table>

**Taula 5 Rols de desenvolupament i tasques**
<table>
<thead>
<tr>
<th>Rol</th>
<th>Sou brut anual</th>
<th>Cost salarial anual</th>
<th>Cost per hora</th>
<th>hores</th>
<th>Cost total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analista</td>
<td>29.000 €</td>
<td>38.570 €</td>
<td>20.09 €</td>
<td>416</td>
<td>8357.44 €</td>
</tr>
<tr>
<td>Programador</td>
<td>19.000 €</td>
<td>25.270 €</td>
<td>13.16 €</td>
<td>664</td>
<td>8738.24 €</td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td></td>
<td></td>
<td></td>
<td>1080</td>
<td><strong>17095.68 €</strong></td>
</tr>
</tbody>
</table>

*Taula.6 Relació costos de recursos humans*

9.4. **Cost total del projecte**

<table>
<thead>
<tr>
<th>Concepte</th>
<th>Quantitat</th>
<th>Cost unitari</th>
<th>Cost total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maquinari</td>
<td>1</td>
<td>187.00 €</td>
<td>612.00 €</td>
</tr>
<tr>
<td>Programari</td>
<td>1</td>
<td>220.48 €</td>
<td>220.48 €</td>
</tr>
<tr>
<td>Recursos humans</td>
<td>1</td>
<td>23418.64 €</td>
<td>17095.68 €</td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td></td>
<td></td>
<td><strong>17928.16 €</strong></td>
</tr>
</tbody>
</table>

*Taula.7 Relació de costos totals*

9.5. **Cost total d’una unitat per a la seva venda**

Considerem que és un sistema que pot tenir cabuda en diferents entorns industrials i acadèmics. Per tant, tenint en compte que podem aconseguir una cartera de clients elevada (sobretot la industria de l’automatització és molt solvent i oberta a les innovacions tecnològiques) ens proposem de vendre tot el conjunt.

pag. 62 | PFC.: Fusió de dades volumètriques per obtenir models 3D
Considerarem de moment una partida de 100 unitats per guanyar en percentatge de descompte dels dispositius, i per altra l’amortització del desenvolupament del sistema. A partir d’aquí obtenim:

<table>
<thead>
<tr>
<th>Concepte</th>
<th>Quantitat</th>
<th>Cost unitari</th>
<th>Cost Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Láser</td>
<td>100</td>
<td>4000 €</td>
<td>400,000 €</td>
</tr>
<tr>
<td>Robot</td>
<td>100</td>
<td>35000 €</td>
<td>3,500,000 €</td>
</tr>
<tr>
<td>desenvolupament</td>
<td>1</td>
<td>17928.16 €</td>
<td>17,928.16 €</td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td></td>
<td></td>
<td><strong>3,917,928.16 €</strong></td>
</tr>
<tr>
<td><strong>TOTAL per unitat</strong></td>
<td></td>
<td></td>
<td><strong>39,179.29 €</strong></td>
</tr>
<tr>
<td><strong>TOTAL per unitat amb 15 % de benefic</strong></td>
<td></td>
<td></td>
<td><strong>45,056.17 €</strong></td>
</tr>
</tbody>
</table>

*Taula 8 Cost de producció i venda del sistema*

Com podem apreciar, el preu dels dispositius i tenint en compte una tirada de producció de 100 unitats, fa irrisori el cost de desenvolupament del sistema.
| pag. 64 | PFC.: Fusió de dades volumètriques per obtenir models 3D |
10. CONCLUSIONS

Una vegada finalitzat el projecte el projecte i havent obtingut una visió més concreta sobre el sistema de modelat3D amb d’utilització de sensors làters. En podria extreure les següents conclusions.

- Crec que es una solució viable per a multitud d’aplicacions, ja que genera uns resultats força concrets. Amb un temps d’escaneig raonable.

  Donant una visió considerablement fiable de l’entorn se’ns presenta com una solució a tenir en compte per aplicacions que requereixin de sistemes fiables per modelar escenes tridimensionals.

  Això fa que sigui factible per a multitud d’usos i industries amb un grau elevat d’automatització. També el fa adequat per laboratoris de desenvolupament de sistemes per a l’automatització i el control.

- El fet que els resultats obtinguts son satisfactoris, ens obra les portes a múltiples ampliacions de funcionalitats (veure l’apartat de possibles millores). Millorant les prestacions del sistema, i per tant trobant més possibles aplicacions.

- Els sensors làters utilitzats, tot i que cada vegada s’estan utilitzant més amb sistemes similars al desenvolupat, encara es pot apreciar que inicialment estaven destinats a un tipus d’aplicació menys exigent en quant a precisió i velocitat de mostreig. Ja que les mesures obtingudes presenten força soroll i oscil·lació.

  Tot i això es pot veure que el mercat dels sensors làser també s’està encarant cap aquest sector, ja que per exemple les interfeccies sèrie PC-Làser s’han substituït per connexions eternet –eliminant el coll d’ampolla que representava la línia sèrie. També estan traient sensors més estables en quan a les mesures.
| pag. 66 | PFC.: Fusió de dades volumètriques per obtenir models 3D |
11. POSSIBLES MILLORES

Una vegada finalitzat el projecte, i completats els objectius, plantejo algunes possibles millors per complementar i algunes possibles aplicacions per trobar-li una utilització pràctica.


Tots els moviments d'escombrat de l'escena que es presenten, es realitzen tenint en compte que l'origen de coordenades està situat just a la font d'emissió del laser, informació que sol ser facilitada pel fabricant del sensor.

Per a que el robot realitzi el moviments tenint en compte aquest punt, hem d'assignar aquestes coordenades al T.C.P. (Centre del punt d'eina) del robot. Indicant al robot que deixi de treballar amb les coordenades del seu canell i prengui les coordenades T.C.P. com a nou punt de referència.

![Diagrama](image)

*Fig.x Coordenades del T.C.P. respecte les coordenades del canell del robot*

Però aquestes coordenades del sensor no sempre ens son facilitades, per tant seria pertinent de completar el sistema per a que el T.C.P. pogués ser calcular automàticament.

Amb l'utilització d'una plantilla tridimensional adient juntament amb les mesures que ens indica el sensor per -amb els moviments adients- calcular les coordenades del T.C.P. respecte les del canell del robot.
11.2. **Partó per al càlcul de l’error del sistema**

Tot i que s’ha validat de forma empírica el correcte funcionament del sistema, en quan a que els resultats entren dins el marge d’error que ens proporcionen els fabricants dels dispositius que intervenen en el sistema. Caldria incorporar una nova funcionalitat que ho verifiqués automàticament.

S’hauria de disposar d’un escenari conegut amb particularitats, com podria ser una plantilla coneguda, que els programari fos capaç de reconèixer dins l’escena. Preferiblement plana, ja que és més fàcil de localitzar i a part ens permetria calcular la variància dels resultats, així com les cotes de la plantilla.

11.3. **Intercalar diferents imatges 3D capturades**

Ens pot ser de molt interès poder fusionar dues o més captures d’un mateix escenari preses des de diferents localitzacions del sensor, per generar una sola representació tridimensional.

Pot ser de molta utilitat a l’hora obtenir un resultat més aproximat a l’escena real, desestimant les zones ocultes que es solen aparèixer cada captura.

També seria aplicable en robòtica mòbil, per tal de generar un mapa tridimensional d’un mateix espai dividit en diversos escenaris.

Això ens permetria, per exemple, que si incorporésim el nostre sistema a un robot mòbil poder generar un sol mapa tridimensional de tot un pis a partir de la fusió de les diferent imatges tridimensionals captades al llarg del recorregut d’aquest, per tant d’escenaris completament diferents.

11.4. **Reconèixer objectes dins d’una escena**

Partint de models tridimensionals d’objectes concrets, desenvolupar un algorisme de certa d’aquests objectes dins de la recreació tridimensional d’una escena. Retornant la posició exacta d’aquests dins de l’escena.

Així mateix es podria eliminar part de l’escena que no correspon a cap dels models d’objectes a identificar.

11.5. **Renderitzar un modelat amb d’incorporació d’una càmera de vídeo.**

Es tractaria de complementar la representació tridimensional obtinguda amb imatges de la mateixa escena provinents d’una càmera de vídeo. Assignant textures a les diferent cel·les que conformen la representació tridimensional.
POSSIBLES MILLORES

Es tractaria de buscar punts representatius comuns entre l'escena tridimensional i la capturada per la càmera per tal de fer-ne una extrapolació que ens permeti colorejar la representació tridimensional amb la textura que li correspon a l'escena real.

Aquest sistema ens aportaria molta més informació de l'escena, permetent-nos crear representacions virtuals d'un escenari.

11.6. Finestres amb més resolució.

Aprofitar que la resolució del sensor làser és parametritzable per realitzar modelats generals a baixa resolució (no es requeririen tants punts, i per tan la captura seria molt més ràpida). Per una vegada tenim un primera visió més general de l'escena, augmentar la resolució per mostrejar les zona que ens siguin de més interès i que requereixin d'una resolució més elevada.

Aquesta nova millora podria complementar el punt 10.3. On la el modelat genèric ens servís per localitzar un objecte determinat dins l'escena, i una vegada localitzat augmentar la resolució per fer un escombrat a alta resolució, sols de l'objecte en qüestió.

11.7. Agrupar els punts segons el pla al que pertanyen.

El fet d'unir els tots els punts d'una mateixa escena ens genera una imatge continu a de l'escena. Com si estenguéssim un llençol sobre l'escena. Per tant, dos objectes situats en plans de l'escena diferents, queden units entre ells.

El que es tractaria ara és de detectar els salts que es produeixen entre punt que pertanyen a plans diferents, i unir sols els punts que pertanyen al mateix pla. Aquest salts ens poden indicar la presència de zones occultes.
| pag. 70 | PFC.: Fusió de dades volumètriques per obtenir models 3D |
12. BIBLIOGRAFÍA

12.1. **Enginyeria del software**
- Enginyeria del software: Especificació - *D. Costal, M.R.Sancho, E. Teniente*
- Enginyeria del software: Disseny I - *C.Gómez, E. Mayol, A.Olivé, E. Teniente*

12.2. **Llenguatge de programació VisualC++**
- Programación en C++ - Enrique Hernández Orallo, José Hernández Orallo
- Programación avanzada con VisualC++ - *D. Kruglinski, G. Shepherd, S. Wingo*
- Visual C++ 6 Unleashed – Mickey Williams, David Bennett

12.3. **Visió per ordinador**
- Visión 3D . José M.Sebastián – *Luis M. Jiménez*
- Viewpoint-Coded Structured Light – *Ravi Ramamoorthi*
- From Wide-baseline Point and Line Correspondences to 3D – *Herbert Bay*

12.4. **VRML**
- RML y meta-ejercicio - *Luis Fernando Lago Fernández*  
- Técnicas de realidad virtual – *Fernando de la Rosa*  

12.5. **OpenGL**
- Programación en OpenGL – *Richard S.Wright, Jr. Michael Sweet*
12.6. **Sensor làser**

- ROTOSCAN RS4-4 Laser Scanner for personal protection and measurement tasks – Technical description – Leuze lumiflex

- RS4soft Configuration and diagnostics software for ROTOSCAN RS4-4 – User manual – Leuze lumiflex

- RS 4 / Software – Information on the RS4 Protocol for external use – *Leuze lumiflex*

12.7. **Robotica**

- Fundamentos de robótica – *Antonio Barrientos, Luis Felipe Peñín*

- Curso de robótica – *J.M. Angulo, Rafael Aviles* Annex A: Sensor làser
rotoscan ROD-4

The rotoscan ROD-4 is an area scanning distance sensor for the detection of objects. The light beam is reflected by a rotating mirror and directed over a semicircular area (190°) with a radius of max. 50m.

- The area is divided into two detection areas, each with a radius of 50m. The size of the area to be evaluated can be freely defined in each detection area.
- It is possible to store 4 detection area pairs in the ROD-4 and to switch between these pairs, for example, to define various heights or allowed overhanging.
- ROD-4 with UL approval
- ROD-4-06 with heating and ROD-4-08 with heating, dust-resistant versions.
- Plug-in dll file for measurement processing on the PC

accessories:
- Mounting systems
- Configuration software
- Various connection cables
- Measurement-DLL for PC

ISO 9001

Leuze electronic GmbH & Co. KG
www.leuze.de

Dimensioned drawing

Optical distance sensors

Measurement principle

180° working range
60° sectors, 0.36° each

Working field: 50m
Detection area 1: max. 50m
Detection area 2: max. 50m

PFC.: Fusió de dades volumètriques per obtenir models 3D
Specifications

Optical data
- Scanning range (per detection area) 0 ... 50cm (ROD-4-08 ... 25m)
- Angular range max. 180°
- Angular resolution 0.36°
- Scanning rate 25 scans/s or 4000 scans
- Transmitter infrared laser diode (eye safe), wavelength 905 nm, beam divergence ± 2mrad, time base = 100 s

Detection area 1 and 2
- Reflectivity from at least 1.8% (matte black)
- ROD-4-08 from 6% (dark grey)
- > 20 mm at distance of 4 m
- > 100 mm at distance of 17 m
- > 40° at least 40° (corresponds to 1 scan)
- Number of detection area pairs 4 (selectable via switching inputs)
- Output 3x PNP transistor output 24 V/0.5mA, 5V
- Measurement value resolution per sector 10 ... 99% diffuse reflection at 4 m distance ± 15 mm ± 20 mm
- Repeatability

Electrical data
- Voltage supply +24VDC ±20% ±90%
- Overcurrent protection via fuse 2 A semi time-lag in the switching cabinet
- Current consumption approx. 100 mA (max. 1A power supply), approx. 2A with heating
- Power consumption < 630 W at 24 V including the outputs
- Overvoltage protection overvoltage protection with protected limit stop

Mechanical data
- Housing diecast aluminium, plastic
- Weight 2.0 kg
- Connection type optionally possible, 2 connectors (used for plug-in, solder connection)

Environmental data
- Ambient temp. (operation/storage)
  - -5° C ... +55° C / -20° C ... +70° C
- VDE safety class II, all installed
- Protection class IP 65
- Laser class 1 (acc. to EN 60825-1)
- Standards applied IEC 60847-5-2

Operating principle

Order guide

<table>
<thead>
<tr>
<th>Designation</th>
<th>Part No.</th>
<th>UL</th>
</tr>
</thead>
<tbody>
<tr>
<td>With heating</td>
<td>ROD-4</td>
<td>500 36010</td>
</tr>
<tr>
<td></td>
<td>ROD-4-06</td>
<td>500 36614</td>
</tr>
<tr>
<td></td>
<td>ROD-4-08</td>
<td>500 41423</td>
</tr>
</tbody>
</table>

Remarks

"RODSoft" configuration software

The configuration software "RODSoft" runs under Windows 95/98/NT/2000/XP and offers the following possibilities:
- Programming of the detection areas
- Parameterisation of other data
- Visualisation of the detection area with measurement values
- Error code display
- Support of various languages

There are various methods with which detection areas can be programmed, for example:
- "Teach-in" function
- Numerical and graphical entry of the detection areas
- "Edit" function

Tables

0609

pag. 74

PFC.: Fusión de dades volumétriques per obtenir models 3D
RS 4 / Software

Information on the RS4 Protocol for external use

<table>
<thead>
<tr>
<th>Title</th>
<th>Information on the RS4 Protocol for external use</th>
</tr>
</thead>
<tbody>
<tr>
<td>Document No</td>
<td>XXX</td>
</tr>
<tr>
<td>Version No</td>
<td>1.0</td>
</tr>
<tr>
<td>Date</td>
<td>01.03.01</td>
</tr>
<tr>
<td>Status</td>
<td>checked (Brunner)</td>
</tr>
<tr>
<td>Author</td>
<td>Martin Kaul</td>
</tr>
<tr>
<td>Filename</td>
<td>rs4_protocolExtern.doc</td>
</tr>
</tbody>
</table>

PFC.: Fusió de dades volumètriques per obtenir models 3D
## History

<table>
<thead>
<tr>
<th>Date</th>
<th>Vers.</th>
<th>Author</th>
<th>Status</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>01.03.2001</td>
<td>1.0</td>
<td>Martin Kaul</td>
<td>checked /</td>
<td>created using documents SER_COMM.DOC</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Brunner</td>
<td></td>
<td>and CMDPROC.DOC</td>
</tr>
</tbody>
</table>

Filename          | Version | Department     | Date     | Page |
------------------|---------|----------------|----------|------|
RS4_protocol_exte-doc | 1.0     | Martin Kaul - LOG | 09.05.01 | 2/2  |
Table of contents

1. Notice ........................................................................................................... 4
2. Purpose of the present document ............................................................... 4
3. Requirements .................................................................................................. 4
   3.1. Universal requirements ........................................................................... 4
   3.2. Type of data ............................................................................................ 4
   3.3. Access restriction .................................................................................... 5
   3.4. Protocol .................................................................................................. 5
   3.5. Realization ............................................................................................... 6
   3.6. Characteristics of the interface ............................................................... 7
4. Analysis ........................................................................................................... 8
   4.1. Protocol .................................................................................................. 8
       4.1.1. Requirement and impact ................................................................. 9
       4.1.2. Specifications of the transmission protocol .................................... 10
   4.2. Structure of the user information ........................................................... 15
       4.2.1. Strategies for the transmission of measurement values .......... 15
       4.2.2. Structure of the transmitted measurement contour ............... 16
       4.2.3. Transmitting errors ...................................................................... 17
   4.3. Command description ............................................................................ 18
1. Notice

This document is an excerpt from the documents SER_COMM.DOC and CMDPROC.DOC for external use, i.e. when an external organisation requires information regarding the structure of the RS4 protocol.

2. Purpose of the present document

This document describes the structure and protocol of the serial communication between the RS 4 and the PC configuration program.

It examines the requirements placed on the serial communication, analyses these requirements, uses this analysis to define a protocol for the serial communication and then designs the serial communication.

3. Requirements

Please find below a list of the different requirements to be met by the serial communication.

3.1. Universal requirements

Please find below a list of the universal requirements:

- Requirement 1000:
  Category: universal
  Description: The serial communication should facilitate a secure transmission of messages (with any content) between the RS 4 and a PC configuration program.

3.2. Type of data

Please find below a list of the requirements which specify the type of information to be transmitted:

- Requirement 2000:
  Category: Type of data
  Description: The serial communication must facilitate the transmission of configuration data.

- Requirement 2010:
  Category: Type of data
  Description: The serial communication must facilitate the transmission of device data.

- Requirement 2020:
  Category: Type of data
  Description: The serial communication must facilitate the output of measurement values.

- Requirement 2030:
  Category: Type of data
  Description: It should be possible to transmit measurement values sector-wise, i.e. it should be possible to suppress the output of measurement values from defined sectors.
3.3. Access restriction

Please find below a list of the requirements which apply to the access restriction of the RS 4:

- Requirement 3000:
  Category: Access restriction
  Description: When transmitting messages from the PC configuration program to the RS 4, the password for the current authorisation level must always be transmitted together with the message. Messages without a password must be ignored by the RS 4.

3.4. Protocol

Please find below a list of the requirements which are required for specifying the transmission protocol:

- Requirement 4000:
  Category: Protocol
  Description: Messages should be transmitted optionally with/without acknowledgement.
  
  The condition, whether the transmission takes place with or without acknowledgement, affects the transmitter (RS4 or PC program) of a message depending on the type of data information to be transmitted.

  Security-critical data are, in this case, always transmitted with an acknowledgement. This setting is permanently programmed into the respective program.

- Requirement 4010:
  Category: Protocol
  Description: Messages are not transmitted within a network.

- Requirement 4020:
  Category: Protocol
  Description: Errors which could occur during the transmission must be detectable by the serial communication.

- Requirement 4030:
  Category: Protocol
  Description: Binary data should be used for the transmission of messages, i.e. the transmission makes use of all user bits of the serial interface.

- Requirement 4040:
  Category: Protocol
  Description: The connection of a terminal over the serial interface of the RS 4 is not intended.
• Requirement 4050:
  Category: Protocol
  Description: The following status information is transmitted each time the RS 4 performs an output function:
  - Operating status
  - Protected fields busy
  - Warning
  - Error
  - Restart-disabled
  - Identifier of the current protected field pair

• Requirement 4060:
  Category: Protocol
  Description: The protocol should transmit information with as little redundancy as possible.

• Requirement 4070:
  Category: Protocol
  Description: It should be possible to transmit measurement values both word-wise (16 bit) as well as byte-wise (8 bit).
  16 bit: The resolution is 2 mm, i.e. 0 ... approx. 65 m
  8 bit: The resolution is 16 mm, i.e. 0 ... approx. 4 m

• Requirement 4080:
  Category: Protocol
  Description: The transmission protocol must be capable of synchronising the RS 4 and PC configuration program on start and stop if an error occurs during a previous transmission.

• Requirement 4090:
  Category: Protocol
  Description: Messages from RS 4 to PC and from PC to RS 4 have the same structure.

3.5. Realization

Please find below a list of requirements which are necessary for realizing the serial communication:

• Requirement 5000:
  Category: Realization
  Description: Only a single transmission buffer should be used in the RS 4 for the output of messages.

• Requirement 5010:
  Category: Realization
  Description: Only a single reception buffer should be used in the RS 4 for the reception of messages.
• Requirement 5020:
  Category: Realization
  Description: The reception buffer is not overwritten until a received message has been processed.

• Requirement 5030:
  Category: Realization
  Description: The transmission buffer is not overwritten until a message to be transmitted has been output.

• Requirement 5040:
  Category: Realization
  Description: The serial communication must function effectively and with an efficient use of resources.

3.6. Characteristics of the interface

Please find below a list of the requirements which are necessary for setting the serial interface.

• Requirement 6000:
  Category: Characteristics of the interface
  Description: The following serial interface settings should be available:
  - Baud rate 4800 baud and more
  - Data format: 8N1
4. Analysis

This chapter analyses the various requirements listed in chapter 0 Requirements and, using this analysis, establishes a transmission protocol for the communication between the RS 4 and the PC configuration program. The analysis is also used to create transmission strategies for the transmission of data.

4.1. Protocol

In general, a distinction must be made between the protocol and the interpretation of the transmitted data. The protocol defines a language which assists the RS 4 and the PC communication program in exchanging data with one another.

In this case it must be irrelevant to the protocol which type of data are being transmitted.

In order to achieve this, the protocol generates a frame for interpreting the data to be transmitted independent of the data. In this way the protocol defines for each message to be transmitted a header, which precedes the actual information, and a footer, which is appended to the actual information. Each message between the two communicating partners is composed of such a package with a header, the information itself and a footer.

The following diagram illustrates this:

Sending a message from the RS 4 to the PC configuration program

```
RS 4       Footer  Actual Information  Header       PC config program
```

Sending a message from the PC configuration program to the RS 4

```
RS 4       Header  Actual Information  Footer       PC config program
```

Protocol relevant data

Not protocol relevant data

Figure 0-1 Protocol relevance

In this chapter the requirements from chapter 0 Requirements are analysed and the header and footer formats specified according to these requirements.
### 4.1.1. Requirement and impact

This chapter lists the relevant requirements and their corresponding impacts. These impacts are then converted into a protocol specification.

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td>3000</td>
<td>The frame must contain a field in which a password can be transmitted together with the message.</td>
</tr>
<tr>
<td>3000 4090</td>
<td>It must be possible to set for each message whether or not a password is to be included in the transmission, as the password is only required for messages transmitted from the PC configuration program to the RS4 and the protocol structure must be identical in both directions. Messages from PC to RS4 thus always include the password. Messages from RS4 to PC include no password. (Option)</td>
</tr>
<tr>
<td>4000</td>
<td>It must be possible to set for each message whether or not an acknowledgement is desired. This setting is permanently programmed into the respective program. (Option)</td>
</tr>
<tr>
<td>4010</td>
<td>An address can be omitted.</td>
</tr>
<tr>
<td>4020</td>
<td>A checksum must be transmitted within the frame in order to detect transmission errors. A line-cut is specified between the individual characters in order to detect a missing message end.</td>
</tr>
<tr>
<td>4030 4040 4080 6000</td>
<td>No extra bits may be used for identifying the frame, as all data bits of the individual characters are required for the data transmission. Therefore, a token must be used for identifying the frame. A token is a special combination of characters which does not occur in the data transmission. As, in principle, all combinations of characters are possible when transmitting binary data, the corresponding combination of characters must be suppressed when transmitting the binary data. It is best when the combination of characters representing a token appears as seldom as possible within the frame. Readability of the messages on a terminal is not necessary, as the data can be transmitted in binary format without worry.</td>
</tr>
</tbody>
</table>
4050
The following additional information must also be transmitted within the frame:

- Operating status
  of at least 4 different states
- act. personal safety field busy
- act. object safety field busy
- 2. personal safety field busy
- 2. object safety field busy
- selected pair of protected fields
- 2. pair of protected fields
- Warning
- Error
- Restart-disable

In order for everything to function quickly, the states of the protected fields + error + warning + restart-disable do not need to be transmitted when they are not set (i.e. "off").

The operating status should always be included in the transmission.

> Option

Table 0-1 Requirements and their impacts

4.1.2. Specifications of the transmission protocol

This chapter describes the specifications of the transmission protocol. For this purpose, the requirements from chapter 0 Requirements and the impacts from chapter 0 Requirement and impact are used.

Each message transmitted by the RS 4 to the PC configuration program or from the PC configuration program to the RS 4 has the following structure:

Two consecutive 0x00 are used as token characters, i.e. identifier of a special value (control character). Within the actual information, the occurrence of a (0x00, 0x00) is extended with a 0xFF to avoid confusing with the token character, i.e. it must be prevented that a 0xFF follows a token in the protocol (by means of an appropriate specification). In this way, the occurrence of a (0x00, 0x00, 0xFF) can be unmistakably resolved as (0x00, 0x00). The additional 0xFF is also used in the calculation of the checksum.

Please find below a description of the structure

<table>
<thead>
<tr>
<th>Name</th>
<th>Num. Bytes</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start</td>
<td>2</td>
<td>The start character consists of the token (i.e. 0x00, 0x00) Together with the following command character, which possesses a value range from 0x01 - 0xFE, the start sequence (consisting of start character and command character) can be unmistakably different from the end character and from a (0x00, 0x00) in the user information.</td>
</tr>
<tr>
<td>Command</td>
<td>1</td>
<td>The command character possesses a value range from 0x01 - 0xFE. The command character specifies the method with which the user information which is to be transmitted must be interpreted. The command character is not analysed by the protocol, but rather according to the given operating status.</td>
</tr>
</tbody>
</table>
**ANNEX B: LÀSER – PROTOCOL RS4 PER A US EXTERN**

Leuze electronic

RS 4 / Software

Information on the RS4 Protocol for external use

<table>
<thead>
<tr>
<th>Name</th>
<th>Numb. Bytes</th>
<th>Description</th>
</tr>
</thead>
</table>
| Option 1 | 1 | The command is specified in chapter 0 Command description.  
The option characters transmit certain control flags, e.g. status information, which is transmitted together with each message.  
The first option character is always transmitted and defines protocol information, e.g. the total number of option characters being transmitted. All other option characters are transmitted only when required by the transmission, i.e. when at least one status bit is set.  
This option character is always transmitted together with the message being transmitted and has the following structure: |
| | | Bit |
| | | Number of option fields, i.e. at least 1 |
| | | Presently operating status |
| | | 0: without password  
1: with password |
| | | 0: without validation  
1: with validation |
| | | Reserved |
| | | The number of option fields is, thus, limited to 1-3 (bit 0 & bit 1)  
The field *current operating status* (bits 2-4) indicates the currently active operating status and has the following structure:  
000 - no data (for messages from PC -> RS 4)  
001 - initialization  
010 - measurement operation  
011 - configuration  
100 - error/fault  
As the number of option fields must be at least 1, the field Option 1 never has the value 0x00. |

| Option 2 | 1 | This option character transmits status information regarding the individual protected fields + error + warning + restart-disable.  
This option character is only transmitted when at least one flag is set, i.e. this field cannot possess the value 0x00. If this field is transmitted, the number of option fields in field Option 1 must be at least 2. |

---

File name: RS4_protocolExtern-e.doc  
Version: 1 0  
Department: Martin Kall / LOG  
Date: 09.06.01  
Page: 11 / 11

**PFC.: Fusió de dades volumètriques per obtenir models 3D**

pag. 85
This option character has the following structure:

```
    7 6 5 4 3 2 1 0
   +-----+-----+-----+-----+-----+-----+-----+-----+
   | 5a  | 4   | 3   | 2   | 1   | 0   |     |
   +-----+-----+-----+-----+-----+-----+-----+
          |      |      |      |      |      |      |
          | Act. personal safety field |      |      |      |      |      |
          | 0: free | 1: busy |
          +-----+-----+
          |      |      |
          |      |      |
          | Act. object safety field |      |      |      |      |
          | 0: free | 1: busy |
          +-----+-----+
          |      |      |
          |      |      |
          | Warning |      |      |      |      |      |
          |      |      |      |      |      |      |
          |      |      |      |      |      |
          | Error |      |      |      |      |      |
          +-----+-----+
          |      |      |
          |      |      |
          | Rupture disable |      |      |      |      |
          | 0: free | 1: busy |
          +-----+-----+
          |      |      |
          |      |      |
          | 2. Personal safety field |      |      |      |
          | 0: free | 1: busy |
          +-----+-----+
          |      |      |
          |      |      |
          | 2. Object safety field |      |
          | 0: free | 1: busy |
          +-----+-----+
          |      |
          |      |
          | 1 (for option 3) |
```

This field has no meaning for messages from the PC to the RS 4.

Bit 7 is set to 1 when Option 3 needs to be transmitted to prevent Option 2 from taking the value 0.
## ANNEX B: LÄSER – PROTOCOL RS4 PER A US EXTERN

**4 Leuze electronic**  
**RS 4 / Software**  
**Information on the RS4 Protocol for external use**

<table>
<thead>
<tr>
<th>Name</th>
<th>Numb.</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Option 3</td>
<td>1</td>
<td>This option character transmits status information regarding the selected protected field pair. This option character is only transmitted when at least one flag is set, i.e. this field cannot possess the value 0x00. If this field is transmitted, the number of option fields in field Option 1 must be at least 3. This option character has the following structure:</td>
</tr>
<tr>
<td></td>
<td></td>
<td><img src="" alt="Diagram of Option 3" /></td>
</tr>
<tr>
<td></td>
<td></td>
<td>As it is possible that two protected field pairs be simultaneously selected during protected field switching, two pieces of information are required for the selected protected field pair. For the selected protected field pairs:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 0: means nothing selected</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 1-4: means protected pair 1-4 selected</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This field has no meaning for messages from the PC to the RS 4.</td>
</tr>
<tr>
<td>Password</td>
<td>8</td>
<td>The password is transmitted in this field when bit 5 of Option 1 is set. The number of characters is always 8. The structure of the individual characters is as follows:</td>
</tr>
<tr>
<td></td>
<td></td>
<td><img src="" alt="Diagram of Password" /></td>
</tr>
<tr>
<td></td>
<td></td>
<td>i.e. bit 7 of the individual characters is always set. This is necessary to prevent consecutive 0x00. The individual characters of the password thus have a value range from 0x00 - 0x7F. If the password has no content (e.g. authorisation level 1), 0xFF is transmitted 8 times. All actual information is transmitted with 8 bits. The actual information is not interpreted by the protocol, but rather depending on the command and the current operating status of a command process.</td>
</tr>
</tbody>
</table>

**Filename:** RS4_protocol_extern-a.doc  
**Version:** 1.0  
**Department:** Martin Kaul / LOG  
**Date:** 09/05/01  
**Page:** 13 / 13

PFC.: Fusió de dades volumètriques per obtenir models 3D  pag. 87
Check characters | 1 | The check character corresponds to an XOR link of all transmitted characters, including command characters, password, option characters, i.e. following the start character to the last character preceding the check character.
In order to prevent confusing the end character, the check character must never possess the value 0xFF. If the result of the XOR link is determined to be a 0x00, a 0xFF is transmitted and considered in the analysis.
Comment: when transmitting configuration data, a CRC signature must be calculated and transmitted for the user information of all messages. This increases the level of security during the transmission of configuration data.

End | 3 | The end character consists of the token (i.e. 0x00 0x00) followed by a 0x00.
The end character can, thus, be unmistakably different from the start sequence (start character and command character) and a (0x00, 0x00, 0xFF).

Table 0-2 Protocol structure

The protocol makes it possible for the transmitter and receiver to mutually synchronize one another. This is because the reception of (0x00, 0x00) followed by a character not equal to 0x00 or 0xFF always identifies the start of a message and the reception of (0x00, 0x00, 0x00) always identifies the end of a message.

A message which could not be correctly transmitted is not repeated by the protocol. The transmitter service is responsible for the repetition of faulty messages (for example, when transmitting the configuration data, the function which generated the message).
4.2. Structure of the user information

This chapter describes the function of the data which are exchanged with the aid of the transmission protocol between the RS 4 and the PC configuration program.

In general, the following types of data are exchanged:

- **Measurement values**
  During measurement operation, the measured values are transmitted from the RS 4 to the PC configuration program.

- **Configuration data**
  The PC configuration program loads the configuration data from the RS 4 and also stores them on the RS 4. i.e. the configuration data are transmitted both from the RS 4 to the PC configuration program as well as from the PC configuration program to the RS 4.

- **Diagnostic data**
  With the aid of the PC configuration program, it is possible to query various diagnostic values from the RS 4, i.e. the diagnostic data are transmitted upon request by the PC configuration program from the RS 4 to the PC configuration program.

Various commands are defined for the transmission of the various data and for the control of the RS 4. The received data can be unmistakably identified by the receiver with the aid of the commands (for a list of the commands, see document Command Processing (Level 2) CMDPROC.DOC)

4.2.1. Strategies for the transmission of measurement values

The RS 4 measures the distance to the next object over an area of 190°. The calculated distance results are internally analysed for protected field monitoring, but may, however, be transmitted for external analysis/display via the serial interface of the RS 4.

Please find below a list of the associated data volumes and transmission times:

- The incremental encoder possesses 1000 edge changes over the entire rotation.
- The measurement range covers 190°, i.e. 529 measurement values are measured in the measurement range, i.e. 529 measurement values are measured per scan.
- The measurement values have a value range of 16 bits, where the resolution is approximately 2 mm, i.e. 2 bytes are required for the transmission of a measurement value.
- The measurement takes place at 25 rotations/sec, i.e. each rotation is 40 ms in duration. As a scan takes place over 190°, each scan is 21.1 ms in duration.
- Within a period of 21.1 ms, 529 measurement values are determined, i.e. 1 measurement value is determined every 40 μs.

Each scan produces a data volume of 2*529 bytes = 1058 bytes.

Plus the protocol overhead of 10 bytes (start+command+option+check character+end) yields a packet of 1068 bytes / scan for the transmission of all measurement values.

At a baud rate of 57600 baud, 185 ms are required for a packet, i.e. at a baud rate of 57600 baud not all measurement values for a scan can be transmitted.

Realization:

With the RS4, all measurement values are transmitted with 16 bits, where bit 0 specifies whether one of the active protected fields has been violated between the previously output measurement value and the given measurement value. These bits can be used on the PC when displaying the locations in the measurement value contour which violate an active protection field.
The resolution and size of the measurement value output can be defined in such a way that the output begins at a particular sector and ends at a particular sector and only every n-th value (beginning with the output start sector) is output. In this way the size of the measurement value output can be reduced.

The measurement value message is output by means of the RS4 protocol without acknowledgment. As the output of an entire measurement value message may take several rotations, depending on the quantity of data and the baud rate, the output of the measurement value message within the area being monitored is interrupted. While a measurement value is being output, scanning can, therefore, continue and the defined area monitored without overwriting the current measurement value output.

4.2.2. Structure of the transmitted measurement contour

This chapter describes the structure of the user data of the communication protocol during the transmission of a measurement contour from the RS4 to the PC configuration program.

Notice: Each measured contour possesses 529 measurement values (angle -5.04° - 185.04° with 0.36° resolution). Each individual measurement value possesses a number, beginning with 0, i.e. the measurement value at position -5.04° has the number 0 and the measurement value at position 0° has the number 14. Within a measurement message, it is not absolutely necessary that all values of the contour being measured be transmitted (see resolution, output start and output stop).

The following data are transmitted within a measurement contour:

<table>
<thead>
<tr>
<th>Address within the user data</th>
<th>Name</th>
<th>Description</th>
<th>Size in bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Scan number</td>
<td>The scan number is incremented by one with each rotation of the RS4. This scan number can be used to determine the temporal separation between two output measurement contours. The actual scan number is 32 bytes in size. In order to prevent a smaller value from creating a double zero during the transmission of 4 bytes, filler bytes with the value 0xFF are inserted between the individual bytes of the scan number. The scan number is thus structured as follows:</td>
<td>3</td>
</tr>
<tr>
<td>8</td>
<td>Resolution</td>
<td>This value specifies the distance between two individually transmitted measurement values. Assuming that the measurement contour is output beginning with measurement value 0 and ending with 529 and the resolution is set to value 1, the following values of the measured contour are then output:</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>Output start</td>
<td>This value specifies with which measurement value the output of the measured contour is to begin.</td>
<td>2</td>
</tr>
</tbody>
</table>

Filename: RS4_protocol_external.doc
Version: 1.0
Department: Martin Kaul / LOG
Date: 09.05.91
Page: 10 - 18

pag. 90 | PFC.: Fusió de dades volumètriques per obtenir models 3D
When transmitting 16-bit values within the message, the high byte is always transmitted first followed by the low byte.

4.2.3. Transmitting errors

Errors and warnings may occur due to various events. In the RS 4 errors are stored in an error buffer.

An error/warning consists of the following parts:

- **Number (2 bytes)**
  - Specifies the given error/warning
- **Parameter (2 bytes)**
  - Additional parameter for the given error/warning number, e.g. in the event of window soiling an indicator of which area of the window is soiled.
- **Error location (2 bytes)**
  - The location at which the error/warning occurred.

For 16-bit values, first the high byte and then the low byte is transmitted.

The error numbers + corresponding parameters are described in a separate realization document.

The PC configuration program can read out the error buffer of the RS 4. For this purpose, the PC configuration program sends a request to the RS 4. The RS 4 then sends the entire content of the error buffer to the PC configuration program.
4.3. Command description

This chapter lists all commands which are transmitted between the RS 4 and the PC configuration program.

The individual commands may possess a value of 0x01 - 0xFE - for further information, see document Serial Communication (Level 3) (SER_COMM.DOC)

Operating mode: K - configuration
M - measurement operation
E - error/fault

<table>
<thead>
<tr>
<th>Command</th>
<th>Group</th>
<th>Significance (HEX)</th>
<th>Description</th>
<th>RS 4 → PC</th>
<th>PC → RS 4</th>
<th>with password</th>
<th>with acknowledgment</th>
<th>min Access Level</th>
<th>Operating mode</th>
<th>with user information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mess. W/RT</td>
<td>Measurement</td>
<td>0x21</td>
<td>Transmission of the measurement values in 16-bit format with current scan number. This number is incremented by 1 with each rotation of the mirror and can be used to determine the distance between two successive measurement value outputs</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>G</td>
<td>x</td>
</tr>
<tr>
<td>Error Occur</td>
<td>Error</td>
<td>0x33</td>
<td>This command is sent by the RS 4 to the PC configuration program in the event of an error. The error message is included in the transmission as a parameter (consisting of error number, parameter and error location)</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>MIRE</td>
<td>x</td>
</tr>
<tr>
<td>Warning Occur</td>
<td>Error</td>
<td>0x54</td>
<td>This command is sent by the RS 4 to the PC configuration program in the event of a warning. The warning message is included in the transmission as a parameter (consisting of warning number, parameter and warning location).</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>MIRE</td>
<td>x</td>
</tr>
</tbody>
</table>

Table 0-3 Commands

If the recipient of a message does not know a command, it is ignored. If, as per the acknowledgement flag of the protocol, the transmitter expects an acknowledgement, the receiver sends in this case a negative acknowledgement.
PFC.: Fusió de dades volumètriques per obtenir models 3D
Small, Powerful and Fast

Compact, powerful IRB 140 industrial robot. Six axis multipurpose robot that handles payload of 6 kg, with long reach (810 mm). The IRB 140 can be floor mounted, inverted or on the wall in any single. Available as Standard, Foundry Plus, Clean Room and Wash versions, all mechanical arms completely IP67 protected, making IRB 140 easy to integrate in and suitable for a variety of applications. Uniquely extended radius of working area due to bend-back mechanism of upper arm, axis 1 rotation of 360 degrees even as wall mounted.

The compact, robust design with integrated cabling adds to overall flexibility. The Collision Detection option with full path retraction makes robot reliable and safe.

Using IRB 140T, cycle-times are considerably reduced where axes 1 and 2 predominantly are used. Reductions between 15-30 % are possible using pure axis 1 and 2 movements. This faster version is well suited for packing applications and guided operations together with PickMaster.

IRB Foundry Plus and Wash versions are suitable for operating in extreme foundry environments and other harsh environments with high requirements on corrosion resistance and tightness. In addition to the IP67 protection, excellent surface treatment makes the robot high pressure steam washable. The white-finish Clean Room version meets Clean Room class 10 regulations, making it especially suited for environments with stringent cleanliness standards.
TECHNICAL DATA, IRB 140 INDUSTRIAL ROBOT

SPECIFICATION

<table>
<thead>
<tr>
<th>Specification</th>
<th>IRB 140</th>
<th>IRB 140T</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reach (mm)</td>
<td>810</td>
<td>810</td>
</tr>
<tr>
<td>Weight (kg)</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>Type</td>
<td>Foundry Plus Protection</td>
<td>Clean Room</td>
</tr>
<tr>
<td>Mod.</td>
<td>IRB 140/CH</td>
<td>IRB 140/T</td>
</tr>
<tr>
<td>Weight (kg)</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>Position (mm)</td>
<td>810</td>
<td>810</td>
</tr>
<tr>
<td>Dimensions</td>
<td>W x D x H</td>
<td>400 x 410 x 400</td>
</tr>
<tr>
<td>Work envelope</td>
<td>960 x 980 x 460</td>
<td></td>
</tr>
<tr>
<td>Weight</td>
<td>995 kg</td>
<td></td>
</tr>
<tr>
<td>Interchangeability</td>
<td>5 - 65°C</td>
<td></td>
</tr>
<tr>
<td>Degree of protection</td>
<td>95%</td>
<td></td>
</tr>
</tbody>
</table>

ELECTRICAL CONNECTIONS

Supply voltage: 200-240 V, 50/60 Hz
Robot power transformer rating: 4.5 kW
Power consumption typical: 0.4 kW

PHYSICAL

Robot mount: any angle
Robot weight: 995 kg

ENVIRONMENT

Ambient temperature: 5 - 65°C
Relative humidity: 95%

SAFETY

Class 10 flammable gas, Class 1 Class 3, ISO 14999

WORKING RANGE AND LOAD DIAGRAM

www.abb.com/robotics

pag. 96

PFC.: Fusió de dades volumètriques per obtenir models 3D
ANNEX D: CONTROLADORA ABB IRC5

15. ANNEX D: CONTROLADORA ABB IRC5

IRC5
Industrial Robot Controller

TECHNICAL DATA, IRC5 INDUSTRIAL ROBOT CONTROLLER

PERFORMANCE
Control hardware: Multi-processor system
Input: 144 inputs
Output: 128 outputs
Program: 9999 program memories
Language: English

Control software: Object oriented system
High-speed I/O handling
Graphs
Portable, interface expandable
PC-AT compatible
Robust software architecture
Pre-built database, Mass storage

SYSTEM CONNECTIONS
Stable voltage: 240 Volts 50-60 Hz
Power supply: Single phase 300-650 Watts

PHYSICAL
Size (W x H x D): 672 x 258 x 780 mm
Weight: 150 kg

Electrical cabinet: 420 x 250 x 760 mm
Robot model: 350 x 250 x 500 mm
Common equipment: 280 x 250 x 760 mm

Environment:
Absolute humidity: 0 to 90% at 40°C
Dust and dirt: IP54
Cooling: Air cooling
Waterproofing: Yes
Ventilation: No

MACHINES AND INTERFACES
Input: 114 inputs
Output: 128 outputs
Graphs: 24 inputs, 24 outputs
Sensors: 12 inputs, 24 outputs
Network: Ethernet (TCP/IP)

Sensors:
- Sensors: 24 inputs, 24 outputs
- Feedback: 24 inputs, 24 outputs

ELECTRICAL POSITION
Switches: 5-volt outputs inverting and linear

Sensors and interface:
- Sensors: 24 inputs, 24 outputs
- Feedback: 24 inputs, 24 outputs

FEATURES AND FUNCTIONALITY
Versatile, expandable, modular and easy to use

ABB

PFC.: Fusió de dades volumètriques per obtenir models 3D

pag. 97
Fifth generation robot controller

Based on more than three decades of industrial experience, the IRC5 industrial robot controller implements proven advancements in motion control technology. It is designed to enhance safety, productivity and accessibility, offering new features and a high level of safety, user-friendliness and ease of installation.

Safety

Operator safety is a central pillar of the IRC5 technology and is in line with regulations, as well as the third party requirements, ensuring the highest performance and safety level. It is the result of a long-term investment on safety applications, being the first level of a new generation of safety applications that ensures a safer robot environment.

Motion control

In order to ensure precise and dynamic positioning, the IRC5 is equipped with the best algorithms for the physically shortest possible time. This means QuickMove, the high-path accuracy algorithm, which allows for superior performance without sacrificing performance and high-precision performance in parallel with the programming.

Modularity

The IRC5 is designed to be modular, allowing for easy expansion and reconfiguration at any time. This flexibility makes it possible to adapt to any configuration and cell layout.

FlexPendant

The IRC5's compact design is characterized by its sleek, clean lines, easy-to-read screens, and intuitive operation. The FlexPendant is an application software interface that provides an easy-to-use, high-performance operator interface, reducing the need for a separate cell HMI.

RAPID

The RAPID programming language provides the user with a comprehensive development environment, including features such as a powerful language, advanced features, extensibility, user-friendly environment, and expert system knowledge control.

Communication

The IRC5 is designed to be readily field-configurable and is well suited for integration into existing systems. Its interface with other robots, machines, and network devices is based on standards for powerful networking technologies.

RobotStudio

The IRC5 is compatible with the RobotStudio, providing a powerful tool for planning, programming, and commissioning features.

MultiMove

Through MultiMove, the IRC5 is able to connect multiple robots from one controller, allowing for a flexible setup that can be adapted to the needs of any application. This feature is ideal for large-scale industrial applications, providing a cost-effective solution for high-end industrial applications.

ABB

pag. 98

PFC.: Fusió de dades volumètriques per obtenir models 3D
Cortona® VRML Client 5.0

User's Guide
### Table of contents

- The Cortona VRML Client Window: 3
- Navigating in Cortona VRML Client: 3
  - Using Viewpoints: 3
  - Moving around: Walk, Fly, and Study: 3
  - Restore, Fit, and Align: 7
  - Using Keyboard for Navigating: 7
- Setting Cortona VRML Client Options: 8
  - General: 8
  - Renderer: 9
  - Navigation: 9
  - Skin: 9
- Interacting with the Scene: 10
- Using Cortona VRML Client in HTML Documents: 10
  - An example of using EMBED: 10
  - An example of using OBJECT: 11
  - Using Cortona VRML Client Automatic Installation: 11
The Cortona VRML Client Window

With Cortona VRML Client you can see and explore 3D worlds. Cortona VRML Client will start automatically when you open a file containing VRML world. There are two parts of the Cortona VRML Client window:

1. Toolbars:
   - The vertical toolbar, which contains buttons used to specify navigation type in a world.
   - The horizontal toolbar, which contains buttons with predefined actions to change your position in a world.
2. The 3D window, which shows VRML world.

There is also pop-up menu, which you access by pressing the right mouse button while the pointer is over toolbars or 3D window.

Some worlds do not allow you navigating in a world, so toolbars may be invisible.

Navigating in Cortona VRML Client

Moving through a 3D space is similar to moving a camera. Think of a video camera that captures images in the real world and converts them into electronic signals for viewing on a screen. It has a position and orientation, and these are independent attributes. Your movements in the world continually position and orient that camera. Use the camera controls on the vertical toolbar to move the camera through the 3D Space.

This concept assumes that there is a real person viewing and interacting with the VRML world. Nevertheless, the VRML author may place any number of viewpoints (or cameras) in the world - interesting places from which the user might wish to view the world. Only one viewpoint may be active at a time. This section describes the mechanisms that Cortona VRML Client provides for navigating in a three-dimensional space.

Using Viewpoints

A viewpoint describes a position and orientation for viewing the scene. The VRML author will probably want to guide the user to the best vantage points for viewing it. Not all worlds contain viewpoints, but when they do, you can use them.

To activate a viewpoint, do one of the following:

- Click the VIEW in the horizontal toolbar or choose viewpoints from the pop-up menu, and then select a viewpoint from the list of the predefined viewpoints.
- Click the arrow buttons to move to the next or previous viewpoint. You can also press Page Down or Page Up keyboard button.

Note:
If there are no predefined viewpoints in a world, the Empty message appears.

Moving around: Walk, Fly, and Study

There are three main navigation modes that Cortona VRML Client offers: WALK, FLY and EXAMINE. You can switch the navigation mode by clicking buttons on the vertical toolbar (click STUDY to enter EXAMINE mode). Each navigation mode may have several options: PLAN, PAN, TURN and ROLL. The combination of navigation mode and its option determines the possible camera motion and its orientation. Please note that the VRML author has an opportunity to specify which navigation paradigm should be used in the scene by default. Some worlds don’t allow the user to use navigation controls, but they may provide on-screen cues to navigation.
You can navigate with the mouse, the keyboard, or both mouse and keyboard. To move around a 3D world using the mouse:

1. Choose a navigation mode.
2. Position the pointer anywhere in the 3D window and press the left mouse button.
3. Move the mouse while holding down its left button. The direction in which you drag the mouse determines the camera motion.
4. Release the left mouse button to stop moving.

Note: The distance that you drag the mouse determines the speed with which the camera moves. If you stop moving the mouse, the camera will continue moving until you release the mouse button. To accelerate the camera's movement or rotation, press SHIFT, CTRL or SHIFT+CTRL.

Use WALK+PLAN to move in a horizontal plane.

- ↑ Forward - move closer
- ↓ Backward - move further
- → Right - turn to the right
- ← Left - turn to the left

Note: Move the mouse forward or backward while holding down the Space key to turn upward, downward.
Move the mouse left or right while holding down the ALT key to move left or right.

Use WALK+PAN to move left or right in a horizontal plane.

- ↑ Forward - move closer
- ↓ Backward - move further
- → Right - move right
- ← Left - move left

Note: When you move the camera right or left, the world will appear to move in the opposite direction.

Use WALK+TURN to change the angle of the camera in a world.

- → Forward - turn upward
Cortona VRML Client User's Guide

- Backward - turn downward
- Right - turn to the right
- Left - turn to the left

Use FLY+PLAN to move left or right.

- Forward - move the camera forward towards its longitudinal axis
- Backward - move the camera backward
- Right - turn the camera to the right around its vertical axis
- Left - turn the camera to the left around its vertical axis

**Note:** The camera's vertical axis may be inclined in a 3D Space.
Move the mouse while holding down the ALT key to switch FLY+PAN.
Move the mouse while holding down the Space key to switch FLY+TURN.
Move the mouse while holding down the ALT+Space keys to switch FLY+ROLL.

Use FLY+PAN to move up, down, left, or right within a single vertical plane.

- Forward - move up
- Backward - move down
- Right - move right
- Left - move left

**Note:** When you move the camera, the world will appear to move in the opposite direction.

Use FLY+TURN to turn the camera.

- Forward - turn the camera upward around its horizontal axis
- Backward - turn the camera downward around its horizontal axis
- Right - turn the camera to the right around its vertical axis
Left - turn the camera to the left around its vertical axis

and

Use FLY+ROLL to incline the camera.

Right - incline to the left

Left - incline to the right

and

Use STUDY+PLAN to examine an object from various angles.

↑ Forward - move the camera forward

↓ Backward - move the camera backward

↑ Right, Left - move the camera around the central point which is defined by the center of bounding box of the geometry in the 3D scene.

and

Use STUDY+TURN to examine an object from various angles.

↑↑ Forward, Backward, Right, and Left - move the camera around the central point which is defined by the center of bounding box of the geometry in the 3D scene.

Note: Move the mouse while holding down the ALT key to switch STUDY+PAN.
Move the mouse while holding down the Space key to switch STUDY+PLAN.
Move the mouse while holding down the ALT+Space keys to switch STUDY+ROLL.

and

Use STUDY+ROLL to incline the camera around the central point which is defined by the center of bounding box of the geometry in the 3D scene.

Right - incline to the left

Left - incline to the right

Note: The VRML author can choose the rotation center in the EXAMINE navigation mode. By default, when in EXAMINE mode, Cortona VRML Client sets the center of rotation in the center of the bounding box of the scene geometry. But it is possible to change this center by specifying three additional values in the avatarSize field of the NavigationInfo node. According to the VRML specification, the first three values
Cortona VRML Client User’s Guide

are the avatar dimensions, while additional values may be used for browser-specific purposes. In Cortona VRML Client the fourth, fifth and sixth values specify the center of scene rotation in the world coordinates X, Y, and Z.

Use GOTO to move close to object in a world. Select GOTO in the toolbar and then click on an object in the world. You’ll move directly to it.

**Restore, Fit, and Align**

Cortona VRML Client provides three mechanisms that can help to re-orient a camera if you have lost your way in a world. Unlike the navigation tools, these buttons invoke predefined actions that take place as you click on them.

Use RESTORE to automatically return to the loaded world’s original active viewpoint.

Use FIT to make the scene view fully visible in the Cortona VRML Client 3D window.

Use ALIGN to position the camera’s horizontal and longitudinal axes parallel to the scene horizontal plane.

**Using Keyboard for Navigating**

You can control the camera using keyboard commands. The functionality for buttons corresponds to the movement of your mouse and depends on the navigation type, its option, and the specified skin. Please note that the following description of keyboard commands is presented for the FLY+PLAN navigation and the Default skin.

- **Arrow Up** - move closer
- **Arrow Down** - move further away.
- **Arrow Right** - turn to the right.
- **Arrow Left** - turn to the left.
- **Arrow Up on the numeric keypad** - move closer.
- **Arrow Down on the numeric keypad** - move further.
- **Arrow Right on the numeric keypad** - move to the right. If the STUDY mode - move the camera around the center of rotation.
- **Arrow Left on the numeric keypad** - move to the left. If the STUDY mode - move the camera around the center of rotation.
- **7 on the numeric keypad** - turn the camera downward around its horizontal axis.
- **9 on the numeric keypad** - turn the camera upward around its horizontal axis.
- **1 on the numeric keypad** - incline to the right. Note: It is not available in WALK mode.
- **3 on the numeric keypad** - incline to the left. Note: It is not available in WALK mode.
- **Gray Plus** - move up in the case of FLY mode.
- **Gray Minus** - move down in the case of FLY mode.
- **1 on the alphanumeric keyboard** - incline to the right. Note: It is not available in WALK mode.
- **2 on the alphanumeric keyboard** - incline to the left. Note: It is not available in WALK mode.
- **3 on the alphanumeric keyboard** - turn downward. If the STUDY mode - move the camera around the center of rotation.
- **4 on the alphanumeric keyboard** - turn upward. If the STUDY mode - move the camera around the center of rotation.
- **Page Down** - next viewpoint.
- **Page Up** - previous viewpoint.
Cortona VRML Client User's Guide

- Pause - Enables or disables CylinderSensor, PlaneSensor, SphereSensor, and TouchSensor in Cortona VRML Client.

To accelerate the camera's movement or rotation: Press SHIFT, CTRL or SHIFT+CTRL and one of the above keys simultaneously.

The ALT and SPACE keys allow the user to quickly change the navigation option to PAN and TURN accordingly in the FLY navigation mode (to ROLL and PLAN in the STUDY mode). The ALT and SPACE keys simultaneously pressed activate ROLL in the FLY navigation mode or PAN in STUDY.

Setting Cortona VRML Client Options

The following options are supported from the pop-up menu of 3D window:

- **Viewpoints.** Activates a list of the predefined viewpoints.
- **Headlight.** Cortona VRML Client automatically includes a light for the viewer in every world. The headlight always shines directly in front of the camera. You can switch the headlight on and off.
- **Navigation.** You can select a navigation mode.
- **Speed.** Controls the rate at which a camera moves through a world.
- **Full Screen.** This hides most screen elements so that you can view the whole scene. To close Full Screen, press ESC or F11.
- **Hide/Show Toolbars.** You can show or hide toolbars.
- **Show/Hide Console.** Shows or hides the VRML console containing errors or warnings (for VRML developers).
- **Preferences.** Modifies settings for Cortona VRML Client such as screen appearance, a renderer mode, and other options.
- **Help.** Lets you view the online Help system and general information about Cortona VRML Client.

You can configure Cortona VRML Client options according to your preferences. To change settings, choose Preferences from the right-button pop-up menu. The ParallelGraphics Cortona Control dialog box is displayed.

**General**

- **Loading.** The Show progress check box determines whether the current state of the loading process is shown. If the Wait for all resources is selected, the scene will not be shown until all resources are loaded. If it is cleared, the scene’s geometry will be shown immediately after the main UI is loaded.
- **Appearance.** You can change a background color of Cortona VRML Client 3D window. Click the Background color box at the General tab, click the desired color from the palette, and then click OK. Please note that VRML authors can also control a color that simulates ground and sky.
- **Display frame rate.** Allows you to display the frame rate on the status bar.
- **Console mode.** Shows or hides the VRML console containing errors or warnings. The ConsoleMode attribute is set to Autolaunch by default. This shows the VRML console if errors or warnings occur.
- **CPU load.** Determines the degree of acceleration for Cortona VRML Client renderer. This allows you to specify the processor usage in the range from 0 (minimum frame rate, maximum the processor idle time) to 100 (maximum frame rate).

**Scene**

- **Scene location.** Shows the currently opened VRML file and lists the files you have recently opened. To quickly reopen one of these files, click it in the Scene location list box and then click Apply. You can also open a document on your computer hard disk or on a network drive that you have a connection to by clicking the Browse button.

---

pag. 106

PFC.: Fusió de dades volumètriques per obtenir models 3D
Cortona VRML Client User's Guide

Renderer

Allows you to select a rendering mode. To draw the 3D image, Cortona VRML Client provides two hardware renderers: OpenGL and DirectX. If your system has hardware acceleration for either OpenGL or DirectX (note: you must have DirectX 5 or greater installed), choose the appropriate renderer. Using hardware renderers can introduce limitations. Select a renderer to compare the performance and visual quality of hardware versus software rendering and set your preferences accordingly.

Renderer options

- **Dither colors if needed.** Controls whether Cortona VRML Client dithers while rendering. Dithering improves the quality of rendering, but may lower performance.
- **Wireframe rendering mode.** The object appears to be outlined with wires rather than solid.
- **Anti-aliasing.** The technique for smoothing out jaggies in showing curves on computer monitor. There are two different modes: Idle-time and Real-time. If the Idle-time option is on, it is applied only for static scene. Whereas selecting Real-time initiates smoothing even in the viewer movement but results in decreasing frame rate.
- **Do not render textures.** To turn the textures on or off.
- **Optimize texture for quality.** All textures are optimized for quality. Note that if you select both Optimize texture for speed and quality, the renderer uses the available resources to optimize speed and quality simultaneously.
- **Limit textures size.** This controls the texture resolution of the image being pushed to the graphics card. If the image resolution is bigger than the value you choose, the renderer will reduce it. The smaller the resolution, the faster the rendering but poor quality.
- **Strict VRML97 compliance.** When selected, Cortona VRML Client renders VRML scene in accordance with the VRML97 Specification. If it is turned off (the default), simplified lighting equations and rendering techniques, which provide higher performance, are used.
- **Enable pixel buffer access.** This option should be used for the DirectX 9.0 renderer only. It is effective in the following cases:
  - When the Idle-time (still scenes) option of the Anti-aliasing group is selected.
  - When Cortona is used within Cortona SDK applications that allow for the production of screenshots.
- **Renderer version.** This is accessible only for Direct3D renderers. Allows you to select a rendering mode in the case you have DirectX 7 or a later version installed.

Navigation

- **Navigation mode.** This allows you to select the navigation mode from a list of available navigation paradigms.
- **Travel speed.** Sets the rate at which the viewer travels through a scene. The following are the typical values for the rates, in metres per second: Slowest=0.0625, Slower=0.25, Normal=1, Faster=4, Fastest=16. If the speed field of the NavigationInfo is specified in the VRML file, the corresponding speed factors are multiplied.
- **Animate viewpoints.** Sets the Viewpoint transition rules that specify how Cortona VRML Client interprets the transition from the old viewpoint to the new one. Auto - defined by the jump field of the Viewpoint node of a new location. Always - a jump with the transition effect. Never - instantaneous transition.
- **Collision detection.** By default, Cortona VRML Client will allow you to pass through objects in your path. To prevent from passing through objects, select Always in the Collision detection box at the Navigation tab. Select Auto to use the collision method specified by VRML author.
- **Headlight on.** Mark the check box to turn the headlight on.
- **Show hidden viewpoint.** This allows you to see all viewpoints of the scene in the VIEW list including those that do not have a description (a Viewpoint's description field is empty).

Skin

Use the skins to change the appearance of the dashboard in the Cortona VRML Client window. This version of Cortona VRML Client includes at least two simple skins: Default and CAD-like.
To apply a skin:

- In the list of skins select the skin you want to apply and then click Apply.

When you apply a skin it is then displayed in the Cortona VRML Client 3D window. Please note that Cortona VRML Client provides the capabilities for creating your own user interface. The description of the types of files you can use to make up a complete skin can be found at http://www.parallelgraphics.com/developer/products/cortona.

**Interacting with the Scene**

Some of the objects in the scene may provide special effects that allow the user to interact with the scene in different ways. As you position the pointer over the object, containing a VRML sensor, the pointer changes:

- ![Touch Sensor](image). Detects a click or the pointer over the object. The sort of activity in the scene is decided by the VRML author.
- ![Anchor](image). Clicking will link to the other object, VRML world or HTML document.
- ![Cylinder Sensor](image). Transforms pointer motion into a rotation of the object around its axis.
- ![Sphere Sensor](image). Transforms pointer motion into a rotation of the object around its two axes.
- ![Plane Sensor](image). Transforms pointer motion into a moving of the object.
- ![Drop Sensor](image). (VRML extension). Handling a drag-and-drop operation. Retrieves an object's uniform resource locator (URL) of an object (resource) dragged to the 3D window.

**Note:**

The following node types are pointing-device sensors: Anchor, CylinderSensor, PlaneSensor, SphereSensor, and TouchSensor. You can enable or disable them in Cortona VRML Client excepting the Anchor sensor by pressing the Pause keyboard button.

**Using Cortona VRML Client in HTML Documents**

To display a VRML world in a Web browser authors should use the EMBED or OBJECT (supported by Internet Explorer) elements with the proper attributes in an HTML document:

**An example of using EMBED**

```
<EMBED SRC="file.wrl"
WIDTH="300"
HEIGHT="300"
PLUGINPAGE="http://www.parallelgraphics.com/cortona"
VRML_DASHBOARD="FALSE"
VRML_BACKGROUND_COLOR="#000077"
CONTEXTMENU="FALSE">
```

where

- SRC specifies the name of the VRML scene to be loaded.
ANNEX E: PLUGIN VRML CORTOLA

Cortona VRML Client User's Guide

- **PLUGINSPAGE** If the user doesn't have a plug-in installed to handle the defined object data type, then it is possible to guide the user to a different URL (the value of the PLUGINSPAGE attribute) to provide instructions on installing the necessary plug-in module.
- **VRML_DASHBOARD** "TRUE" - turns on horizontal and vertical toolbars. "FALSE" - turns off.
- **VRML_BACKGROUND_COLOR** "#rgrgbh" specifies the background color of the 3D window (hexadecimal).
- **CONTEXTMENU** "TRUE" - enables context menu in the 3D window; "FALSE" - disables.

For more information about the Cortona plug-in specific attributes, see (http://www.parallel graphics.com/developer/products/cortona/html/attrib.html).

An example of using OBJECT

```xml
<Object
CLASSID="CLSID:86A88967-7A26-11d2-8EDA-006008185EDB1"
WIDTH="300"
HEIGHT="300">
<PARAM NAME="Scene" value="file.wrl"/>
</Object>
```

where

- **CLASSID** identifies the Cortona ActiveX control for the browser. The value must be entered exactly as shown.
- **WIDTH, HEIGHT** standard formatting attributes.
- **Scene** specifies the name of the VRML scene to be loaded.

For more information about the Cortona VRML Client properties, see (http://www.parallel graphics.com/developer/products/cortona/html/object/).

Using Cortona VRML Client Automatic Installation

Internet Explorer can automatically download Cortona VRML Client if it is not already installed. It gives the opportunity to reduce download size and choose Cortona VRML Client as VRML viewer despite the fact that the other VRML browser is installed on the user's computer. This technology provides to developers the CAB file for HTML based Internet installations of Cortona VRML Client.

An example of Using Cortona VRML Client Automatic Installation

```xml
<!-- Instantiates Cortona VRML Client components for MS IE -->
<Object
CLASSID="CLSID:86A88967-7A26-11d2-8EDA-006008185EDB1"
CODEBASE="http://www.parallel graphics.com/bin/cortvrml.cab"
WIDTH="300"
HEIGHT="300">
<PARAM NAME="Scene" value="file.wrl"/>
</Object>
```

where

- **CODEBASE** identifies the location of the Cortona VRML Client (CAB file cortvrml.cab, size: 1189KB) so that the browser can automatically download it if it is not already installed.
- **Scene** specifies the name of the VRML scene to be loaded.
