



Proyecto final de carrera  
Ingeniero de telecomunicaciones

***Título: “Diseño de un sistema  
virtual de tanques de fluidos”***

Autor:  
Francisco González Jurado

Tutor:  
Jorge Chávez  
Departamento de electrónica

*Escuela Superior de Ingenieros de Sevilla  
Universidad de Sevilla  
Octubre 2010*

## Indice de contenidos

Indice de figuras.....	III
Indice de tablas.....	V
Agradecimientos.....	VI
Introducción.....	1
Concepto y objetivos del proyecto.....	4
Descripción de la solución propuesta.....	5
CAPITULO 1. Arquitectura del sistema.....	8
1.1 Hardware y protocolos.....	8
1.1.1 Protocolo Profibus.....	8
1.1.2 Modelo Siemens ET.....	11
1.1.3 Profibus A71 DP-PC-SL (PCI).....	12
1.2 Sistema real de tanques.....	14
CAPPITULO 2. Herramientas software.....	16
2.1 Componentes Software, esquema de conexión.....	16
2.2 Comparativa CAD software.....	17
2.3 Flux Studio.....	18
2.4 WinMod 5.0.....	18
2.4.1 Configuración del software WinMod.....	18
2.4.2 Comunicación con WinMOD 3D.....	19

2.5 WinMOD 3DView.....	22
2.5.1 Propiedades de WinMOD 3Dview.....	22
2.6 Codesys V2.3.....	24
2.6.1 Propiedades de CoDeSys V2.3, configuración general.....	24
2.6.2 Esquema de conexión entre Codesys y WinMod .....	26
2.7 Simatic Manager.....	27
2.7.1 Propiedades de Simatic Manager.....	27
CAPÍTULO 3. Diseño del modelo virtual.....	30
3.1 Diseño del CAD con Flux Studio .....	30
3.1.1 Visualización atractiva.....	30
3.1.1 Conexión con WinMod.....	31
3.2 Modelo del Sistema en WinMod.....	33
3.3 Comunicación exterior PROFIBUS.....	38
CAPÍTULO 4: Sistemas de control.....	40
4.1 Introducción.....	40
4.1 Control a través de CoDeSys2.3.....	41
4.3 Control con Siemens S7. Simantic Manager.....	44
CAPÍTULO 5: Conclusiones .....	46
5.1 Herramienta en uso.....	46
5.2 Esquema general.....	48
5.3 Problemas encontrados .....	49
5.4 Trabajo futuro.....	50
5.5 Diagrama de Gannt.....	50
Bibliografía.....	52

## Índice de figuras

Fig 1.1: Esquema Bus Profibus DP.....	9
Fig 1.2: Esquema comunicación Master Class 2.....	10
Fig 1.3: Esquema comunicación a través de Publisher.....	11
Fig 1.4: Siemens S7, CPU .....	11
Fig 1.5: WinMod internal.....	13
Fig 1.6: Tarjeta Profibus PC .....	13
Fig 1.7: Sistema real.....	14
Fig 2.1: Comunicación Profibus entre PC's.....	18
Fig 2.2: Configuración comunicación exterior de WinMod .....	19
Fig 2.3: Unidad de Comunicacion.....	20
Fig 2.4: Configuración WinMod 3D comm.....	21
Fig 2.5: Configuración de las propiedades de WinMod3D.....	22
Fig 2.6: Comunicacion con WinMod.....	23
Fig 2.7: Importando señal.....	24
Fig 2.8: CodeSysV2.3 propiedades generales.....	25
Fig 2.9: Opciones generales. Opciones de Codesys.....	25
Fig 2.10: Atributos de objetos.....	26
Fig 2.11: Comunicación CodeSys a través de Profibus.....	26
Fig 2.12: Esquema Profibus en Simatic Manager.....	27
Fig 2.13: Configuración DC-PC-SL Siemens .....	28
Fig 2.14: Lista de simbolos S7.....	29
Fig 3.1: Diseño 3D .....	31
Fig 3.2: Archivo de texto para al comunicación 3D.....	32
Fig 3.3: Modelo final en 3D.....	32
Fig 3.4: Esquema del modelo de un tanque.....	33
Fig 3.5: Macro del Tanque 1, en WinMod.....	35
Fig 3.5: Macro de una válvula, en WinMod.....	35

Fig 3.5: Macro de una bomba, en WinMod.....	35
Fig 3.6: El modelo en WinMod.....	37
Fig 3.8: Control PAD.....	38
Fig 3.9: Transmisión DP-PC-SL.....	39
Fig 4.1: Pantalla principal de codesys y Pad de control.....	41
Fig 4.2: Control Manual en CoDeSys2.4.....	42
Fig 4.3: Arquitetura del Sistema.....	43
Fig 4.4: Pantalla principal Simantic Manager.....	44
Fig 4.5: Operando manualmente con S7-300.....	44

## Índice de tablas

Tabla 1.1: Tabla líneas Profibus.....	9
Tabla 1.2: Tabla de especificaciones de las CPUs de Siemens.....	12
Tabla 2.1: Tabla comparativa software CAD.....	17
Tabla 3.1: Macro en WinMod.....	34

## **Agradecimientos**

Ante todo quiero agradecer al Dr. Robert Kremer, un gran profesor y persona, quien me dió la oportunidad de tomar parte en este proyecto en la Fachhochschule Konstanz. Gracias por la atención y por darme la oportunidad de volver al Bodensee un lugar difícil de olvidar.

También quiero agradecer al Profesor Dr. Wihelm Fromm, quien ha sido mi tutor durante todo el proceso y a pesar del poco tiempo del que disponía, siempre atendió mis dudas con gran rapidez y eficiencia, he aprendido mucho al lado de este profesor.

Muchas gracias también a Doris Heiser, quien me ayudó con muchos de los problemas que me fueron surgiendo a lo largo de mi estancia en Alemania.

Todo lo ocurrido alrededor del desarrollo de este proyecto ha sido muy enriquecedor, sin duda una etapa de mi vida me ha aportado muchos conocimientos, experiencia, y sobre todo, una nueva actitud ante la vida.

También me enorgullece agradecer a mis padres el apoyo que me ha dado para que pudiera realizar los estudios y posteriores prácticas, aquí en la ciudad del Lago.

**Este trabajo ha sido apoyado y supervisado por “Hochschule  
Konstanz Technik, Wirtschaft und Gestaltung”, Universidad de  
ciencias aplicadas**



# **INTRODUCCIÓN**

## **Presentación del problema y justificación**

Las limitaciones de espacio físico, personal especializado y presupuesto para la dotación, implementación, operación y mantenimiento de laboratorios de control de procesos basado en plantas piloto supone una razón de peso suficiente para justificar el desarrollo e implementación de soluciones basadas en herramientas virtuales. Con tal estrategia se consigue la ampliación de equipos y la obtención de un ratio (alumno/unidad de prácticas) equilibrado acorde con las necesidades deontológicas enfocadas a la adquisición de habilidades y destrezas en relación con el aprendizaje y/o la investigación en el campo de la ingeniería de control de procesos industriales.

La herramienta que este proyecto muestra, se presenta como una versátil alternativa de apoyo en el desarrollo de las prácticas de laboratorio, así como prácticas de adquisición de experiencia en el manejo de plantas de proceso industrial.

Esta herramienta brinda al alumnado la posibilidad de flexibilizar el esfuerzo y dedicación en relación con la gestión del tiempo, estableciendo libremente su ritmo de adquisición de conocimiento. Supone tácitamente el aprovechamiento del tiempo de forma óptima ya que evita desplazamientos físicos desde el lugar de estudio hasta el lugar de ubicación física del laboratorio de prácticas que es el caso del uso de la herramienta de manera telemática (que es el fin que persiguen los profesores en el laboratorio de automática de la universidad de Konstanz), evitará desplazamientos físicos innecesarios desde el lugar de estudio (hogar, cyber-aulas, cyber-cafés, etc) hasta el lugar de ubicación física del laboratorio de prácticas.

El desarrollo de herramientas abiertas de software para simular, visualizar y programar equipos o instalaciones costosas, permitirán a estudiantes, profesores maestros e investigadores, mejorar el proceso de aprendizaje e investigación.

Desde una perspectiva global en este laboratorio se tiene como principal objetivo el

desarrollo de aplicaciones que permitan la adquisición de conocimientos sobre el manejo y visualización de experimentos tele-operados, así según sus propias palabras: *"pretendemos antes de finales del año 2013 poner a disposición de los estudiantes un laboratorio virtual completo, al cual los alumnos puedan acceder desde sus propias casas, o desde una estación remota"* Prof. W Fromm. En un contexto o marco general, el desarrollo de SCADAs y aplicaciones de supervisión de procesos captan el enfoque de aplicaciones de aprendizaje de mayor peso.

Para la realización de las prácticas virtuales tanto locales como remotas, se dispone de un aula dotada de estaciones de trabajo basadas en PC's, que se encuentra situada a escasos metros del laboratorio y permite el trabajo in situ bajo la supervisión y control del profesorado así como autónomo. En los casos de aprendizaje en modo local, si bien no es estrictamente necesaria la presencia física del profesorado o tutores, una vez explicado el método operativo de la sesión de prácticas, es preferible la tutorización completa del proceso para obtener el máximo rendimiento. Al alcanzar determinado nivel de conocimientos, los alumnos administran su tiempo en modo remoto obteniendo mayor rendimiento en las tareas de aprendizaje.

## **Localización**

En el Laboratorio de Automática de la facultad de Ingenieros de Konstanz hay un sistema industrial compuesto por dos tanques de líquido de 500l cada uno, un pequeño tanque de mezcla en el medio de 20l, y tuberías que unen los tanques entre sí. El flujo de líquido entre los tanques se produce a través de esas tuberías y es controlado por válvulas ON/OFF que van conectadas en periferia descentralizada a un *ET200* de Siemens, que cuenta con tarjetas de ampliación de entradas y salidas. Esta CPU está conectada a través de Ethernet con un PC desde donde se cargan los programas de control. A este controlador están conectados también todos los sensores del sistema que ofrecen información acerca del nivel, así como sendas bombas de presión, situadas debajode cada tanque, son analógicas de modo que se puede establecer la amplitud de su efecto.

En el Semestre de verano de 2008 un grupo de estudiantes finalizaron el conexionado físico de todos los sensores con el Hardware de control. Esto dejaba finalmente operativo el conjunto de tanques. Había sido instalada, también durante ese semestre, una pantalla

táctil desde donde se podía acceder a todo el abanico de tareas que ofrecía el control. El programa se carga en la CPU de Siemens, y era manejado a través de la pantalla táctil. También quedó en proceso de construcción un panel de control con botones y niveles para, esta vez de manera manual, controlar cada una de las válvulas y la potencia de los motores. Este sistema permite a los alumnos entrar en un primer contacto con sistemas de control distribuidos, comunicados con líneas *PROFIBUS* y con un control centralizado en una CPU, en este caso un PLC de Siemens.

Al ponerlo en práctica, el profesor W. Fromm, quien ha sido mi tutor durante todo el semestre, se percató de que la herramienta es limitada, ofreciendo un entorno de trabajo adecuado, pero para un número de estudiantes muy limitado (sólo un grupo por sesión), además necesitaban supervisión constante, por problemas que pudieran surgir con este equipamiento tan costoso. Por otro lado, ocurrieron durante los últimos meses del semestre de verano, cuando el sistema completo estaba ya marcha, varios accidentes provocados por una manipulación indebida. Estos problemas llevan al Profesor Fromm en una reunión con el departamento a reflexionar sobre el tema, llegando a la conclusión de que una herramienta virtual interactiva para el aprendizaje y preparación al posterior uso real, podría ser la solución a todos los problemas. De esta manera la limitación por uso dependería de los puestos en el laboratorio (20 grupos por sesión), que constan de ordenador de sobremesa y una CPU de Siemens 315b. Además se solucionarían los problemas derivados de un uso indebido ya que los materiales usados por los alumnos son de menor coste y con menor riesgo de colapso.

Al comenzar el semestre de invierno, entre los meses de Octubre 08 y Febrero 09 es desarrollado el presente proyecto, guiado por el Prof. D. W. Fromm, que pretende ser una herramienta para estudiantes, para que puedan aprender más y con mejores condiciones acerca de la comunicación Profibus, control de sistemas distribuido, utilización de PLCs para control industrial, como control principal de un modelo virtual, modelos virtuales y cómo desarrollar programas de control para los mismos.

## Concepto y objetivos del proyecto

La motivación de este proyecto parte de la necesidad de implementar una herramienta de trabajo, que simule el sistema de tanques de la figura, que utilice como sistema de comunicación líneas *PROFIBUS*, y que trabaje con el hardware de control de Siemens S7, además hay que programar un scada en CodeSys. De esta manera se pretende simular el sistema real de tanques sito en el laboratorio de automática de la universidad con ayuda de un PC y un controlador que será instalado en un *PLC S7 300*.



La utilización directa del sistema real por parte de manos inexpertas lleva, ha llevado en algunos casos a errores de manipulación, que dieron como resultado el deterioro de varias piezas que tuvieron que ser sustituidas. Por este motivo el Profesor W. Fromm, profesor de automática y robótica en la facultad, pensó en la creación de esta herramienta virtual, que simularía el sistema de tanques real, con una estructura de comunicación fiel, con la intención de crear un entorno interactivo que será usado por los estudiantes como herramienta de trabajo en posteriores cursos, para realización de prácticas. Estos alumnos no trabajarán con el sistema de tanques real para probar sus propuestas de control, sino que lo harán a través de la herramienta que será el resultado final de este proyecto. Así como también podrán investigar acerca de las características de la comunicación vía *PROFIBUS* usando un *PLC* como sistema de control, investigar acerca de la repercusión de retrasos en la señal control o de datos...

Para lograr este objetivo, hay que comenzar con el diseño del modelo virtual coherente con el sistema real. Debemos programar el mismo sistema de comunicación emulado y las escalas de medida en sensores y actuadores, de manera que los programas de control desarrollados con la herramienta, sean aplicables directamente en el sistema real.

### **Descripción de la solución propuesta**

El sistema será modelado con la ayuda del software *WinModv5.0* de "Media Machines". Paralelamente será diseñado un modelo en 3D que conectaremos con el modelo en WinMod, para esto haré uso de un software de diseño *CAD* que queda a mi elección. Combinando ambos modelos probaremos el sistema para afinar nuestro modelo en busca de un comportamiento coherente con el sistema real, para esto haré uso del software de control *CodeSys2.3*, con su ayuda será diseñado un *Scada* que servirá de controlador mientras alcanzamos el paralelismo entre los sistemas real y virtual. Esta parte es una parte obligatoria del proyecto, ya que así la herramienta será más versátil, pudiendo también hacer pruebas conectando dos PC's a través de PROFIBUS, en uno de ellos estará el modelo y en el otro el controlador. Así ofrecemos la posibilidad de realizar pruebas con el protocolo de comunicación Profibus entre dos ordenadores evaluando por ejemplo, los efectos de retrasos en las señales de control (práctica realizada en diciembre curso 2009/2010).

Una vez el modelo esté terminado, y esté terminado también el control con CodeSys, configuraré la conexión con el PLC de Siemens S7-300, para hacerlo solidario con el controlador real, utilizando las mismas variables que están programadas en la CPU que ejecuta el programa de control en el sistema real. De este modo, será sencillo y directa la aplicación de los controladores diseñados con el modelo virtual, en el modelo real.

La representación en 3D estará en comunicación con el modelo de WinMod, pudiéndose observar las variaciones del sistema.

Como resultado tendremos una herramienta lectiva, visualmente atractiva para los alumnos, quienes tendrán mayor interés en aprender acerca de estas disciplinas

Para finalizar este proyecto será necesario completar las siguientes etapas:

- Completar un modelo con WinMod v5.0 del sistema real.
- Programar y controlar el modelo en WinMod mediante CoDeSys
- Refinamiento del modelo y pruebas
- Programar un control con Simatic Manager o Siemens S7-300 como PLC.
- Investigar y probar aplicaciones en 3D
- Hacer un modelo CAD del sistema y convertirlo a tipo de archivo VRML97 para su posterior configuración para hacerlo dinámico.
- Probar el sistema completo y asegurarse de la coherencia con el sistema real.

*" Con el advenimiento de las nuevas tecnologías, el énfasis de la profesión docente está cambiando desde un enfoque centrado en el profesor y basado en clases magistrales, hacia una formación centrada principalmente en el alumno dentro de un entorno interactivo de aprendizaje. El diseño e implementación de programas de capacitación docente que utilicen las TICs efectivamente es un elemento clave para lograr reformas educativas profundas y de amplio alcance. "*  
(UNESCO-2004)

## **Descripción de la estructura de la memoria**

La memoria se compone de 5 capítulos principales. Globalmente primero se define el marco donde se desarrolla el proyecto, hardware y software que se va a utilizar y por qué, se estudia la configuración específica de estos elementos y pasamos a describir el trabajo realizado y el funcionamiento de la herramienta. A continuación se resume brevemente la composición de cada uno de estos capítulos por separado:

**Capítulo 1:** presenta la arquitectura del sistema empezando por los protocolos y el hardware que serán utilizados, que son PROFIBUS y la gama Siemens S-300 con la que he trabajado durante todo el semestre, también se presenta seguidamente el sistema de tanques real que será el objetivo a simular.

**Capítulo 2:** muestra las herramientas software que vamos a utilizar, así como el análisis de distintas posibilidades de software que quedaban a elección propia. Tras la comparativa se elige la mejor opción. Se muestra la lista de programas que se van a utilizar en el

diseño y desarrollo del proyecto. Seguidamente se detallan las características y la configuración determinada que necesitaremos para trabajar con ellos cumpliendo los requisitos del presente proyecto.

**Capítulo 3:** muestra el proceso de diseño y las características más importantes del modelo virtual, tanto del modelo en WinMod como de la representación en 3D que está preparada para la conexión con el modelo anterior. Se explica detalladamente cómo funcionan los distintos módulos del programa, así como la conexión entre el modelo y la representación en 3D, aspecto necesario para la correcta visualización del estado de las máquinas virtuales.

**Capítulo 4:** muestra los aspectos de control, tanto en Codesys como con Step7 de Siemens. Se describen los procesos de control en ambas plataformas. Los códigos se muestran en el apéndice C. Tras esto se exponen las conclusiones y un diagrama de Gantt del trabajo realizado.

**Capítulo 5:** en este capítulo se explica el funcionamiento de la herramienta y se describen las conclusiones una vez terminado el trabajo.

## **Antecedentes**

La automatización de procesos industriales es ya una realidad presente en casi todos los grandes centros de fabricación y tratamiento de materiales. Los paneles de control son programados cada vez de manera más intuitiva para facilitar su uso. La virtualización de estos procesos es el siguiente paso permitiendo una correcta manipulación en un entorno que se asemeja al real. Este avance permite a los manipuladores realizar pruebas en sistemas virtuales y analizar las consecuencias para así prever posibles errores en la realidad. Estas nuevas herramientas son especialmente útiles para el aprendizaje de estas técnicas de control y manipulación mediante PLCs industriales, minimizando el uso de recursos por puesto, ofreciendo la posibilidad a un grupo de aprendices de interactuar con un mismo sistema al mismo momento y sin riesgo de poner en peligro el sistema real.

# Capítulo 1

## ARQUITECTURA DEL SISTEMA

### 1.1 Hardware y Protocolos

#### 1.1.1 Protocolo Profibus

Profibus es un bus de campo abierto independiente del fabricante. Su área de aplicación abarca manufactura, procesos y automatización de edificios. La independencia del fabricante y el ser un sistema abierto está garantizado por el estándar Profibus EN 50 170. Con Profibus los dispositivos de diferentes fabricantes pueden comunicarse sin necesidad de adaptaciones mediante interfaces especiales. Profibus puede ser empleado tanto para transmisiones de datos de alta velocidad y tiempos críticos, como para tareas intensivas de comunicación compleja.

La familia Profibus consiste en tres versiones compatibles:

**PROFIBUS-DP:** Optimizado para alta velocidad y enganche económico. Esta versión de Profibus está diseñada especialmente para comunicación entre sistemas automáticos de control y E/S distribuidos a nivel de campo. Puede ser empleado para reemplazar transmisiones paralelas de señales con 24V o 0 a 20mA.

**PROFIBUS-PA:** Está especialmente diseñado para automatización de procesos. Permite que sensores y actuadores puedan ser conectados a un bus común en áreas intrínsecas de seguridad. Permite comunicación de datos y transporte de energía sobre el mismo bus empleando tecnología de dos cables, acorde con el estándar internacional IEC 1158-2. Este Profibus reemplaza los 4 –20 mA.

**PROFIBUS-FMS:** (Fieldbus Message Specification) está diseñado para tareas de comunicación a nivel superior de dispositivos de campo. Es decir, tareas como el intercambio de información entre controladores y dispositivos de campo inteligentes,

donde la funcionalidad es más importante que el tiempo de reacción del sistema. Habitualmente el intercambio de información es acíclico, a petición del proceso.

Sub familia	Principal aplicación	Principal ventaja	Características más relevantes
PROFIBUS-FMS	Automatización para propósitos generales	Universal	<ul style="list-style-type: none"> <li>• Gran variedad de aplicaciones</li> <li>• Comunicaciones multi-maestro</li> </ul>
PROFIBUS-DP	Automatización de factorías	Rápido	<ul style="list-style-type: none"> <li>• Plug and Play</li> <li>• Eficiente y efectivo en costo</li> </ul>
PROFIBUS-PA	Automatización de procesos	Orientado a aplicación	<ul style="list-style-type: none"> <li>• Suministro de energía a través del propio bus</li> <li>• Seguridad intrínseca</li> </ul>

Tabla 1.1: Tabla líneas Profibus

La versión más utilizada de éstas es la Profibus DP (Periferia Distribuida; Decentralized Peripherals), es utilizada en los sistemas de automatización, fue desarrollada en 1993. Profibus tiene, conforme al estándar, cinco diferentes tecnologías de transmisión, que son identificadas.

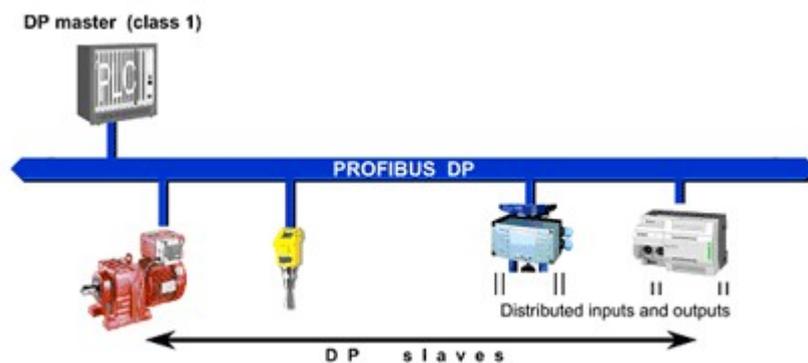


Fig 1.1: Esquema Bus Profibus DP

El RS-485 utiliza un par de cobre trenzado apantallado, y permite velocidades entre 9.6 kbps y 12 Mbps. Hasta 32 estaciones, o más si se utilizan repetidores. MBP (Manchester Coding y Bus Powered) es transmisión sincrónica con una velocidad fija de 31.25 Kbps [3].

Las versiones IS son intrínsecamente seguras, utilizadas en zonas clasificadas.

Fibra óptica incluye versiones de fibra de vidrio multimodo y monomodo, fibra plástica y fibra HCS. Profibus DP está actualmente disponible en tres versiones:

- DP-V0. Provee las funcionalidades básicas incluyendo transferencia cíclica de datos, diagnóstico de estaciones, módulos y canales, y soporte de interrupciones
- DP-V1. Agrega comunicación acíclica de datos, orientada a transferencia de parámetros, operación y visualización
- DP-V2. Permite comunicaciones entre esclavos. Está orientada a tecnología de drivers, permitiendo alta velocidad para sincronización entre ejes en aplicaciones complejas

Desde el punto de vista del control de las comunicaciones, el protocolo Profibus es maestro esclavo, pero permite:

- Aplicaciones mono maestro. Un sólo maestro está activo en el bus, usualmente un PLC. Los demás dispositivos son esclavos. Este esquema es el que permite los ciclos de lectura más cortos
- Aplicaciones multi maestro. Permite más de un maestro. Pueden ser aplicaciones de sistemas independientes, en que cada maestro tenga sus propios esclavos. U otro tipo de configuraciones con dispositivos de diagnóstico y otros

En un ambiente multimaestro, pueden haber dos tipos de maestros:

- DPM1. DP Master Class 1. Es un controlador central que intercambia información con sus esclavos en forma cíclica. Típicamente un PLC.
- DPM2. DP Master Class 2. Son estaciones de operación, configuración o ingeniería. Tienen acceso activo al bus, pero su conexión no es necesariamente permanente

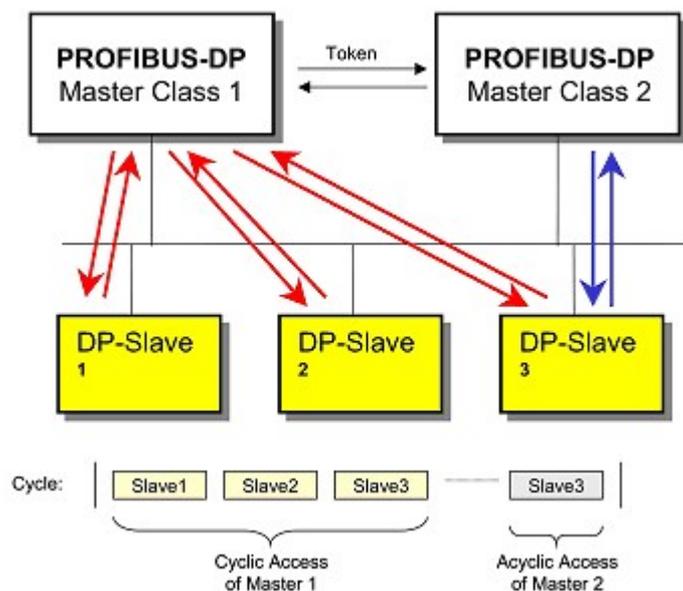


Fig 1.2: Esquema comunicación Master Class 2

En la situación más simple, un solo maestro DPM1 y usa DP-V0. Este proceso pasa por tres fases: Parametrización, configuración y transferencia de datos. Durante los dos primeros se define la forma en que se hará la comunicación, y esto es verificado por errores y consistencia, para luego pasar a la fase de transmisión. Adicional a los comandos a una estación, el DPM1 puede enviar comandos a un grupo o a todos los esclavos.

Estos se conocen como comandos multicast, y son utilizados para funciones de sincronización. DPV1 por otro lado, agrega las transferencias acíclicas. Éstas son realizadas en paralelo a la comunicación cíclica, pero en un esquema de menor prioridad. DPV2 permite a su vez comunicaciones esclavo-esclavo.

Uno de los esclavos se define como emisor ("publisher") y uno o más esclavos como suscriptores. El emisor envía un mensaje, que es recibido directamente por todos los suscriptores, sin necesidad de pasar por el maestro.

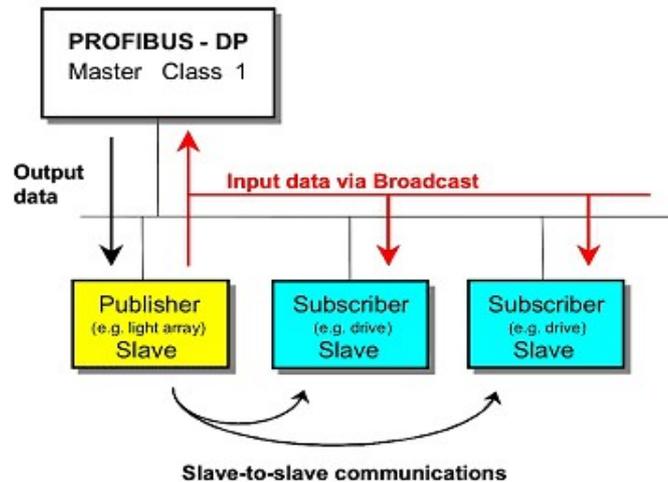


Fig 1.3: Esquema comunicación a través de Publisher

DPV2 tiene otras funciones, como comunicación isocrónica, control de reloj, y carga y descarga de datos.

### 1.1.2 Modelo ET – Siemens

La gama de productos SIMATIC ET 200 es parte de TIA y utilizada para conexiones descentralizadas. La gama ofrece todo, unidades compactas o modulares, con meras E/S digitales, con componentes tecnológicos o sistemas completos que incluyen accionamientos así como elementos neumáticos. Las unidades pueden instalarse tanto dentro de un armario eléctrico como fuera del mismo ya que resisten el ambiente industrial [7].

- Simatic ET 200 M IM 153-2

153-2AA01-0XB0

(dirección de bus apunta a "2". Podemos elegir entre en direccionamiento de 0 - 128)

- SM 321 DI 16xDC 24 V / 0,5 A

321-1BH01-0AA0

Tarjeta digital de entradas con 16 entradas simples

- SM 322 DI 16xDC 24 V / 0,5 A

322-1BH01-0AA0 Tarjeta digital de salidas con 16 salidas simples



Fig 1.4: CPU S7 – Siemens

La siguiente tabla representa las distintas CPU's de Siemens, los distintos modelos con las propiedades que tiene cada uno. En nuestro caso el modelo de CPU de que disponemos es la CPU 315-2 DP que integra interfaces de conexión DP y DP/MPI.

Design	CPU	Start of delivery	Integrated Interfaces	Integrated I/O	Integrated technological functions
Standard CPUs	CPU 312		MPI		
	CPU 314		MPI		
	CPU 315-2 DP		DP, MPI		
	CPU 317-2 DP (New)	3rd quarter 03	DP, DP/MPI		
	CPU 318-2 DP		DP, DP/MPI		
Fail-safe CPUs	CPU 315F-2 DP		DP, MPI		Fail-safety
	CPU 317F-2 DP (New)	3rd quarter 03	DP, DP/MPI		Fail-safety
Compact CPUs	CPU 312C		MPI	Digital	Counting
	CPU 313C		MPI	Digital, analog	Counting
	CPU 313C-2 PtP		PtP, MPI	Digital	Counting
	CPU 313C-2 DP		DP, MPI	Digital	Counting
	CPU 314C-2 PtP		PtP, MPI	Digital, analog	<ul style="list-style-type: none"> <li>● Counting</li> <li>● Positioning</li> </ul>
	CPU 314C-2 DP		DP, MPI	Digital, analog	<ul style="list-style-type: none"> <li>● Counting</li> <li>● Positioning</li> </ul>
Technology CPU	CPU 317TC-2 DP (New)	4rd quarter 03	DP, DP/MPI	Digital	<ul style="list-style-type: none"> <li>● Synchronism</li> <li>● Travel to fixed stop</li> <li>● Print mark control</li> <li>● Cam switching</li> </ul>

Tabla 1.2: Tabla de especificaciones de CPU's de Siemens

La CPU 315-2DP es un autómata que tiene integrada una interfase PROFIBUS DP. Para dicha CPU, se dispone de los siguientes protocolos PROFIBUS:

- Interfase DP como Maestro, según EN 50170.
- Interfase DP como Esclavo, según EN 50170.

PROFIBUS-DP (Periferia Distribuida) es el protocolo de conexión de elementos de campo o periféricos a una CPU con un rápido tiempo de respuesta.

Otra característica de este protocolo es que la CPU es capaz de direccionar estos módulos de periferia como si fueran módulos propios de entrada/salida.

Esta CPU es por tanto suficiente como elemento de formación, disponiendo de las siguientes características:

- 16K de instrucciones. 48K de memoria de trabajo 80K de espacio disponible.
- 1024 Bytes de Entradas/Salidas Digitales
- 128 Bytes de Entradas/Salidas Analógicas
- 0,3 ms / 1K instrucciones
- 64 contadores
- 128 temporizadores
- 2048 bits de Marca

### 1.1.3 Profibus A71 DP-PC-SL (PCI)

El grupo DP-PC-SL simula un esclavo Profibus DP con un rango escalable de datos con un máximo de 242 Bytes Entrada /242 Bytes de Salida, juntos son 368 Bytes I/O (2944 bits). La frecuencia en la comunicación Profibus puede ser configurada a distintos valores: 9.6K, 19.2K, 93.75, 187,5K, 500K, 750K, 1.5M.

Se necesita un PC con un puerto AT-Bus (ISA) libre, ahí conectaremos la tarjeta Profibus.

La configuración A71 fija el acoplamiento entre el sistema de automatización externo (AS, PLC, IPC, PLS) y los drivers específicos del sistema de software de WinMod, donde residirá el modelo diseñado.

Un esquema de esta comunicación podemos verlo en la siguiente figura.

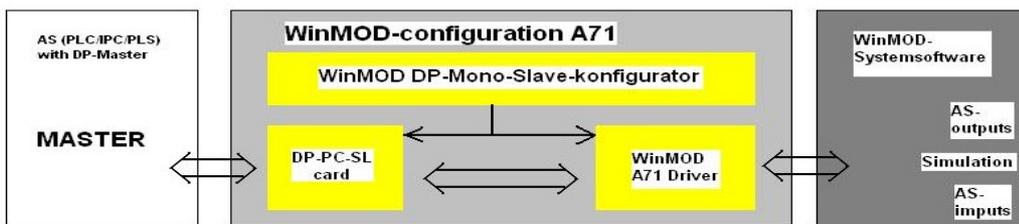


Fig 1.5: interno a WinMod

Para conectar por medio de Profibus un PC, necesitamos una tarjeta PCI de Profibus para ordenador, esta tarjeta tiene sigue un esquema como el de la figura 1.4

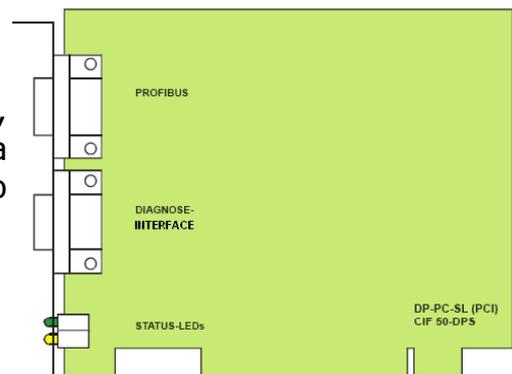


Fig 1.6: Tarjeta PC Profibus

## 1.2 Sistema real de Tanques

El sistema de tanques sito en el laboratorio de automática de la universidad de Konstanz es el objetivo que hemos de virtualizar. Un esquema de los tanques puede verse en la *figura 1.1*. En este esquema aparecen todos los elementos que habrá que simular. Las válvulas son Hvi y Mvi, y son los actuadores del sistema, junto con las bombas de presión que no aparecen en este esquema. Todos los sensores aparecen en Amarillo, y están conectados por líneas profibus al armario central, donde están todas las Cpu's y módulos de entrada/salida. En el sistema real las válvulas, se accionan por encendido/apagado, interruptor que está controlado por el control central instalado en una CPU en el armario.

Como se puede observar hay distintos sensores cuya finalidad es la misma, todos los que están situados en la parte superior de los tanques y en la parte baja son sensores que miden la cantidad de líquido de distintas maneras, por presión, midiendo el desfase de ondas...

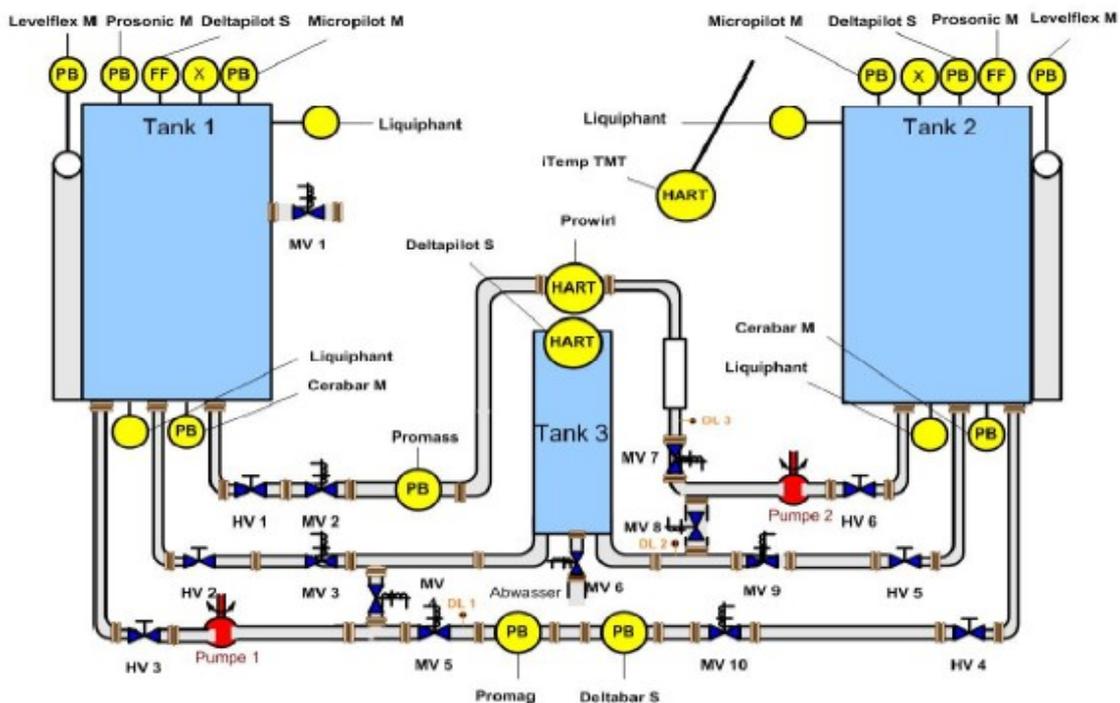


Abbildung 1: eine detaillierte Abbildung der Tankanlage<sup>1</sup>

Fig 1.7 Sistema real

En la sección *Concepto y objetivos del trabajo* pueden ver una imagen del sistema de tanques completo. Así, la misión que tiene este proyecto es la de llevar el sistema real, a una virtualización que permita el manejo y el aprendizaje por parte de un número más elevado de alumnos, con las mismas herramientas que lo hace el trabajo directo con los tanques. De este modo, crearé un modelo de los tanques con WinMOD, que utilizando el protocolo de comunicación PROFIBUS, sea controlado a través de una CPU Siemens de la familia S300. Además del modelo en WinMod es imprescindible el desarrollo de una representación virtual de los tanques fiel a los reales, para hacer la herramienta más atractiva, de modo que los alumnos se interesarán más por el aprendizaje.

Para diseñar el modelo, primero realizo una serie de mediciones que me indican cómo se comporta el sistema ante las distintas entradas.

## Capítulo 2:

### HERRAMIENTAS SOFTWARE

#### 2.1 Componentes Software

La inclusión en los proyectos de software no libre encarece el proyecto llegando a ser, en muchos casos una parte económica de peso dentro del mismo. Es por esta razón que por la que, en la medida de lo posible, se busca utilizar herramientas Freeware, de manera que durante la puesta en marcha en la práctica del proyecto no resulte necesaria la adquisición de las licencias de los mismos.

Aquí adelanto una lista del software que será utilizado durante el desarrollo del proyecto, así como la finalidad para que tendrá dentro del marco del proyecto presente. Posteriormente describiré con mayor detalle la labor que será realizada con cada uno de ellos, así como la configuración necesaria de cada uno de ellos para la correcta elaboración del proyecto.

- **WinMOD v5.0**

Centro de simulación en tiempo real para procesos de automatización. Será usado para programar el modelo del sistema. Es un Software de "Media Machines". Se podía haber optado por hacer el modelo con ayuda de Matlab, en este caso este software venía propuesto por ser un software de código libre no siendo necesaria ningún tipo de licencia siendo utilizado en nuestro proyecto.

- **WinMOD-3D View**

Aplicación de WinMOD para la interactividad del modelo realizado con su representación en 3D diseñada con ayuda de un software CAD. Lo usaremos para conectar el modelo de WinMod al archivo VRML correspondiente al diseño en 3D.

- **CAD software**

Software a elegir. Para diseñar el modelo en 3D del sistema de tanques. A la hora de elegir el software valoraré altamente el hecho de que el software sea de distribución libre.

- **CoDeSyS V2.3**

Entorno de desarrollo para la programación de controladores conforme con el

estándar industrial internacional IEC 61131-3. El término CoDeSys es un acrónimo y significa Sistema de Desarrollo de Controladores.

Dentro de este software se encuentran aplicaciones que usaremos también como son: CoDeSyS SP RTE: como Software SPS (realtime), CoDeSyS OPCConfig: utilizado para adaptar el servidor OPC.

CodeSys es también software de libre acceso, siendo ésta una característica importante en el marco de nuestro proyecto.

USO: para programar y manejar el control con software SPS

· **Simatic Manager:**

Simatic Manager es un entorno profesional flexible que permite programar un gran número de autómatas de la familia SIEMENS. Para ello está dotado de una colección de librerías que almacenan las características específicas HW de cada autómata y los elementos de conexionado para configuraciones en red.

Además de los lenguajes de representación ya conocidos (KOP, AWL y FUP), STEP 7 posee un potente lenguaje gráfico denominado S7-GRAPH que permite programar directamente usando Grafcats. Es habitual que el programador realice la codificación de la parte 'estándar' del automatismo empleando S7-GRAPH para después efectuar las modificaciones de detalle en ensamblador (AWL).

USO: para programar y manejar el control en Hardware SPS (siemens S7 CPU-315-2-DP)

## 2.2 Comparativa de CAD software.

Queda a elección libre, el programa de diseño en 3D que será usado para representar el sistema y poder visualizar su estado. Para decidir cuál es el más apropiado aquí presento una tabla de características de distintos programas de diseño. Al no tener conocimientos previos en esta materia, no tengo ningún software preferido, de modo que buscaré el programa que más encaje con las características que necesito.

Al menos necesito un software con el que pueda diseñar el sistema que tenga la opción de convertir a formato VRML y de importar desde VRML97:

CAD Tools	a VRML97	Desde VRML97	Dificultad	Freeware	Estructura de nodo tipo árbol
Wings 3D	SI	NO	OK	SI	NO
Flux Studio 2.1	SI	SI	MUY FÁCIL	SI	SI
Solid Works	SI	SI	SENCILLO	NO	NO
OpenFX-1.7	SI	NO	OK	SI	NO
White Dune 1.4	SI	SI	OK	SI	SI

Tabla 2.1: comparativa CAD

Como podemos observar los programas que más se ajustan a mis necesidades son Flux

Studio 2.1, Solid Works y White Dune 1.4. En el caso de White Dune 1.4, la versión más actual está ya obsoleta y no ofrece un entorno amigable y fácil de comprender. Nos quedamos con Solid Works y Flux Studio, ambos programas de diseño me ofrecen todo lo que necesito, ambos son programas fáciles de utilizar, la estructura de nodo tipo árbol no es necesaria pero si muy aconsejable, esta característica y el detalle de que Flux Studio es Free-software harán que me decante por éste. Al ser un software libre su uso en un proyecto o en el desarrollo de una aplicación también lo es.

## 2.3 Flux Studio

Con ayuda de este programa desarrollaré el modelo en 3D y tras esto lo convertiré a VRML97 file, para poder trabajar con él con WinMod 3D. La versión utilizada es la última que sacó *Media Machines*, Flux Studio 2.1 que salió en Mayo del 2007.

Propiedades de Flux Studio:

- Ventana Standard (Windows OS).
- Sombra realista.
- Fácil de animar.
- Función undo/redo inteligente.
- Función de "Operaciones Boleanas".
- La estructura de archivos de los CAD en Flux Studio es similar a los archivos VRML97.

## 2.4 WinMod 5.0

La simulación de los procesos internos del sistema será realizada usando el software de desarrollo WinMOD 5.0 by Mewes & Partner.

La visualización en 3D será llevada a cabo también por este programa al incluir una aplicación, WinMod 3D view, que permite conectar una simulación de un sistema con una representación 3D del mismo, para esto es necesario que el archivo del modelo en 3D tenga formato VRML97.

**2.4.1 Configuración de WinMod.-** El modelo del proceso del que será responsable WinMOD, no será ejecutado en el mismo ordenador que el control automático. En un primer momento el programa de control se ejecutará en otro ordenador, éstos distintos sistemas deben ir conectados a través de una línea Profibus DP.



Para conectar un control externo para el modelo diseñado en WinMod, éste último debe configurarse en modo mono-esclavo. Por esta razón la herramienta "Profibus Monoslave Configurator A71" debe ser iniciada en WinMod al inicio. Aquí, es configurada la

Fig 2.1: Comunicación Profibus entre PC's

base de datos que se va a transmitir, también será configurada exactamente de igual manera en el SPS que lo controle. La periferia descentralizada es construida de esta manera como en un sistema real.

En las propiedades de la configuración PROFIBUS elegimos la velocidad 1500kb/s, se elige esta velocidad para ajustarnos a la misma con la que se transmiten los datos en el sistema real. De este modo podremos trabajar con los tanques virtuales en busca de fallos causados por retrasos, de la misma manera que lo haríamos con el sistema real.

La configuración queda de esta manera:



Fig 2.2: Configuración comunicación WinMod

La configuración del grupo DP-PC-SI que será nuestro puerto virtual para la comunicación Profibus queda de la siguiente manera, 10 Bytes de señales de salida agrupados en 5 palabras, 3 de ellas serán la información del nivel de los tanques que irán representadas con 16 bits cada una, además irán en el esquema de configuración 6 Bytes de entrada agrupados en 3 palabras, que serán los responsables de recibir datos venidos desde el sistema de control, acciones para los actuadores, y también la potencia de las bombas, señales que serán enviadas de manera analógica representadas con 8 bits cada una, de esta manera obtendremos una mayor precisión en el control del sistema.

#### 2.4.2 Parámetros para la comunicación con WinMod 3D-View

La comunicación entre WinMOD y WinMOD 3D-View será establecida y monitorizada a través de la *unidad de comunicación*, un elemento de WinMOD al cual irán conectados todas las entradas que queramos darle al diseño en 3D. Para hacer esto posible hay que configurar previamente la unidad de comunicación. La configuración por defecto es la que enlaza un VRML97 con WinMOD, sólo hay que establecer las variables que vayamos a utilizar.

La siguiente imagen muestra la ventana de la herramienta de comunicación, sensores, actuadores, las bombas, los niveles de los tanques, todo va conectado a la ventana de comunicación, de esta manera enviará WinMOD a WinMOD 3D view las señales que gobernarán el diseño en 3D.

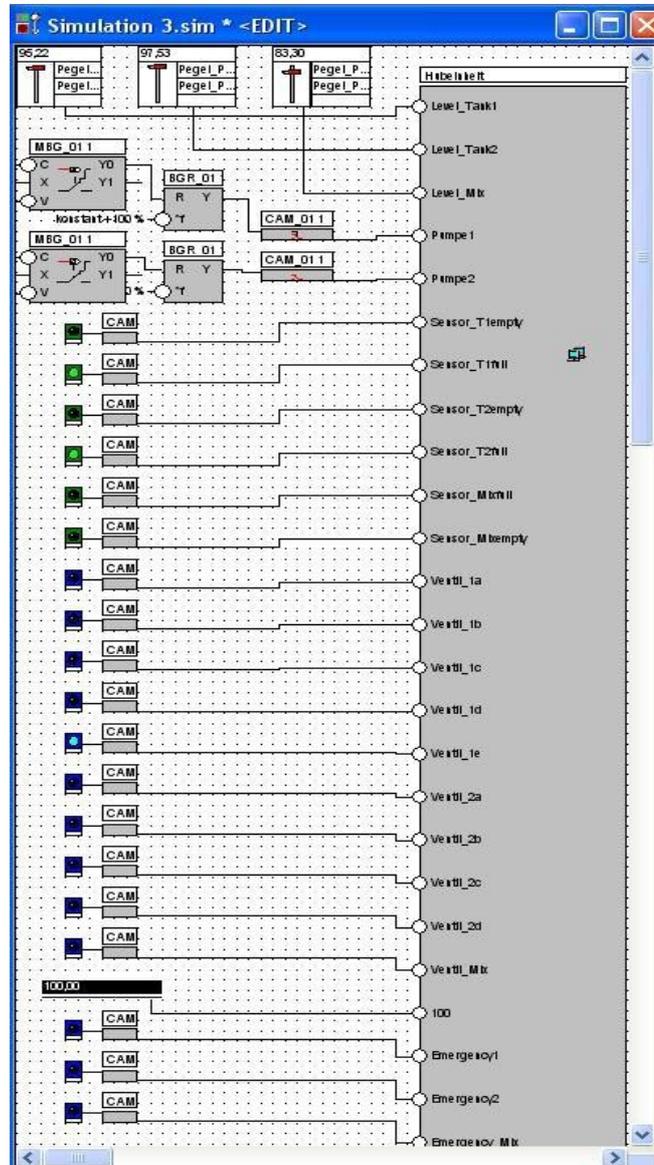
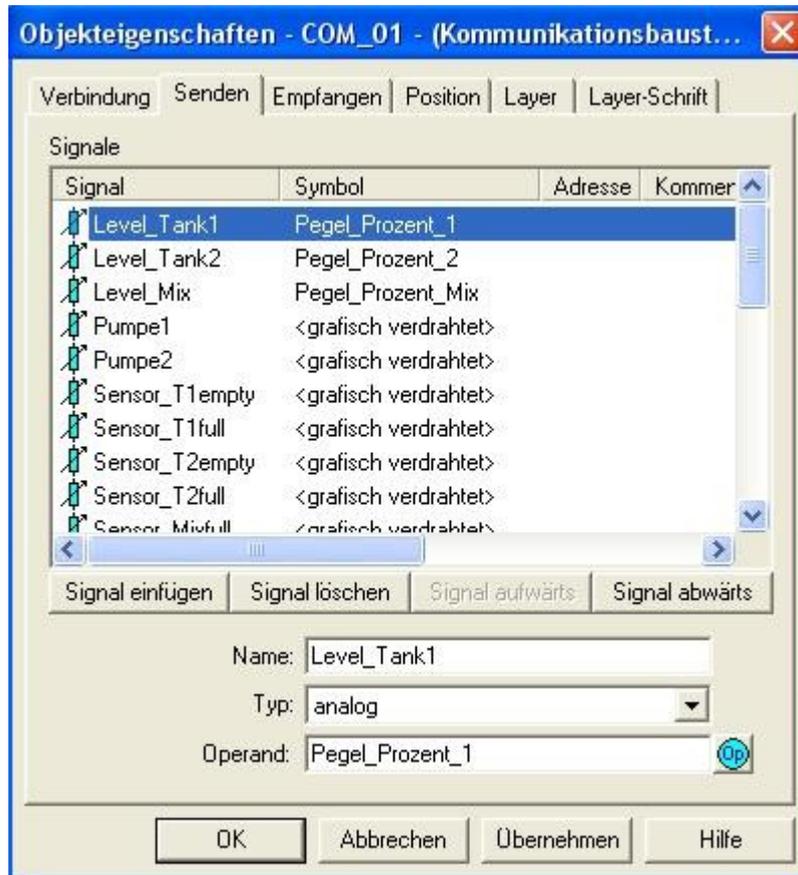


Fig 2.3: Unidad de Comunicación

WinMOD 3Dview identifica cada variable con un tipo y un nombre, éste no debe ser ambiguo porque tendremos que trabajar con él. Este nombre debe decirnos algo acerca de la variable a la que representa

En el diálogo de propiedades, pulsando sobre el botón SEND, podemos definir una señal de entrada, con el botón, insertar señal, en la lista de señales será creada una nueva entrada.

En la *figura 2.4* podrán ver el cuadro de diálogo donde editamos la lista de señales en la configuración del WinMOD 3D-View:



*Fig 2.4: Configuración de WinMod 3D comm*

Importante también es definir el tipo de variable, que debe ser coherente con el formato que esté programado en el modelo, así una señal analógica no podrá representada por una variable definida de otro tipo.

La conexión entre aplicaciones será via TCP/IP, para que esta comunicación se establezca debemos introducir las propiedades en el cuadro de diálogo de configuración del proyecto. En la siguiente figura se muestra cómo deben de configurarse las distintas opciones.

Para abrir el cuadro: **Configuration / Project Properties**

Ajustamos el tiempo a 10ms que corresponde aproximadamente con el retraso que sufre el sistema real en la comunicación entre sus sensores y el programa de control, de esta manera el modelo se ajustará más a la realidad.

En la figura 2.5 vemos como queda establecida la configuración de los parámetros de la

comunicación entre WinMOD y WinMOD 3D-View:

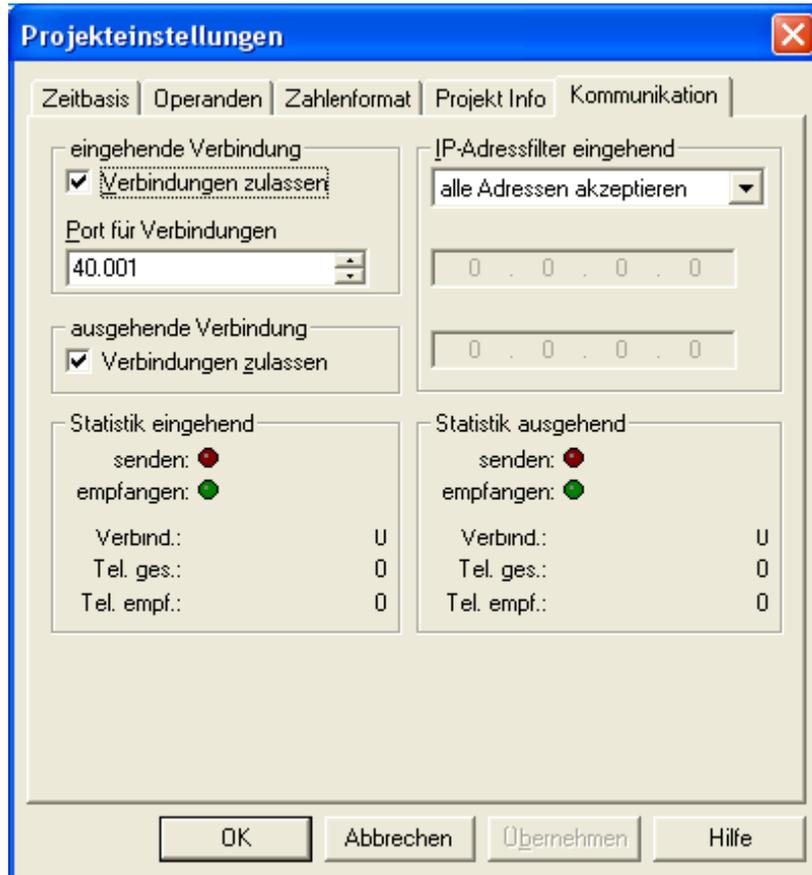


Fig 2.5: Configuración de las propiedades en WinMod3D

Si el CAD está finalizado y guardado en formato VRML97 podemos ya enlazarlo con el modelo en WinMOD.

## 2.5 WinMOD 3D-View

WinMOD-3D View (Versión 1.00) es una aplicación añadida (un add-on), software de Mewes&Partner. Con este programa podemos conectar y ver representado fácilmente un diseño en 3D (dinamizado) con un el modelo virtual de un proceso de automatización. El diseño ha de ser realizado con un programa de diseño CAD y el modelo debe ser programado con ayuda de WinMOD.

### 2.5.1 Propiedades de WinMOD 3D-View

Para establecer la comunicación entre un modelo y un diseño, es necesario primero conectar las aplicaciones (al margen de los archivos que queramos enlazar). Esto es, porque, WinMOD3D-View ofrece la posibilidad de enlazar un CAD con un modelo que esté ejecutándose en otro ordenador. Tenemos, entonces que saber, con qué ordenador y qué propiedades de comunicación se establecerán. Para hacer la conexión con el ordenador de trabajo, debemos escribir "localhost" en el campo "Computer", así WinMOD 3D-View

entenderá que se tiene que comunicar con el mismo ordenador y buscará el modelo en WinMOD ejecutándose dentro de los procesos de nuestro ordenador.

Tras entrar en WinMOD 3D-View abrimos el diálogo de propiedades de comunicación: **File->Load VRML-Files**. Los valores de Puerto y Time-Out no tenemos que cambiarlos. La siguiente imagen muestra cómo debe estar la configuración:

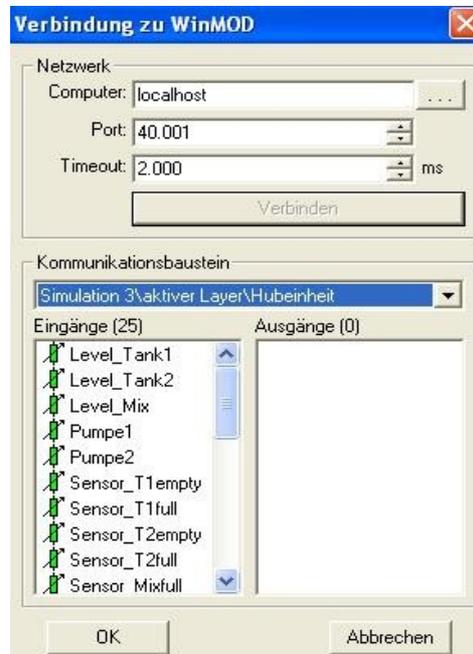


Fig 2.6: Comunicación con WinMod

Si todo se ha ajustado debidamente, tras pulsar sobre el botón **Connection** se debe abrir el módulo de comunicación de visualización 3D. Aparecerá una lista de Entradas y Salidas que serán todas las variables que le queramos pasar.

La aplicación WinMOD3D-View pide un archivo de texto que tiene que ser editado, donde estará escrita una lista con un formato específico, de todos los nodos que queremos animar y, acompañándolos el tipo de animación que se va a realizar, rango de valores que va a tomar cada variable, en definitiva, el comportamiento de cada objeto. No será igual la animación que le daremos al nivel del tanque, que consistirá en la representación del líquido subiendo o bajando de nivel dentro del tanque, que la de un sensor o una válvula que cambiaremos de color para indicar que están abiertos o cerrados.

Lo siguiente será importar la lista antes mencionada de un archivo de texto. En la pantalla principal de WinMOD 3D-View accedemos al menú: **Configuration / Signal scheduling Import/Export (Text) / Import files**. Elegimos el archivo de texto que previamente a debido de ser confeccionado, e importamos la información. La siguiente imagen muestra la carga con éxito de todos los nodos de la lista.

Veremos más en detalle a cerca de este archivo de texto en el siguiente capítulo.



Fig 2.7: Importación de Señal

Tras finalizar la importación, será inicializado el Import-Wizar, desde donde podremos nuevamente cambiar algunos parámetros en nuestra comunicación. En nuestro caso, aquí no necesitamos cambiar nada. Después de terminar este proceso, el enlace entre WinMOD y WinMOD3D-View está completado. En el siguiente capítulo veremos en profundidad algunos detalles de este enlace.

## 2.6 CoDeSys V2.3

Será desarrollado un Scada en CoDeSys, con la intención de probar nuestro modelo en busca de hacerlo robusto y coherente al sistema real. También hará de la herramienta virtual un recurso lectivo más completo, ofreciendo además de la simulación del sistema real, una alternativa de control en CoDeSys, un software muy utilizado también en procesos industriales. Al ser un software libre y de calidad profesional, su difusión ha sido muy rápida convirtiéndose en poco tiempo una de las más solicitadas herramientas de control Software en el marco industrial.

La versión utilizada en este proyecto es CoDeSys V 2.3.7.2. Trabaja bajo el reglamento IEC 61131-3 para entender el lenguaje de programación. Tiene numerosas aplicaciones de estadísticas y control de datos. Ofrece también un entorno amigable para la programación de Scada's convirtiendo el trabajo en un proceso divertido.

### 2.6.1 Propiedades de CoDeSys V2.3.- **Propiedades generales**

Al abrir un nuevo proyecto, la ventana "*Target system properties*" se abre al inicio. La configuración inicialmente debe ser ajustada a: **3S CoDeSys SP RTE**. Que es el driver de CoDeSys para procesos que simulan en tiempo real. Así también podrán ser estudiados con ayuda de CoDeSys los efectos de retrasos, en un sistema de comunicación a través Profibus simulado.

La imagen 2.8 muestra la configuración de este driver.

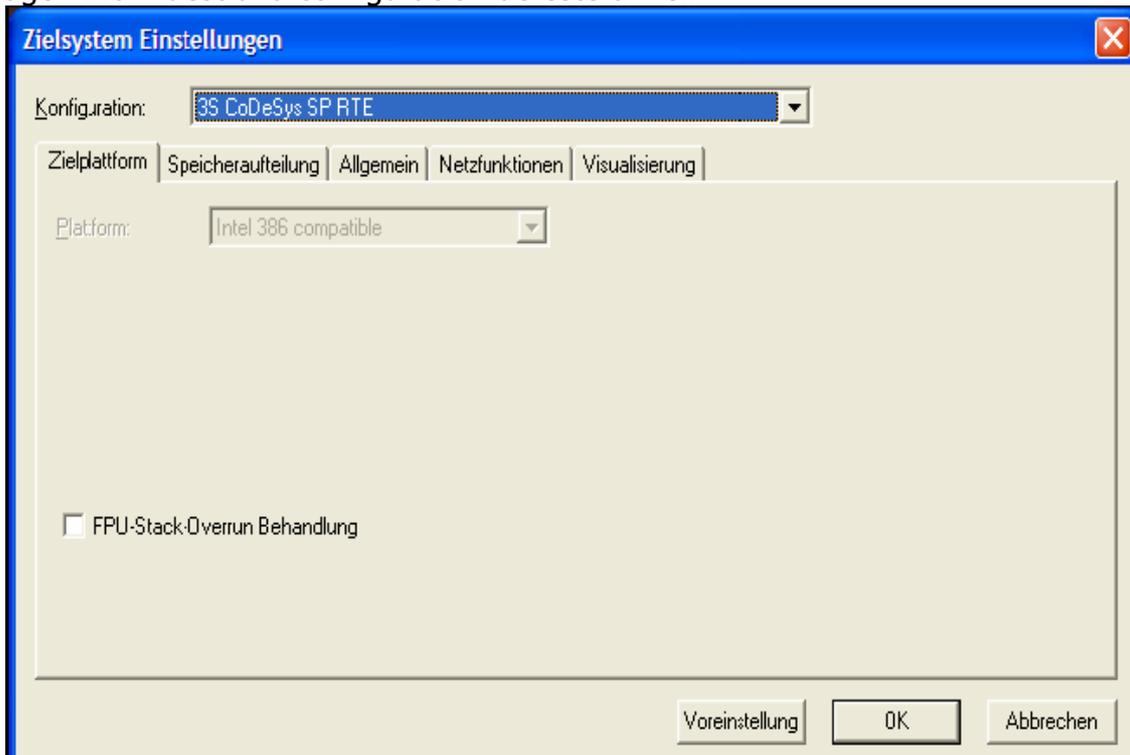


Fig 2.8: CoDeSys V2.3: Sistema objeto, Propiedades generales

En el menú de opciones podemos configurar distintos parámetros de la comunicación. El resto de propiedades, las mantendremos por defecto. Debemos activar **Symbol-items-generate** y **Symbol-table-generate**.

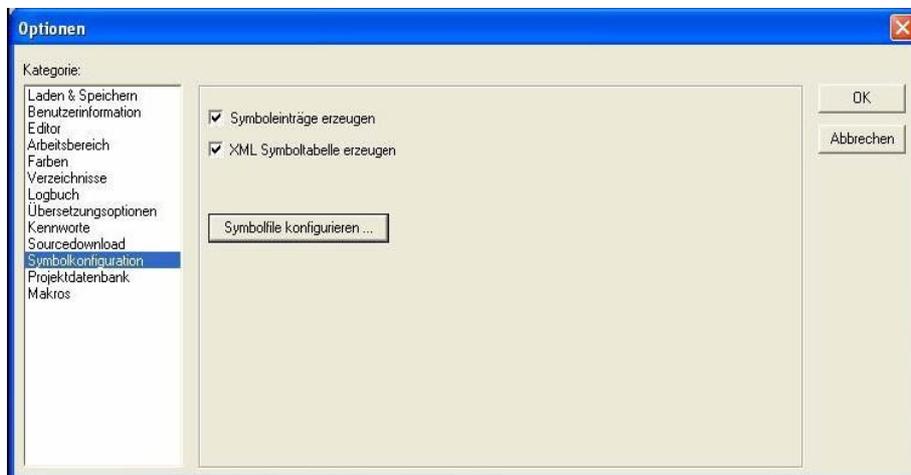


Fig 2.9: Opciones generales en CoDeSys

A través del botón **Symbol-file configuration** abrimos el diálogo *insert Objects-attribute*, aquí debemos marcar todas las pestañas. Esta ventana de opciones está representada en la siguiente imagen

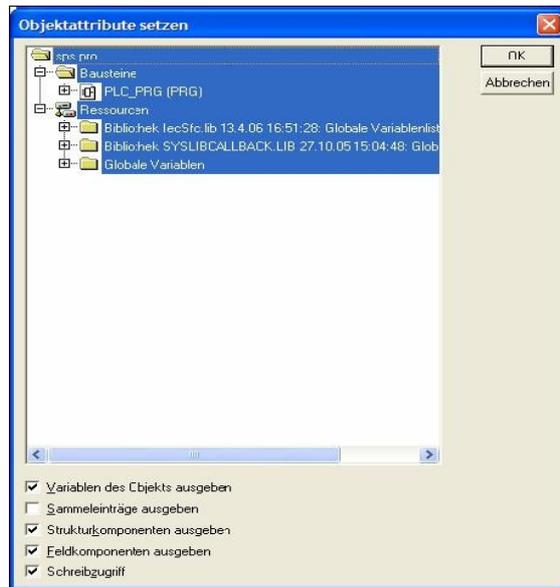


Fig 2.10: Atributos de Objeto

## 2.6.2 Propiedades para conectar CoDeSys con WinMod

Para tener conexión vía Profibus entre PC's ambos sistemas deben tener instalada la tarjeta. La configuración de la comunicación del sistema a controlar (diseñado en WinMOD), ya está realizada, lo siguiente será configurar la conexión Profibus con CoDeSys. Esta configuración en el controlador debe ser totalmente coherente con la del modelo, de lo contrario la comunicación no se establecerá correctamente.

Abrimos el cuadro "WinMod Profibus DP Configuration". Aquí serán representados los bits que serán transmitidos, de entrada y de salida. En la siguiente imagen se muestra cómo resulta:

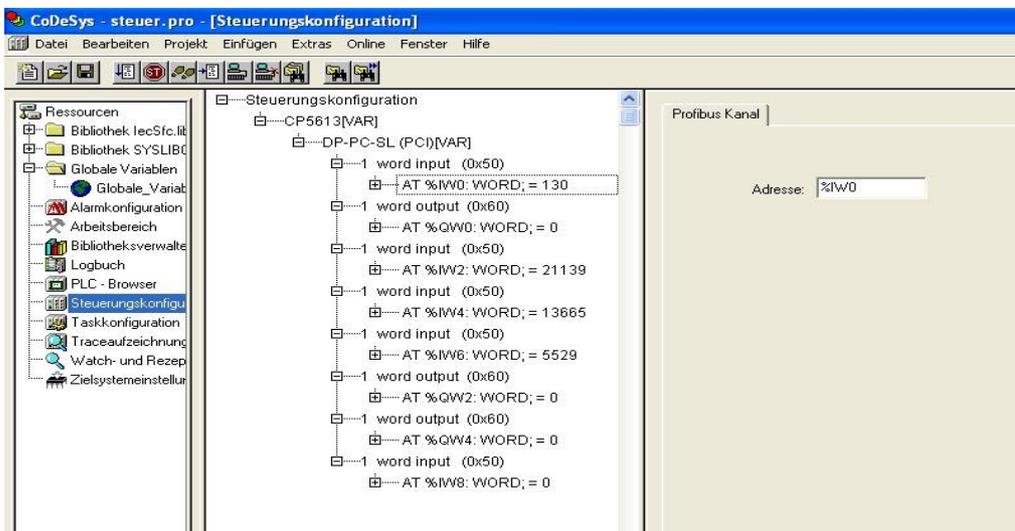


Fig 2.11: Configuración Profibus en CoDeSys

Elegimos la velocidad CP.5613 a 1500 Kbps, es la velocidad común a todos los sistemas en este laboratorio, así optimizaremos la comunicación.

## 2.7 Simatic Manager

Para controlar nuestro modelo en WinMOD mediante Hard-SPS , usamos el Simatic Manager, STEP7, este es un potente lenguaje para especificar comportamientos secuenciales. Un lenguaje parecido utilizado en la escuela de Sevilla es el GrafCet.

### 2.7.1 Propiedades Simatic Manager

Lo primero aquí será configurar en el Step 7 el esquema Hardware que vamos a utilizar. Este entorno de programación es muy flexible permitiendo conexión con todo tipo de sistemas y un sin fin de configuraciones a elegir. Elegimos entonces la Cpu con la que vamos a trabajar, que en nuestro caso es la CPU 315-2DP. Tenemos que configurar también la conexión que será a través de Profibus DP así como las direcciones necesarias de los puertos para WinMOD Profibus DP. La siguiente figura muestra cómo esta programada la estructura Hardware del sistema:

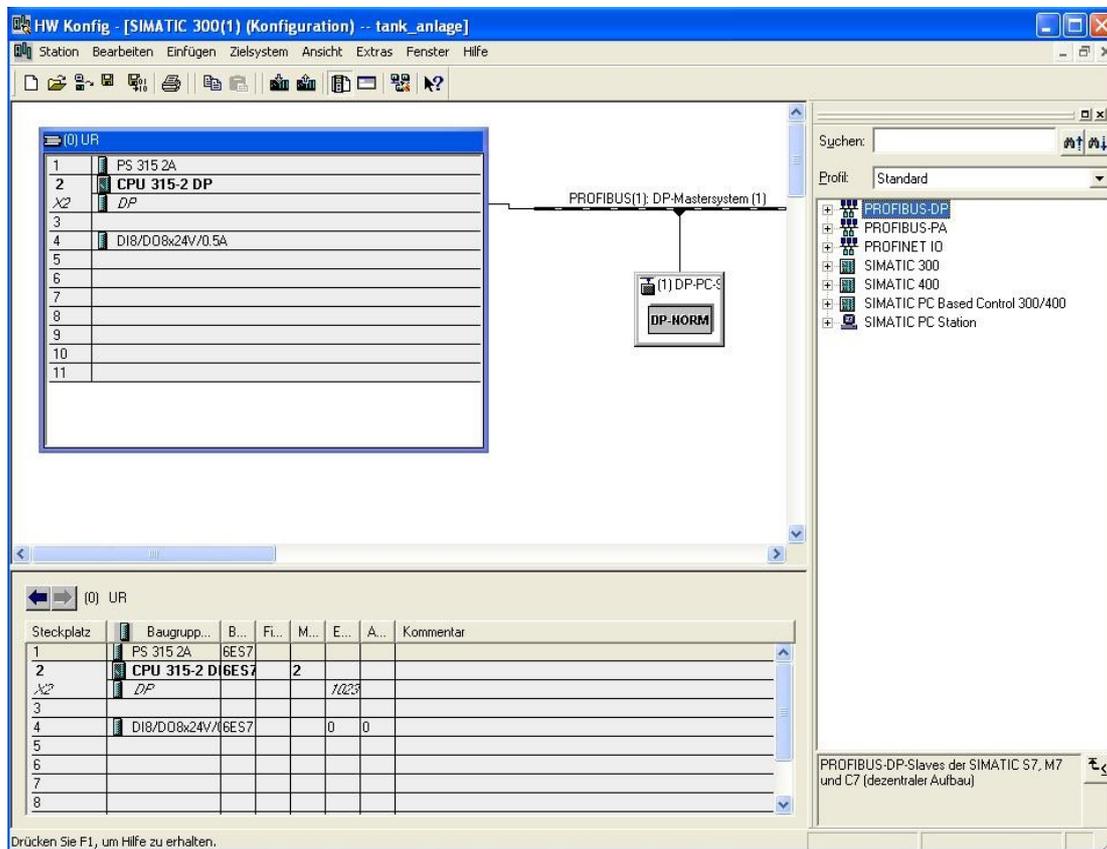


Fig 2.12: Esquema Profibus en Simatic Manager

Encontramos la configuración Profibus A71 en el catálogo del Hardware:

**PROFIBUS-DP/Next/MachineField/General/DP-PC-SL (PCI)**

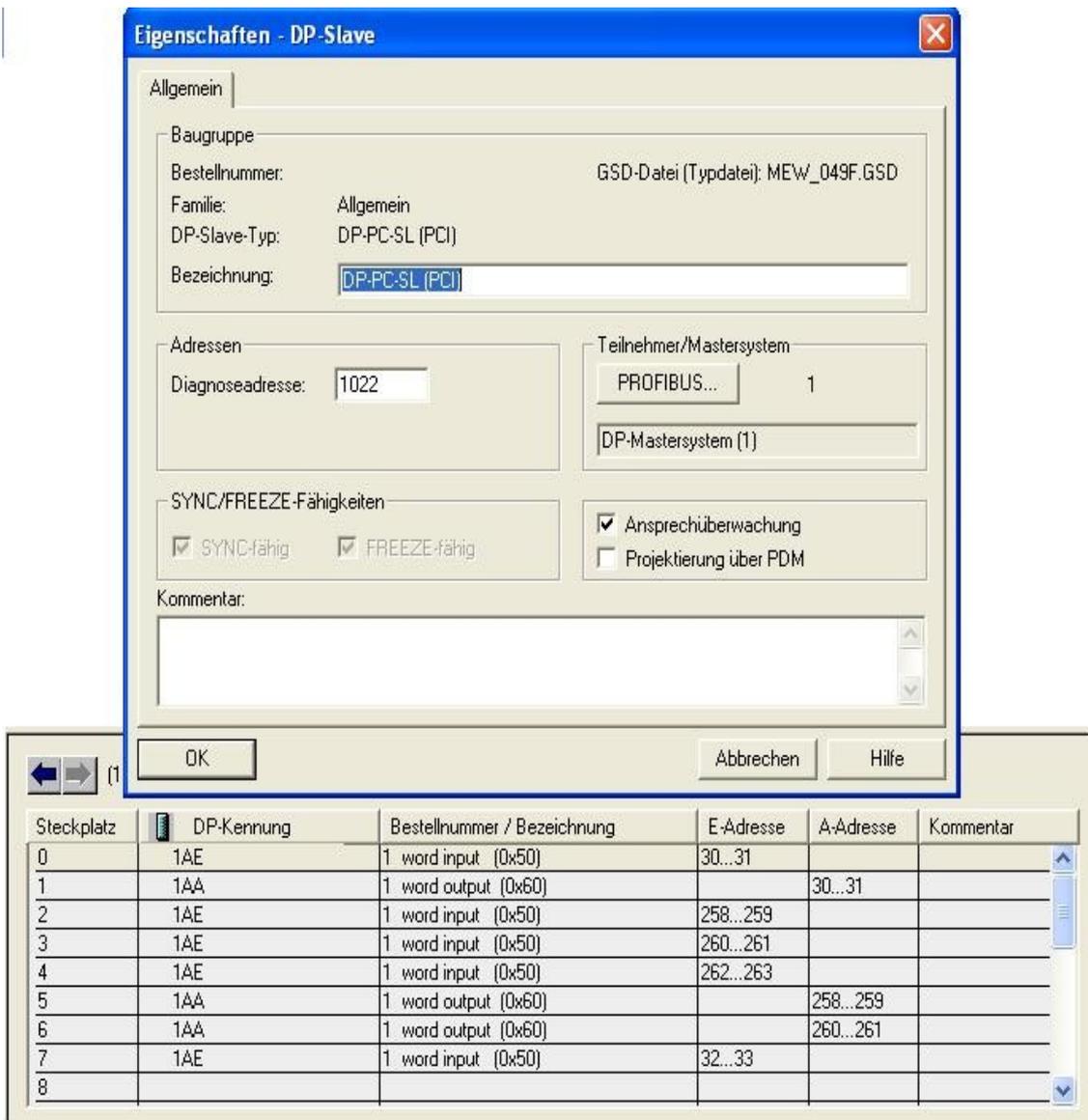


Fig 2.13: Configuración DC-PC-SL Siemens

Al igual que en CoDeSys ahora en Simatic Manager tenemos que elegir los elementos que van a comunicar. Elegimos las direcciones que tendrá cada elemento en la CPU. La velocidad de transmisión también es un parámetro que tiene que ir fijado a 1,5Mbps.

Para clarificar la programación y buscando la compatibilidad con el sistema real, será creada una Tabla de Símbolos idéntica a la programada en la CPU que controla el sistema real. Todos los sensores y actuadores, así como las señales de los niveles y de la potencia de las bombas irán descritas en esta tabla tal y como están en CoDeSys y en WinMOD, acompañadas de sus direcciones DP. Al ser esta tabla de símbolos idéntica a la del sistema real, el control desde step7 será transparente al sistema que estemos usando, será idéntico utilizar el sistema virtual o el real. Aunque ciertamente sí que hay alguna simplificación. He reducido el número de sensores que dan información acerca del nivel de los tanques a uno por tanque, en la realidad, hay tres sensores que, al mismo tiempo proporcionan información del nivel de líquido de los tanques, el programa de control actual, sólo utiliza uno de ellos, por esta razón sólo he implementado uno en mi trabajo,

ampliar en este sentido es bastante sencillo, sólo hay que añadir una variable WORD más por tanque, y programarla para que transmita el level de los tanques.

En la figura 2.14 se muestra esta tabla con todas las variables implicadas.

	Status	Symbol	Adresse	Datentyp	Kommentar
1		Air1	A 31.5	BOOL	output signal to activate air1
2		Air2	A 31.6	BOOL	output signal to activate air2
3		Air3	A 31.7	BOOL	output signal to activate air3
4		level Mix	EW 262	WORD	Level from Mix
5		level Tank1	EW 258	WORD	Level from Tank1
6		level Tank2	EW 260	WORD	Level from Tank2
7		level_Pumpe1	AWV 258	WORD	intensity pumpe1
8		level_Pumpe2	AWV 260	WORD	intensity pumpe2
9		Main Area	OB 1	OB 1	
10		Mix_empty_sensor	E 30.7	BOOL	Sensor indicates that Mix is empty
11		Mix_full_sensor	E 30.6	BOOL	Sensor indicates that Mix is full
12		Pumpe1	A 30.6	BOOL	Actuator that activates pumpe1
13		Pumpe2	A 30.7	BOOL	Actuator that activates pumpe2
14		SPS_Ventil_Mix	A 30.5	BOOL	output signal to activate ventil_mix_evac - To evacuate Mix
15		SPS_Ventil1a	A 30.0	BOOL	output signal to activate ventil1a - Ventil in Tank1 (from Tank1 to Tank2)
16		SPS_Ventil1b	A 30.1	BOOL	output signal to activate ventil1b - Ventil in Tank1 (from Tank1 to Mix)
17		SPS_Ventil1c	A 30.2	BOOL	output signal to activate ventil1c - Ventil in Tank1
18		SPS_Ventil1d	A 31.0	BOOL	output signal to activate ventil1d - Ventil in Tank1 (from Tank2 by Mix)
19		SPS_Ventil1e	A 31.1	BOOL	output signal to activate ventil1e - Ventil in Tank1 (fill up Tank1)
20		SPS_Ventil2a	A 30.3	BOOL	output signal to activate ventil2a - Ventil in Tank2 (from Tank2 to Tank1)
21		SPS_Ventil2b	A 30.4	BOOL	output signal to activate ventil2b - Ventil in Tank2 (from Tank2 to Mix)
22		SPS_Ventil2c	A 31.2	BOOL	output signal to activate ventil2c - Ventil in Tank2
23		SPS_Ventil2d	A 31.3	BOOL	output signal to activate ventil2d - Ventil in Tank2 (from Tank1 by Mix)
24		Tank1_empty_sensor	E 30.2	BOOL	Sensor indicates that Tank1 is empty
25		Tank1_full_sensor	E 30.0	BOOL	Sensor indicates that Tank1 is full
26		Tank2_empty_sensor	E 30.5	BOOL	Sensor indicates that Tank2 is empty
27		Tank2_full_sensor	E 30.3	BOOL	Sensor indicates that Tank2 is full
28		Ventil_mix_evac	E 31.6	BOOL	input signal to activate ventil_mix_evac - To evacuate Mix
29		Ventil1a_vorort	E 31.0	BOOL	input signal to activate ventil1a - Ventil in Tank1 (from Tank1 to Tank2)
30		Ventil1b_vorort	E 31.1	BOOL	input signal to activate ventil1b - Ventil in Tank1 (from Tank1 to Mix)
31		Ventil1c_vorort	E 31.2	BOOL	input signal to activate ventil1c - Ventil in Tank1
32		Ventil1d_vorort	E 32.0	BOOL	input signal to activate ventil1d - Ventil in Tank1 (from Tank2 by Mix)
33		Ventil1e_vorort	E 32.1	BOOL	input signal to activate ventil1e - Ventil in Tank1 (fill up Tank1)
34		Ventil2a_vorort	E 31.3	BOOL	input signal to activate ventil2a - Ventil in Tank2 (from Tank2 to Tank1)
35		Ventil2c_vorort	E 31.5	BOOL	input signal to activate ventil2c - Ventil in Tank2
36		Ventil2d_vorort	E 32.2	BOOL	input signal to activate ventil2d - Ventil in Tank2 (from Tank1 by Mix)
37		Ventil2b_vorort	E 31.4	BOOL	input signal to activate ventil2b - Ventil in Tank2 (from Tank2 to Mix)
38					

Fig 2.14: lista de símbolos S7

Hasta ahora, han sido listados los programas que serán utilizados para virtualizar el sistema de tanques, además han sido descritas las configuraciones básicas de los parámetros de los mismos en el marco de una comunicación PROFIBUS DP. Haré ahora una descripción del sistema de Tanques sito en el laboratorio de automática, enumerando cada uno de sus componentes, y una descripción del trabajo realizado con cada uno de los programas que necesitaré a lo largo del desarrollo de la herramienta.

## Capítulo 3:

### Diseño del modelo virtual

#### 3.1 Diseño del CAD con FLUX 3D STUDIO

El diseño en 3D ha sido una parte del proyecto que me obliga a aprender una disciplina nueva y a utilizar herramientas que desconocía completamente hasta ese momento. Comencé a diseñarlo paralelamente a la programación del modelo en WinMod, utilizaba los periodos en los que más atascado estaba, de manera que diseñar me ayudó a relajarme y despejarme, aunque me llegó a ocupar gran parte del tiempo productivo en las fases más intensas. Tras elegir la herramienta de diseño más adecuada, utilizando el método de prueba-error, diseñé lentamente las partes del sistema. Pronto conseguí un primer modelo que más tarde fue evolucionando, ya que estuve varios meses trabajando sobre el modelo introduciendo "mejoras", utilizando momentos de mayor tensión en el trabajo en el laboratorio. El resultado final pueden verlo en las siguientes imágenes, además del video demostrativo con el que presento el trabajo. Adjunto en el Cd el archivo vrm197.

El software FluxStudio me permite guardar el archivo con formato VRML97, este formato es requerido por WinMod para establecer la comunicación.

##### 3.1.1 Visualización atractiva

Algunos detalles hacen del diseño final una pantalla amigable y atractiva para quienes utilicen la herramienta. Entre otros detalles se pueden ver cómo los tanques se llenan y vacían viendo la cantidad de líquido que hay en cada uno de ellos. Se puede ver también que la bomba está funcionando, además ésta, realizará un movimiento que depende de la fuerza con la que bombea.

Las válvulas que se abren cambian de color, y existen figuras que aparecerán en casos de emergencia. En estos casos (*en los que no hay control o el control no ha funcionado bien*) aparecen unos símbolos encima de los tanques que indican que han rebosado, o debajo si se han quedado sin líquido.

En la siguiente imagen se muestra el diseño desde 4 puntos de cámara distintos, la

imagen corresponde casi idénticamente al sistema real del laboratorio. El azul del que están coloreados los tanques sería cuando todos están llenos.

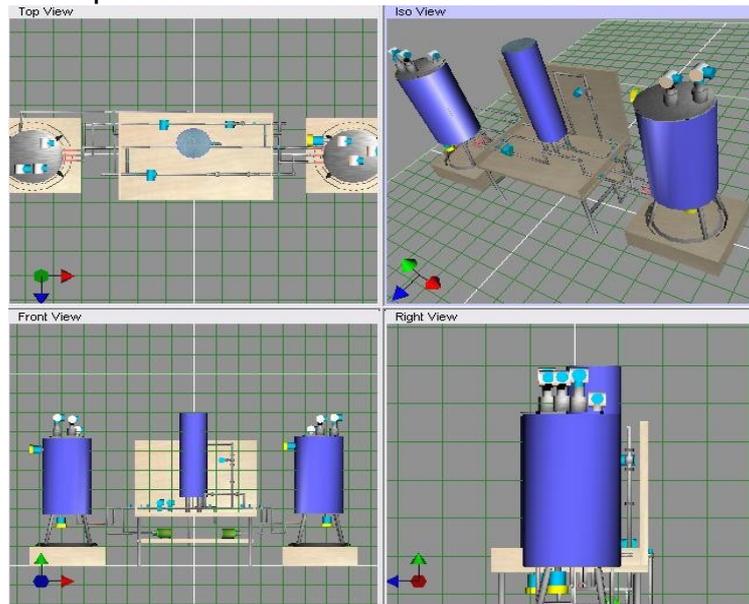


Fig 3.1: diseño 3D

### 3.1.2 Conexión con WinMod

Ésta fue sin duda la parte donde más complicaciones encontré en el desarrollo del trabajo, quizás no por la dificultad intrínseca de la actividad, pero sí por la falta de información, no hay mucha información libre al respecto y quizás (sobre todo en los inicios), falta de destreza en la búsqueda de información de esta disciplina, ya que tampoco conocía bien el lenguaje técnico. A la hora de dinamizar el diseño, me ví atascado y con poca ayuda, ya que mi tutor, tampoco tenía conocimientos de diseño, estaba sólo. Tras estudiar algo de lenguajes de programación en 3D especialmente buscando con el que trabaja WINMOD, VRML, también a través de información obtenida de otros trabajos y mucha intuición ayudada con algo de teoría que encontré por la red, conseguí finalmente establecer las conexiones para dinamizar el diseño. Así, debía de crear un archivo de texto, en donde aparecen los nodos que van a "cambiar", había que especificar el tipo de cambio que sufriría la pieza (de color, de tamaño, movimiento...), el rango en el que cambiaría, y el tipo y el nombre de la variable de programa (en este caso WinMOD) que controlaría este cambio.

Gracias a la técnica de prueba/error alcancé mi propósito configurando los parámetros. A continuación se muestra una impresión por pantalla del archivo de texto. Del resto se encarga WinMod, sólo hay que enlazar con un módulo de comunicación que WinMOD provee, todas las variables que serán necesarias para controlar los movimientos.

```

Zeilennummer;Signalname;Signaltyp;Knotenname;Feldtyp;Feldname;Feldindex;min;max
1;Level_Tank1;AI;dad_Liquido1;SFVec3f;scale;1;-2;2
2;Level_Tank1;AI;dad_Liquido1;SFVec3f;translation;1;0.7;4.7
3;Level_Tank2;AI;dad_Liquido2;SFVec3f;scale;1;-2;2
4;Level_Tank2;AI;dad_Liquido2;SFVec3f;translation;1;0.77;4.7
5;Level_Mix;AI;dad_LiquidoMix;SFVec3f;scale;1;-2.2;2.2
6;Level_Mix;AI;dad_LiquidoMix;SFVec3f;translation;1;1.37;5.8
7;Pumpe1;AI;dad_Cylinder50;SFVec3f;scale;2;0;0.5
8;Pumpe2;AI;dad_Cylinder41;SFVec3f;scale;2;0;0.5
9;Sensor_T1empty;AI;dad_leer1;SFVec3f;scale;0;-0.3;0.25
10;Sensor_T1full;AI;dad_full1;SFVec3f;scale;0;-0.3;0.25
11;Sensor_T2empty;AI;dad_leer2;SFVec3f;scale;0;-0.3;0.25
12;Sensor_T2full;AI;dad_full2;SFVec3f;scale;0;-0.3;0.25
13;Sensor_T1empty;AI;dad_leer1;SFVec3f;scale;1;-0.3;0.25
14;Sensor_T1full;AI;dad_full1;SFVec3f;scale;1;-0.3;0.25
15;Sensor_T2empty;AI;dad_leer2;SFVec3f;scale;1;-0.3;0.25
16;Sensor_T2full;AI;dad_full2;SFVec3f;scale;1;-0.3;0.25
17;Sensor_T1empty;AI;dad_leer1;SFVec3f;scale;2;-0.3;0.25
18;Sensor_T1full;AI;dad_full1;SFVec3f;scale;2;-0.3;0.25
19;Sensor_T2empty;AI;dad_leer2;SFVec3f;scale;2;-0.3;0.25
20;Sensor_T2full;AI;dad_full2;SFVec3f;scale;2;-0.3;0.25
21;Emergency1;AI;dad_Tank1overload;SFVec3f;scale;0;-1;1
22;Emergency1;AI;dad_Tank1overload;SFVec3f;scale;1;-1;1
23;Emergency1;AI;dad_Tank1overload;SFVec3f;scale;2;-1;1
24;Emergency2;AI;dad_Tank2overload;SFVec3f;scale;0;-1;1
25;Emergency2;AI;dad_Tank2overload;SFVec3f;scale;1;-1;1
26;Emergency2;AI;dad_Tank2overload;SFVec3f;scale;2;-1;1
27;Emergency_Mix;AI;dad_Mixoverload;SFVec3f;scale;0;-1;1
28;Emergency_Mix;AI;dad_Mixoverload;SFVec3f;scale;1;-1;1
29;Emergency_Mix;AI;dad_Mixoverload;SFVec3f;scale;2;-1;1

```

Fig 3.2: Texto para comunicación 3D

Y éste es el resultado final, así se visualiza el sistema de tanques virtual. El video demostrativo será proyectado en la presentación del trabajo, se pueden apreciar los sensores y las válvulas, su estado, así como el nivel aproximado de cada tanque.

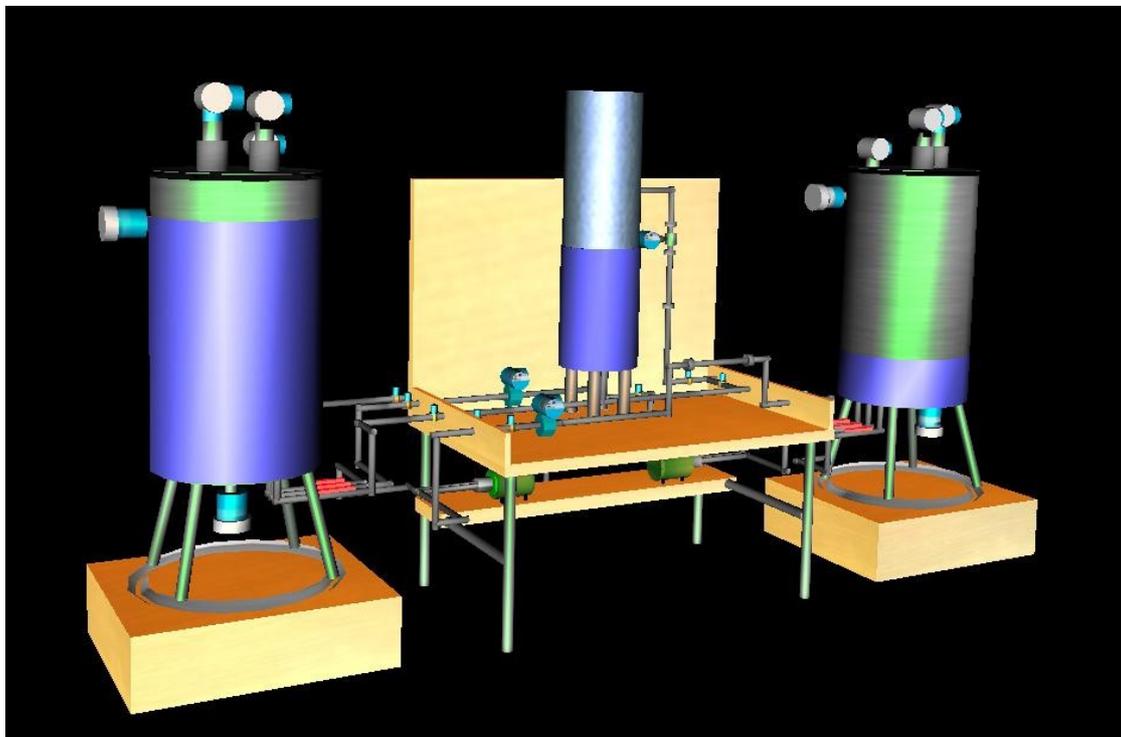


Fig 3.3: modelo en 3D final

## 3.2 Modelo del sistema en WinMod

La primera tarea a resolver en este trabajo fue la de aproximar un modelo virtual del sistema de tanques con ayuda del software WinMod.

Para modelar cada uno de los tanques utilicé el módulo *glied AG101* del software, que permite la simulación de un sistema lineal con una entrada digital y una salida analógica entre 0 y 1. La salida analógica es escalable de modo que fue fácil encontrar la manera de que el comportamiento del tanque real y del simulado fuera aproximadamente el mismo, así, por ejemplo los problemas de control que ocurran por retardos serán similares en la realidad y en la pantalla. Queda así configurada la velocidad de respuesta ante una entrada positiva y negativa. Para configurar el factor de escala del glied he medido la velocidad con la que se llena el sistema real dependiendo del nivel de potencia de la bomba, de este modo he configurado la velocidad de llenado de los tanques virtuales de manera que sea lo más realista posible.

La velocidad del flujo de líquido de los tanques laterales es muy parecida, aunque no es la misma, porque sólo el tanque de la izquierda tiene la entrada de agua del exterior, es decir, el tanque derecho recibe líquido únicamente a través del tanque izquierdo, o del de mezcla, esta configuración es inusual en un sistema de tanques de este tipo que se utilicen en tareas industriales, ya que lo que se pretende es nivelar la proporción de líquido de cada uno de dos tanques distintos que formarán la mezcla central, que más tarde pasará a otro proceso. De este modo, la configuración de los glied que simulan ambos tanques es parecida, aunque no exactamente la misma. El tanque central de mezclado es más pequeño, por lo tanto se llena más rápido en relación con los otros dos ante un flujo de entrada similar. Para configurar el módulo que emula el tanque central también he seguido el mismo proceso, midiendo la velocidad de llenado en varias ocasiones y con diferentes niveles de presión de las bombas.

En la figura siguiente se muestran las propiedades de este módulo:

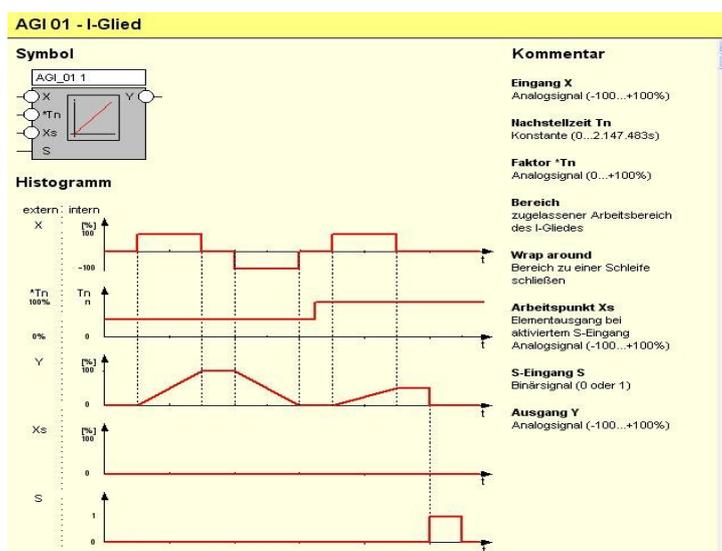


Fig 3.4 Esquema de la macro del tanque

Estructura del programa principal:

En el programa principal se pueden distinguir 15 cajas o macros que serán las más importantes dentro de nuestra simulación, cada uno de los tres tanques es una Macro, así como las bombas, las válvulas, y la macro de comunicación, que más adelante explicaremos con detenimiento. En la siguiente tabla se muestran cada una de ellas y la función que tiene dentro del programa.

<u>Makro</u>	<u>Function</u>
<b>Tank1</b>	<i>To emulate Tank 1. Tank's makro recibes sensors information of the orders.</i>
<b>Tank2</b>	<i>To emulate Tank 2</i>
<b>Mix</b>	<i>To emulate Mix-Tank</i>
<b>Each motor (2)</b>	<i>To emulate each motor, each motor allows processes in each part of the system</i>
<b>Each valve (9)</b>	<i>To emulate each valve, we can active or desactive it by two ways, SPS control or Manual.</i>
<b>Comm makr</b>	<i>To comunicate the WinMod Process Model with the VRML97 file. Recives all the information witch the CAD will work.</i>

*Tabla 3.1: Macros utilizadas en WinMod*

En el modelo del sistema ha sido programada una variable de control que salvo para pruebas o excepciones (dado que el sistema real no dispone de esta herramienta, no sería coherente instalarla en el sistema virtual si queremos ser fieles a la realidad), estará a valor 0. Si esta variable toma el valor 1, actuará de alarma bloqueando todo proceso en caso de emergencia, mientras esto no ocurra, el sistema se comportará como normalmente lo hace. Un caso de emergencia típico sería el llenado completo de alguno de los tanques, si el sistema de control no hace nada, continuar la acción provocaría que rebose el líquido.

En la siguiente imagen se puede ver de cómo están diseñados por dentro los macros de cada tanque, y justo debajo, el Glied que elegimos para como sistema *tanque*. A la izquierda están las variables de entrada que tiene el sistema, a la derecha las variables de salida, el nivel que marca, y algunas variables binarias que se necesitan para acciones fuera de esta macro, posteriormente para el programa de control o la representación en 3D. El módulo alargado al que llegan todas las entradas es el módulo que suma las acciones que se estén llevando a cabo, así, si se está llenando el tanque derecho, la entrada X1 será positiva, y el resto estarán a 0.

Los siguiente si seguimos en sentido a la derecha, es un cuadro informativo del estado del tanque y sus variables, lo siguiente son los sensores que hemos instalado dentro del Macro que nos dan información digital acerca de si el tanque en cuestión está vacío, medio lleno, o lleno.

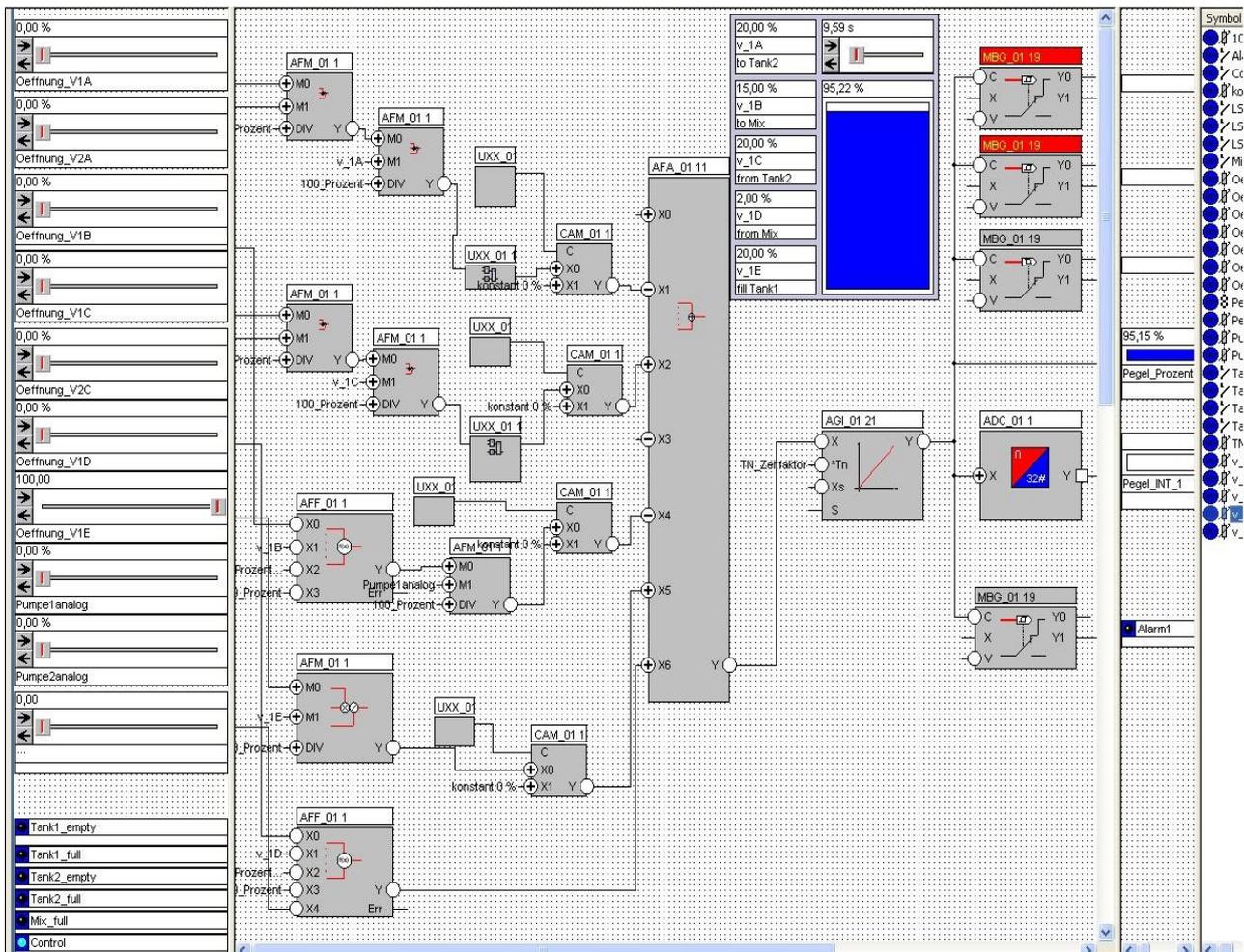


Fig 3.4: macro del tanque1, modelo de proceso en WinMod

Finalmente a la derecha vemos todas las variables incluidas las auxiliares que son utilizadas en el macro del tanque.

Ejemplo de una macro de cada válvula. Las válvulas también las he modelado con el mismo módulo que los tanques, son sistemas que ante una entrada binaria 0 o 1, tardan un cierto tiempo en reaccionar. Así como en el sistema real, las válvulas virtuales tardan aproximadamente 0,3 segundos en abrirse por completo. Para su modelado he utilizado el mismo bloque que el usado para los tanques, pero con una constante de llenado/apertura mucho más elevada.

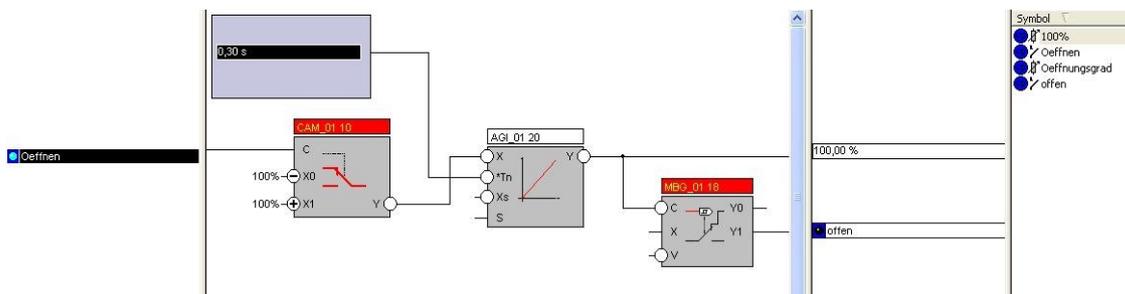


Fig 3.5: Macro de válvula en WinMod

Las bombas tienen un diseño similar al de las válvulas, la diferencia radica en que ésta última se abre siempre al máximo, y las bombas tienen distintos niveles de amplitud, y dependiendo del seleccionado, bombeará con mayor o menor fuerza.

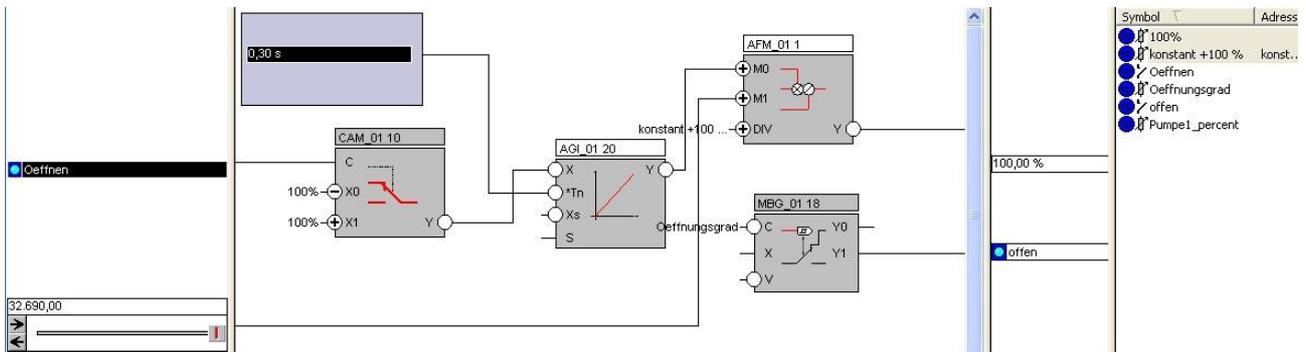


Fig 3.6: Macro de bomba en WinMod

En la siguiente figura he fotografiado la pantalla principal del modelo en WinMod, así se tendrá una visión más general de la simulación que se ha llevado a cabo, tras estudiar detenidamente cada una de sus partes. En esta imagen aparecen las macros de los tanques con la información interna representada, las válvulas, bombas, así como todos los sensores instalados.

Para hacer las primeras pruebas he utilizado una herramienta de WINMOD, que permite activar y desactivar variables de estado manualmente, esto es muy útil y a la vez peligroso, porque si activamos una variable dentro de una macro para hacer pruebas hasta conseguir que funcione, existe el riesgo de olvidar que esa variable está activada, y si no entramos de nuevo en la macro para desactivarla, no recibimos información de que esto es así y nos puede llevar a confusión cuando probamos el programa completo. Varias fueron las ocasiones en que perdí un tiempo precioso buscando fallos que no existían, derivados de este descuido.

La siguiente imagen es una instantánea de la pantalla principal del modelo. En esta pantalla se pueden ver todas las variables, de entrada y salida del sistema, así como también se puede interactuar con él para hacer pruebas. Gracias a la opción "Force" del software que permite establecer el valor de una variable, ya sea analógica o digital. Así podremos probar todas las posibilidades que queramos viendo reflejada la consecuencia en la representación 3D.

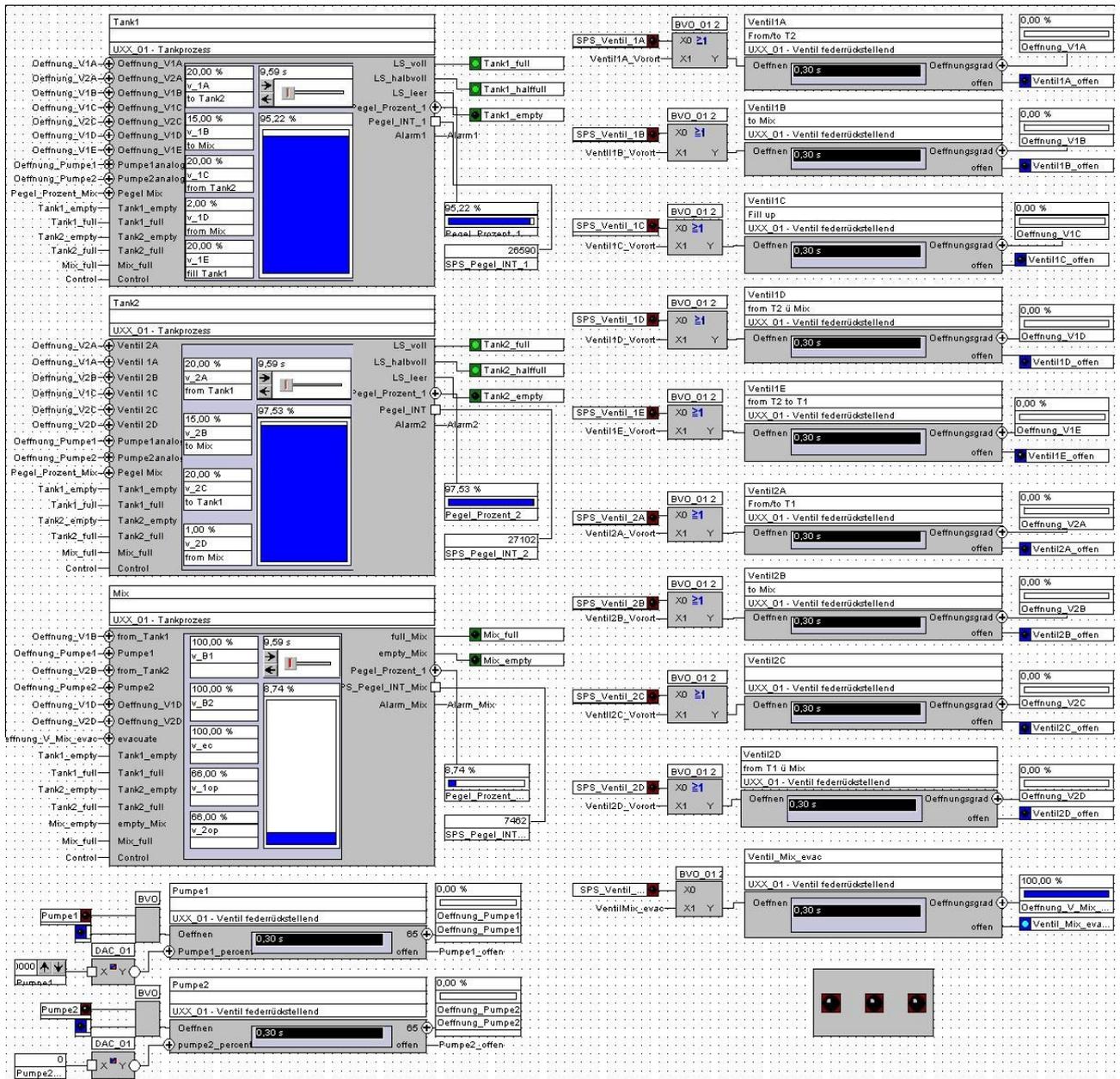


Fig 3.7: esquema modelo en WinMod

La siguiente imagen muestra el aspecto del mando de control programado en WinMod para aplicar un control autónomo, la utilidad de este mando es meramente para hacer pruebas ya que el sistema real no tiene ningún control de este estilo. En el modo de operación normal este mando estará oculto y deshabilitado, sólo estando accesible en el modo de operación pruebas. Este mando de control lanza órdenes a un nivel por encima que cualquier controlador que instalemos externo a WinMod, esto es, siempre tendrá prioridad este controlador sobre cualquier otro, por ello lo dejamos por defecto desactivado.

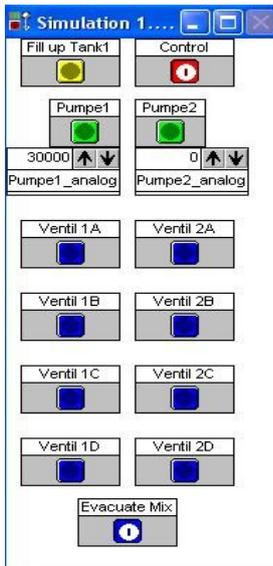


Fig 3.8 PAD control

Con ayuda de este módulo, podemos controlar todos los actuadores de nuestra simulación de manera manual desde el simulador. Esto significa que la acción es prioritaria a los sistemas de control que podamos aplicar de manera externa. Hay que prestar atención en el uso de este controlador ya que dejar aquí un actuador activado o algún valor en particular guardado, puede dar lugar a errores de operación, lo que hace que busquemos arreglar algo un problema que no existe.

### 3.3 Comunicación exterior, PROFIBUS

Esta parte del proyecto es crítica, ya que deben estar todos los parámetros bien configurados, de lo contrario no existirá comunicación. En este apartado nos dedicaremos a mostrar cómo ha sido realizada la configuración de los distintos parámetros. A la hora de configurar no tenemos herramientas para detectar cual es el fallo, si lo hay y se produce un error, debemos revisar cada uno de los parámetros, en busca de aquel que sea disonante con la configuración del periférico con el que nos pretendemos comunicar.

Como la conexión entre los equipos será a través de líneas Profibus el primer paso fue la velocidad de comunicación. Ésta quedó configura a 1,5 Mb/s. Esta es la velocidad a la que actualmente se comunica el sistema real con su controlador instalado en los PLCs. Como puerto, el número 4. Aquí simplemente hay que elegir un puerto libre, y tiene que ser coherente con el puerto elegido en el sistema destino.

Las variables de sistema, los sensores, actuadores, niveles de tanques e intensidad de bombas, tienen que venir representadas en la tabla DP-PC-SL en la configuración Profibus de WinMod. Para sensores binarios y válvulas, he usado 2 Doubleword Long, una de ellas es para variables de entrada (de color verde y desplegadas en la figura de la izquierda) y otra para las salidas del sistema virtual (en color rojo, son el segundo grupo de variables).

Para representar en la comunicación PROFIBUS los niveles de los tanques, así como la presión de las bombas, he usado una palabra doble. Estas bombas son también analógicas aunque, en el sistema real, el controlador que hay instalado sólo ofrece 4 posibles niveles de potencia. En la virtualización he querido reflejar tal y como está en la realidad, de manera que se podrá desarrollar un control que aproveche la versatilidad de las bombas mejor que el actual, pero el controlador básico se comportará como el que hay ahora instalado, además realizaré un controlador avanzado que demuestre algunos avaces en relación con el que actualmente se está usando.

He de señalar aquí que en esta etapa del proyecto me atasqué unos días, debido a que los valores numéricos que me proporcionaba WinMod cuando me sacaba el nivel de los tanques eran valores irreales debido a una mala configuración interna del modelo. Finalmente descubrí el error, tuve un despiste en la declaración de variables teniendo declaradas dos veces una misma variable de manera *analógica* y otra *entera*, parece algo sin importancia, pero a lo largo de la realización del programa tuve que declarar un número elevado de variables, internas y externas al modelo en Winmod, además al ser la primera vez que manejaba la herramienta, no tenía experiencia para resolver rápidamente, esto hizo que esta etapa fuera especialmente tediosa.

Realicé 3 modelos distintos antes de empezar con el que actualmente le estoy presentando. Estos modelos, si bien conseguían cumplir con muchas de las características incluso llegar a ser válidos en cuanto a su funcionamiento, tenían un problema, especialmente el anterior a la versión final, era muy complicado de manipular y cambiar,

cuando tenía que hacer alguna modificación (cosa que al principio era bastante usual), tenía que cambiar todo el modelo, de manera que no resultaban versátiles y por lo tanto los deseché.

Recordemos que el programa está finalmente diseñado igual que el sistema de tanques real, de manera que si necesita algún tipo de ampliación, quien la ejecute deberá poder ser capaz de hacerlo fácilmente, esto es debido a la sencillez de representación y conexionado de elementos de los tanques. El modelo virtual, debe de ser solidario con esta característica también, ya que forma parte de una herramienta de trabajo que se prevee siga evolucionando con el paso del tiempo.

A la izquierda puede observar la configuración del esquema DP-PC-SL, que recordemos, debe ser la misma en los sistemas de control que queramos implantar. Tanto en Codesys para probar nuestro sistema como más adelante en Step 7, la configuración de estos elementos debe ser la misma. En el caso contrario no establecerá la conexión Profibus.

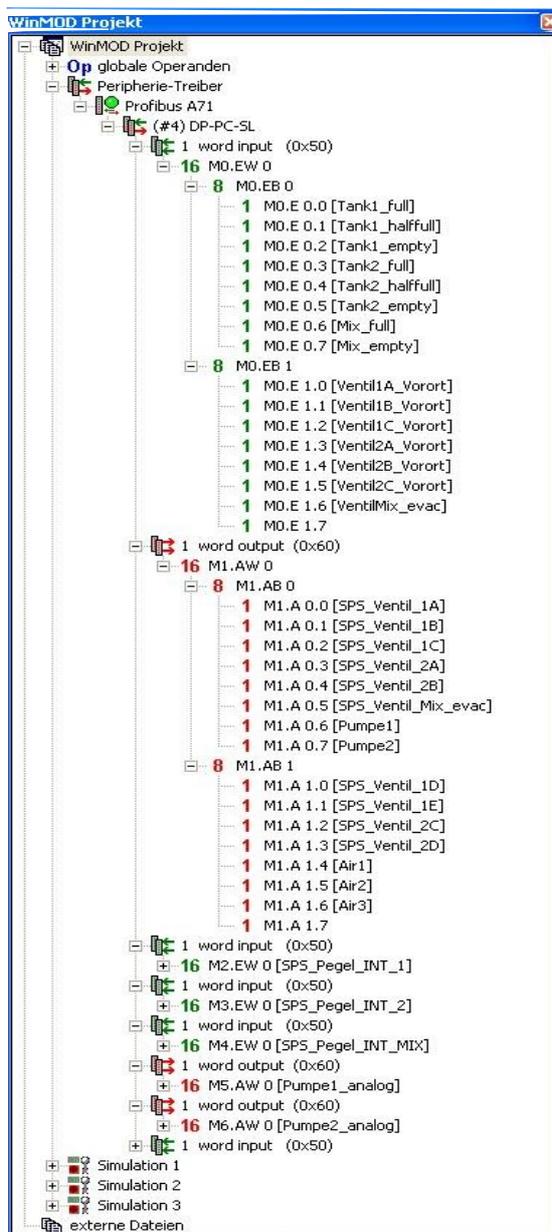


Fig 3.9: transmisión DP-PC-SL

# Capítulo 4:

## Sistemas de Control

### Introducción

Aquí comenzamos la programación del control del modelo. Hasta ahora llevamos la construcción del modelo en WinMod, la representación en 3D que está enlazada con el modelo animada según los valores de las distintas variables, y la comunicación PROFIBUS configurada con todas las variables críticas representadas. En este punto sólo nos queda hacer un control válido y completo que simule el control instalado actualmente en el sistema real. La finalidad es que ambos sistemas ante la misma entrada ofrezcan un mismo comportamiento.

Primero hacemos el control con CodeSys2.3, entorno de programación introducido ya en el capítulo 2: herramientas software. Esto es necesario en el proyecto por deseo del tutor quien utiliza también este entorno en sus clases de control. Esta parte también es intrínseca de la herramienta lectiva que será este proyecto. Mas tarde entraremos en detalle con el controlador diseñado para el control directo a través de PLCs S3 de Siemens.

Para el desarrollo del control son necesarios varios aspectos:

- 2 tipos de control: he de instalar controles *automático y manual*.
- Inclusión del *Botón reset o parada*: que bloquea el sistema en cualquier momento.
- Características similares al sistema real en cuanto a valores de las variables de los sensores.

## 4.1 Control con CoDeSys2.3

Primeramente se realiza la configuración Profibus, ya ampliada en el *capítulo 2: Herramientas software*, tras haber establecido la configuración correctamente comenzamos con el programa de control. Han sido programadas dos rutinas de control, como sugería el plan del proyecto, por defecto el control seleccionado es el automático, esta variable puede cambiar en cualquier momento, pulsando en el cuadro de control.

La representación en Codesys es también requisito del programa, además esclarece de un modo directo la situación del sistema, la siguiente imagen es una impresión de pantalla del diseño en codesys del sistema. Se puede ver cómo el líquido sólo entra en el sistema por el tanque1 y sólo sale de él por el Tanque de Mezcla a través de la válvula **Evacuate**. Las tuberías que por las que pasa el líquido en cada momento cambian de color y las válvulas también cuando están abiertas. Además se pueden ver gráficas de cómo ha variado el nivel de cada tanque en los últimos 3 minutos.

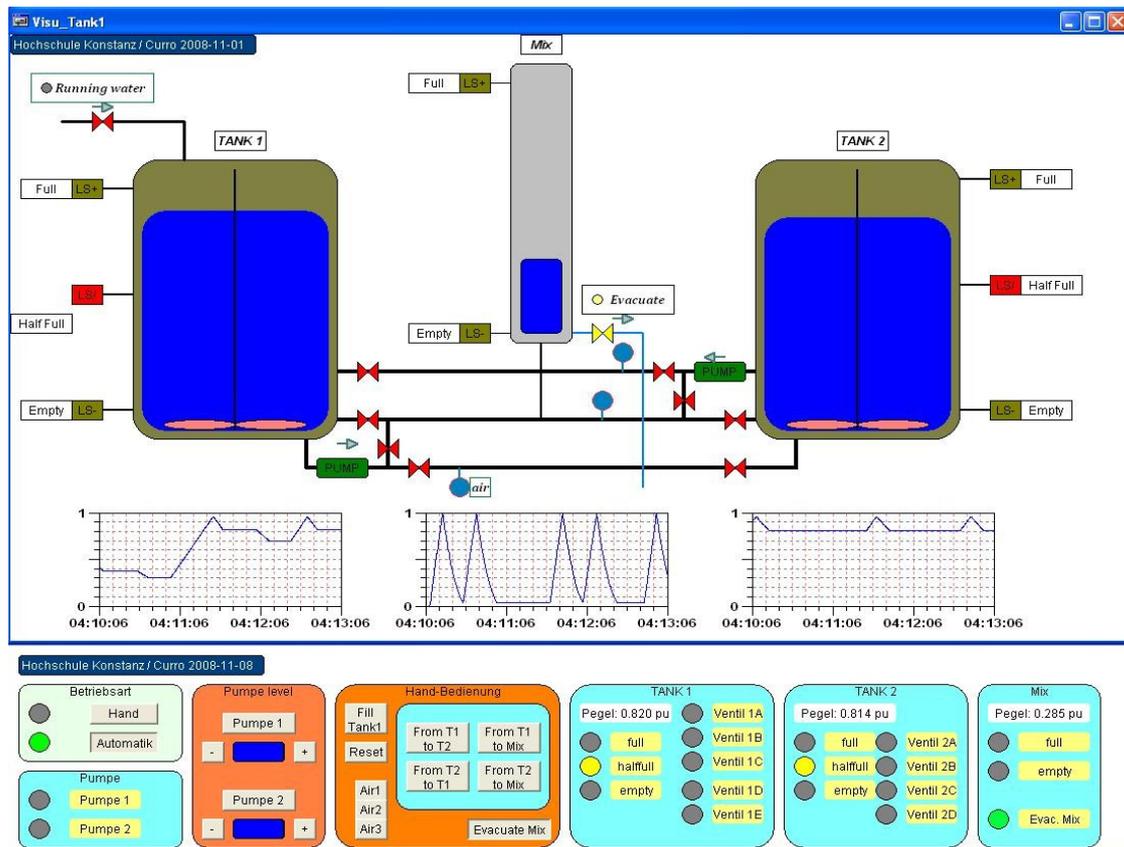


Fig 4.1: Representación gráfica en Codesys

Todos los programas están en el Apéndice B

Al comenzar el programa principal espera a que se elija el tipo de control que va a realizar, **manual** o **auto**. Si la opción es manual, el programa se comporta ejecutando todas las rutinas de manera secuencial, esperando condiciones para cambiar de estado,

dependiendo del estado que tenga, así se comportará de una manera u otra. Un ejemplo de estado es: *Transfusión\_1\_2* : en este estado las válvulas abiertas son las necesarias para que el líquido fluya del tanque 1 al 2, el resto de válvulas están cerradas. En el modo auto el programa comienza un ciclo parecido al ciclo que hace el sistema real aunque algo más largo y con más acciones que éste, pasando por diferentes procesos que podemos hacer con nuestro sistema, se llena el Tanque 1, luego desde el Tanque2 al mixto, desde el tanque1 al mixto directamente, y por último evacuamos a la salida la mezcla, entonces se vacía el mixto y vuelve a comenzar la rutina.

En la figura 4.1 se puede ver representado el control automático, si uno se fija en las gráficas puede ver cómo se han ejecutado los procesos. El siguiente dibujo muestra cómo ocurre el control manual, haciendo uso del mando de control resentado también en la parte baja de la figura

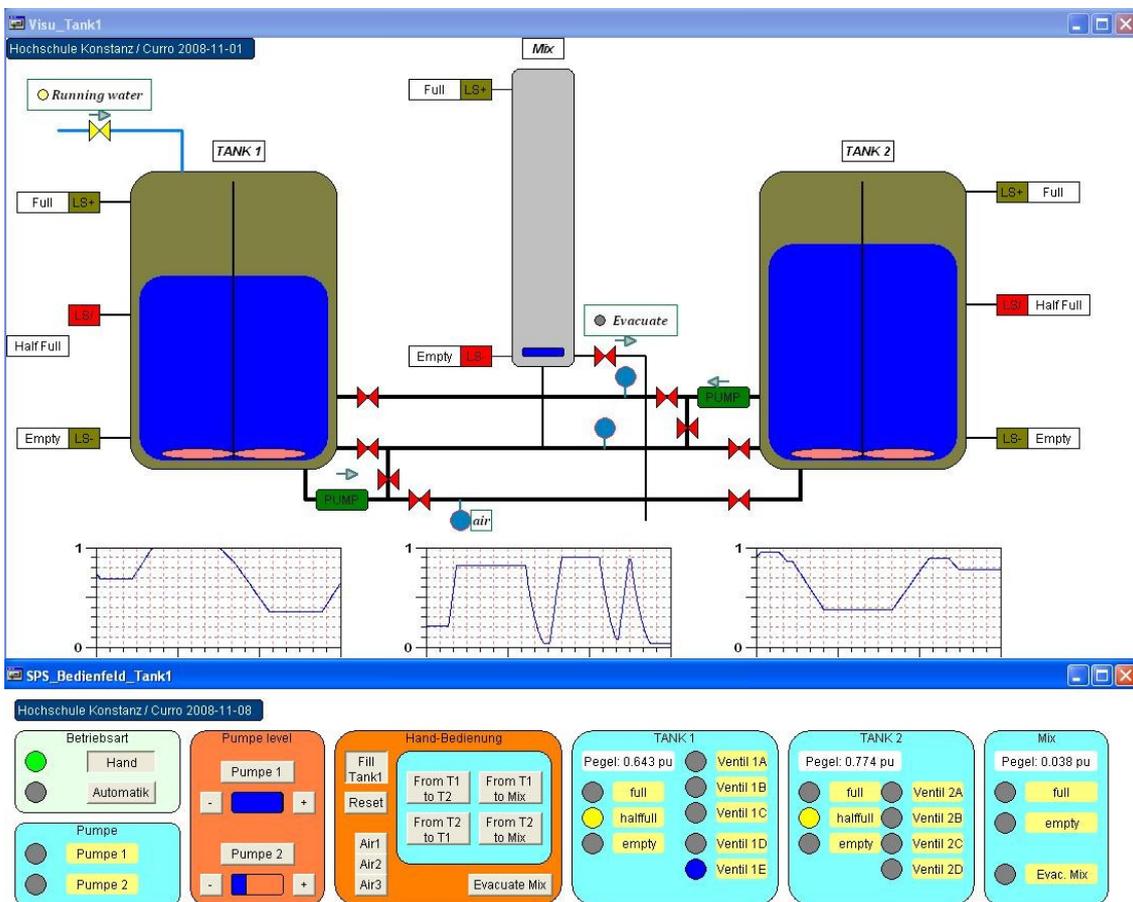


Fig. 4.2: Control Manual con CodeSys2.4

En este control no encontraremos problemas con los retardos, por el margen que se deja y por la rapidez de la ejecución de la rutina de control en relación con la velocidad de los procesos que tienen lugar.

## 4.2 Control con Siemens S7. Simatic Manager



El esquema en el que se han conectado los elementos es el siguiente:

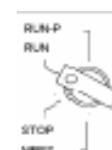
El sistema real es virtualizado, entonces conectado a una CPU s300 donde está instalado el programa de control, con un módulo entradas y salidas.



Fig 4.3: Arquitectura del sistema

Para ejecutar el control con Siemens S7-300, debemos cargar primero el programa en la CPU, para hacerlo seguimos los siguientes pasos:

- Encendemos el S7 – SPS.
- Colocamos el switch de la CPU en STOP
- Abrimos el programa en Simatic Manager (STEP 7)
- Abrimos el HW-config, para cargar la configuración del sistema, que en nuestro caso es una conexión Profibus con un sistema DP-PC-SL
- Cargamos el programa de control en la CPU 315-2 DP.
- Activamos el modelo en winMOD.
- Encendemos la CPU, colocando el interruptor en RUN
- Si todo ha ido bien ambos sistemas quedan correctamente conectados.



El control con el SPS es un control secuencial que se maneja con las entradas del módulo instalado junto a la CPU, en el caso del sistema real, la introducción de las acciones viene dada por un panel de control táctil, donde se ejecuta un programa que ofrece la posibilidad de cambiar de acción. En nuestro caso todos los procesos están controlados. En el modo auto, se ejecutan secuencialmente varias tareas, de manera similar a lo programado en CodeSys. En el modo manual, cada entrada del módulo indicará una acción que será tomada por el modelo en WinMod. Todos los programas están en el Apéndice B.

Hay dos funciones programadas, OB1, es la función principal donde están programadas todas las acciones, FB1 es la función auxiliar donde hay programadas algunas pequeñas secuencias que son usadas por la principal.

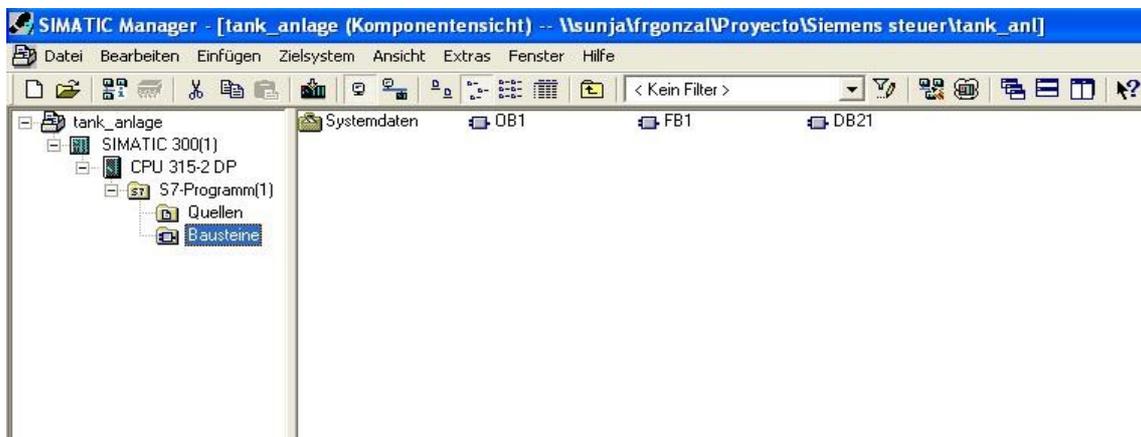


Fig 4.4: pantalla principal del programdor de Simatic manager

La siguiente imagen muestra la correspondencia entre las entradas del módulo Input/output y las acciones tomadas

Signal	Action
E 0.0	fill-up T1
E 0.1	from T1 to T2
E 0.2	from T2 to T1
E 0.3	from T1 to mix
E 0.4	from T2 to mix
E 0.5	from T1 to T2 by mix
E 0.6	from T2 to T1 by mix
E 0.7	evacuate mix

Tabla: 4.1: Operando manualmente con S7-300

*E0.0: Llenar el Tanque 1 hasta que el sensor de lleno se active o se desactive el botón.*

*E0.1: Llenado de Tanque1 al Tanque 2 directamente*

*E0.2: Llenado directo del Tanque 2 al Tanque 1*

*E0.3 Llenado del tanque de mezcla desde el tanque 1*

*E0.4 Llenado del tanque de mezcla desde el tanque 2*

*E0.5 Llenado del tanque 2 desde el tanque 1 a través del tanque de mezcla*

*E0.6 Llenado del tanque 1 desde el tanque 2 a través del tanque de mezcla*

*E0.7 vaciar el Tanque de mezclas*

El programa principal arranca en manual, de esta manera podemos, a través del panel I/O de la CPU, dar las órdenes. En el sistema real hay conectado una pantalla táctil, desde donde se dan órdenes. Se deja como ampliación a este proyecto la programación de las rutinas necesarias para poder manejar también con la pantalla táctil, el sistema virtual. La programación del modelo es solidaria con el sistema real, exportando el mismo tipo de señales en su rango, por lo que esa ampliación se adecuará al modelo existente a la perfección.

Hay programada también un control automático, en el que se ejecutan las tareas en serie mostrando el funcionamiento y todas las posibilidades. Para ejecutarlo hay que cambiar una variable inicial en el programa principal.

Todos los programas están incluidos en el CD, y los códigos en el apéndice.

# Capítulo 5

## Conclusiones

### 5.1 Herramienta en uso

La utilización de la herramienta es muy sencilla e intuitiva, no se necesita de conocimiento previo de la materia para poder manejar el simulador.

Lo primero es preparar las estaciones de trabajo, ambas deben disponer de tarjetas de comunicación PROFIBUS indicadas en el capítulo 1, punto 1.1.3. En al menos una de ellas, tendremos instalado el Software *WinMOD V5* de *Media Machines*, en otro PC, se instalará el software *CodeSys2.3*, Los archivos necesarios para el modelo están en el directorio: *//Programas/WinModV5/* y los necesarios para el controlador software *//Programas/Codesys2.3/*. Si bien los necesarios para el control mediante PLC están en *//Programas/Step7/*. Para utilizar el control PLC, lo primero será cargar el programa en la CPU S300, así, tras cargar los datos en memoria, se procede como se indica en el capítulo 4.2.

Como se indica a lo largo de este proyecto, esta herramienta dispone de dos modos de funcionamiento, su diferencia principal es el controlador del simulador, podemos elegir utilizar un control software instalado en otra máquina, realizado con *CodeSys2.3*, principalmente para estudiar problemas derivados de los retrasos en la comunicación a la hora de controlar sistemas automáticos, y el otro modo de control es mediante hardware PLC Siemens S300, a través del cual estudiamos un comportamiento muy similar al comportamiento real del sistema, la finalidad lectiva principal de este modo de control es su uso como entrenador del sistema real.

Para inicializarlo, es necesario conectar todo el sistema adecuadamente, comunicación se establece en cualquier caso utilizando PROFIBUS DP. Elegimos el modo de trabajo, en un primer caso analizamos el modo de control por software, donde están dos computadoras conectadas a través de sus tarjetas PROFIBUS mencionadas en el Capítulo 1. Una vez conectados los equipos, abrimos el archivo *Tanques.wmp* utilizando el programa *WinMod*, seguidamente se abrirá automáticamente la visualización en 3D y se coloca en la ventana principal de *WinMod* un esquema completo del modelo con un controlador que inicialmente está deshabilitado (este controlador es para pruebas directamente con el modelo). En la otra computadora abrimos el archivo *steuer.sim* que se abre con *CodeSys*, seguidamente arrancamos el modelo en la primera computadora y activamos el control en la segunda. Ya

tenemos funcionando en modo automático el control (por defecto), y podemos empezar a realizar pruebas.

Para el modo de trabajo con controlador Hardware, conectamos adecuadamente los equipos, esta vez, una computadora con una CPU de la familia Siemens S300 al cual, previamente le ha sido cargado en su sistema de archivos, el programa de control. Abrimos primero el modelo en la computadora utilizando el archivo *Tanques.swp*, encendemos el procesador y colocamos el modo RUN. Entonces, arranca el controlador en modo manual, esperando órdenes a través del panel del I/O de la CPU. Así, ésta es la principal diferencia con el sistema real, el cual está conectado a una pantalla táctil para su control. Un apartado opcional era programar el control para que fuera posible trabajar con la pantalla táctil, lamentablemente no me dió tiempo en el periodo que estuve allí, de completar este aspecto.

La herramienta tiene un uso principalmente lectivo, permitiendo a los alumnos interaccionar con un sistema virtual de la misma forma que lo harían en el sistema real, así aprenderán de una manera amigable las distintas configuraciones posibles y las nociones básicas, antes de entrar en contacto con el sistema real, sito en el laboratorio de automática de la facultad de ingeniería de Konstanz.

## 5.2 Esquema general

Este proyecto tiene 4 partes bien diferenciadas,

- 1) la construcción del modelo en WinMod y su exportación a una línea Profibus DP,
- 2) el diseño del sistema en 3D y siguiente conexión con el modelo en WinMod (donde he tenido problemas),
- 3) el diseño del control del sistema con ayuda del software CoDeSys2.3, muy utilizado en sistemas complejos controlados por ordenador, y la posterior configuración de conexión con el modelo, a través de la línea Profibus. Con este controlador he hecho la mayor parte de las pruebas comprobando que el modelo estaba correctamente diseñado. Y por último,
- 4) la configuración de la conexión con la CPU Simatic S7 300, y posterior diseño del control con STEP7 de Siemens, quedando el modelo en WinMod controlado por el PLC S7 300.

A lo largo del trabajo he aprendido diversas técnicas:

A diseñar modelos con el software WinMod (muy parecido a Simulink de Matlab), estos modelos pueden llegar muy precisos. En mi caso he tenido en cuenta muchos detalles, no obstante, el modelo final, aproxima algunos valores como por ejemplo, el flujo del fluido, y de cantidad de líquido que entra en los tanques. He aproximado esos valores teniendo en cuenta que no era necesaria una mayor precisión para llevar a cabo el proyecto.

También he aprendido a manejar varios sistema de control, como son el Software CoDeSys2.3, dado cuenta de detalles acerca del delay que la comunicación entre sistemas conlleva y los problemas que pueden surgir. Solucionar los problemas surgidos por fallos en el modelo o el controlador.

He aprendido a diseñar en 3D y a conectar el diseño con un modelo informático. Cosidero importante este punto, porque el diseño en 3D establece un modo de comunicación entre el cliente y el proyecto, visual y muy fácil de entender. De manera que, viendo el cliente cómo funciona su proyecto en una representación en 3D animada, tendrá una mejor idea y se convenza más rápido de la viabilidad del proyecto.

Por último he aprendido a utilizar otro dispositivo PLC, en este caso el Siemens S7-300, utilizado en complejos sistemas controlados automáticamente. El mismo sistema de tanques real, está controlado por CPUs S7 300 y módulos auxiliares entrada/salida, que permiten el conexionado de todos los elementos. Hasta ahora había trabajado con PLC's, pero nunca antes, había controlado sistemas externos a la CPU.

Encontré los principales problemas en:

Aprender a utilizar bien la herramienta WinMod, que es la base del proyecto, y por

lo tanto, la parte más delicada del mismo. Tras un primer diseño, me dí cuenta de que era inviable continuar por ese camino, tras cerca de 3 semanas de trabajo, deseché lo que había hecho hasta el momento, y comencé de nuevo, esta vez con un modelo más versátil que permitía incluir todas las características necesarias de manera más dinámica y sencilla.

El desarrollo del controlador en CodeSys, fue más sencillo, este software de programación es muy visual e intuitivo, por lo que no tardé en solucionar los problemas que iban surgiendo.

Las partes más costosas fueron sin duda, la conexión de la representación 3D con el modelo en WinMod y la configuración y posterior programación con Step7, que a pesar de ser también una herramienta fácil de utilizar, la versatilidad del mismo hacen de él un gran laberinto donde tantas opciones acaban por despistar.

### **5.3 Problemas encontrados**

Entre los problemas que he encontrado he de resaltar el aprendizaje del uso del programa de diseño de sistemas WinMOd2.4, también fue la primera parte del proyecto, y al principio siempre es más difícil, debido a que la visión del proyecto completo aún no la tenía en la cabeza, además el resto del proyecto depende de la veracidad de este modelo diseñado respecto al sistema real, de manera que fue un punto cuello de botella importante en el desarrollo del trabajo, no podría avanzar sin terminar este paso. Después de terminar el primer modelo, me percaté de la necesidad de volver a empezar, las dependencias internas que tuve que programar para que fuera parecido al sistema real habían hecho del modelo algo poco coherente y difícil de manejar. Después de esto, hice otro modelo más coherente con una estructura más clara, aunque este tampoco sería el modelo final. Después de estar trabajando cerca de un mes con el programa fue cuando realmente estaba en disposición de diseñar un modelo fácil de entender, simplificado y coherente con el sistema real. Tras borrar todo lo que había hecho anteriormente me dispuse a comenzar de nuevo para, finalmente hacer un modelo con el que me quedara satisfecho.

El desarrollo del control con Codesys fue algo más rápido, ya sabía realmente qué era lo que quería, y si bien aún no había trabajado con Codesys en esos momentos, mi experiencia con software de control hizo que el aprendizaje con esta herramienta fuera algo mucho más ameno que en el caso de WinMod. El Software Codesys es visualmente atractivo y muy fácil de comprender, no tuve muchos problemas para resolver los atascos en los que a menudo me hallé.

Lo más problemático con respecto a Step 7 fue la configuración de la conexión entre el PLC y el sistema de control. También llevándome un par de semanas, debido también a que el profesor no estaba en la facultad, y no podía resolver mis dudas. El programa STEP 7 también es muy visual y, realmente fácil de usar, pero también es fácil perderse entre el universo de opciones que presenta.

## 5.4 Trabajo futuro

La herramienta diseñada es perfectamente operativa en cualquiera de los puestos del laboratorio de automática. En este momento, 30 alumnos, pueden trabajar, al mismo tiempo con un sistema de tanques virtual que se corresponde con el real. Anteriormente trabajan al mismo tiempo 1 sólo grupo de 2 o 3 alumnos. Ahora, cuando acaban sus programas y han acabado de depurarlos, pueden ir a probarlos al sistema real. La herramienta todavía no tiene acceso telemático a través de internet. La proyección de la herramienta a una plataforma virtual, requiere un volumen de trabajo que no dió tiempo a asumir, y hasta ahora, ningún alumno se ha encargado de esta tarea.

El panel de control manual también está por terminar, se queda pendiente también la misión de configurarlo utilizando mi herramienta, y comprobada luego que, efectivamente, funciona en el sistema de tanques real.

Queda también planteado como trabajo futuro, la inclusión de esta herramienta virtual en un portal telemático completo que será el laboratorio de automática de la universidad de Konstanz. Es un proyecto ambicioso pero se sigue trabajando en ello.

## 5.5 Diagrama de Gantt

<i>diagrama de Gantt</i>	
PROYECTO	Proyecto virtualización
UNIDAD DE TIEMPO	DIAS
FECHA DE INICIO	01/10/2008

ACTIVIDAD	NOMBRE	DURACIÓN	ACTIVIDAD PRECEDENTE	INICIO	FINALIZACIÓN
A	Estudio	10		01/10/2008	10/10/2008
B	Hardware	10	A	11/10/2008	20/10/2008
C	Programas	15	B	21/10/2008	04/11/2008
D	CAD Iemen	40	C	05/11/2008	14/12/2008
E	CAD design	20	D	15/12/2008	03/01/2009
F	Modelo_1	30	C	05/11/2008	04/12/2008
G	Modelo_2	40	F	05/12/2008	13/01/2009
H	Configurar	15	G	14/01/2009	28/01/2009
I	Control_1	10	H	29/01/2009	07/02/2009
J	Control_2	10	I	08/02/2009	17/02/2009
K	Redactar_1	40	I	08/02/2009	19/03/2009
L	Present.	5	K	20/03/2009	24/03/2009
M	Traducción	40	K	20/03/2009	28/04/2009
N					

**Estudio:** comprende la fase de documentación y estudio de los objetivos para buscar la solución más eficiente

**Hardware:** comprensión del hardware a utilizar, conexionado de elementos y protocolos de comunicación

**Programas:** trabajo con el software, al ser todo software nuevo, esta etapa fue de pruebas para hacerme con el interfaz.

**CAD\_lernen:** aprendizaje de diseño en 3D básico, muchas horas dedicadas.

**CAD\_design:** diseño de la visualización 3D

**Modelo\_1:** diseño del primer modelo, fue desechado cuando descubrí una solución más simplificada. Al desconocer el software que estaba utilizando, no tenía todas las skills necesarias para encontrar las soluciones que mejor se ajustan a cada tarea

**Modelo\_2:** diseño de la segunda versión, más versátil de modificar, y mucho más accesible de comprensión.

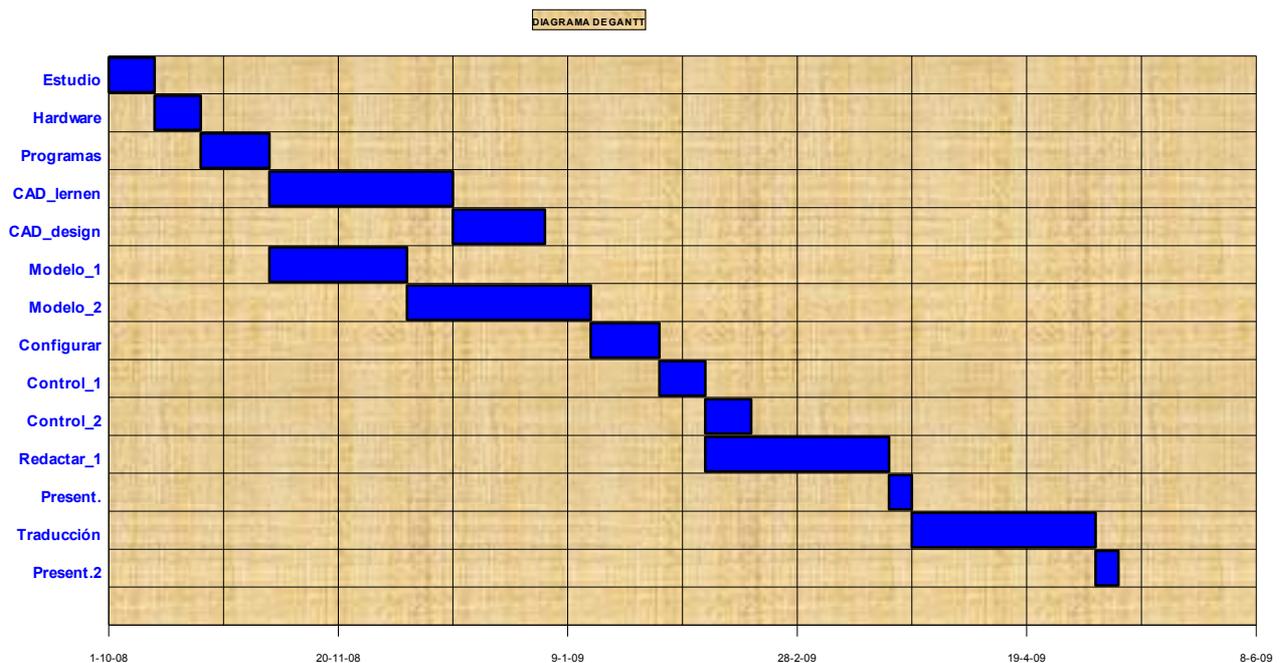
**Configurar:** esta etapa es la de configuración del proyecto del modelo, para que se comunicaran la representación en 3D y el modelo diseñado. Debido a mi inexperiencia y a la escasa documentación en este tema, dediqué muchas horas de prueba y error para comprender cómo funcionaba.

**Control\_1:** Diseño del controlador software, con codeSys2.3 además de una representación propia del programa.

**Control\_2:** Diseño del controlador Step7 y configuración de la comunicación.

**Redactar:** Fase en que redacté el proyecto escrito

**Traducción:** El proyecto fue escrito inicialmente en inglés, tras presentarlo en la facultad de ingeniería de konstanz, lo he traducido y éste es el resultado.



## 8. Bibliografía

- [1] *“Implantación virtual de un sistema de supervisión y coordinación para el control de procesos”*  
autor: Jesús Calderón-Vielma  
web site: <http://www.ing.ula.ve/~jesusc/Page5.html>
- [2] *“Manual de formación para soluciones generales en automatización Totally Integrated Automation (T I A)” MÓDULO A4 Programación de la CPU 315-2 DP*  
autor: SIEMENS  
anexo en *Apéndice B*
- [3] *“Profibus protocol manual”*  
autor: Maguire WSB Blender MLAN Gateway  
anexo en CD en nodo raíz *../documentacion/Profibus\_Protocol.pdf*
- [4] *“WinMOD-Tutorial Vers. 1.0”*  
autor: O. Kümmel / P. Malinowski  
anexo en CD en nodo raíz *../documentacion/WinMOD\_Tutorial.pdf*
- [5] *“Start-Up-Projekt - WinMOD-I/O-Simulation”*  
autor: Mewes & Partners  
anexo en CD en nodo raíz  
*../documentación/WinMOD\_Handbuch\_Systemsoftware\_Anlage2\_de.pdf*
- [6] *“Automatisierung der Tankanlage mit Den Messgeräten der Firma Endress+Hauser”*  
Proyecto fin de carrera de Abdulrahman Abdo en Konstanz, 17.10.2008  
autor: Abdo Abdulrahman  
anexo en CD en nodo raíz: *../documentacion/praxissemesterbericht.pdf*
- [7] *“Siemens Industry Automation and Drive Technologie”*  
autor: Siemens  
web site: <http://support.automation.siemens.com>
- [8] *“Realtime 3D Computer Graphics - Virtual Reality”*  
Autor: Marc Erich Latoschik - AI & VR Lab; Artificial Intelligence Group  
University of Bielefeld  
anexo en CD en nodo raíz: *../documentacion14.RT3DCGVR-X3D-VRML.pdf*
- [9] *“Iniciación al Software de programación Codesys2.3”*  
autor: CODESYS  
anexo en CD en nodo raíz: *../documentacion/infoPLC\_net\_CodeSys\_Iniciacion.pdf*

- [10] “*Flux Studio 2.1 als VRML-Tools für WinMOD-3DView*”  
autor: HTWG-Konstanz University of Applied Sciences  
anexo en CD ../documentacion/SchnellEinstieg-FluxStudio.pdf
- [11] “*Apuntes PLC Simatic Siemens S7*”  
autor: A. Roldán  
anexo en CD ../documentacion/ApuntesS7-200\_KOP.pdf
- [12] *Guia-STEP7*  
autor: Siemens  
anexo en CD ../documentacion/Guia-STEP7.pdf
- [13] “*VRML Primer and Tutorial*”  
Autor: Universidad de Ginebra  
web-site: <http://tecfu.unige.ch/welcome.html>
- [14] “*VRML/X3D (Virtual Reality Modeling Language) unter Unix/Linux*”  
autor: J. Scheurich  
Web-site: <http://vrml.cip.ica.uni-stuttgart.de/linuxtag/einf.html>
- [15] “*Dynamische virtuelle Welten mit VRML 2.0*”, 1. Aufl., 1997  
autor: Hans-Lothar Hase
- [16] “*Diseño de Laboratorios Virtuales y/o remotos. Un caso práctico*”  
autor: J.M. Andújar Márquez, T.J. Mateo Sanguino  
Revista iberoamericana de automática e informática industrial. ISSN: 1697-7912. Vol 7,  
Núm. 1, Enero 2010, pp 64.72  
anexo en CD en nodo raíz ../docuemntacion/Laboratorio virtual.pdf