

DESARROLLO E IMPLANTACIÓN DE PLATAFORMA ROBÓTICA MÓVIL EN ENTORNO DISTRIBUIDO



Autor: Francisco J. Arjonilla García

Tutor: Ricardo Sanz Bravo

Co-tutor: Carlos Hernández Corbato

INDUSTRIALES
ETSII | UPM

1. Introducción
2. Análisis de requisitos
3. Hardware
4. Entorno de desarrollo
5. Software
6. Pruebas y validación
7. Conclusiones



1. Introducción

1) Marco del proyecto

2) Objetivos

2. Análisis de requisitos
3. Hardware
4. Entorno de desarrollo
5. Software
6. Pruebas y validación
7. Conclusiones



Proyecto ASys del Laboratorio de Sistemas Autónomos (ASLab)

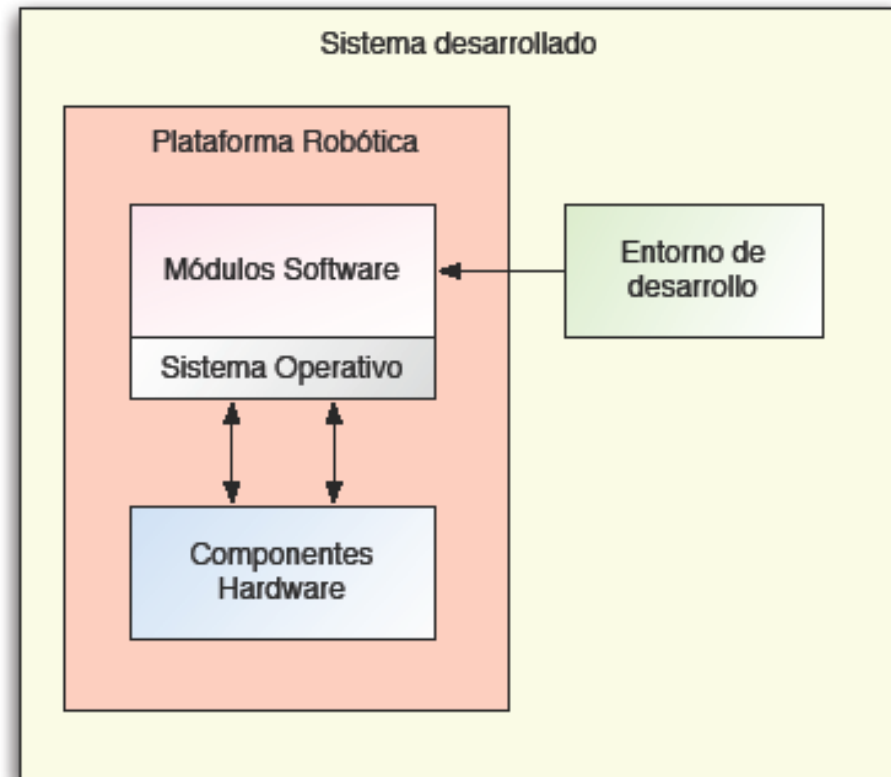
- Tecnología de sistemas autónomos
- Temas:
 - Ontologías
 - Control basado en modelos
 - Arquitecturas de control integradas
 - Procesamiento **distribuido** y en **tiempo real**
 - Sistemas **conscientes**
 - Implicaciones filosóficas

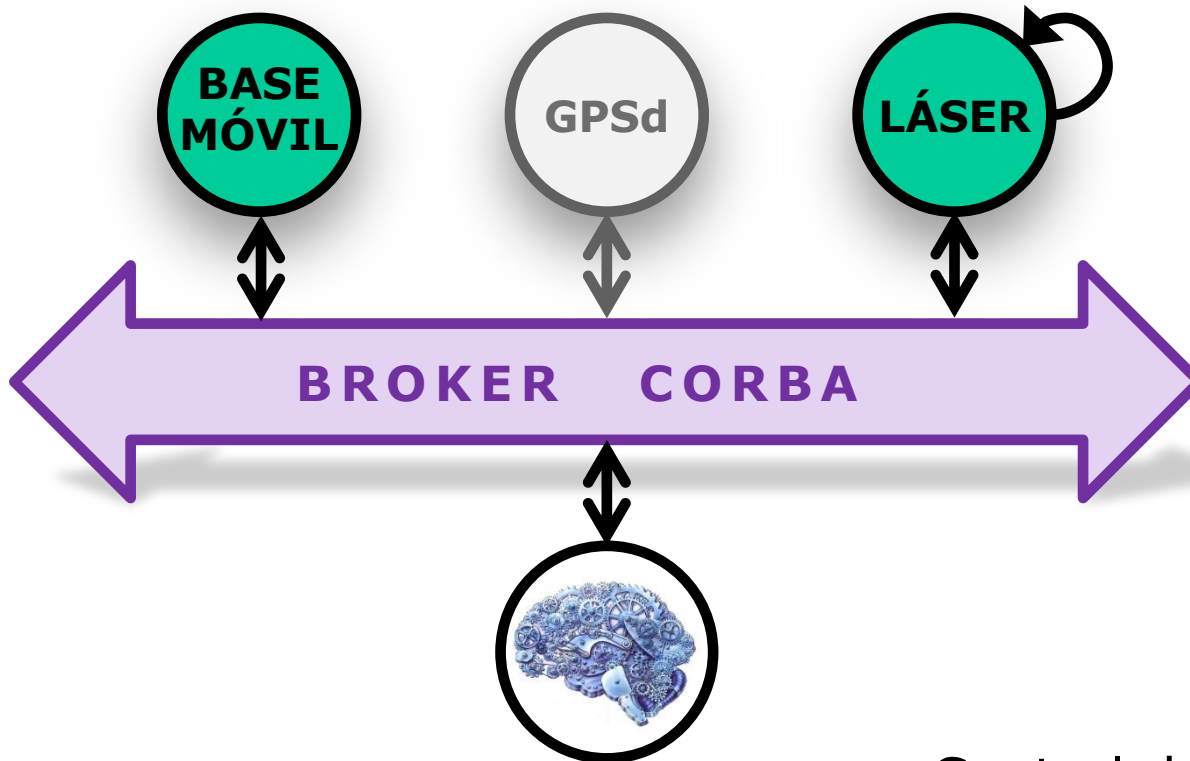


Objetivo principal:

→ Implantar una plataforma de pruebas sobre la que sea posible probar en entornos reales los sistemas de control desarrollados dentro del marco del proyecto ASYS.

Este proyecto proporciona un *cuerpo* donde probar *mentes* artificiales

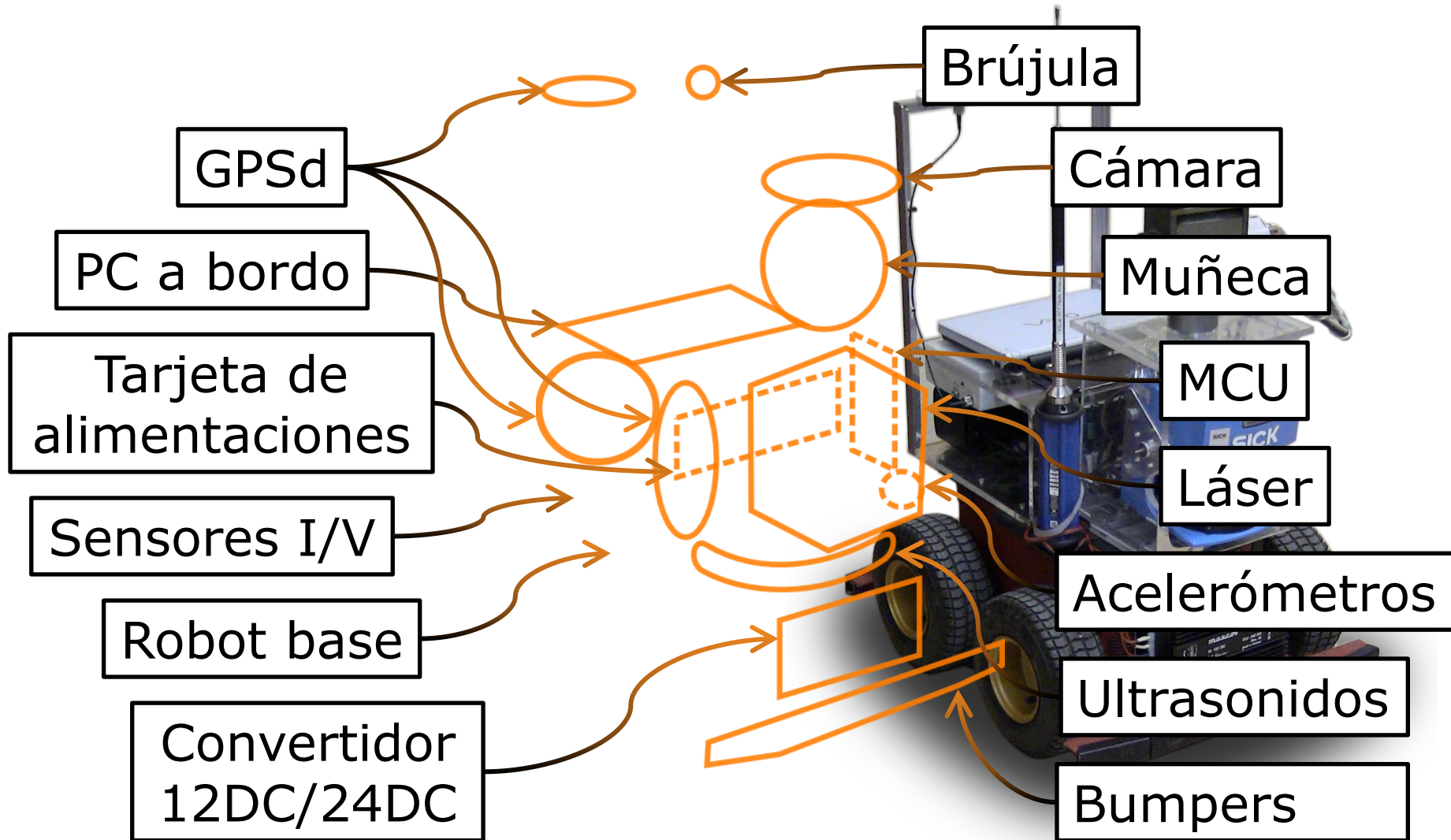




- Control de dispositivos
- Disponibilidad no garantizada
- Capacidad de auto-observación

1. Introducción
- 2. Análisis de requisitos**
3. Hardware
4. Entorno de desarrollo
5. Software
6. Pruebas y validación
7. Conclusiones





Requisitos	Sistema de partida		
	Adquirido	Integ. HW	SW
Robot base	●	●	Falta integ.
Cámara estereoscópica	●	●	Falta integ.
Control de cámara (muñeca)	●	●	Iniciado
Medida de carga y consumo	●	●	●
Tarjeta de adquisición de datos	Falta brújula	Parcial	Parcial
Sensor láser	●	●	●
Apagado manual y automático	●	●	●
Receptor GPSd	●	●	●
Ordenador de a bordo	●	●	●

Leyenda:

- Disponible en sistema de partida.
- Pendiente por desarrollar y/o implantar

Requisitos	Sistema de partida
Sistema operativo en tiempo real	Obsoleto
Configuración y arranque de módulos SW	●
Restablecimiento de sistemas tras fallos	●
Uso de estándares y optimización del desarrollo	●
Uso de CORBA	●
Funcionamiento robusto e integrado	●
Documentación: Manual de usuario y del desarrollador	●

R 0.0.0

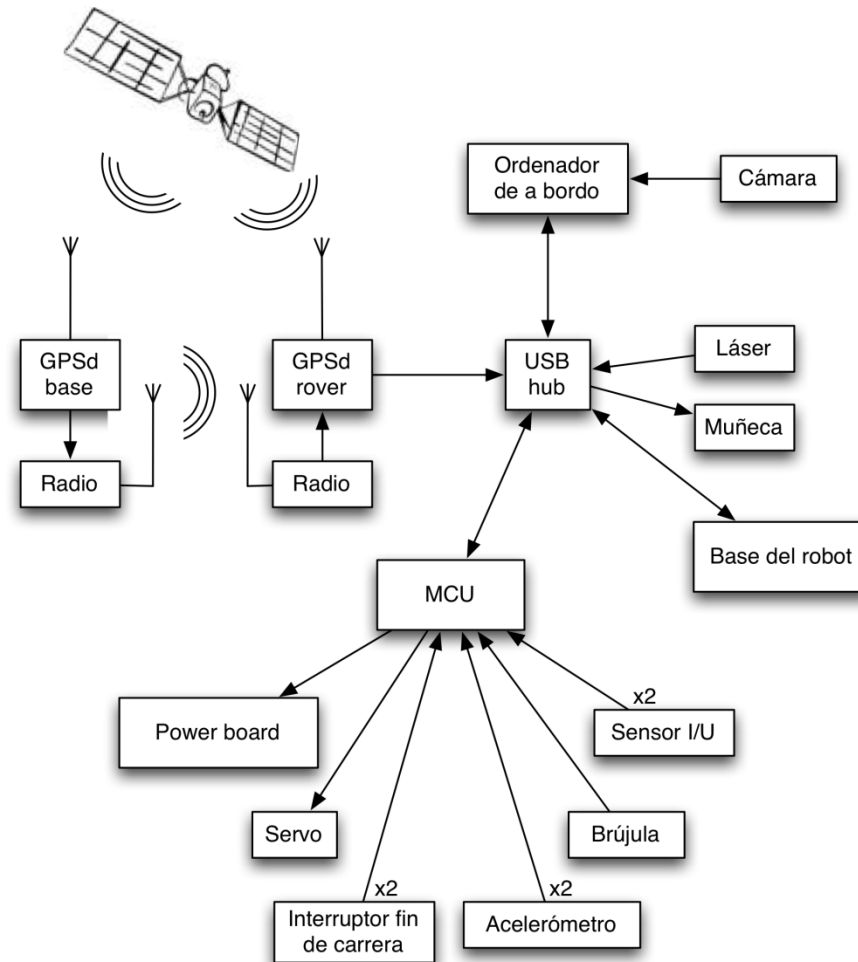
Notación utilizada para referenciar requisitos

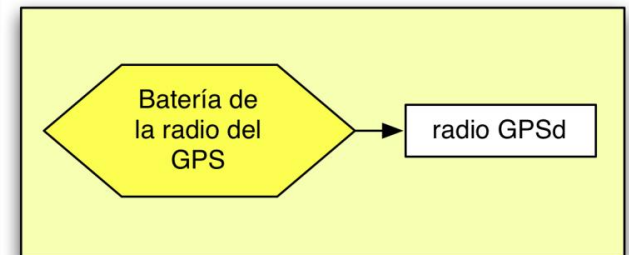
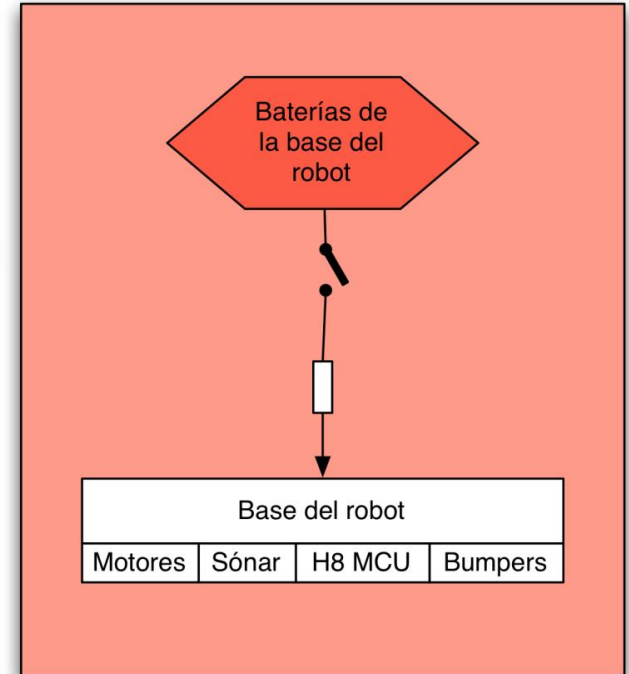
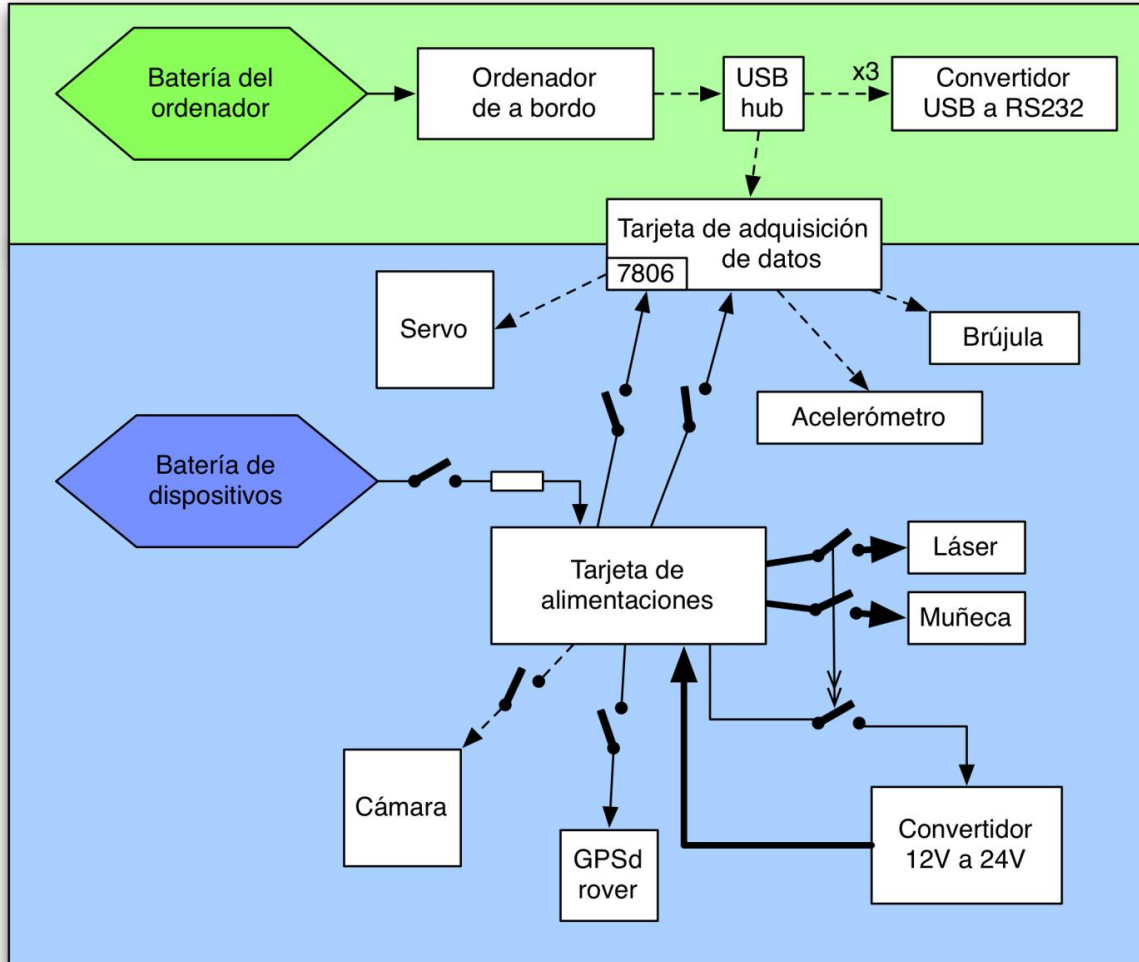
Leyenda:

- Disponible en sistema de partida.
- Pendiente por desarrollar y/o implantar

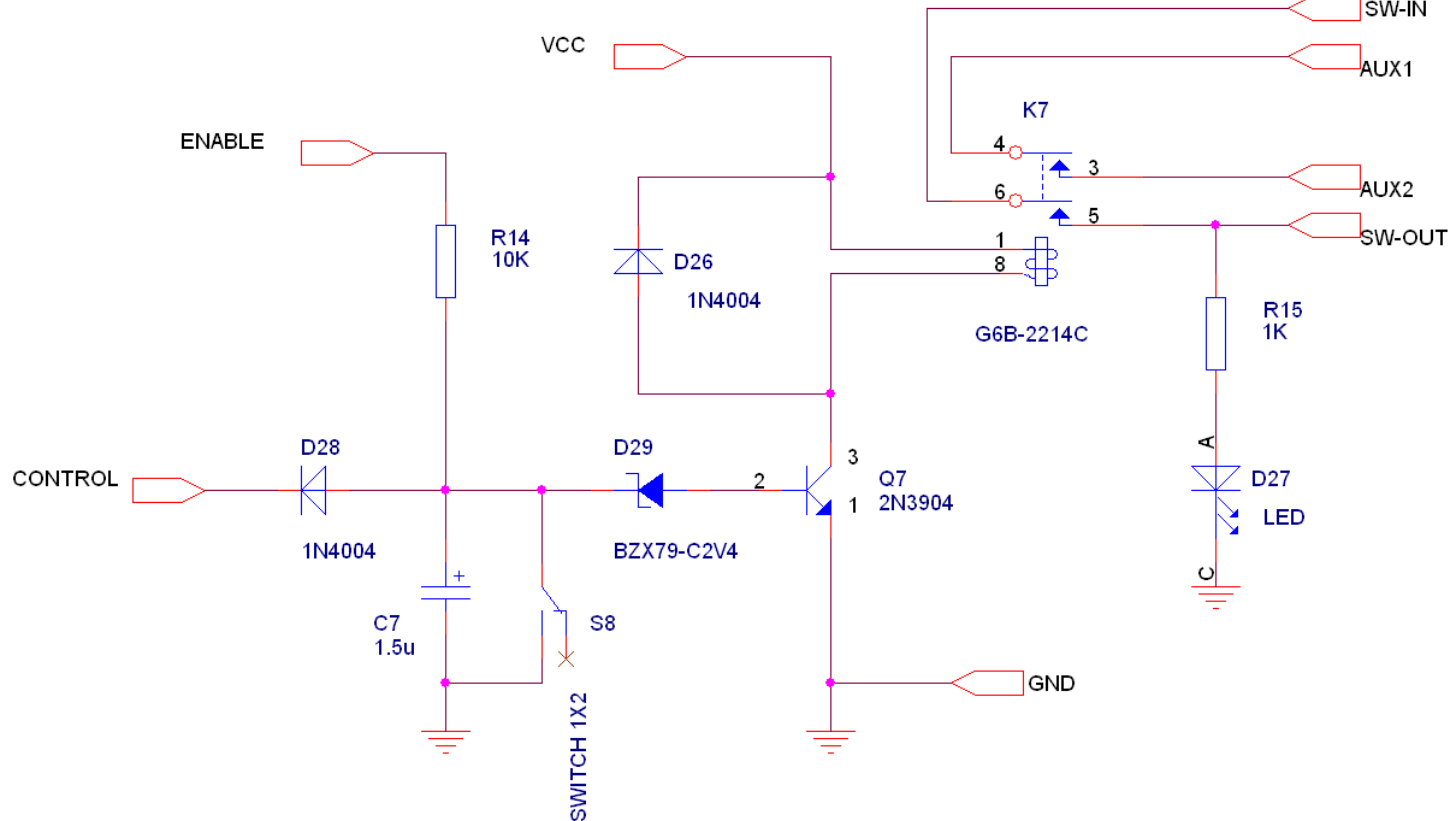
1. Introducción
2. Análisis de requisitos
- 3. Hardware**
 - 1) Cableado y estructuras**
 - 2) Tarjeta de alimentaciones**
 - 3) Sensor tensión/intensidad**
4. Entorno de desarrollo
5. Software
6. Pruebas y validación
7. Conclusiones







R 1.2.1 Los dispositivos se controlarán individualmente.

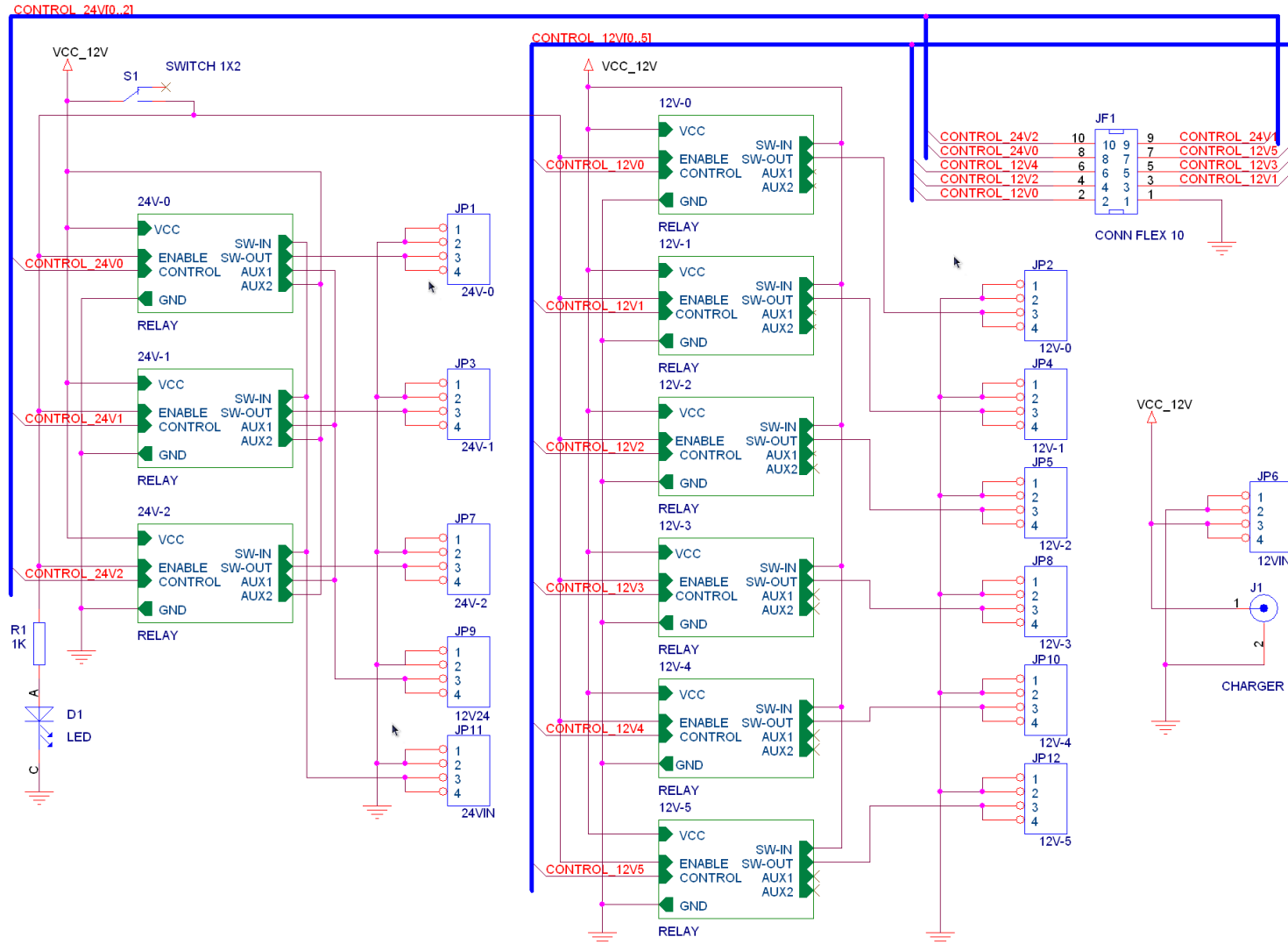


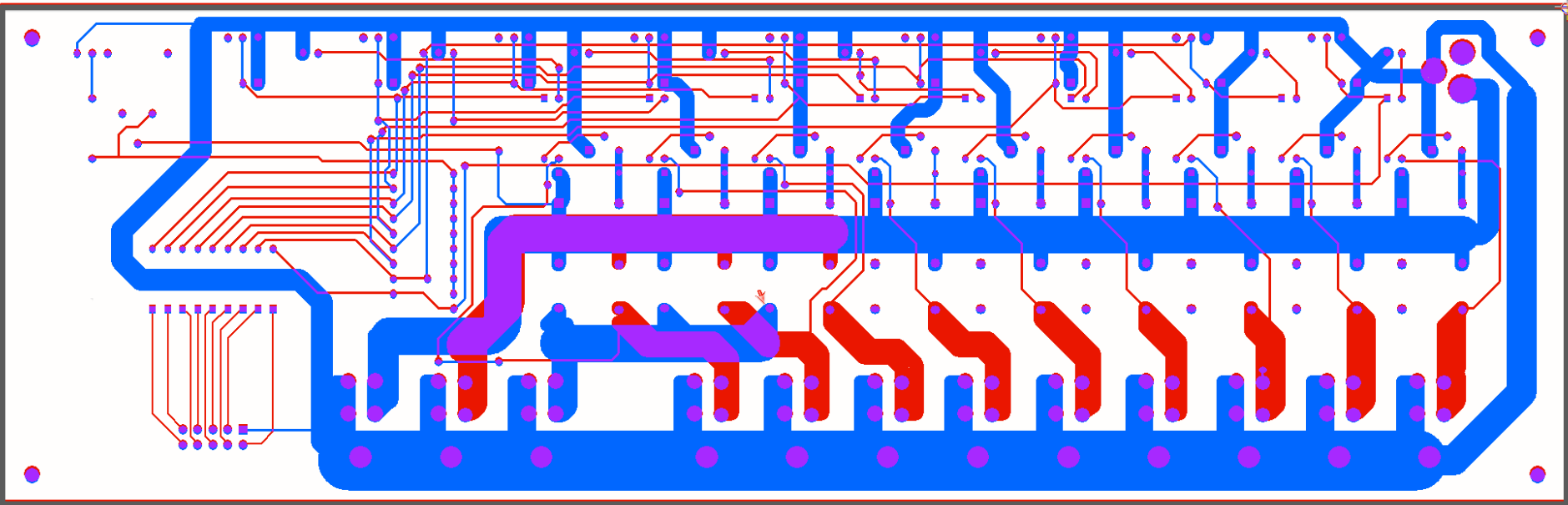
- El control de la polarización del transistor es una puerta AND cableada.
- Relé DPST con caída de tensión máxima de 0,1V.

R 1.2.2

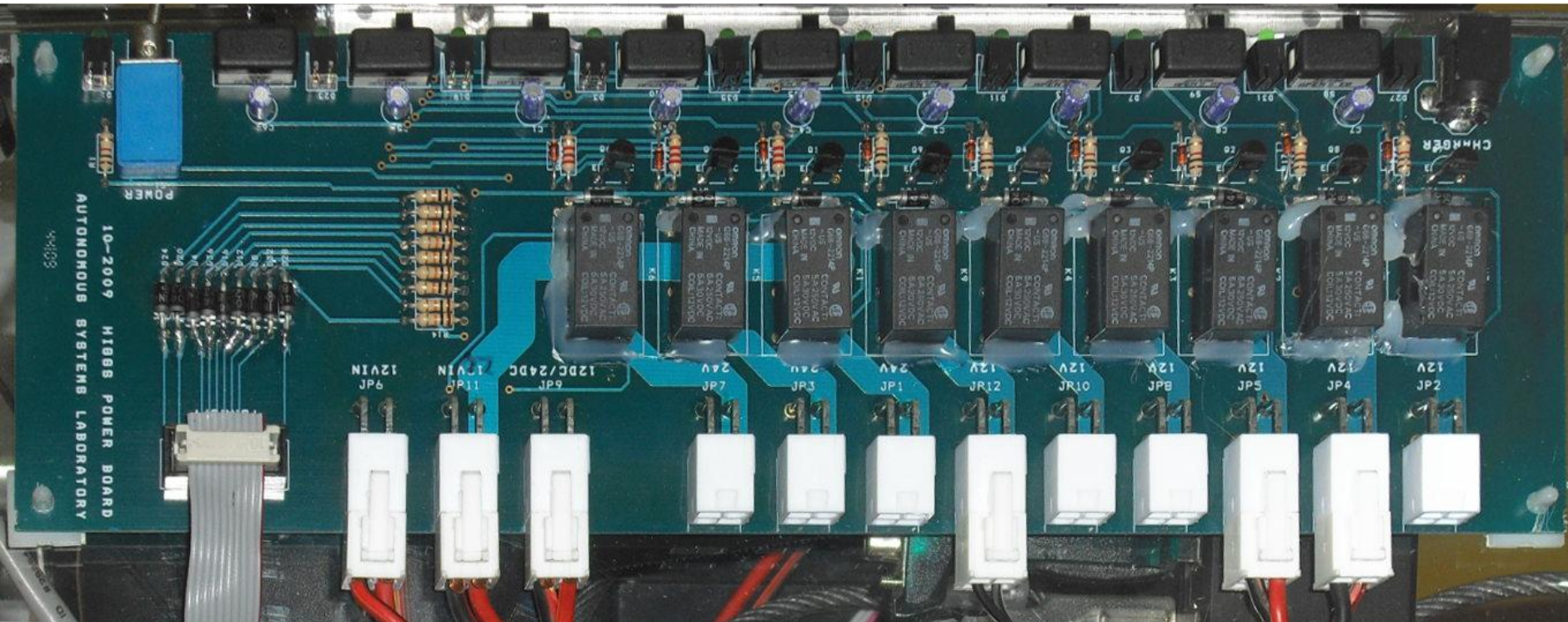
R 1.2.3

Apagado manual y automático de dispositivos.





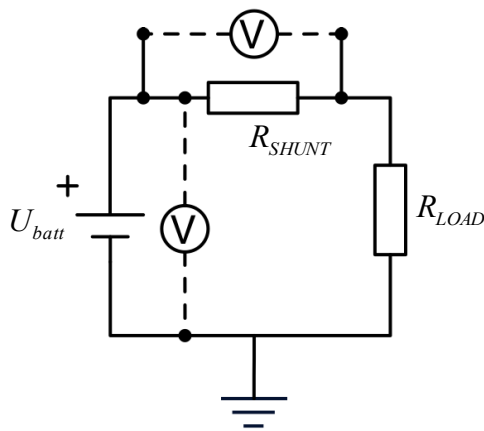
- ✓ 6 canales para dispositivos de 12V y 3 de 24V
- ✓ Apagado manual de cada dispositivo mediante interruptor
- ✓ Apagado automático mediante conexión por cable plano
- ✓ Control del convertidor DC12V/DC24V
- ✓ Posibilidad de conectar el cargador



- ✓ 6 canales para dispositivos de 12V y 3 de 24V
- ✓ Apagado manual de cada dispositivo mediante interruptor
- ✓ Apagado automático mediante conexión por cable plano
- ✓ Control del convertidor DC12V/DC24V
- ✓ Posibilidad de conectar el cargador

R 1.2

Desarrollo de la tarjeta de alimentaciones



$$\begin{cases} \frac{U^+ - U_e}{R_1} = \frac{U_e}{R_2} \\ \frac{U^- - U_e}{R_3} = \frac{U_e - U_0}{R_4} + \frac{U_e}{R_5} \end{cases}$$

$$U_0 = U^+ \cdot \frac{R_2 R_4}{R_1 + R_2} \cdot \left(\frac{1}{R_3} + \frac{1}{R_4} + \frac{1}{R_5} \right) - U^- \cdot \frac{R_4}{R_3}$$

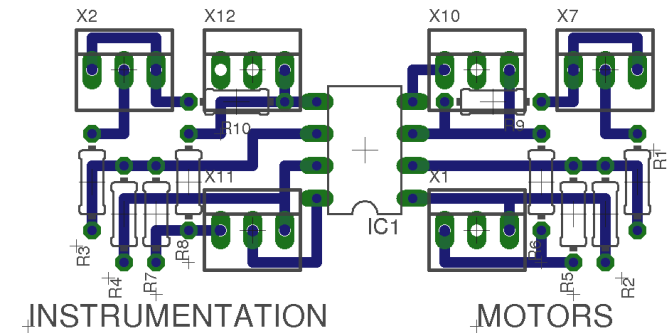
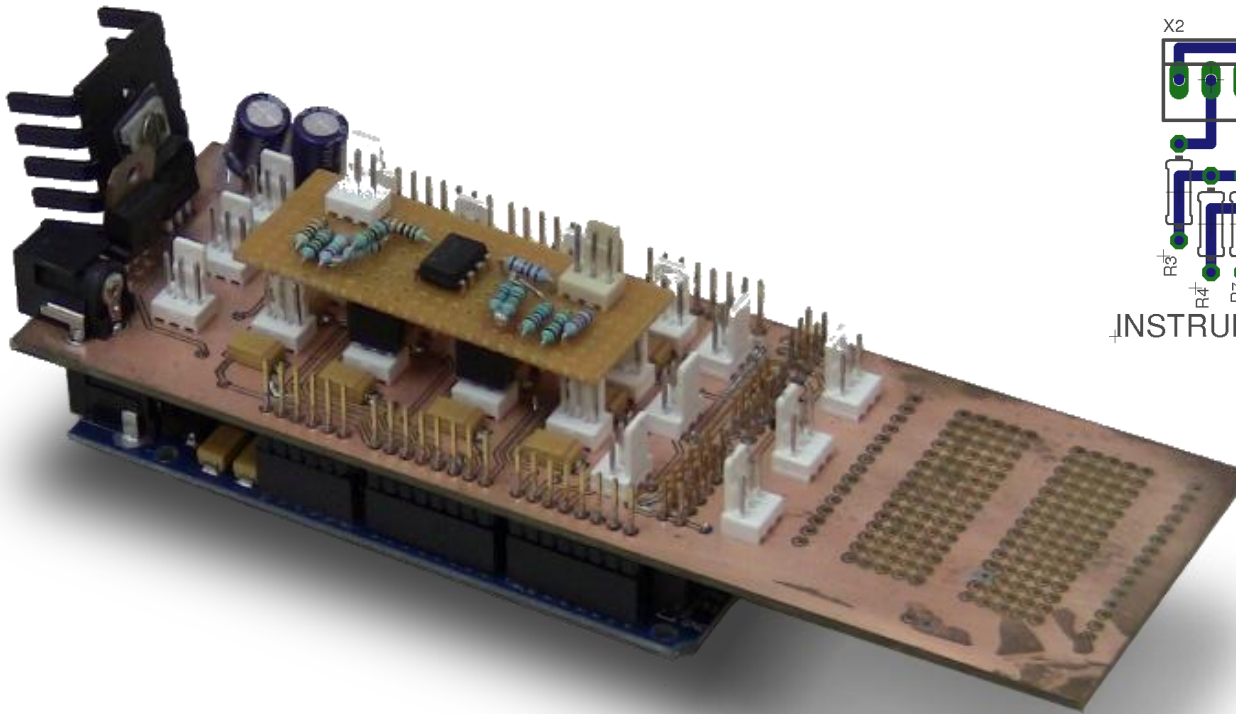
$$U_0 = K \cdot (U^+ - U^-) \Rightarrow \frac{R_1}{R_2} = \frac{R_3}{R_4 \parallel R_5}$$

Sustituyendo valores,
con una tolerancia de
resistencia de un 1%:

$$R'_2 = 1,01 \cdot R_2 \Rightarrow U'_0 = 1,32 \cdot U_0$$

R 1.2

Integración de sensores de tensión e intensidad



- ✓ Realización artesanal
- ✓ Corrección lineal de factores por software
- ✓ Montado sobre la tarjeta de adquisición de datos

1. Introducción
2. Análisis de requisitos
3. Hardware
- 4. Entorno de desarrollo**
 - 1) Sistema operativo**
 - 2) Compilación y librerías**
5. Software
6. Pruebas y validación
7. Conclusiones



¿Porqué Ubuntu con Linux-RTAI?

- Por ser de código abierto
- **R 3** Estándar POSIX
- Está en constante desarrollo
- Por su rapidez de instalación
- Homogeneidad en ASLab



- ✓ Ubuntu Desktop 10.04 LTS (Long Term Support)
- ✓ Configuración para funcionamiento autónomo
- ✓ Configuración y compilación del núcleo RTAI 2.6.32
- ✓ Utilización del sistema Upstart para configuración y ejecución de los módulos. **R 1.1.2**

R 1.1.1

Sistema operativo en tiempo real.

Sistema de compilación basado en CMake.

- Generación de proyectos en IDEs.
- Extendido para soportar interfaces CORBA (IDLs).

Gestión de versiones con Subversion.

- Traspaso de código fuente a este sistema.
- Transformación de la estructura de directorios.

Creación de librerías

- Acelera la creación de nuevos módulos SW.
- Facilita el mantenimiento de los ya existentes.

```
#include "implementationC.h"
#include "CosNamingC.h"
#include "../..../lib/CORBA_utils.h"
int main(int argc, char * argv[]) {
    CORBA_BEGIN_CLIENT(argc, argv);
    CORBA_GET_REFERENCE(module::impl_t,
                        impl, "IMPL");

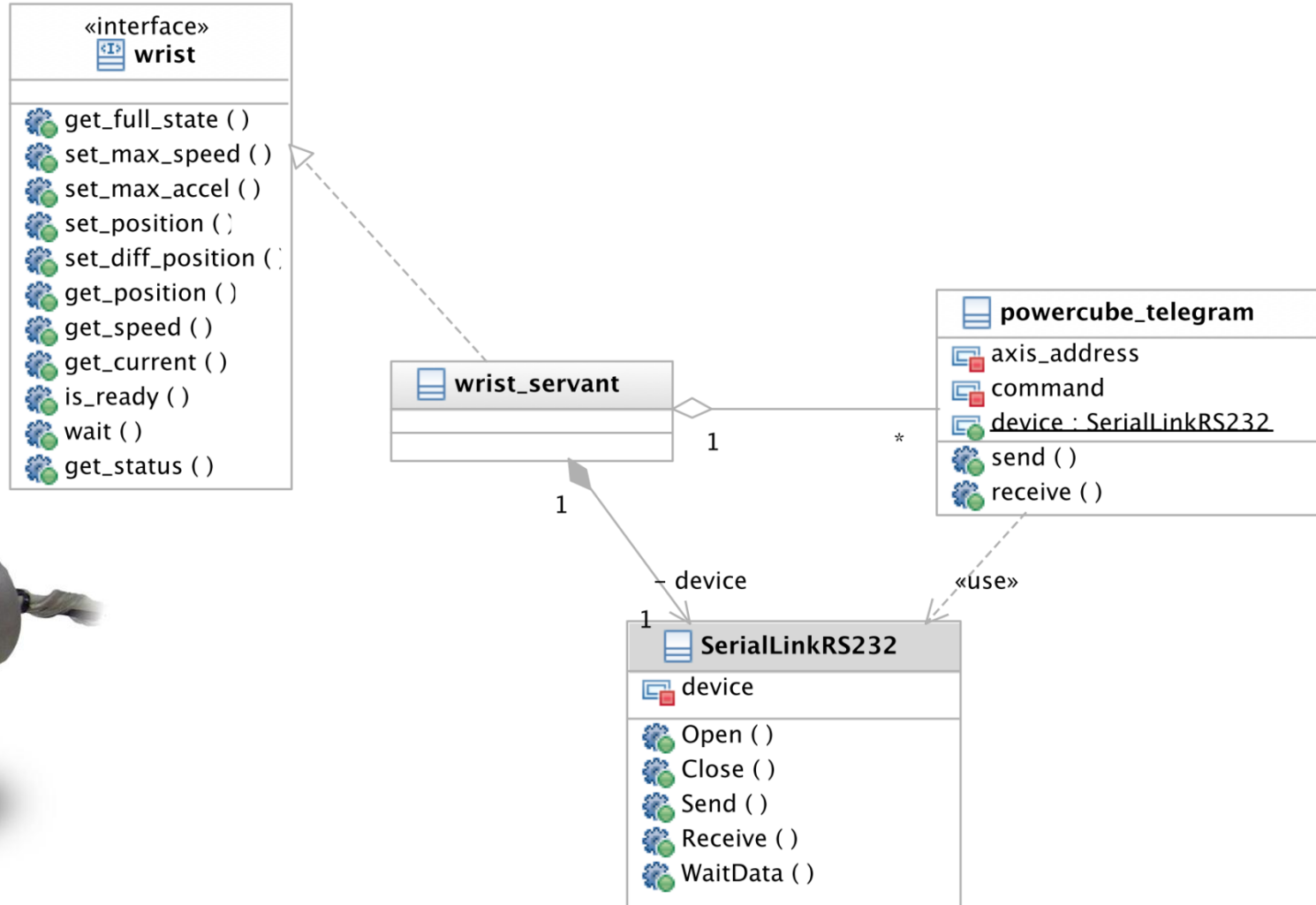
    impl->hacer_algo();
    CORBA_END_CLIENT;
    return 0;
}
```

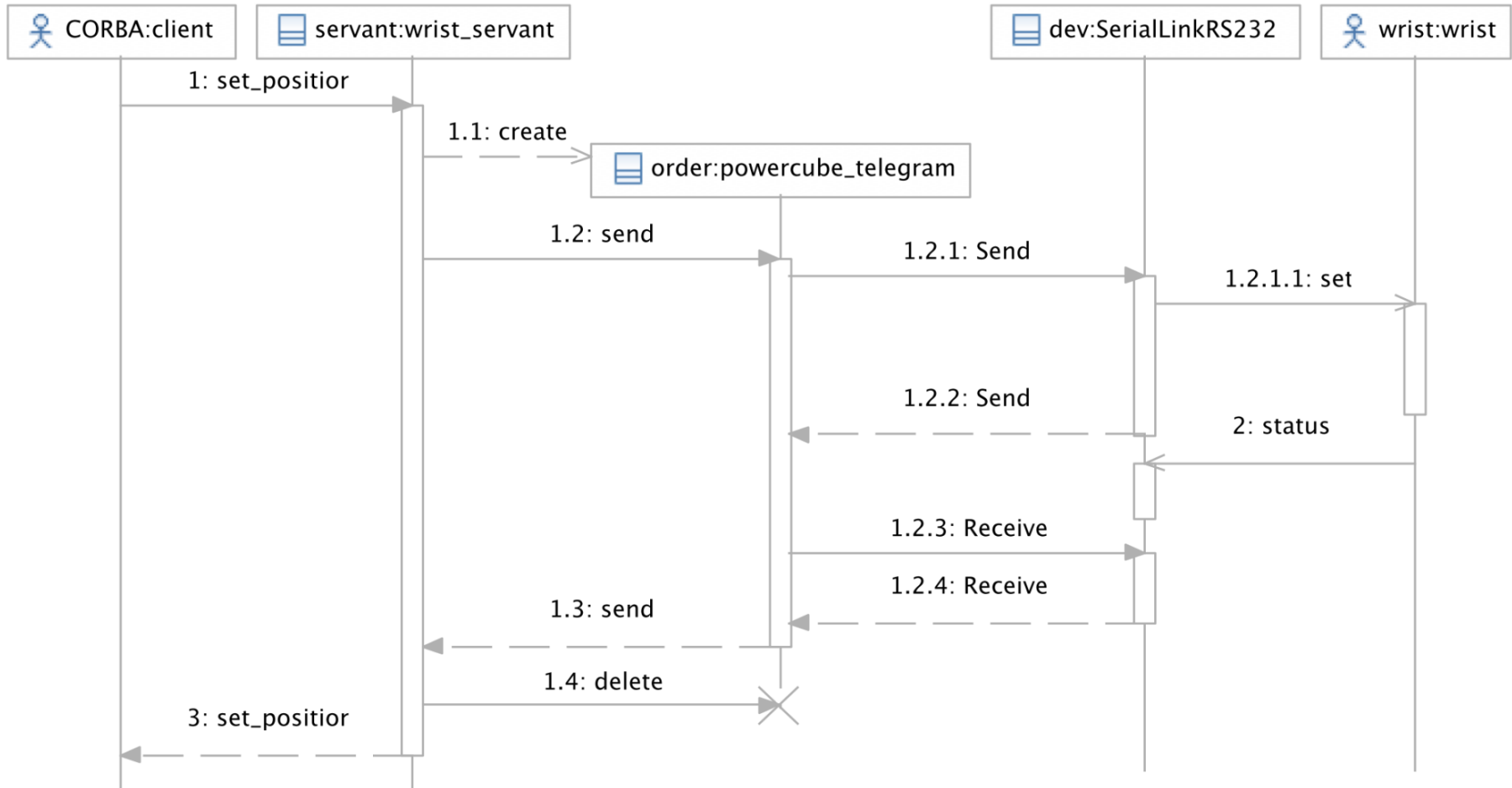
R 3

Uso de estándares y optimización en el desarrollo.

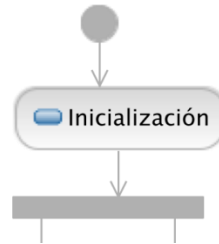
1. Introducción
2. Análisis de requisitos
3. Hardware
4. Entorno de desarrollo
- 5. Software**
 - 1) Controlador de la muñeca**
 - 2) Mecanismo de autorecuperación**
6. Pruebas y validación
7. Conclusiones

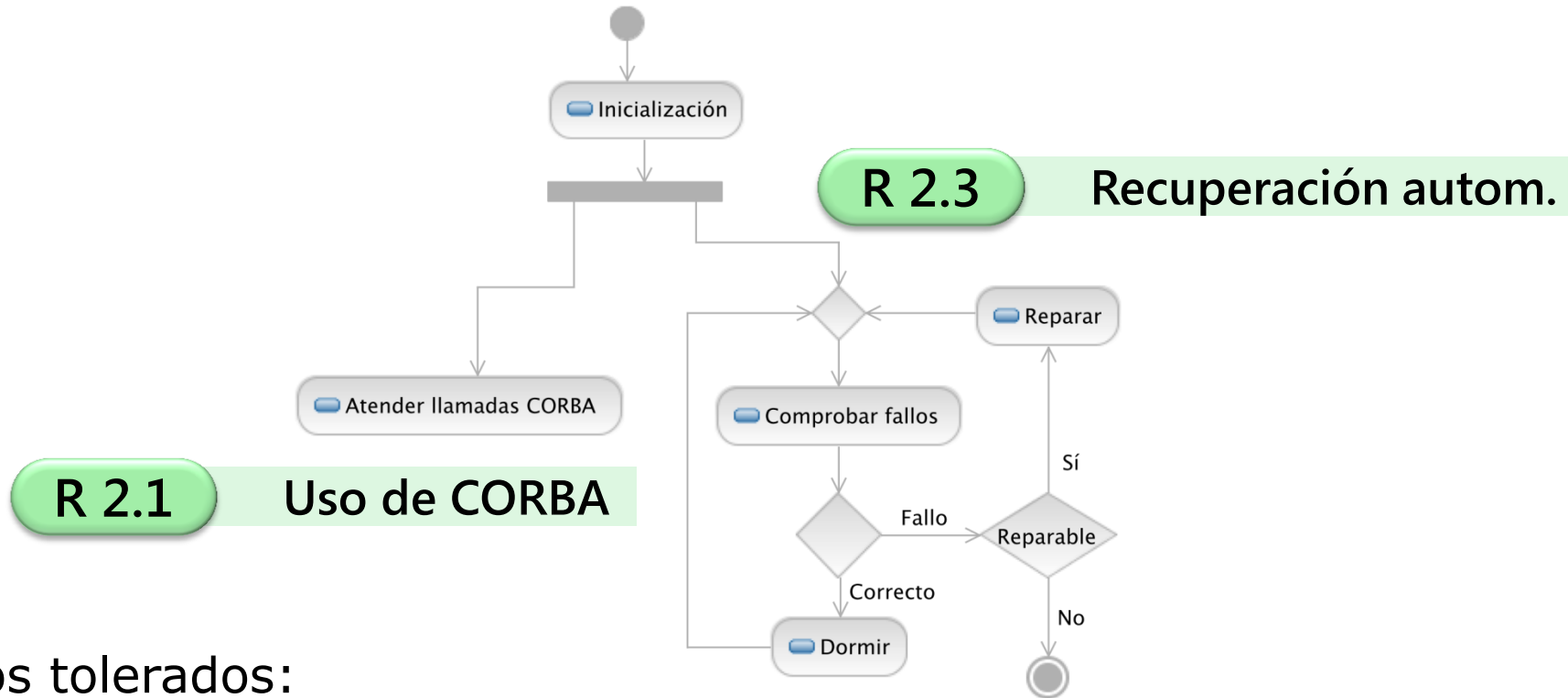






R 2.5.1.2 Integración de la muñeca.





R 2.1 Uso de CORBA

Fallos tolerados:

- Sobrecorrientes
- Tensión de alimentación fuera de rango
- Dispositivo apagado o desconectado

R 2.2 Disponibilidad de objetos CORBA cuando el dispositivo lo esté

Características del módulo SW

- ✓ Más de 3000 líneas de código C++
- ✓ Implementación completa del protocolo
- ✓ Tolerancia a fallos
- ✓ Extensible a más grados de libertad



Módulos SW similares desarrollados y/o mejorados:

- ✓ GPS
- ✓ Arduino (Tarjeta de adquisición de datos)
- ✓ Láser
- ✓ Cámara estereoscópica
- ✓ Pioneer (Base móvil)

R 2

Integración de sensores y actuadores

1. Introducción
2. Análisis de requisitos
3. Hardware
4. Entorno de desarrollo
5. Software
- 6. Pruebas y validación**
7. Conclusiones



Validación de requisitos

- ✓ Clientes de prueba para cada dispositivo.
- ✓ Pruebas de recuperación de funcionalidad ante fallos

Pruebas de integración:

- ✓ Interiores: Base, láser, brújula, cámara, sensor i/v.
- ✓ Exteriores: Base, brújula, láser, GPSd, acelerómetros.



1. Introducción
2. Análisis de requisitos
3. Hardware
4. Entorno de desarrollo
5. Software
6. Pruebas y validación
- 7. Conclusiones**



Requisitos	Sistema de partida		
	Adquirido	Integ. HW	SW
Robot base	●	●	●
Cámara estereoscópica	●	●	▲
Control de cámara (muñeca)	●	●	●
Medida de carga y consumo	●	●	●
Tarjeta de adquisición de datos	●	●	●
Sensor láser	●	●	●
Apagado manual y automático	●	●	●
Receptor GPSd	●	●	●
Ordenador de a bordo	●	●	●

Leyenda:



● Disponible al finalizar el proyecto



▲ Algunas funcionalidades no se han implementado

Requisitos	Sistema de partida
Sistema operativo en tiempo real	●
Configuración y arranque de módulos SW	●
Restablecimiento de sistemas tras fallos	●
Uso de estándares y optimización del desarrollo	●
Uso de CORBA	●
Funcionamiento robusto e integrado	▲
Documentación: Manual de usuario y del desarrollador	●

Leyenda:



● Disponible al finalizar el proyecto

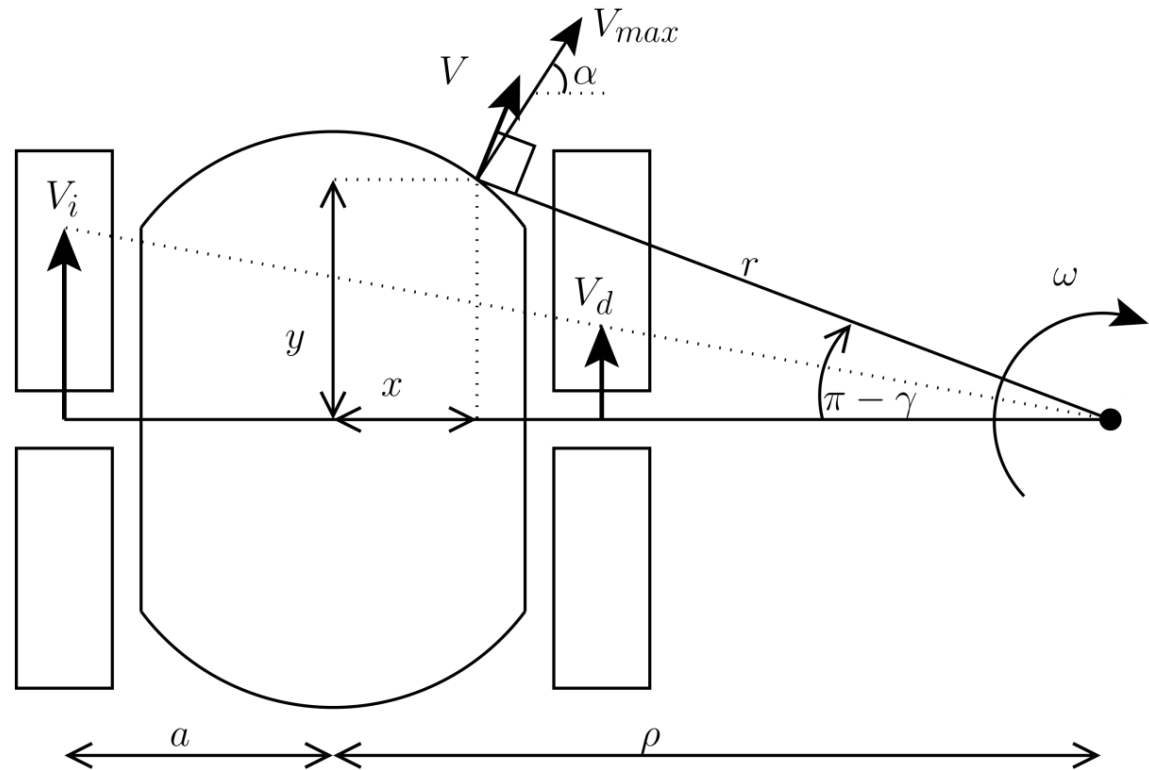


▲ Algunas funcionalidades no se han implementado

$$\begin{cases} V_i = -(\rho + a) \cdot \omega \\ V_d = -(\rho - a) \cdot \omega \end{cases}$$

$$\begin{cases} \omega = \frac{V_d - V_i}{2 \cdot a} \\ \rho = a \cdot \frac{V_i + V_d}{V_i - V_d} \end{cases}$$

$$\begin{cases} r = \sqrt{y^2 + (\rho - x)^2} \\ \pi - \gamma = \tan^{-1} \frac{y}{\rho - x} \end{cases}$$

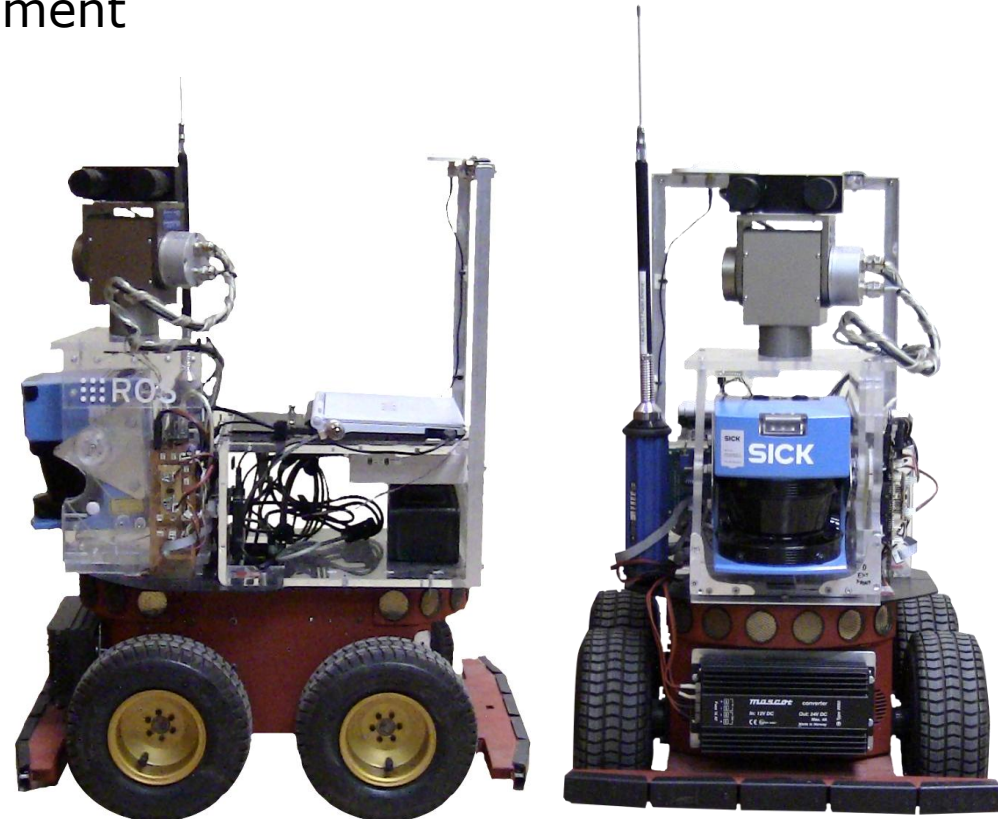


Con $\theta = \gamma - \alpha + \frac{\pi}{2}$, si $V \cdot \cos \theta > V_{max} \Rightarrow V = \frac{V_{max}}{\cos \theta}$

- ✓ Se han cumplido los objetivos del proyecto
 - Módulos funcionales (muñeca, GPS, láser, etc.)
 - Mecanismos de self-awareness
 - Mecanismos de self-management

- ✓ Se han realizado desarrollos adicionales no contemplados en los requisitos:
 - Librería executive
 - Anticipador de colisiones

- ✓ Colaboración con otros grupos dentro del departamento.



DESARROLLO E IMPLANTACIÓN DE PLATAFORMA ROBÓTICA MÓVIL EN ENTORNO DISTRIBUIDO



Autor: Francisco J. Arjonilla García

Tutor: Ricardo Sanz Bravo

Co-tutor: Carlos Hernández Corbato

INDUSTRIALES
ETSII | UPM

ESTRUCTURA DE DESCOMPOSICIÓN DEL PROYECTO
DESARROLLO E IMPLANTACIÓN DE PLATAFORMA
ROBÓTICA MÓVIL EN ENTORNO DISTRIBUIDO

