

# Robot Móvil de Arquitectura Abierta (RoMAA-II) Desarrollo y aplicación de los lenguajes C y C++



Gonzalo F. Pérez Paina

Centro de Investigación en Informática para la Ingeniería  
Universidad Tecnológica Nacional, FRC

<http://cii.frc.utn.edu.ar>



Informática II – 2016

¿Por qué? y ¿Para qué?

¿Por qué? y ¿Para qué?

- Robot experimental (robótica móvil y visión por computadoras)

¿Por qué? y ¿Para qué?

- Robot experimental (robótica móvil y visión por computadoras)
- Flexibilidad en montaje de sensores y actuadores

# Robot Móvil de Arquitectura Abierta, RoMAA

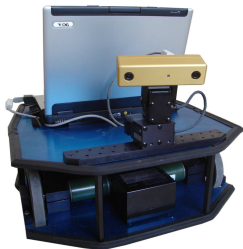
## Prototipo y evolución

- JAR2008, JAR2010, CASE2011, CASE2012, ARGENCON2014
- Proyecto PID-UTN 2010-2013 (Director: Dr. Araguás)

# Robot Móvil de Arquitectura Abierta, RoMAA

## Prototipo y evolución

- JAR2008, JAR2010, CASE2011, CASE2012, ARGENCON2014
- Proyecto PID-UTN 2010-2013 (Director: Dr. Araguás)



RoMAA



RoMAA-II

## Robot Móvil de Arquitectura Abierta, RoMAA



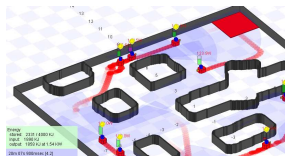
# RoMAA, “Arquitectura Abierta”



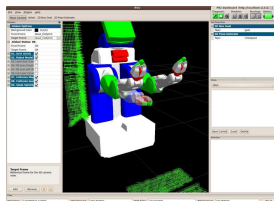
# RoMAA, “Arquitectura Abierta”

## FOSS - Free and Open Source Software

- Player/Stage
- OpenCV (Open Source Computer Vision)
- ROS (Robot Operating System), Gazebo



Player Stage



ROS



# RoMAA, “Arquitectura Abierta”

## OSHW - Open Source Hardware

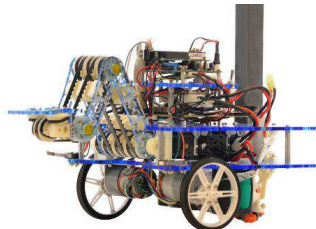
- Arduino, Impresoras 3D, etc.
- Robots: humanoides, manipuladores, UAV (drones), etc.



e-puck



Verter



OHMM

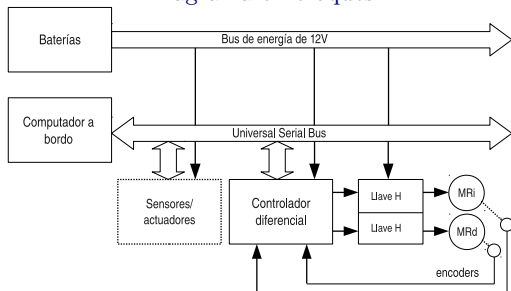
## RoMAA-II... más detalles



## RoMAA-II... más detalles



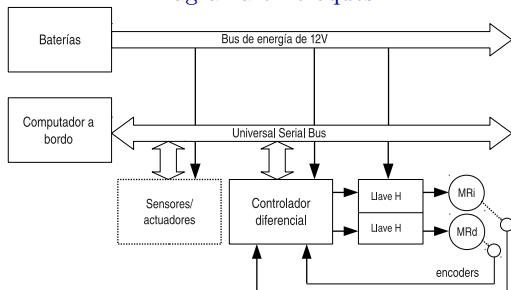
### Diagrama en bloques



## RoMAA-II... más detalles



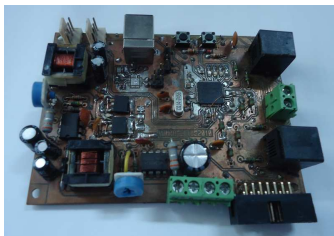
### Diagrama en bloques



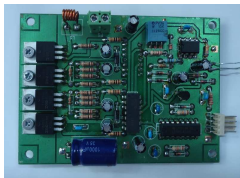
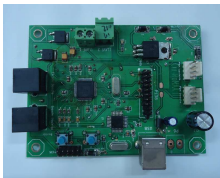
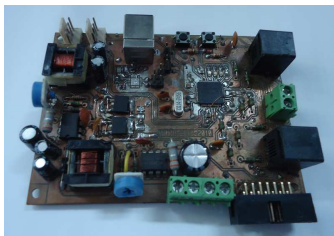
Algunas características:

- Motores de tracción: 2000r.p.m., 60W
- Peso aproximado: 30Kg
- Velocidad máxima: 2m/s

## Electrónica de control de bajo nivel



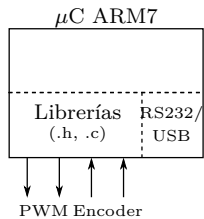
# Electrónica de control de bajo nivel



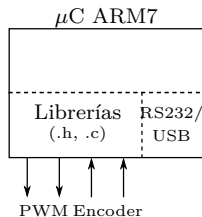
# Desarrollo y aplicación de los lenguajes C y C++



# Desarrollo y aplicación de los lenguajes C y C++

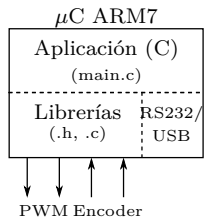


# Desarrollo y aplicación de los lenguajes C y C++



Módulos {  
- TIMER, GPIO, UART, IRQ, PWM  
- Encoder, PID, Comun.

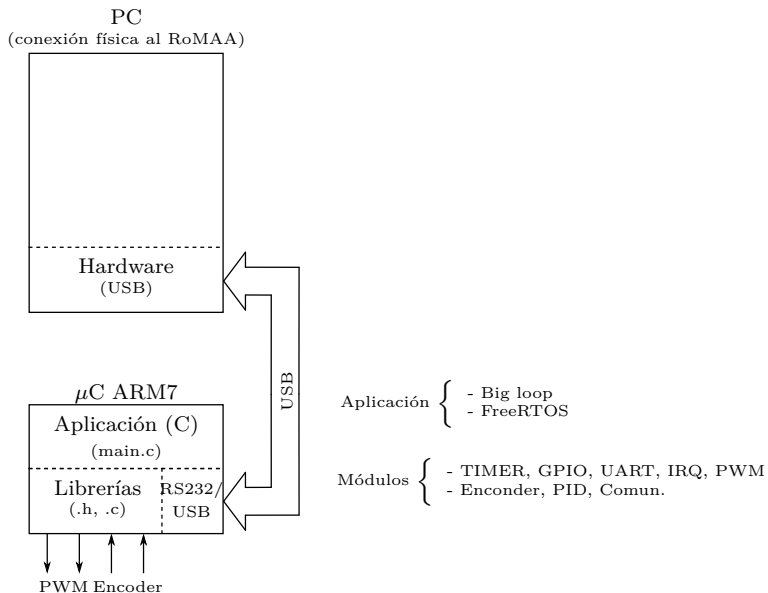
# Desarrollo y aplicación de los lenguajes C y C++



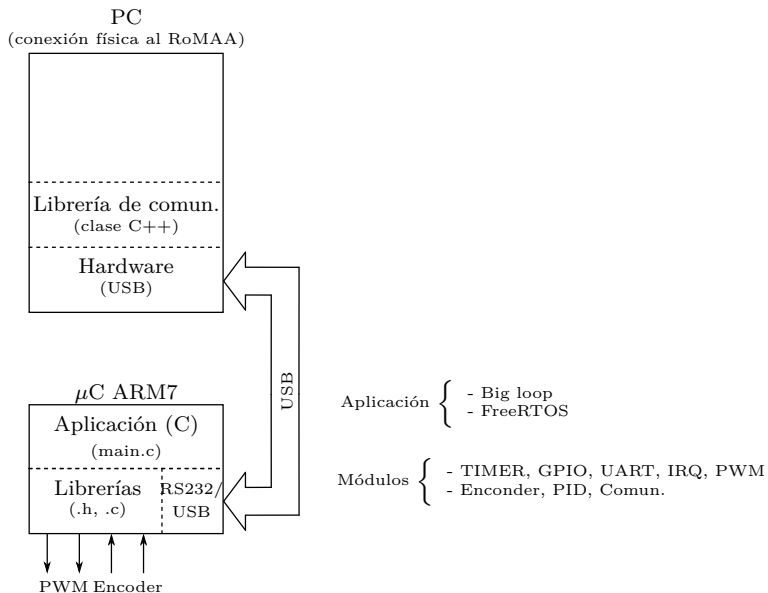
Aplicación {  
- Big loop  
- FreeRTOS

Módulos {  
- TIMER, GPIO, UART, IRQ, PWM  
- Encoder, PID, Comun.

# Desarrollo y aplicación de los lenguajes C y C++

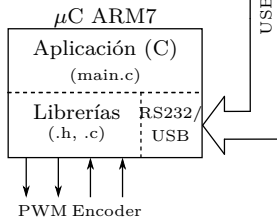
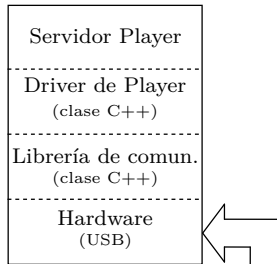


# Desarrollo y aplicación de los lenguajes C y C++



# Desarrollo y aplicación de los lenguajes C y C++

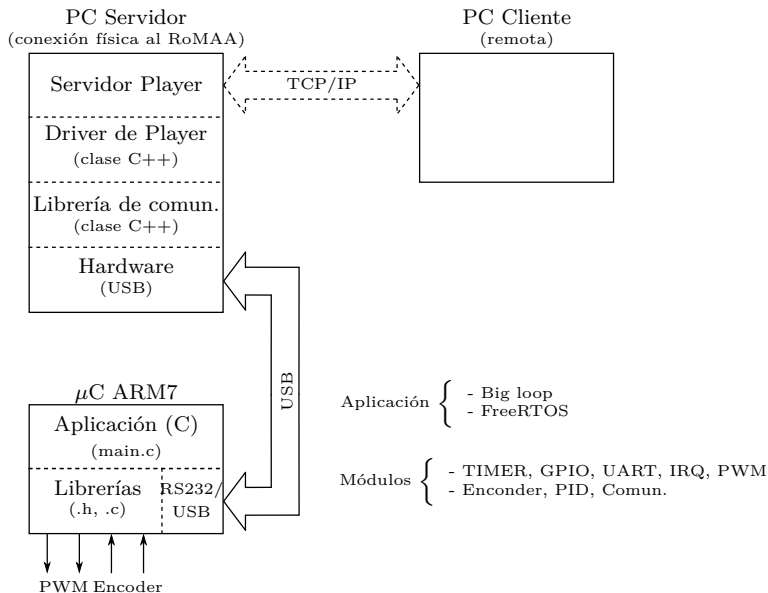
PC Servidor  
(conexión física al RoMAA)



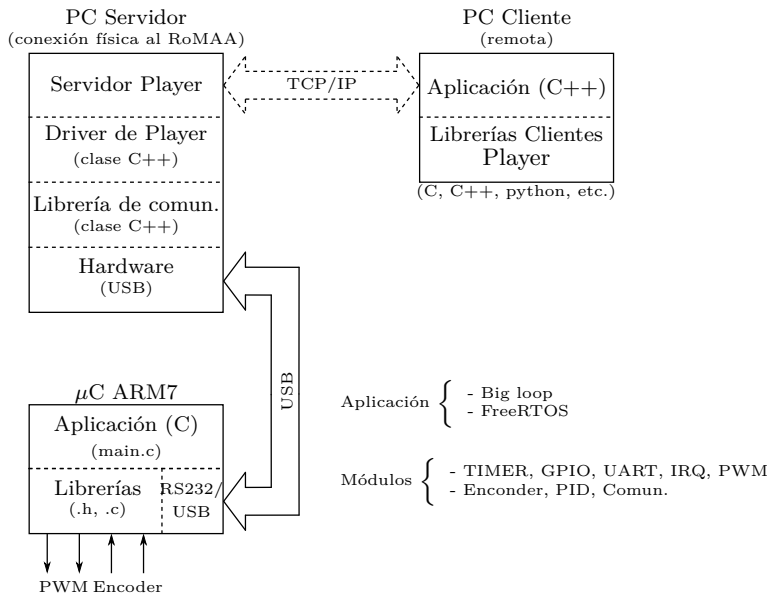
Aplicación {  
- Big loop  
- FreeRTOS

Módulos {  
- TIMER, GPIO, UART, IRQ, PWM  
- Encoder, PID, Comun.

# Desarrollo y aplicación de los lenguajes C y C++

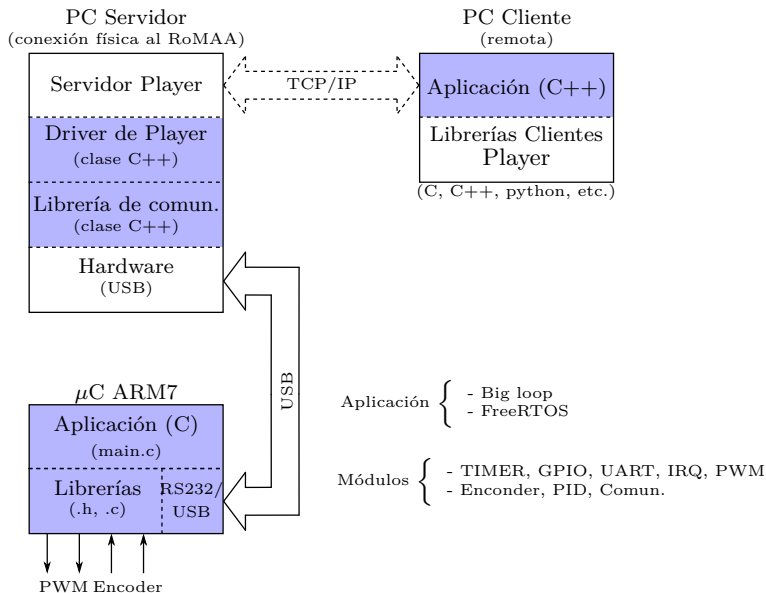


# Desarrollo y aplicación de los lenguajes C y C++





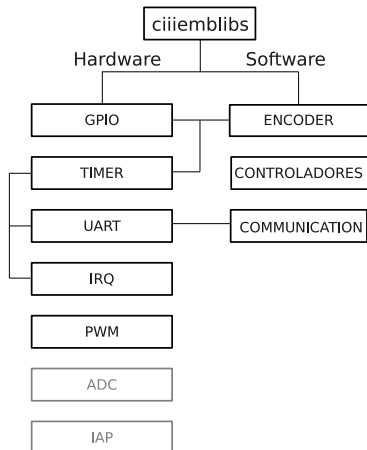
# Desarrollo y aplicación de los lenguajes C y C++



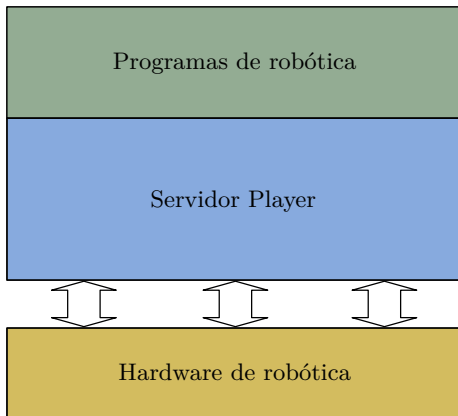
# Librerías del Firmware

- Desarrolladas en lenguaje ANSI-C
- Separadas en módulos (.c y .h)
- Divididas en dos grandes grupos
  - ▶ módulos para periféricos o hardware
  - ▶ módulos especiales o de software
- Estilo de nombre unificado

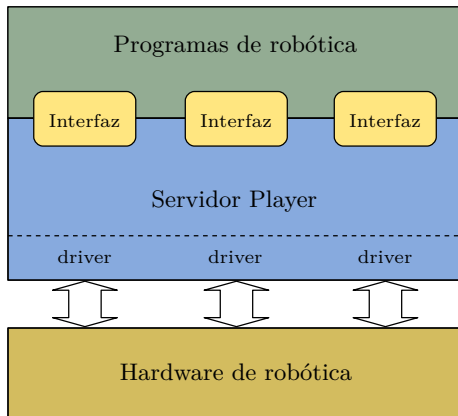
Ejemplos: `gpio_init()`, `pwm_init()`,  
`com_init()`, `timer_init()`



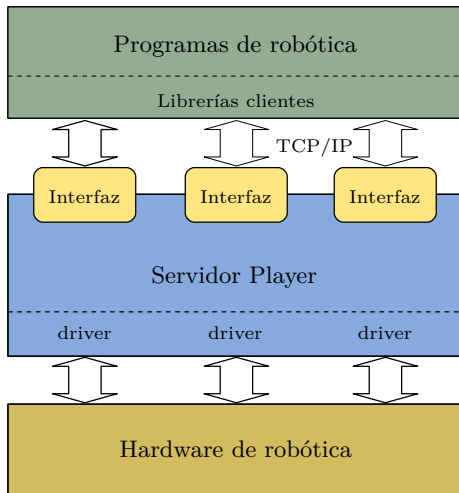
## Entorno de desarrollo de robótica Player/Stage



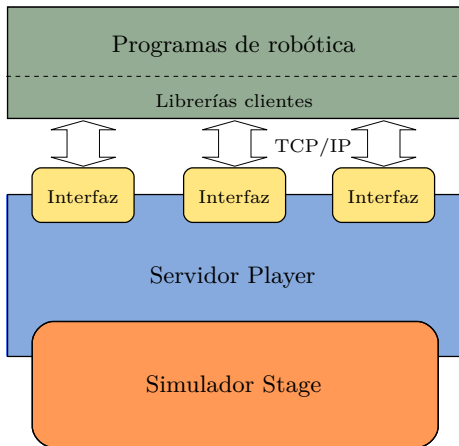
## Entorno de desarrollo de robótica Player/Stage



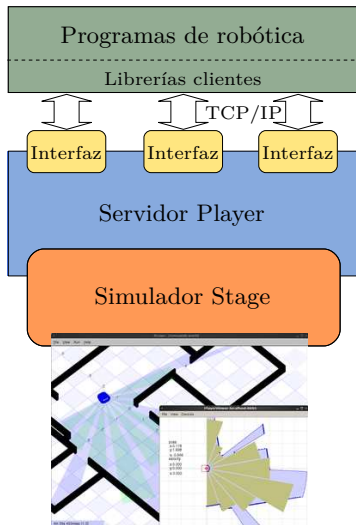
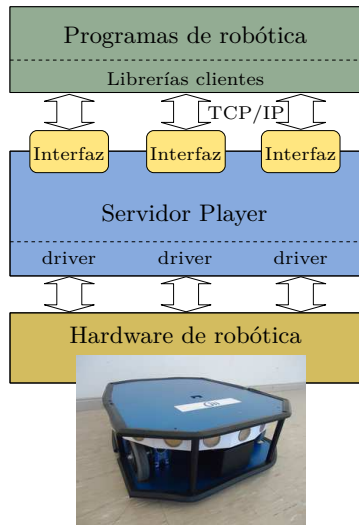
## Entorno de desarrollo de robótica Player/Stage



## Entorno de desarrollo de robótica Player/Stage



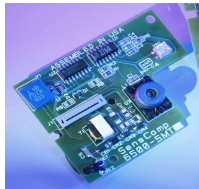
# Entorno de desarrollo de robótica Player/Stage



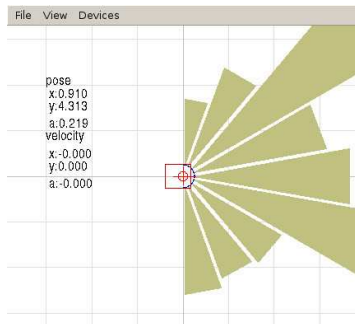
# Reutilización de Software - Módulo de sensores



## Reutilización de Software - Módulo de sensores



- Clase de comunicación en C++:  
Implementa métodos análogos a c/u de los comandos de bajo nivel.
- Driver para Player:  
Permite utilizar el módulo de forma abstracta en redes TCP/IP.



# Desarrollo y aplicación de los lenguajes C y C++

