

Curso de Especialización en Sistemas Embebidos

Ingeniería de Software

12va Cohorte 2020

Trabajo práctico 4. Arquitectura y Diseño

“Implementación de aprendizaje por

refuerzo en un Robot”

Alumno:

Pablo Daniel Folino

Docentes:

Esp. Ing. Alejandro Permingeat

Esp. Ing. Esteban Volentini

Índice

**Historial de cambios** 2

**1. Introducción** 3

1.1. Propósito 3

1.2. Ámbito del Sistema 3

1.3. Definiciones, Acrónimos y Abreviaturas 4

1.4. Referencias 4

1.5. Visión General del Documento 4

**2. Arquitectura** 4

2.1. Patrones 4

2.1.1 Patrón Arquitectura en capas 5

2.1.2. Patrón Capa de abstracción de hardware (HAL) 5 2.1.3. Patrón Control ambiental 5

2.2. Componentes 6

2.3. Interfaces 6

**3. Diseño preliminar** 7

1. Módulo acelerómetro. 7
2. Módulo control de motores. 7
3. Módulo de comunicaciones WIFI. 7
4. Módulo de posicionamiento (GPS). 7
5. Módulo sensor de ultrasonido. 8
6. Módulo de administrador de energía. 8
7. Módulo principal. 8

# Historial de cambios

|  |  |  |
| --- | --- | --- |
| **Fecha** | **Versión** | **Cambio** |
| 10/08/2020 | 1.0 | Versión 1 de Arquitectura de software |
| 12/08/2020 | 2.0 | Se agrega descripción de interfaces de software |
|  |  |  |
|  |  |  |
|  |  |  |

1. Introducción

**1.1. Propósito**

1. El propósito de este documento es el proporcionar una descripción completa de la arquitectura de software y el diseño detallado para un Robot móvil al que se le aplicará algoritmos de aprendizaje por refuerzo.
2. Este documento está dirigido a desarrolladores que se ocupen de implementación, así como también a usuarios finales que deseen saber características mas detalladas del sistema robótico.

**1.2. Ámbito del Sistema**

El sistema llevará el nombre de ROBOT\_RL y formará parte del trabajo final de la Carrera de Especialización en Sistemas Embebidos (CESE) y deberá:

1. Adquirir información del entorno a través de sensores.
2. Controlar los actuadores de robot para lograr los movimientos deseados.
3. Relacionar la información obtenida para lograr una correcta navegación.
4. Permitir el control remoto del robot, por comunicación inalámbrica.

**1.3. Definiciones, Acrónimos y Abreviaciones**

|  |  |
| --- | --- |
| CIAA | Computadora Industrial Abierta Argentina |
| GPS | Sistema de Posicionamiento Global |
| GIT | Software de control de versiones diseñado por Linus Torvald |
| I/O | Entrada /Salida |
| LCD | Pantalla de cristal líquido |
| LED | Light Emitting Diode |
| N/A | No Aplica |
| PID | Control Proporcional, Integral y Derivativo |
| PWM | Modulación por ancho de pulso |
| RL | Aprendizaje por refuerzo (reinforcement learning) |
| TBD | A ser definido (to be defined) |
| TBC | A ser confirmado (to be confirmed) |
| TTL | Transistor-Transistor Logic |
| WIFI | Sistema de Conexión Inalámbrica |
| Checksum | Sistema de chequeo de información mediante sumas |
| WatchDog  **(WDT)** | Técnica para evitar que un procesador quede “colgado” |

**1.4. Referencias**

Este documento está vinculado con la especificación de requerimientos de software para el proyecto “*Implementación de aprendizaje por refuerzo en un Robot*”: [Folino\_ROBOT\_RL-ER-0001]

**1.5. Visión General del Documento**

* Este documento describe el tipo de arquitectura utilizada
* Se describen los componentes de software, sus responsabilidades e interfaces
* Se detalla también cada uno de los componentes de software.

1. Arquitectura
2. **Patrones**

Para este software se emplearán dos patrones arquitectónicos:

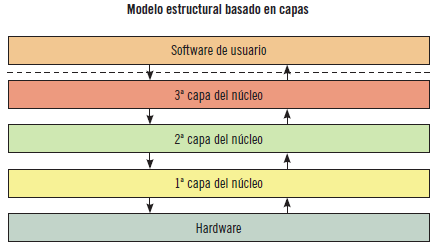
**2.1.1. Arquitectura en capas**

**2.1.2. Capa de abstracción de hardware (HAL)**

**2.1.3. Control ambiental**

1. **Patrón Arquitectura en capas**

Este patrón es utilizado cuando se desea separar la funcionalidad del software por niveles de abstracción. Cada capa proporciona servicios a la siguiente capa superior.



En este proyecto se emplearán las siguientes capas

* **Capa de Aplicación**
* **Capa de Sistema operativo**
* **Capa Abstracción de hardware (HAL)**

1. **Patrón Capa de abstracción de hardware (HAL)**

La capa de abstracción de hardware (en inglés, Hardware Abstraction Layer o HAL) funciona como interfaz entre el software y el hardware del sistema, de modo que el software no accede directamente al hardware sino que lo hace a la capa abstracta provista por la HAL.

Se utilizará como hardware la placa de desarrollo EDU-CIAA-NXP, se empleará el firmware de la CIAA versión 3.0 como capa de abstracción de hardware.

Aplicación

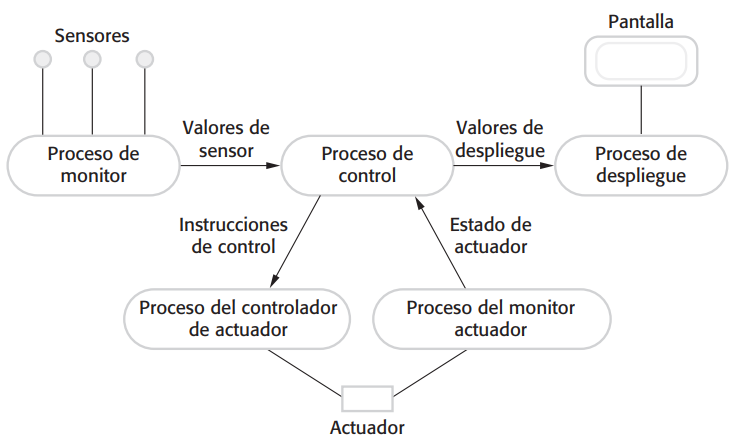
HAL(sAPI)

Hardware EDU-CIAA\_NXP

1. **Control ambiental**

Se utilizará un patrón de Control Ambiental, ya que se trata de un patrón de control general que incluye procesos de sensor y actuador. El software controla la operación del equipo, con base en estímulos del entorno del equipo.

Se empleará este patrón arquitectónico para la capa aplicación.



**2.2. Componentes**

* + 1. Cada capa de software es considerada un componente de software. Con lo cual se poseen los siguientes componentes

**2.2.1.1. Capa Abstracción de hardware (HAL)**

**2.2.1.2. Capa de Sistema operativo**

**2.2.1.3. Capa de Aplicación**

**2.2.2**. A su vez, la “*Capa de Aplicación*” estará compuesta por los siguientes componentes de

software:

1. Módulo acelerómetro.
2. Módulo control de motores.
3. Módulo de comunicaciones WIFI.
4. Módulo de posicionamiento (GPS).
5. Módulo sensor de ultrasonido.
6. Módulo de administrador de energía.
7. Módulo principal.
8. Módulo encoder.

**2.3. Interfaces**

1. *Módulo acelerómetro:* recibe los datos del hardware externo del tipo MPU-9250 se conecta mediante un bus I2C (inter integrated circuits) con la placa EDU-CIAA, y envía la información a el *módulo principal*.
2. *Módulo control de motores*: del *módulo principal* recibe la velocidad de los dos motores a controlar, y envía al hardware externo (puente H, tipo L298) la información mediante dos señales PWM..
3. *Módulo de comunicaciones WIFI*: recibe y envía información entre el módulo principal y el hardware externo (del tipo ESP 8266) de comunicación WIFI. Con el hardware externo se comunica con una comunicación TTL de 3,3v serie. La información debe estar codificada en formato de tramas para poder prevenir errores de comunicación.
4. *Módulo de posicionamiento* (GPS): brinda al *módulo principal* información de posicionamiento, se comunica con el hardware externo (del tipo Neo 6 Ublox), mediante una comunicación serie TTL de 3,3v o 5v.
5. *Módulo sensor de ultrasonido*: envía información al *módulo principal* proveniente de un hardware externo (del tipo HC-SR04), se comunica con éste mediante dos líneas de I/O de propósitos generales.
6. *Módulo de administrador de energía:* en función de los requerimientos del *módulo principal*, este modulo de software administra la energía de la placa EDU-CIAA.
7. *Módulo principal*: es una máquina de estado y se comunica con todos los módulos del sistema.
8. *Módulo de encoder*: mediante un sensor de cuadratura recibe pulsos TTL de 3,3v para registrar la odometría de cada motor, que se reenvía al *módulo principal.*
9. **Diseño preliminar**
10. Módulo acelerómetro.

TBD

1. Módulo control de motores.

Se decodifica el valor absoluto

Se decodifica el sentido de giro

Para qué motor

Se envía valor de velocidad absoluto y sentido de giro al motor **izquierdo**

Se envía valor de velocidad absoluto y sentido de giro al motor **derecho**

1. Módulo de comunicaciones WIFI.

TBD

1. Módulo de posicionamiento (GPS).

TBD

1. Módulo sensor de ultrasonido.

Se envía pulso de disparo (>10mseg)

Se lanza un contador

No

Se recibe pulso en la entrada

Si

Se calcula la distancia

Se envía la distancia al Módulo Principal

1. Módulo de administrador de energía.

TBD

1. Módulo principal.

TBD