BOTBLOQ: Ecosistema integral para el diseño, fabricación y programación de robots DIY

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI)
EXPEDIENTE: IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

ACRÓNIMO DEL PROYECTO: BOTBLOQ

ENTREGABLE E.4.2. Diseño de la arquitectura modular.

RESUMEN DEL DOCUMENTO

Listado de modelos clásicos de robots que permitan obtener un amplio abanico de aplicaciones.

Diseño de los módulos. Mecánicos, Eléctricos, Electrónicos, Otros.

Descripción de los modos de diseño: Generación automática o predefinida, Generación semiautomática, Generación manual.

---

5 de febrero de 2016

Índice

1. Introducción **4**

2. Diseño de módulos **4**

2.1. Diseño estructural 6
2.2. Geometría de los módulos 8
2.3. Sistema de conexión entre módulos 9
2.4. Estructura módulos activos 12
2.4.1. Módulo principal 12
2.4.2. Módulo de infrarrojos 13
2.4.3. Módulo de ultrasonidos 14
2.4.4. Módulo de servo par bajo 15
2.4.5. Módulo de servo par medio 17
2.4.6. Módulo de servo par alto 18
2.5. Estructura módulos pasivos 18
2.5.1. Uniones rectas 19
2.5.2. Uniones en ángulo 19
2.5.3. Ruedas 20
2.5.4. Mano 20
2.5.5. Pie 21
2.6. Diseño eléctrico/electrónico 22
2.7. Electrónica Módulo Principal 24
2.8. Electrónica Módulos Sensores 25
2.8.1. Sensor de proximidad 26
2.8.2. Sensor de distancia 27
2.8.3. Sensor de luz 28
2.8.4. Sensor de sonido 29
2.9. Electrónica Módulos actuadores 30
2.9.1. Módulo Servo Par Bajo 30
2.9.2. Módulo Servo Par Medio 33
2.9.3. Modulo Servo Par Alto 35
2.10. Alimentación 37
2.11. Conexión electrónica entre módulos 37

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

3. Descripción de los modos de diseño **38**
3.1. Generación predefinida . . . 38
3.2. Generación manual . . . 39
3.3. Generación Semi-Automática . . . 39
3.4. Otras consideraciones del esquema de diseño . . . 40
3.4.1. Definición de la estructura física . . . 41
3.4.2. Definición semántica . . . 41
3.4.3. Control del Robot . . . 42

4. Listado de robots **42**
4.1. Robots industriales . . . 43
4.2. Robots educativos . . . 45
4.2.1. Robots con ruedas . . . 45
4.2.2. Robots con patas . . . 47
4.3. Robots de investigación . . . 47

5. Anexo I. Métodos de comunicación entre módulos **49**
5.1. Powerline Communication . . . 50
5.2. SISTEMAS DC POWERLINE UTILIZANDO TRANSFORMADORES DE TRANSMISIÓN LINEAL. . . . 51
5.3. BIAS-T . . . 54
5.4. 1-WIRE . . . 59
5.5. DIGITAL COMMAND CONTROL . . . 60
5.6. FILTRO PASO ALTO/PASO BAJO ENTRE TERMINALES DE ALIMENTACIÓN Y DATOS . . . 61

6. Anexo II. Demostrador ROS en EDISON con Robot Renacuajo **65**
6.1. Introducción . . . 65
6.2. Instalación de Ubilinux en la Edison . . . 65
6.2.1. Introducción . . . 65
6.2.2. Instalación de Ubilinux . . . 66
6.2.3. Instalación de ROS (Indigo) . . . 68
6.3. Multiple machine . . . 71
6.4. Demostrador con el PrintBot Renacuajo . . . 71
6.4.1. Generación del Workspace . . . 72

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

6.4.2. Instalación del paquete rosserial 72
6.4.3. Instalación de las librerías ros_lib 73
6.4.4. Compilado del programa 74
6.5. Ejecución del Demostrador 75
6.6. Simulación en Gazebo del Demostrador 75

7. Anexo III. Ejemplos de robots prediseñados 76
7.1. URDF 76
7.2. SDF 78
7.3. Rviz 79
7.4. Gazebo 81
7.5. Simulaciones 81
7.5.1. Servo 82
7.5.2. Serpiente 85
7.5.3. Zowi 85
7.5.4. Manipulador 85
7.5.5. Robots móviles 85

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

1. Introducción

En este documento presentamos el desarrollo de la estructura modular realizado para Botbloq hasta la fecha.

El la Sección 2 presentamos en conjunto la arquitectura modular que proponemos para Botbloq separando entre diseño físico y diseño electrónico. En la Sección 2.1 describimos la estructura general física de los módulos, separando en módulos con electrónica o activos (apartado 2.4) y módulos pasivos sin electrónica (apartado 2.5). En el apartado 2.6 presentamos el diseño electrónico general, centrándonos en apartados siguientes en las descripciones en detalle de los diferentes tipos de módulos. Por último en el apartado 2.11 mostramos como se realizará la conexión electrónica entre módulos que permita su comunicación y alimentación.

En la Sección 3 describimos los diferentes modos de diseño que están planeados el desarrollo de robots que utilicen nuestra estructura modular. Por último, en la Sección 4 presentamos un listado de todos los robots más característicos cuya estructura, a nuestro entender, debería ser posible emular, total o parcialmente, con nuestra estructura modular.

2. Diseño de módulos

El diseño de nuestro sistema modular gira entorno a dos premisas: un diseño estructural de los modulos que permita construir la mayor variedad de robot posibles y un diseño electrónico que permita darle la actuación, sensorización e inteligencia suficiente para controlar cualquiera de estos robots.

Tras el estudio de las diferentes estructuras modulares y de los requisitos de nuestro sistema vistos en el Entregable E4.1, se ha decidido diseñar la estructura modular de una manera *heterogénea abierta*, esto es: un conjunto de módulos activos con número limitado más un grupo de módulos pasivos con un posible número ilimitado. Los módulos activos tienen una geometría y dimensión determinadas (ver apartado 2.4), mientras que los pasivos son parametrizables y por lo tanto pueden existir infinitas variantes (ver apartado 2.5). Ambos tipos de módulos serán imprimibles en 3D. Los módulos activos llevan una electrónica que detallamos en el apartado 2.6. Esta electrónica gira entorno a un microcomputador basado en la plataforma Intel Edison que permita la integración de los módulos en el framework

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

de ROS (*Robot Operating System*). En realidad, la integración en ROS es un requisito que trasciende varios puntos fundamentales en el diseño de la arquitectura hardware. En primer lugar, ROS es un referente hoy en día el desarrollo de robots tanto de investigación como comerciales. Su estructura hace posible una comunicación entre diferentes elementos del robot de manera transparente al programador. De la misma manera, el programador puede utilizar una gran cantidad de librerías robóticas desarrolladas anteriormente por multitud de investigadores y compañias. Estas librerías van desde planificadores de trayectorias, sistemas de visión, inteligencia artificial etc. Además ROS propone una manera de definir los robots que será utilizada en nuestro proyecto para la descripción semántica y sintáctica de los robots modulares creados.

En la Figura 1 resumimos el esquema general que tendrá nuestra arquitectura modular teniendo en cuenta la estructura modular, la integración en ROS y los modos de diseño que se verán en el apartado 3.

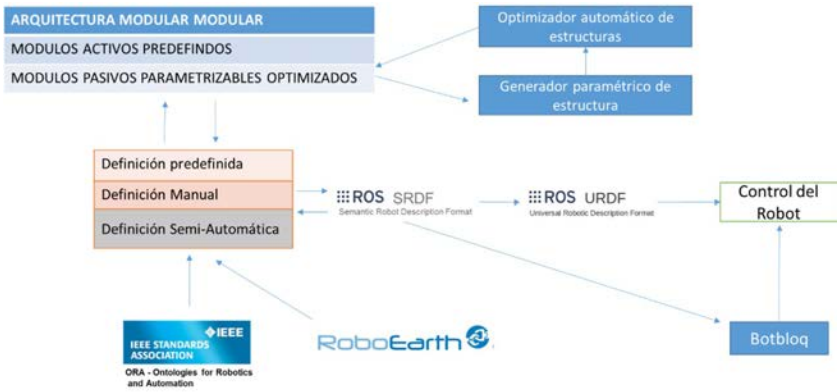


Figura 1: Esquema general de la arquitectura modular propuesta.

Se comprueba pues que nuestro diseño gira alrededor de ROS. En concreto, ROS nos va a servir tanto como para definir los robots que creemos como para controlarlos. Utilizaremos los siguientes aspectos:

- Control del Robot. Los robots generados por la estructura modular estarán integrados en ROS. Además de poder controlarlos con ROS utilizaremos las herramientas de definición que trae ROS. El uso de ROS implicará un hardware que permita correr Linux embebido.

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

- ROS URDF. Para definir la estructura física del robot aprovecharemos el Formato de Descripción Universal de Robots (URDF) que integra ROS. A este nivel tendremos una especificación de bajo nivel de cómo están conectados y relacionados los diferentes bloques del robot

- ROS SRDF. El significado semántico estará definido utilizando un formato parecido al SRDF. No utilizaremos el propio SRDF pues es propio de GAZEBO y no es totalmente compatible con todos los robots. En cambio nos crearemos nuestro propio meta lenguaje en el que tendremos definido semánticamente un robot: piernas, brazos, manos, chasis, etc

Es por ello que el uso de ROS ha sido una de las razones principales de utilizar un sistema con un Linux embebido como es la Intel Edison tal como se detalle en le diseño electronico del apartado 2.6. En el Anexo II se presenta un demostrador de la utilización de ROS con la Intel Edison para el control del robot Renacuajo de BQ.

En los siguientes subapartados presentamos en detalle el diseño estructural y el diseño electrónico llevado a cabo hasta la fecha. Aunque es de esperar que este diseño se mantega en su mayoría en versiones futuras es posible que algunos detalles sean cambiados o mejorados a medida que se avance con desarrollo del proyecto BOTBLOQ.

2.1. Diseño estructural

En esta sección se presenta la versión actual del desarrollo estructural de la plataforma modular que se propone para el proyecto BotBloq. Para la creación de los módulos se han seguido una serie de premisas:

- Los módulos deben poder crearse con una impresora 3D con tecnología de Modelado de Fusión por Deposición (FDM).

- Reducir al mínimo métodos de sujeción no impresos, como tornillos o tuercas.

- Crear módulos cuya relación de tamaño sea proporcional entre si, de forma que sea más sencillo crear cualquier estructura.

- Diseñar un sistema de unión entre módulos que sea imprimible y sencillo de usar.

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

- Todas las caras de los módulos deben poder unirse entre si.

Estas premisas tienen como principal limitación el uso de la tecnología de FDM, ya que introduce un gran número de limitaciones. El problema principal de esta tecnología es la tolerancia dimensional, ya que depende de múltiples factores como la temperatura de extrusión, la temperatura ambiente de la sala, el estado del hotend de la máquina, la escala del objeto e incluso del color (por los pigmentos utilizados) del material plástico que se está empleando. Otra limitación importante se debe al principio en el que se basa esta tecnología: la deposición de un filamento de plástico fundido. Este fundamento limita notablemente las geometrías realizables con acabados dentro de los márgenes aceptables, por ejemplo, las estructuras en voladizo se completan de forma muy defectuosa. Este hecho limita los posibles diseños realizables.

Los módulos creados se dividen en dos tipos: módulos activos, que contienen los componentes electrónicos; y módulos pasivos, que actúan como eslabones de unión entre módulos. Los módulos activos encapsulan a los componentes presentados en sección de diseño hardware de esta memoria (2.6), por tanto se dispone de módulos para la placa de control (módulo principal), sensores y actuadores. En la Figura 2 puede verse un ejemplo de un sistema ensamblado con los módulos desarrollados. Todos estos módulos serán descritos más detalladamente en las siguientes secciones de este documento.

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

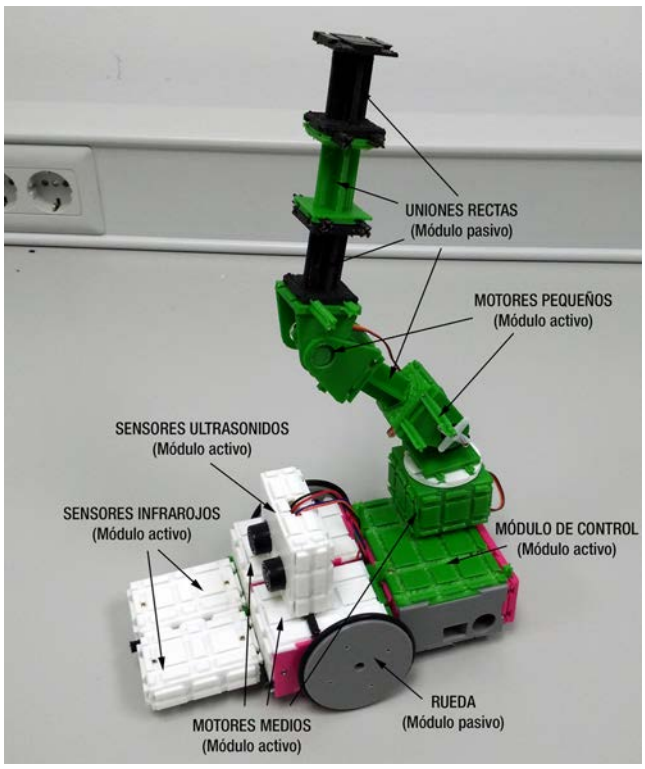


Figura 2: Ejemplo de sistema modular.

2.2. Geometría de los módulos

La primera consideración para el diseño es la geometría de los módulos. Tras el análisis de estado del arte de los robots modulares existentes se observa que las geometrías más comunes son prismas simples cuadrangulares (por lo general cubos), aunque en algunos casos se emplean geometrías más complejas como dodecaedros. Este tipo de geometrías basadas en prismas cuadrangulares son ideales para su impresión 3D, ya que no necesitan de la impresión de partes en voladizo, por lo que se decide adoptarlas en este proyecto.

Como se indicó en la introducción, un requisito de la arquitectura es que la relación de tamaños de módulos sea proporcional entre sí. Por ello cada cara del prisma será de un tamaño a o $n * a$ siendo n un número entero.

En la Figura 3 pueden verse tres paredes de distinto tamaño para módulos. Como se observa, cada una es proporcional a las demás.

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

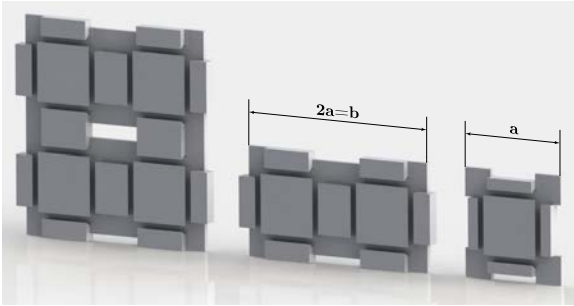


Figura 3: Ejemplo de sistema modular.

Como ya se ha comentado, los módulos deben crearse mediante impresión 3D, lo que limita las posibilidades de diseño. El módulo debe tener un sistema de unión con los otros módulos y debe contener la electrónica que le corresponda (en el caso de los módulos activos). Por todo esto se considera que la forma más efectiva de crear los módulos consiste en fabricar cada una de las caras del prisma rectangular de forma independiente uniéndola posteriormente. Como se observa en la Figura 4, las caras se crean de forma que puedan ensamblarse entre si fácilmente. El diseño se ha realizado de forma que con unos tornillos el módulo permanece ensamblado de forma robusta.

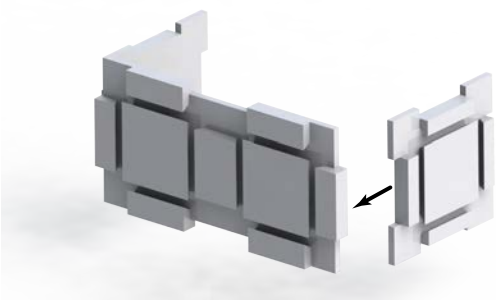


Figura 4: Ejemplo de sistema modular.

2.3. Sistema de conexión entre módulos

El factor más critico del diseño de los módulos es la conexión entre los mismos. La unión entre módulos debe cumplir la siguiente serie de especificaciones que se han propuesto en este proyecto:

- Fabricarse con una impresora 3D con tecnología de FDM.

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

- Montarse con facilidad, para una rápida conexión y desconexión de los módulos.
- No permitir el movimiento relativo entre módulos. La conexión debe unir los módulos de forma firme.

Existen gran cantidad de métodos de unión entre objetos, habiéndose utilizado algunos de ellos durante siglos. El reto de esta parte del proyecto consiste desarrollar una forma de unión que sea realizable mediante impresión 3D. Las limitaciones de la tecnología FDM, restringen el tipo de fijaciones realizables. Durante la fase de diseño se han realizado distintos prototipos con diferentes sistemas de unión. Los diseños iniciales para las uniones han sido muy diversos, y se enumeran a continuación.

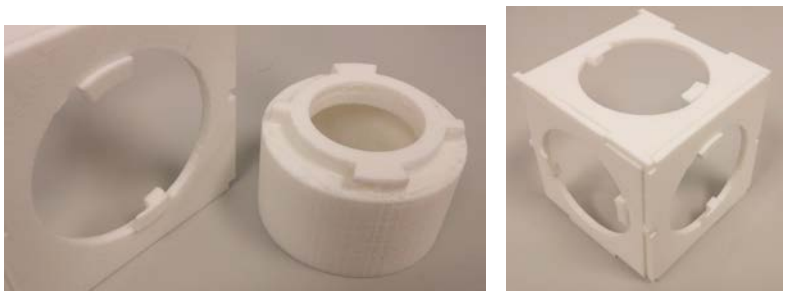
Uno de los primeros diseños se basa en la conexión de las caras mediante agujeros sobre las mismas que son ocupados por salientes de las caras de otro módulo (puede verse un ejemplo en la Figura 5). Uno de los problemas de esta fijación es que mantiene los módulos fijos en el plano de la cara, pero pueden moverse de forma perpendicular a ese plano, además las caras quedan rotadas 45º entre sí.

(a) Caras separadas. (b) Caras unidas.

Figura 5: Primer prototipo de sistema de unión.

El segundo sistema de unión propuesto se basa en el uso de pestañas en las dos caras que, mediante un giro, se mantienen unidas. En la Figura 6 puede verse el sistema de unión. El principal problema de este mecanismo es la fragilidad de las pestañas que pueden romperse con facilidad, además requiere alta precisión en la fabricación de las pestañas para mantener un ajuste con apriete, la tecnología FDM difícilmente alcanza esta precisión con dimensiones tan reducidas.

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020



(a) Cara y módulo pasivo con sistema de pestañas.

(b) Módulo con sistema de pestañas.

Figura 6: Segundo prototipo de sistema de unión.

El tercer sistema de unión diseñado emplea una acanaladura donde se inserta un saliente que la recorre, de forma similar a como lo hace un conector BNC, por último, un vástago que atraviesa las dos partes del mecanismo fija la unión. En la Figura 7 puede verse el mecanismo. El principal problema de este mecanismo es la construcción de la acanaladura, así como su resistencia tras múltiples usos. Además el sistema emplea formas distintas en cada cara, lo que no permite conectar conectar cualquier cara de dos módulos entre si.

(a) Partes del sistema.

(b) Ssitema unido.

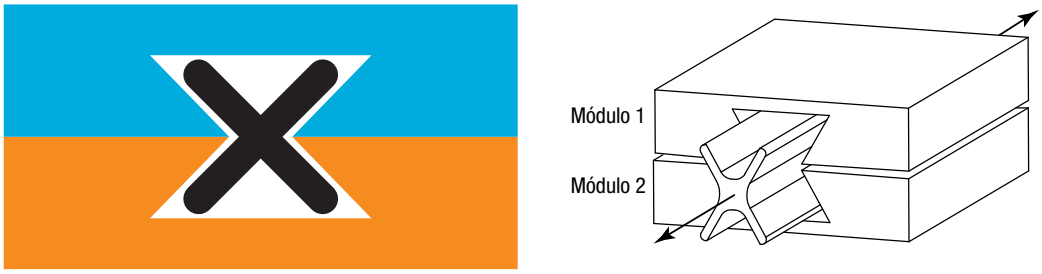
(c) Esquema.

Figura 7: Tercer prototipo de sistema de unión.

Finalmente el mecanismo de unión entre módulos que se ha decidido utilizar es un sistema similar a la cola de milano. La Figura 8(a) muestra una vista frontal. Como se observa, los módulos disponen de una acanaladura en sus caras cuyas paredes se encuentran en ángulo agudo de forma que al insertar una pieza con forma de cruz los módulos permanecen unidos entre si. En la Figura 8(b) se observa de forma más clara este mecanismo. Las ventajas de este sistema son que puede imprimirse sin grandes problemas de tolerancias y que todas las caras emplean el mismo sistema de unión. Además es notablemente robusto y su ensamblado y

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

desensamblado es relativamente sencillo.



(a) Vista frontal del sistema de unión. (b) Vista tridimensional del sistema.

Figura 8: Cola de milano.

La Figura 2 muestra varios módulos unidos mediante este mecanismo, como puede observarse, es posible crear estructuras de alta complejidad.

2.4. Estructura módulos activos

Contienen los componentes electrónicos del sistema modular. Por un lado está el módulo central, que se encarga del control de todo el sistema, y la batería encargada de la alimentación de todos los componentes. Por otro lado están los sensores, que ofrecen información del entorno que rodea al robot. Se ha decidido que todos estos módulos serán completamente cerrados encapsulando los componentes, dejando accesible únicamente los conectores y las partes que deban interactuar con el exterior.

A continuación se muestra cada unos de los módulos activos y se explica su composición.

2.4.1. Módulo principal

Es el encargado de albergar la electrónica para controlar el resto de módulos. En su interior se encuentra la placa de control del sistema (microcomputador Intel Edison que se describe en secciones posteriores). En la Figura 9 puede verse el módulo principal, dispone de una serie de aperturas para la conexión con el exterior.

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

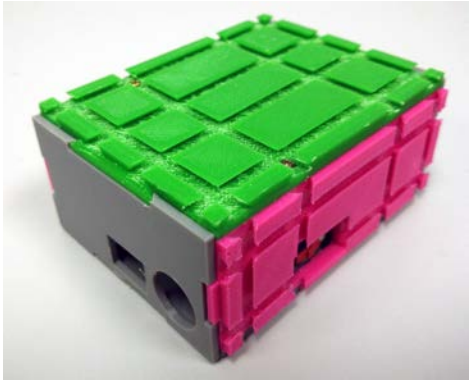


Figura 9: Módulo principal.

En la Figura 10 se observa que todas las caras de este módulo disponen de un sistema de unión, por tanto se pueden unir módulos tanto activos como pasivos en cualquiera de sus caras.

Figura 10: Caras del módulo principal.

2.4.2. Módulo de infrarrojos

Este módulo contiene un sensor de infrarrojos, así como un microcontrolador encargado de las comunicaciones que se describirán en la sección 2.8. En la Figura 11 puede verse la distribución de los componentes en su interior.

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

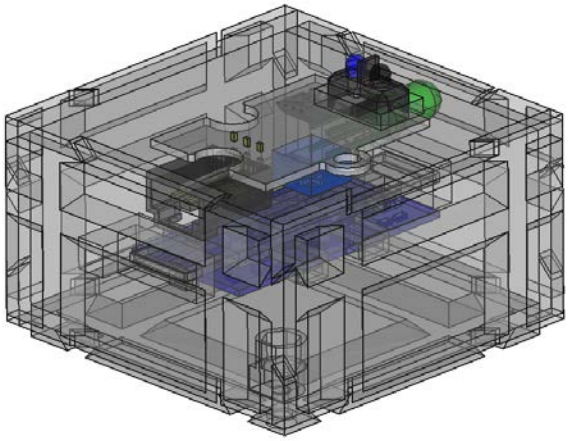


Figura 11: Esquema del módulo de infrarojos.

La Figura 12 muestra un módulo real de infrarrojos. Todas las caras disponen del sistema de unión salvo la cara en la que se encuentra el fotoemisor y fotorreceptor ya que es la encargada de interactuar con el exterior.

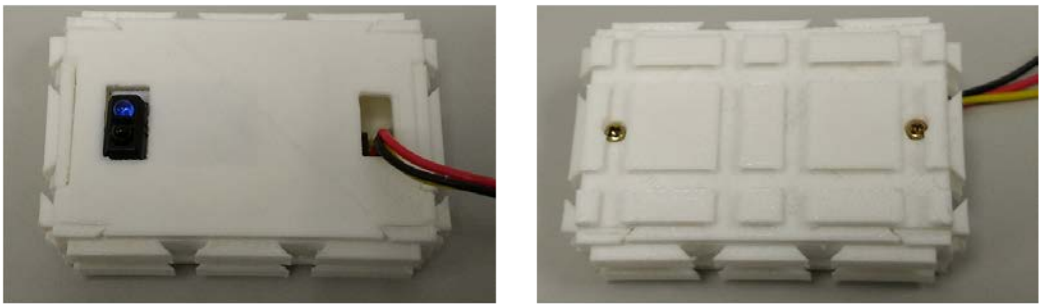


Figura 12: Módulo de infrarojos.

2.4.3. Módulo de ultrasonidos

En este módulo se encuentra el sensor de ultrasonidos, así como, nuevamente, un microcontrolador encargado de las comunicaciones. La Figura 13 muestra un esquema del interior del módulo. En ella puede verse la disposición del sensor y el microcontrolador que lo componen.

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

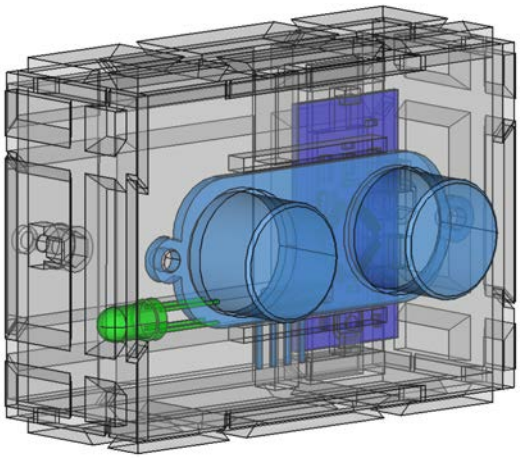


Figura 13: Esquema del módulo de ultrasonidos.

El módulo real aparece en la Figura 14. Nuevamente todas las caras disponen del sistema de unión salvo la cara en la que se encuentra el emisor/receptor de ultrasonidos para permitir al módulo interactuar con el entorno.

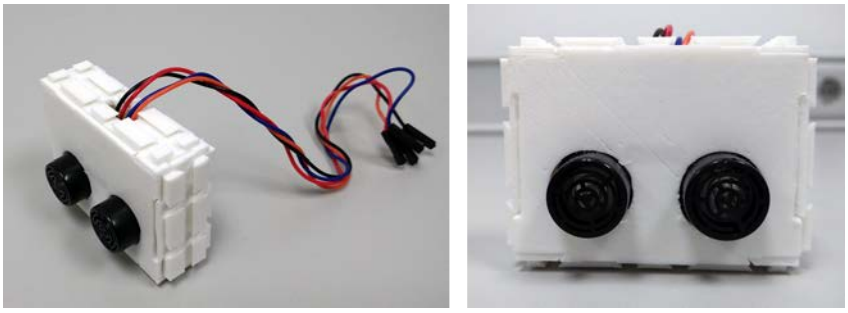


Figura 14: Módulo de ultrasonidos.

2.4.4. Módulo de servo par bajo

Este módulo encapsula un motor que se encarga de mover pequeñas cargas, como por ejemplo módulos sensores o pequeñas partes pasivas. Se han diseñado dos variantes de este módulo: de rotación estándar y de rotación continua. La Figura 15 muestra un esquema de este módulo empleado para giro axial, en ella se observan tanto el servomotor como el microcontrolador encargado de las comunicaciones.

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

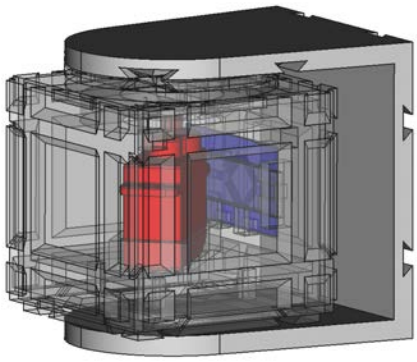


Figura 15: Esquema módulo de miniservo de giro axial.

En las figuras 16 y 17 se muestran módulos reales. La primera Figura muestra el módulo en una configuración de giro axial, mientras que la segunda permite el giro radial.

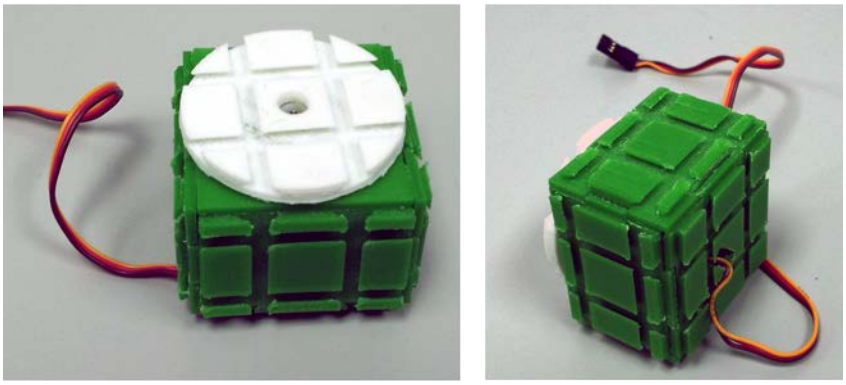
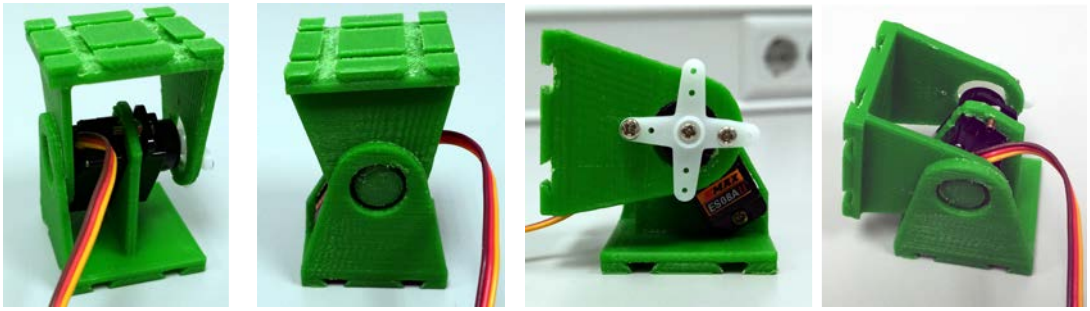


Figura 16: Módulo de miniservo de giro axial.

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa
Operativo Plurirregional de Crecimiento Inteligente 2014-2020



(a) Posición 1 (b) Posición 2

Figura 17: Módulo de miniservo giro radial.

2.4.5. Módulo de servo par medio

Este módulo es de mayor tamaño que el anterior y se encarga de mover la estructura principal del robot. Como en el caso anterior esta formado con un motor, pero en esta ocasión su par es mayor. Se distinguen igualmente dos tipos: rotación estándar y rotación continua. En la Figura 18 se muestra la distribución interna de las distintas partes del módulo.

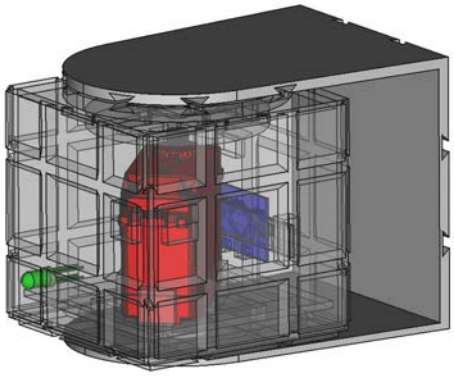


Figura 18: Esquema módulo de motor de giro axial.

En la Figura 19 puede verse un módulo real de servo de par medio. Se pueden conectar módulos en todas sus caras salvo en la de salida del eje del motor.

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

Figura 19: Módulo de servomotor.

2.4.6. Módulo de servo par alto

Estos módulos serán los de mayor tamaño junto con el cerebro del robot. Su función es la de desplazar y levantar cargas y/o, en casos puntuales, mover la estructura principal del robot si éste fuese de grandes dimensiones. La estructura es similar a la del servo de par medio con la salvedad de que sus dimensiones son mayores.

2.5. Estructura módulos pasivos

Los módulos pasivos permiten añadir funcionalidades al sistema o colocar módulos activos en posiciones concretas. Por ejemplo, el uso de un módulo Rueda, el cual permite al robot desplazarse mediante rodadura, añade la funcionalidad de locomoción. La colocación de módulos activos en posiciones determinadas permite generar robots distintos con la misma configuración cualitativa, e.g. un robot antropomorfo de alcance 15 cm o de alcance 20 cm puede generarse con uniones rectas pasivas de 5 cm de diferencia en longitud para colocar el siguiente servo.

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

2.5.1. Uniones rectas

El módulo recto de unión permite colocar módulos desplazados unos de otros una distancia L, siendo L la longitud del módulo pasivo de unión recta. Este módulo se puede crear con la longitud que se desee en función de la aplicación para la que se emplee. En la Figura 20 puede verse un ejemplo de un módulo de real.

Figura 20: Módulo de unión recta.

2.5.2. Uniones en ángulo

Al igual que los módulos anteriores, los módulos de unión en ángulo permiten la conectar dos módulos, pero es esta ocasión formando un determinado ángulo entre sí. Gracias a que estos módulos se crean mediante impresión 3D, es posible generarlos con el ángulo más adecuado para la estructura que se desea crear. En la Figura 21 puede verse este módulo con un ángulo de 30º.

Figura 21: Módulo de unión en ángulo.

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

2.5.3. Ruedas

El módulo pasivo rueda permite dotar de capacidades de rodadura al sistema, así como cualquier otra función que se puede realizar con una rueda. Es un módulo cuyo radio y grosor puede variar en función de las necesidades. Puede verse un ejemplo en la Figura 22.

(a) Rueda (b) Rueda con servomotor

Figura 22: Módulo rueda.

2.5.4. Mano

La creación de determinados tipos de robots hace necesaria la existencia de módulos específicos, por ejemplo, para crear un robot humanoide hacen falta manos. Este módulo pasivo es una mano que se puede emplear para la creación de robots humanoides. En la Figura 23 pueden verse varias imágenes de un módulo de mano real.

(a) Rueda (b) Rueda con servomotor

Figura 23: Módulo mano.

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

2.5.5. Pie

Este módulo, como en el caso anterior, es específico. Nuevamente, para la creación de un robot humanoide se necesitan pies que permitan apoyarse en el suelo. El módulo pasivo pie realiza esta función. En la Figura 24 puede verse un ejemplo de un módulo de real.

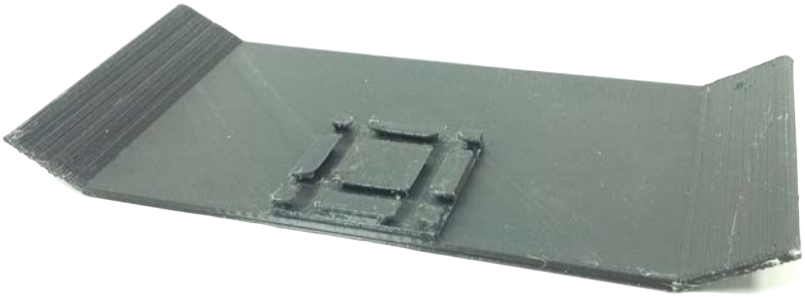


Figura 24: Módulo pie.

La conjunción de estos módulos con algunos de los anteriormente presentados permiten la creación de un robot humanoide tal y como se muestra en la Figura 25.

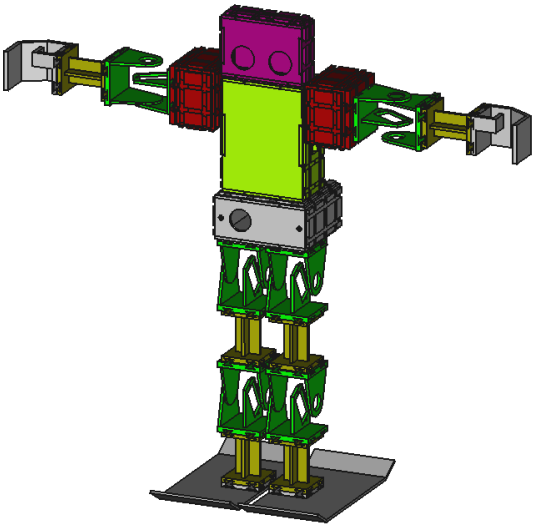


Figura 25: Representación de un robot humanoide.

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

2.6. Diseño eléctrico/electrónico

En este documento se pretende recoger una revisión inicial del hardware que formará el robot modular resultante del proyecto BOTBLOQ. A la hora de seleccionar el hardware del robot se diferencian tres grupos principales: Cerebro, Módulos Sensores y Módulos Actuadores. En la Figura 26 se puede observar un esquema general de dichos módulos, así como de la alimentación del robot. Todos estos módulos serán descritos más detalladamente en las siguientes secciones de este documento. En esta primera revisión se ha optado por utilizar tecnología propia de bq en la medida de lo posible.

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

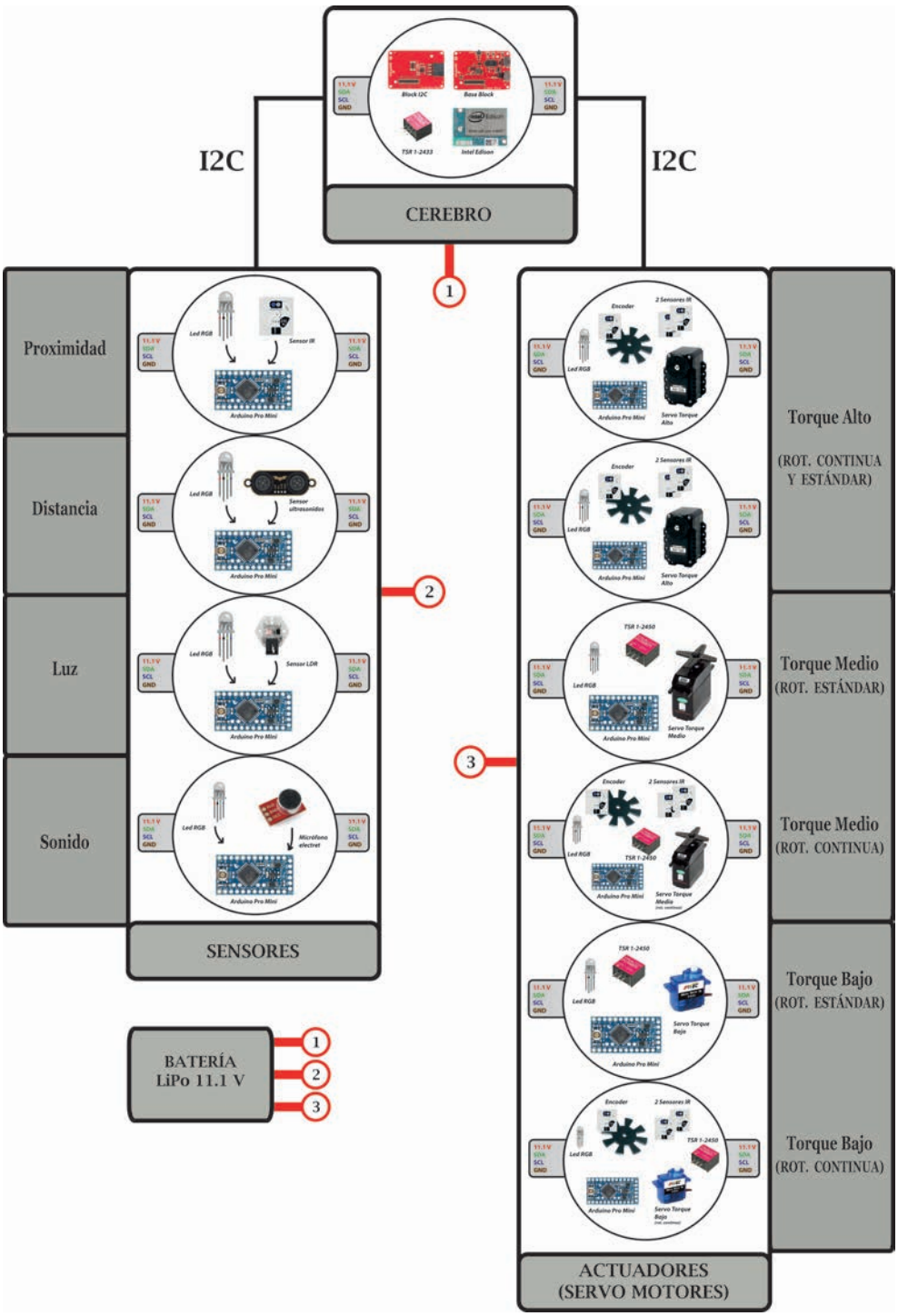


Figura 26: Esquema general de los módulos.

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa
Operativo Plurirregional de Crecimiento Inteligente 2014-2020

2.7. Electrónica Módulo Principal

Será el módulo principal del robot, también denominado "Cerebro". Será el encargado de comandar el resto de módulos del robot y, por tanto, de controlar las acciones del robot. Recibirá información de los módulos sensores y enviará órdenes a los módulos actuadores para actuar en consecuencia. Estará compuesto por (ver Figura 27):

- Intel Edison como cerebro. Este microcomputador dispone de SoC Intel Atom con CPU doble núcleo de 500 MHz, 1 GB de memoria RAM, 4 GB de almacenamiento interno, 40 pines de propósito general (GPIO) y Wifi de banda dual integrado. Esta última característica la diferencia del resto de microcomputadores de prestaciones similares.

- SparkFun Block for Intel Edison - I2C. Este dispositivo permite acceder al puerto I2C de la Intel Edison. Además dispone de un conector de 70 pines que permite acoplar la Intel Edison.

- SparkFun Block for Intel Edison – Base. Este dispositivo dispone de 2 puertos micro USB que permiten tanto alimentar la Intel como acceder a ella a través de una consola de comandos. Dispone también de un interruptor que permite encender, apagar o poner la Intel en estado de bajo consumo, así como del conector de 70 pines que permite acoplar la Intel.

- Regulador de tensión por conmutación TSR 1-2433. Este regulador permite adaptar la tensión de alimentación del módulo a la necesaria para alimentar debidamente la Intel Edison. Tiene una mayor eficacia que los reguladores lineales y por tanto produce menos calor, lo cual es un factor importante a considerar a la hora de encapsularlo en un módulo cerrado. Ofrece 3.3 V y hasta 1 A en su salida, pudiendo ser alimentado a una tensión de entre 4.75 V y 36 V.

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

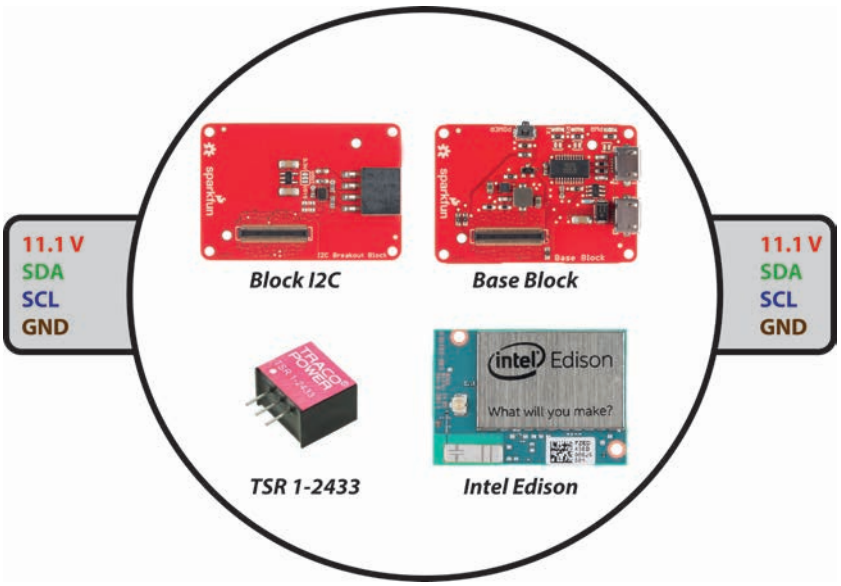


Figura 27: Esquema del módulo principal - Cerebro

La idea general es acoplar la Intel Edison al Bloque Base de tal forma que pueda ser alimentada mediante el puerto USB así como encendida y apagada a través del interruptor. Dicho Bloque Base se acopla al Bloque I2C a través del conector de 70 pines de tal forma que la Intel Edison pueda aprovechar las funciones de ambos. Con esta disposición se dispone de un módulo cuyo cerebro es la Intel Edison, utilizada y programada vía Wifi o USB, alimentada vía USB y comunicada con el resto de módulos vía I2C. El bus I2C recorre todos los módulos activos del robot, siendo el módulo de la Intel el maestro de dicho bus y el resto de módulos los esclavos.

2.8. Electrónica Módulos Sensores

Ofrecen información del entorno que rodea al robot. Se ha decidido que estos módulos serán completamente cerrados y por tanto cada uno de ellos vendrá con un firmware propio (en principio no actualizable) que permitirá hacer uso de sus funcionalidades propias así como de comunicarse con otros módulos y con el cerebro del robot.

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

2.8.1. Sensor de proximidad

Este módulo da información de si se encuentra muy próximo a un objeto o no, en un rango de distancias menor a 1 cm. Contiene (ver Figura 28):

- Arduino Pro Mini haciendo las veces de cerebro del módulo. Las características de este modelo de Arduino están resumidas en la Tabla 1

Microcontroller	ATmega328
Operating Voltage	3.3V or 5V (depending on model)
Input Voltage	3.35-12V (3.3V model) or 5-12 V (5V model)
Digital I/O Pins	14 (6 PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
Flash Memory	32 kB (0.5 kB for bootloader)
SRAM	2 kB
EEPROM	1 kB
Clock Speed	8 MHz (3.3V model) or 16 MHz (5.5V model)

Tabla 1: Especificaciones Arduino Pro Mini

  Es importante destacar que el modelo elegido es el que funciona a 5V. Por tanto la alimentación del mismo debe realizarse a una tensión entre 5-12V y la frecuencia del mismo será 16 MHz.

- Led RGB indicador del estado del módulo (conectado, vinculado, funcionando).

- Sensor de proximidad *Octopus* de bq. Ofrece una salida digital (0 ó 1). Tiene un tamaño y forma predeterminados.

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

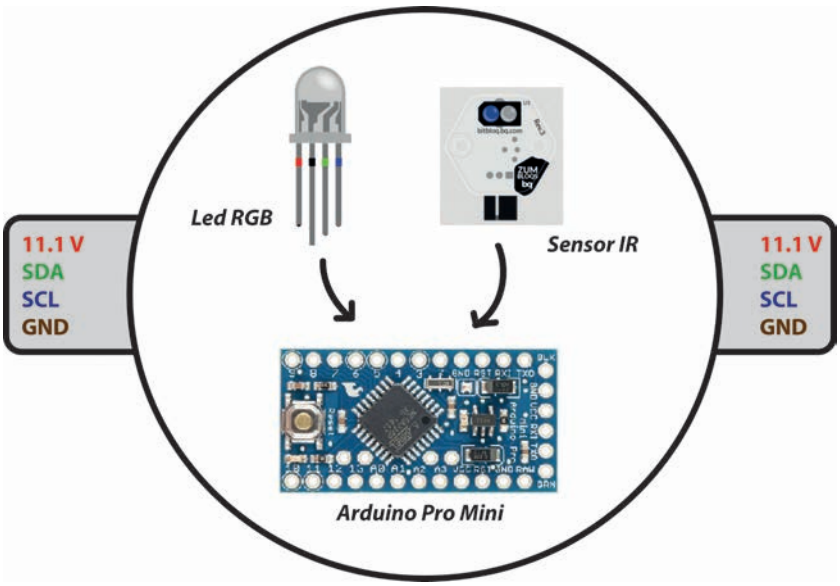


Figura 28: Esquema del módulo de proximidad

2.8.2. Sensor de distancia

Este módulo da información de la distancia a la que se encuentra un objeto dentro del rango de funcionamiento del sensor. Contiene (ver Figura 29):

- Arduino Pro Mini haciendo las veces de cerebro del módulo.
- Led RGB indicador del estado del módulo (conectado, vinculado, funcionando).
- Sensor de ultrasonidos de bq. Similar al sensor HC-SR04, tiene un rango de funcionamiento de entre 2 y 400 cm aproximadamente.

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

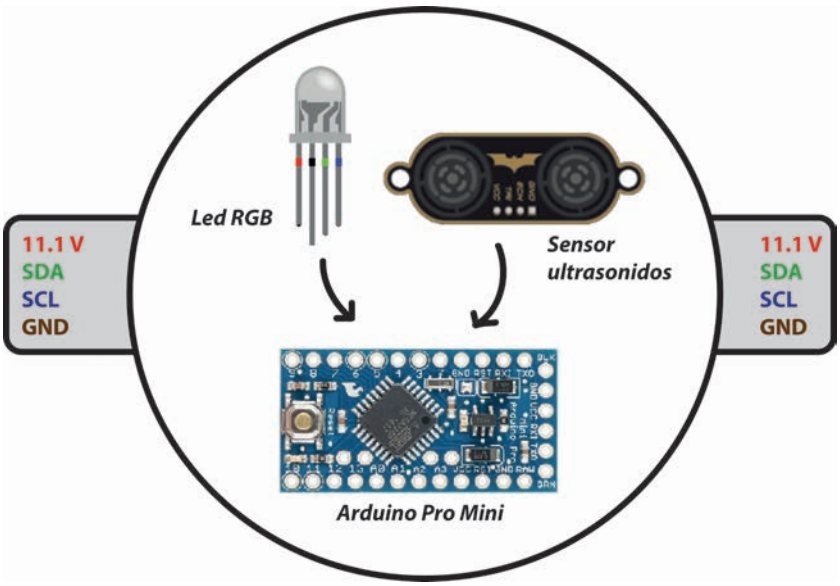


Figura 29: Esquema del módulo de distancia

2.8.3. Sensor de luz

Este módulo da información sobre la iluminación a la que está expuesto. Contiene (ver Figura 30):

- Arduino Pro Mini.
- Led RGB indicador del estado del módulo (conectado, vinculado, funcionando).
- Sensor LDR de bq. Entrega una señal analógica.

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

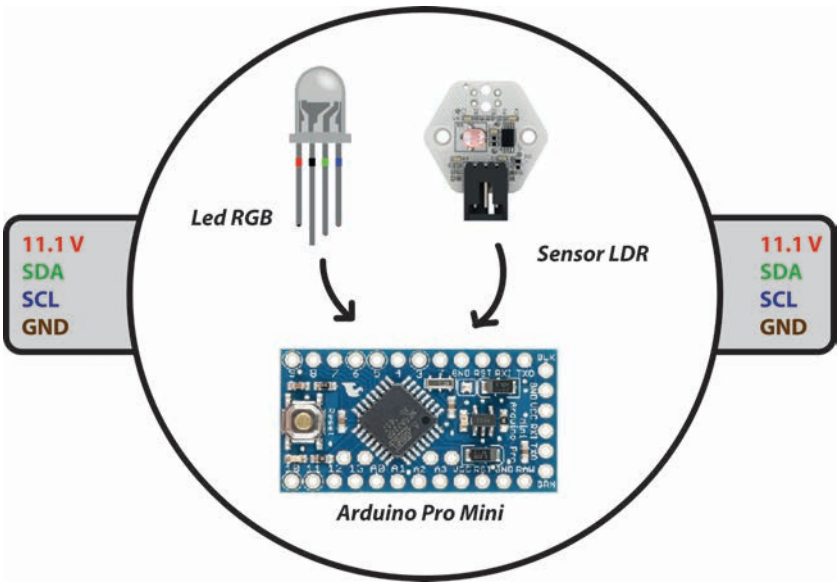


Figura 30: Esquema del módulo de luz

2.8.4. Sensor de sonido

Este módulo da información sobre la existencia o no de sonido alrededor del mismo y su amplitud. Contiene (ver Figura 31):

- Arduino Pro Mini.
- Led RGB indicador del estado del módulo (conectado, vinculado, funcionando).
- Micrófono electret preamplificado con el amplificador operacional OPA344.

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

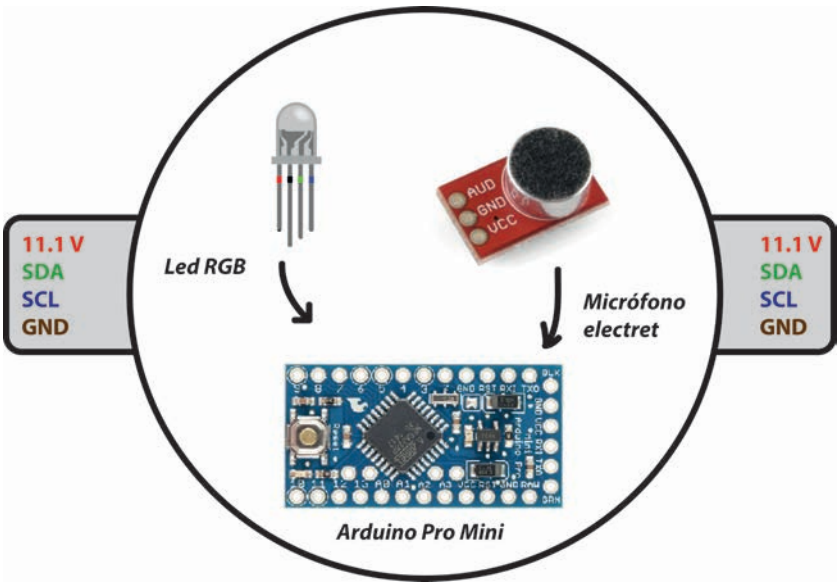


Figura 31: Esquema del módulo de sonido

2.9. Electrónica Módulos actuadores

Son los módulos encargados de realizar las acciones del robot, las cuales en su mayoría implican movimiento de uno o varios de los módulos restantes del robot. Al igual que ocurría con los módulos sensores, estos módulos también llevarán un firmware propio que permitirá su control a través del bus I2C.

2.9.1. Módulo Servo Par Bajo

Este módulo se encarga de mover pequeñas cargas, como por ejemplo otros módulos sensores o pequeñas partes pasivas. Se distinguen al menos dos tipos: de rotación estándar y de rotación continua.

- Rotación continua. Contiene los siguientes elementos (ver Figura 32):
    - Arduino Pro Mini como cerebro del módulo.
    - Led RGB indicador del estado del módulo (conectado, vinculado, funcionando).
    - Mini Servo modelo FS90. Se puede alimentar a una tensión comprendida entre 4.8 y 6 V. En función de dicho voltaje, el servo tendrá distintos

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

parámetros de funcionamiento. En la Tabla 2 se muestran dichos parámetros cuando el servo es alimentado a 4.8 V ó 6 V.

Tensión (V)	Par (Kg·cm)	Velocidad de giro (rpm)
4.8	1.3	100
6	1.51	130

Tabla 2: Especificaciones Servo FS90.

- Sensores de proximidad *Octopus* de bq. Este módulo integrará también dos sensores infrarrojos colocados en caras opuestas haciendo la función de sendos finales de carrera. De esta forma será posible utilizar el módulo como un actuador lineal.
- Regulador de tensión por conmutación TSR 1-2450. Este regulador permite adaptar la tensión de alimentación del módulo a la necesaria para alimentar debidamente el servo. Tiene una mayor eficacia que los reguladores lineales y por tanto produce menos calor, lo cual es un factor importante a considerar a la hora de encapsularlo en un módulo cerrado. Ofrece 5V y hasta 1.5A en su salida, pudiendo ser alimentado a una tensión de entre 6.5V y 36V.
- Encoder mediante sensor de proximidad *Octopus* y pieza circular dentada. El sensor irá colocado en la parte superior del servo, junto al eje. La pieza circular dentada irá acoplada al eje y sobre el sensor. De esta manera el sensor detectará cada vez que un hueco de la pieza dentada pase sobre él y se podrá determinar el número de revoluciones que el servo ha girado.

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

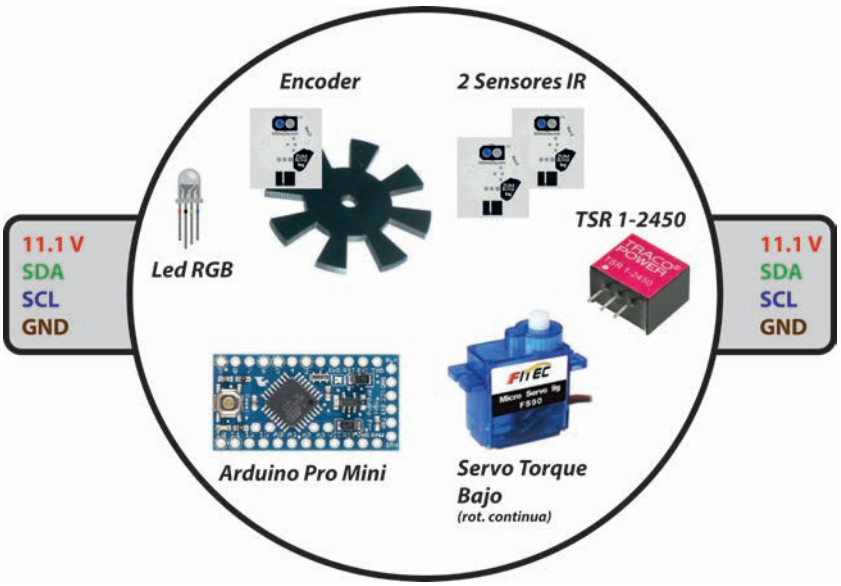


Figura 32: Esquema del módulo actuador par bajo y rot. continua.

- Rotación estándar
  - Arduino Pro Mini como cerebro del módulo.
  - Led RGB indicador del estado del módulo (conectado, vinculado, funcionando).
  - Mini servo modelo ES08A. La Tabla 3 muestra sus características.

Tensión (V)	Par (Kg·cm)	Velocidad de giro (rpm)
4.8	1.5	83.3
6	1.8	100

Tabla 3: Especificaciones Servo ES08A

  - Regulador de tensión por conmutación TSR 1-2450.

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

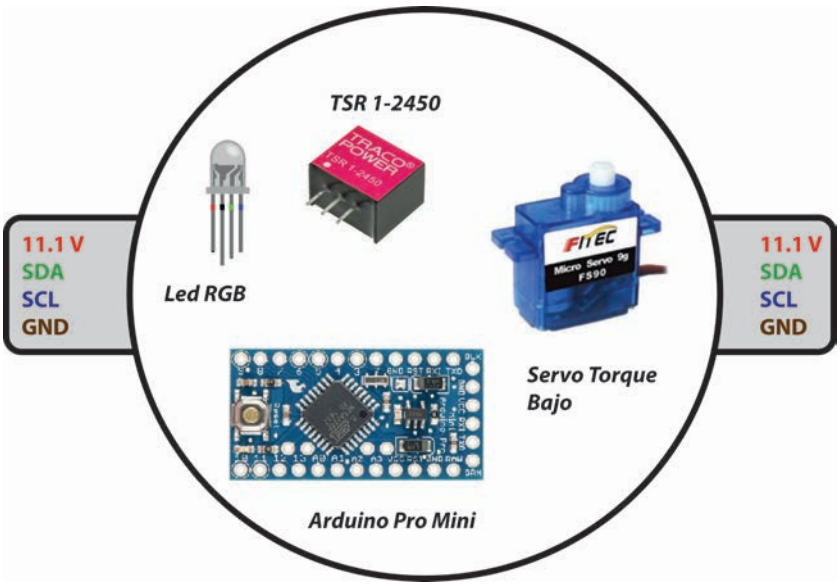


Figura 33: Esquema del módulo actuador par bajo y rot. estándar.

2.9.2. Módulo Servo Par Medio

Este módulo es de mayor tamaño que el anterior y se encarga de mover la estructura principal del robot. Se distinguen igualmente dos tipos: rotación estándar y rotación continua.

- Rotación continua
  - Arduino Pro Mini.
  - Led RGB indicador del estado del módulo (conectado, vinculado, funcionando).
  - Servo SM-S4303R, modelo utilizado por bq en su kit de robótica (características en Tabla 4).

Tensión (V)	Par (Kg·cm)	Velocidad de giro (rpm)
4.8	3.3	60
6	4.8	70

Tabla 4: Especificaciones Servo SM-4303R.

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

- Sensores de proximidad *Octopus* de bq. Este módulo integrará también dos sensores infrarrojos colocados en caras opuestas haciendo la función de sendos finales de carrera. De esta forma será posible utilizar el módulo como un actuador lineal.
- Regulador de tensión por conmutación TSR 1-2450.
- Encoder mediante sensor de proximidad *Octopus* y pieza circular dentada. El sensor irá colocado en la parte superior del servo, junto al eje. La pieza circular dentada irá acoplada al eje y sobre el sensor. De esta manera el sensor detectará cada vez que un hueco de la pieza dentada pase sobre él y se podrá determinar el número de revoluciones que el servo ha girado.

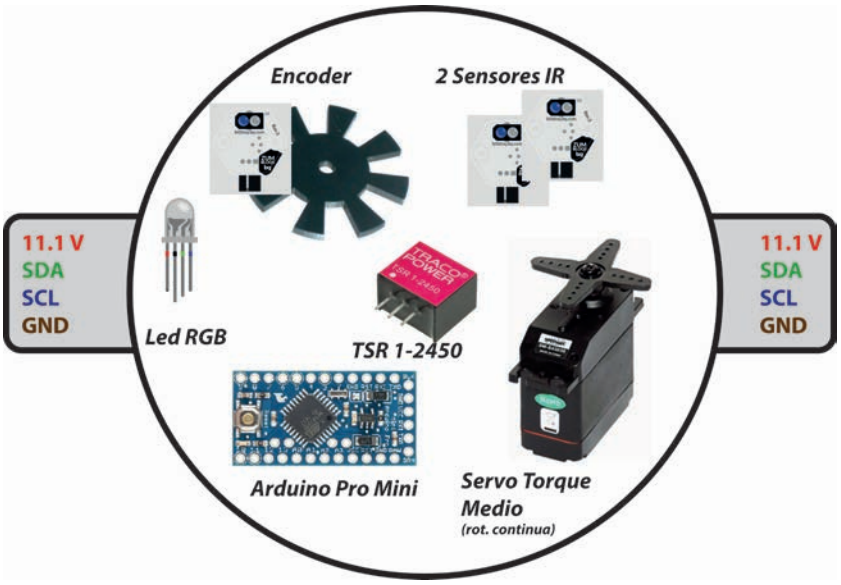


Figura 34: Esquema del módulo actuador par medio y rot. continua.

- Rotación estándar
  - Arduino Pro Mini.
  - Led RGB indicador del estado del módulo (conectado, vinculado, funcionando).
  - Servo modelo 3001HB. Características en Tabla 5.

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

Tensión (V)	Par (Kg·cm)	Velocidad de giro (rpm)
4.8	3.5	66.6
6	4.4	83.3

Tabla 5: Especificaciones Servo 300HB.

- Regulador de tensión por conmutación TSR 1-2450.

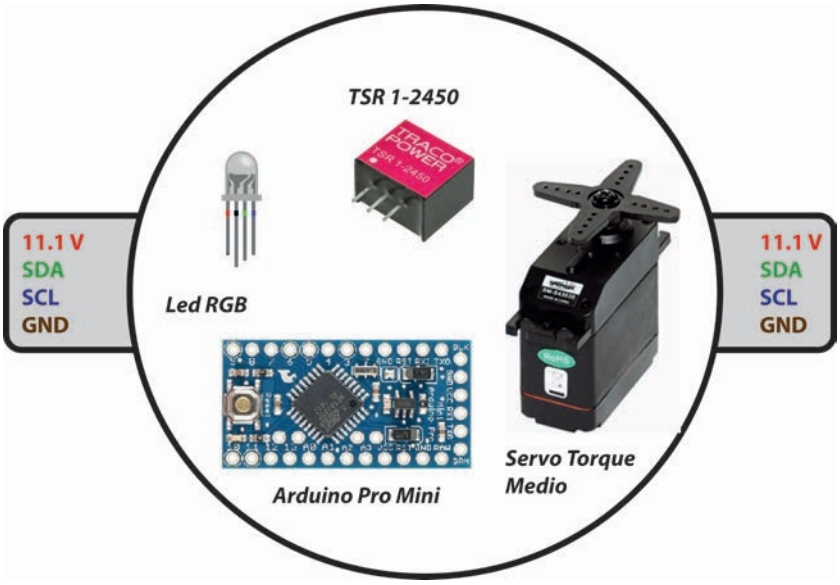


Figura 35: Esquema del módulo actuador par medio y rot. estándar.

2.9.3. Modulo Servo Par Alto

Estos módulos serán los de mayor tamaño junto con el cerebro del robot. Su función es la de desplazar y levantar cargas y/o, en casos puntuales, mover la estructura principal del robot si este fuese de grandes dimensiones. Para este caso particular se utilizará un único tipo de módulo, el cual podrá actuar como servo de rotación continua o de rotación estándar. Estará compuesto por (ver Figura 36):

- Arduino Pro Mini.
- Led RGB indicador del estado del módulo (conectado, vinculado, funcionando).

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

- Servo Dynamixel AX-12A (ver Tabla 6). Este servo motor permite alternar entre dos modos de movimiento: rotación continua y rotación estándar.

Tensión (V)	Par (Kg·cm)	Velocidad de giro (rpm)
12	12.29	59

Tabla 6: Especificaciones Servo Dynamixel AX-12A.

- Sensores de proximidad *Octopus* de bq. Este módulo integrará también dos sensores infrarrojos colocados en caras opuestas haciendo la función de sendos finales de carrera. De esta forma será posible utilizar el módulo como un actuador lineal.

- Encoder mediante sensor de proximidad *Octopus* y pieza circular dentada. El sensor irá colocado en la parte superior del servo, junto al eje. La pieza circular dentada irá acoplada al eje y sobre el sensor. De esta manera el sensor detectará cada vez que un hueco de la pieza dentada pase sobre él y se podrá determinar el número de revoluciones que el servo ha girado.

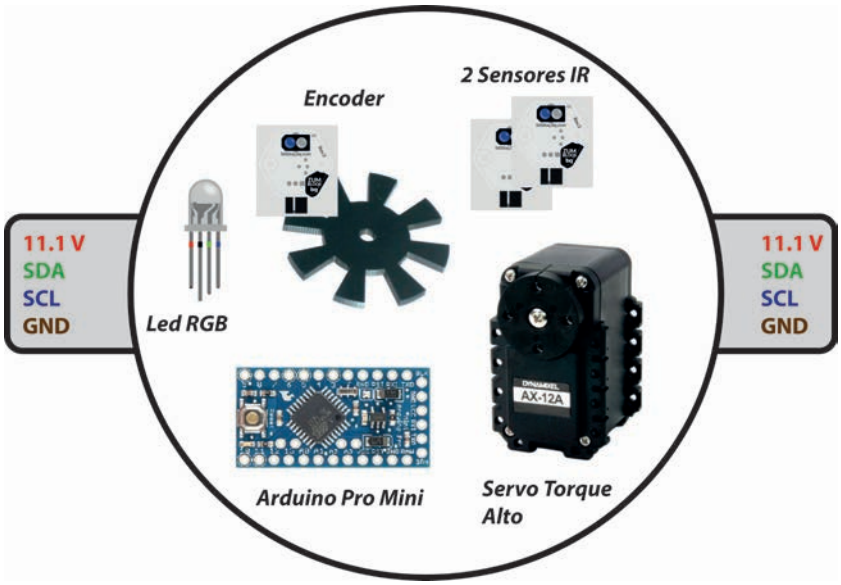


Figura 36: Esquema del módulo actuador par alto.

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

2.10. Alimentación

La batería que alimentará todos los módulos ofrecerá 11.1 V a su salida. Este voltaje es típico en baterías LiPo de 3 celdas y, por tanto, facilitará la selección de una batería cuya capacidad se ajuste al consumo final del robot. No obstante, de manera provisional se ha seleccionado una batería LiPo de 11.1 V y 2200 mAh con un conector tipo DEAN.

Dicho voltaje de salida debe ser adaptado a las necesidades particulares de algunos componentes (ver Tabla 7). Los módulos actuadores de par bajo y medio así como el cerebro deberán adaptar la tensión de 11.1V.

Módulo	Tensión (V)
Cerebro	3.3
Servo par Bajo	5
Servo par Medio	5

Tabla 7: Componentes que requieren adaptación de alimentación.

En el caso de los módulos actuadores de bajo par y medio el módulo será alimentado a 11.1V y el servo en particular será alimentado a 5V haciendo uso del regulador de tensión TSR 1-2450. El módulo Cerebro será alimentado a 3.3V previa adaptación de la tensión de la batería mediante el regulador TSR 1-2433. El resto de módulos irán alimentados a la tensión de la batería (11.1V) ya que la tensión de alimentación del Arduino Pro Mini, el cual actúa como cerebro de todos ellos, lo permite.

2.11. Conexión electrónica entre módulos

Otros de los aspectos eléctricos en los que se ha prestado atención es la comunicación entre módulos y su alimentación. Como se ha visto a lo largo de este documento, se ha optado por una comunicación I2C y una alimentación a 11.1V. Esto implica conexiones de 4 líneas (+,GND,SDA,SCL). Se han estudiado alternativas, descritas en Anexo 5.1, en las que se reduce un cable. La complejidad del diseño hace que estas propuestas se dejen como mejoras futuras. Para asegurar que los módulos se conectan correctamente se ha propuesto conectarlos con un sistema

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

tipo diana, como el de la Figura 37. Esto asegura que nunca se cortocircuiten líneas y permite 4 configuraciones de conexión a 90º.

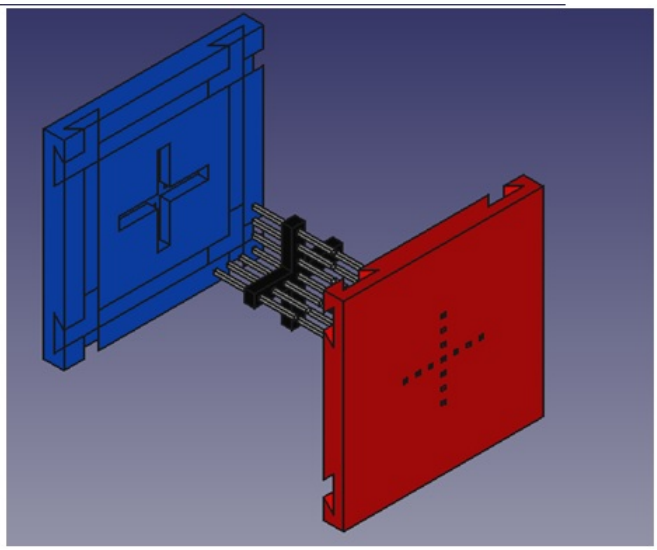


Figura 37: Sistema de conexión eléctrica entre módulos.

3. Descripción de los modos de diseño

Se permitirán tres modos de diseño según el nivel del usuario: generación predefinida, semi-automática y manual. De este modo se pretende hacer accesible la plataforma a un amplio espectro de público, desde alumnos de enseñanza primaria hasta investigadores en el campo de la robótica. Todos los diseños serán programables mediante la interfaz gráfica BOTBLOQ que está siendo creada por BQ, y que se basa a su vez en la IDE Bitbloq [1], la cual utiliza programación por bloques y está diseñada para uso educativo, pero también podrá ser programada directamente en interfaces de programación clásica similares a la IDE de Arduino. A continuación se indican algunas de las principales consideraciones respecto a cada uno de los modos de diseño.

3.1. Generación predefinida

El modelo completo del robot, con estructura física y su estructura semántica, estará definida de antemano. Con ambos elementos predefinidos, los bloques de

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

programación de Botbloq también estarán definidos de antemano (por ejemplo, en un robot industrial, mover el extremo en coordenadas cartesianas).

El usuario podrá elegir de una librería el tipo de robot que desea generar y, opcionalmente, proporcionar una serie de parámetros de configuración para determinar características escalables de ese tipo. Por ejemplo, en el caso de un robot serpiente se podría elegir el número de servos con que se desea construir (o, alternativamente, la longitud de la serpiente). Los tipos de robots predefinidos podrán ser los que se indican en el apartado 4.

3.2. Generación manual

Un usuario podrá crear desde cero un robot seleccionando los bloques básicos de diseño de una librería y ajustando sus parámetros manualmente. Para ello se generará una interfaz que permita realizar dicha construcción. La interfaz será similar a un programa de CAD en el que se puedan manipular los bloques para conectarlos unos con otros y cambiar el tamaño de aquellas partes que sean escalables, tales como los bloques pasivos (eslabones).

Botbloq en este caso no creará bloques de programación asociados a funcionalidades del robot, sino que el propio usuario podrá realizar la programación desde bajo nivel (e.g. mover articulación 2). Los bloques necesarios para programar los robots generados manualmente estarán incluidos ya en las librerías actuales de Bitbloq, (e.g. Servo), incluidas cualesquiera nuevas funcionalidades debidas a sensores o actuadores adicionales que no estuviesen incluidos en las versiones actuales de Bitbloq y se añadan al diseño de la arquitectura modular.

3.3. Generación Semi-Automática

La generación se realizará en base a las tareas que el usuario desee que realice el robot. El usuario determinará, por ejemplo que desea construir un robot que pueda jugar al fútbol. Esa información se pasará a través de una base de datos de conocimiento que determine las implicaciones morfológicas que tiene esa tarea:

- *Debe ser capaz de desplazarse*, por lo que necesitará una base móvil. El sistema inteligente daría la elección al usuario de crear un robot con ruedas o patas.

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

- *Debe poder golpear una pelota*, por lo que también necesitará un mecanismo de golpeo. En caso de ser un robot con patas, las propias patas podrían servir. Si se solicitó un diseño con ruedas, sería necesario añadir un mecanismo adicional.

- *Debe detectar el objeto* que tiene que golpear, por lo que además necesitará un sistema de percepción, ya sea un sistema de visión, un micrófono para detectar un sonido proveniente de la pelota o un sistema de localización basado en sensores ultrasonido, por ejemplo. Este sistema puede necesitar información adicional, como el color de la pelota, el sonido que produce o su tamaño para poder identificarla.

Para generar este sistema inteligente, desarrollaremos una ontología, basada en el estándar IEEE ORA [9, 6] que nos permita relacionar acciones con elementos de un sistema robótico a través de un proceso de inferencia. Un sistema similar es el proyecto RoboEarth [13], un internet para robots que permitía a cualquier robot descargar recetas [11] para la realización de tareas, las cuales se desglosaban en acciones y a su vez en primitivas que tenían asociados unos requisitos mínimos por parte del robot que quería realizar la acción. De este modo se podía determinar si un robot podía realizar una tarea o no, e.g. cocinar tortitas de maíz [7]. Este motor de inferencia que determina si un robot puede hacer una tarea y cómo hacerla se conoce como KnowRob [10]. Nuestra tarea, sin embargo, consiste en desarrollar el proceso de inferencia inverso: queremos crear un robot que tenga unas determinadas funcionalidades.

Además de la generación del robot, se añadirán una serie de parametrizaciones opcionales que puedan ser indicadas por el usuario en caso de así desearlo, tales como la velocidad del movimiento del robot, o el tamaño que se desea tener. Una vez generado el robot, BOTBLOQ asociará los bloques correspondientes a las acciones que el robot pueda realizar y creará bloques de tarea como programas de acciones de modo que el robot pueda ser programado de la manera más sencilla.

3.4. Otras consideraciones del esquema de diseño

Para evitar diseñar por completo el sistema de representación de robots, así como el control a bajo nivel de los componentes que los componen, se ha decidido

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

apoyarse para ello en ROS (*Robot Operating System* [5]). Este es un entorno de trabajo de código libre que ha tomado un gran auge en la última década y que trata de unificar el conocimiento robótico en un único repositorio, de modo que existan un gran número de algoritmos avanzados disponibles, e.g. SLAM (*Simultaneous Location and Mapping*), para cualquier usuario sin necesidad de volver a programarlos. Esto unido a la existencia de drivers para la mayor parte de los componentes habituales en robótica móvil (e incluso controladores para robots industriales como el KUKA LWR) y a su carácter multiplataforma (es un entorno nativo de Linux Ubuntu, pero existen versiones para otros SSOO, algunos embebidos), hace que sea una plataforma ideal y que la comunidad que trabaja en ROS aumente con cada nueva versión. A continuación se comentan algunos de los elementos que utilizaremos de ROS.

3.4.1. Definición de la estructura física

La generación del robot incluirá la creación de los ficheros URDF, (*Unified Robot Description Format*) y SRDF (*Semantic Robot Description Format*) que lo describen. Para definir la estructura física del robot aprovecharemos el formato URDF que incorpora ROS. Existen numerosas aplicaciones y entornos que también utilizan ese estándar, por lo que resulta exportable a otros entornos (e.g. *My Model Robot* [4]). En él tendremos una especificación de bajo nivel de cómo están conectados y de las relaciones cinemáticas entre los diferentes módulos del robot. En el Anexo III de este documento se detalla más información acerca de los ficheros URDF y se muestran algunos ejemplos ya desarrollados que serán parte de la librería de robots prediseñados de la arquitectura modular. Actualmente ROS dispone del paquete *MoveIt!* [3] que permite generar estos ficheros URDF de forma sencilla y que ya cuenta con descripciones de robots como Baxter de Rethink Robotics, PR2 de Willow Garage o YouBot de KUKA, entre otros muchos.

3.4.2. Definición semántica

El significado semántico estará definido utilizando un formato basado en SRDF. Sin embargo, puesto que es bastante limitado, será necesario ampliarlo o bien crear nuestro propio metalenguaje. En él se definirá semánticamente un robot (piernas, brazos, manos, chasis, etc.) según la ontología que se va a desarrollar, de modo

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

que podamos asociar a cada elemento semántico las acciones que son realizables por él, garantizando que dicho robot pueda realizar las tareas que son requeridas en la generación semi-automática.

3.4.3. Control del Robot

Los componentes de los robots generados mediante la estructura modular serán controlados en ROS. Esto implica que la placa controladora del robot debe incluir un hardware que permita correr un Linux embebido que tenga distribución de ROS. Como ya se ha indicado, existen varias opciones con las que se está trabajando actualmente en paralelo, como Raspberry PI, Beagle Bone Black e Intel Edison, siendo esta última la preferida por el momento debido a su factor de forma. El proceso de instalación de Linux (Ubilinux) y ROS (Indigo) se detalla en el Anexo II de este documento. A este respecto, se han diseñado aplicaciones en ROS para controlar un robot de cinemática diferencial (Renacuajo de BQ) y se está trabajando en la interfaz BOTBLOQ-ROS para la generación automática del código que tiene que ejecutarse en la placa controladora del robot a partir de la programación de bloques.

4. Listado de robots

En base al estudio realizado en la Sección 3 del Entregable E4.1: Informe detallado del estado del arte, se ofrece un listado de una serie de robots que se consideran representativos para el espectro de usuarios potenciales. ya que estos requisitos tienen su reflejo en las características de los robots actuales. En concreto, se ha realizado una clasificación de los robots siguiendo la propuesta de la IFR (*International Federation of Robotics*). Una vez realizada, se han detectado las características comunes de diferentes clases de robots y los parámetros que los definen. Estos modelos serán simulados en GAZEBO, el simulador interactivo de ROS (*Robot Operating System*) como base para ser reproducidos con nuestra estructura modular, primero mediante simulación y luego con una implementación física.

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

4.1. Robots industriales

La clasificación de los robots industriales según su morfología y sus gdl (grados de libertad) es la siguiente (para una información más detallada, consultar la Sección 3 del Entregable E4.1):

- **Robot antropomorfo**: (6 gdl) Tres articulaciones de torsión (1, 4 y 6) y 3 articulaciones angulares (2, 3 y 5).
- **Robot cartesiano**: (3 gdl + efector) Tres articulaciones prismáticas que producen movimiento en los ejes cartesianos (XYZ) y un efector final específico de la aplicación.
- **Robot SCARA**: (4 gdl) Tiene dos articulaciones de rotación angular y una articulación cilíndrica.
- **Robot paralelo**: (6 gdl) Poseen varias cadenas cinemáticas cerradas. Dos tipos: Delta, posee 3 o 4 cadenas cinemáticas cerradas que se unen en la exrtremidad del robot con articulaciones de rotación angular; y Plataforma de Stewart, con seis cadenas cinemáticas paralelas formadas por una articulación prismática y una articulación esférica.
- **Robot cilíndrico**: (3 gdl + efector) Una articulación cilíndrica y otra prismática. Sistema de coordenadas cilíndricas.
- **Robot esférico o polar**: (3 gdl + efector) Una articulación de torsión, otra angular y una última prismática. Sistema de coordenadas esféricas.

La robótica colaborativa [2] persigue la idea de la HRI de la forma más versátil y segura posible. Si bien esta idea es la base de la robótica de servicios, en la robótica industrial no se ha incorporado hasta hace relativamente poco tiempo al diseño de robots, e.g. Baxter de Rethink Robotics.

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

(a) Antropomorfo: Stäubli-TX60

(b) SCARA: OMRON-SCARA-XG

(c) Paralelo: ABB-FlexPicker

(d) Cilíndrico: Hudson-PlateCrane

(e) Esférico: UNIMATE-2000

(f) Colaborativo: Baxter

Figura 38: Robots industriales.

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

4.2. Robots educativos

Esta sección se subdivide en robots con patas y robots con ruedas, que es la división habitual de los robots móviles.

4.2.1. Robots con ruedas

A continuación se enumeran los tipos de robots rodantes clasificados en función de su número de ruedas/apoyos con el suelo. Una descripción más completa se puede encontrar en la Sección 3 del Entregable E4.1 o en [8].

- **Robots de una rueda**: Un solo contacto con el suelo, por lo que necesitan de un sistema activo de control (e.g. Murata Girl). Existen variantes cuyo punto de apoyo es una esfera (e.g. BB-8).

- **Robots de dos ruedas**: Dos configuraciones principales:

  En línea: Son bicicletas automáticas (e.g. Murata Boy).

  En paralelo: Pueden tener el centro de gravedad por encima del eje, en cuyo caso son péndulos invertidos (e.g. Segway), o en línea, con lo que son estables (e.g. Parrot Jumping Sumo).

- **Robots de tres ruedas**: Existen diferentes configuraciones según qué ruedas sean tractoras y/o directrices. Las configuraciones más habituales (diferencial o triciclo) no son holonómicas, por lo que no se pueden desplazar en cualquier dirección (e.g. Renacuajo). Sin embargo, existen configuraciones holonómicas (omnidireccional o synchro-drive) que sí lo permiten (e.g. Rovio de WowWee Robotics).

- **Robots de cuatro ruedas**: Tienen un mayor polígono de estabilidad, pero presentan otros problemas: deben estar muy bien ajustados o poseer un sistema de suspensión para evitar que una de las cuatro ruedas quede en el aire; deben poder sincronizar los centros de giro o se desplazarán con deslizamiento (e.g. Nanokhod). También existen configuraciones holonómicas de cuatro ruedas (e.g. KUKA YouBot, RWI B21).

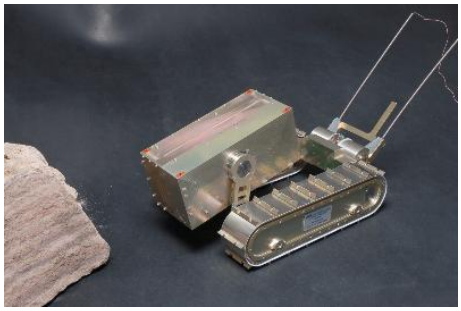
- **Robots de seis ruedas**: Se utilizan más ruedas para conseguir que el robot sea capaz de superar obstáculos (e.g. rovers espaciales como Curiosity).

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

(a) Una rueda: BB-8

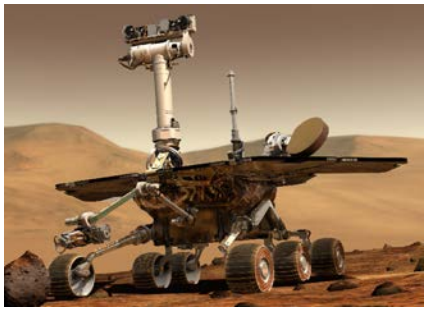
(b) Dos ruedas: Segway

(c) Tres ruedas: Renacuajo



(d) Oruga: Nanokhod

(e) Cuatro ruedas: YouBot



(f) Seis ruedas: Curiosity

Figura 39: Robots educativos con ruedas.

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

4.2.2. Robots con patas

Seguidamente se realiza una clasificación de los robots con patas en función del número de apoyos con el suelo. De nuevo, se puede encontrar una descripción más detallada en la Sección 3 del Entregable E4.1.

- **Robots de una pata**: Poca aplicabilidad. Son robots de investigación como Hopper del MIT.
- **Robots bípedos**: Los robots con dos patas son muy interesantes por su similitud con la locomoción humana. Existen modelos sencillos (menos articulaciones), pero necesitan al menos 3 servos por pata para tener un movimiento más o menos realista.
- **Robots con tres patas**: Son muy poco habituales, ya que no tienen estabilidad estática, y su movimiento es muy particular, e.g. STriDER.
- **Robots cuadrúpedos**: Son la versión más estable de los robots bípedos, ya que poseen estabilidad estática si se mueve una sola pata en cada paso. La complejidad de la pata suele ser algo menor que en ellos, teniendo habitualmente hasta un máximo de tres servos por pata.
- **Robots hexápodos**: Tienen la ventaja de aumentar el polígono de estabilidad hasta un hexágono. El movimiento se suele hacer mediante tríos de patas a la vez.

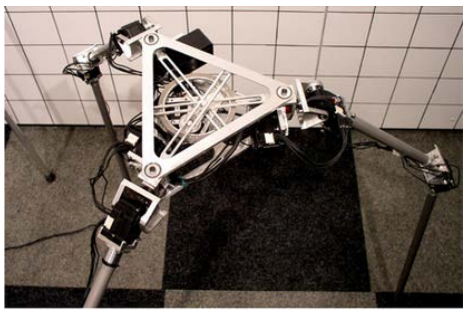
4.3. Robots de investigación

Se comentan varios tipos de robots utilizados en investigación y que poseen características particulares en cuanto a su morfología, lo que les hace interesantes para el análisis.

- **Robots serpiente**: Son robots ápodos eminentemente modulares. Se forman por la concatenación de unos módulos básicos con un servo, y sus movimientos se generan mediante el desfase en la oscilación de los servos de cada módulo básico, e.g. Anna Konda.

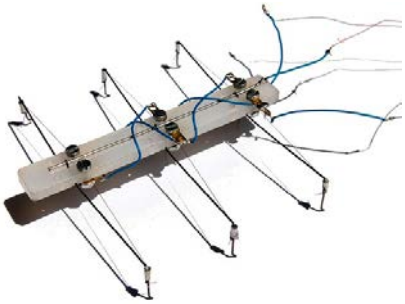
Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

(a) Dos patas: RoboBuilder



(b) Tres patas: STriDER

(c) Cuatro patas: Aibo



(d) Seis patas: Stiquito

Figura 40: Robots educativos con patas.

- **Robots flexibles**: Estos robots aprovechan la flexibilidad en algún sentido, ya sea como sensor para medir distancia en función de su deformación, para facilitar el ensamblado gracias a la elasticidad en las articulaciones o para conseguir un método de locomoción particular, como es el caso de RHex.

- **Robots humanoides realistas**: Robots de tamaño humano real con actuadores hidráulicos. Están siendo objeto de investigación intensiva en la actualidad para poder colocarlos a trabajar en entornos diseñados para humanos, bien de modo colaborativo, o bien en entornos peligrosos, e.g. Atlas, HUBO.

- **Robots cuadrúpedos**: Son versiones más rápidas y fuertes de los cuadrúpedos educativos. Se utilizan como métodos de transporte de carga, e.g. LS3 (BigDog).

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

(a) Serpiente: Anna Konda

(b) Patas flexibles: X-HRex

(c) Humanoide: HUBO

(d) Cuadrúpedo: BigDog

Figura 41: Robots de investigación.

5. Anexo I. Métodos de comunicación entre módulos

Los sistemas robóticos presentan un gran número de sensores y actuadores, que deben conectarse entre ellos mediante cables. Obviamente la presencia de cables provoca limitaciones a la hora de diseñar y fabricar la estructura de un robot, así como problemas de fiabilidad y mantenimiento.

Estos problemas son más acuciantes si cabe en un robot modular, donde debe primar la libertad a la hora de construir cualquier forma posible con los diferentes bloques de los que se dispone.

El cableado puede dificultar los métodos de ensamblado entre bloques activos.

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

También a lo largo de los bloques pasivos habrá que extender cables. Por estas razones, se van a proponer algunas soluciones para reducir en la medida de lo posible el cableado. La principal es unificar las señales de datos y de alimentación en un único cable, permitiendo así poder realizar comunicaciones con tan solo dos cables, alimentación/datos y GND.

En el caso del protocolo de comunicaciones I2C, el más extendido en robótica modular, contamos con cuatro cables (VCC, SCL, SDA,GND), por lo que se podría reducir a tres cables, uniendo SDA y VCC.

Para ver si es posible llevar a cabo esto de una manera óptima y que permita la integración en nuestro proyecto, se va a realizar un estado del arte relatando tecnologías ya existentes que tengan principios de funcionamiento similares a lo que buscamos, así como se van a analizar otras soluciones propuestas.

5.1. Powerline Communication

Powerline Communication (PLC) se trata probablemente del método para transmitir información sobre una línea de alimentación más extendido del mercado. Esta tecnología permite enviar datos a través de la red eléctrica, permitiendo utilizar así toda la infraestructura eléctrica existente.

El protocolo de comunicación PLC no es la solución que buscamos ya que emplea señales AC para propagar la información en lugar de señales DC, pero debido a su importancia merece la pena comentarlo brevemente.

Los cables eléctricos transmiten señal a 50 o 60 Hz como fuente de alimentación, mientras que para transmitir datos la señal PLC va modulada entre 1,6 y 40 MHz dependiendo del sistema. El funcionamiento básico esta descrito en la Figura 42.

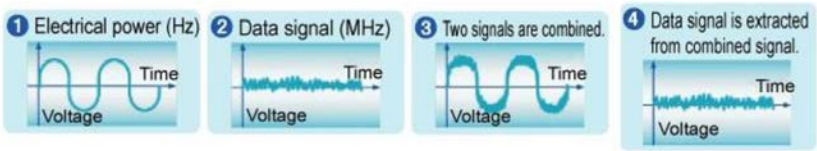


Figura 42: Esquema del funcionamiento de PowerLine

Principalmente se usan tres tipos de modulación, DSSSM (Direct Sequence Spread Spectrum Modulation), OFDM (Orthogonal Frecuency Division Multiplex) y GMSK (Gaussian Minimum Shift Keying).

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

Debido a su gran velocidad de transmisión, de entre 14 Mbps a 45 Mbps, los usos de la tecnología PLC son muy variados, por ejemplo para cables de telefonía, televisión por cable, domótica (siendo especialmente conocido el sistema X-10), acceso a internet, sustituyendo a las redes Ethernet, etc.

Existen multitud de soluciones comerciales para implantar redes PLC, principalmente dispositivos en forma de modem.La gran desventaja que presenta la tecnología PLC es que depende del ancho de banda de las redes eléctricas, que están reguladas por normativa vigente en cada país, siendo necesario modificar la tecnología dependiendo de la zona en la que nos encontremos. Por ejemplo, en Europa el comité CENELEC permite transmisiones en el rango de 3 a 148.5 kHz, mientras que en EEUU las transmisiones se hacen en el rango de 10 a 450 kHz.

5.2. SISTEMAS DC POWERLINE UTILIZANDO TRANSFORMADORES DE TRANSMISIÓN LINEAL.

En [12] se propone una solución para crear redes con un gran número de nodos (sensores y actuadores) usando solamente una línea de corriente continua, cuya principal aplicación puede ser la implementación en la robótica.

Este método de comunicación se aplica a un sistema cerrado en el que todos los nodos comparten la misma fuente de alimentación DC. Los datos se codifican, se modulan y se superponen a la señal de alimentación en forma de señal de alta frecuencia. Para ello cada módulo debe contar con su propio modem. Estos modem deben ser construidos específicamente para cada nodo, teniendo en cuenta sus características de carga, impedancia y ruido. A la hora de transmitir datos sobre una señal DC hay que tener en cuenta las siguientes consideraciones:

- La señal de datos puede ser atenuada significativamente debido a que las fuentes de DC presenten una baja impedancia, lo cual puede provocar que la corriente del sistema fluye por la propia fuente en lugar de por la señal de transmisión. Además,los datos también pueden verse afectadospor los reguladores de la fuente. Para solucionar esto, se propone utilizar frecuencias de transmisión elevadas (por encima de los 100 kHz), ya que se ha demostrado que la impedancia de los componentes lineales de las fuentes DC tienen cierta dependencia de la frecuencia. De esta manera, a frecuencias elevadas se presentará una impedancia elevada que solucionará los problemas de

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

atenuación.

- La presencia de ciertos componentes en las fuentes DC generan ruido, provocando que la señal de datos pueda corromperse. Experimentalmente se ha demostrado que los niveles de ruido generado dependen también de la frecuencia del sistema, y por esto pueden ser predichos. Basta con elegir frecuencias de trabajo en la que la presencia de ruido sea lo menor posible.

Una vez determinado que la comunicación a través de una señal DC es posible, se presenta un esquema de la arquitectura del sistema (ver Figura 43)

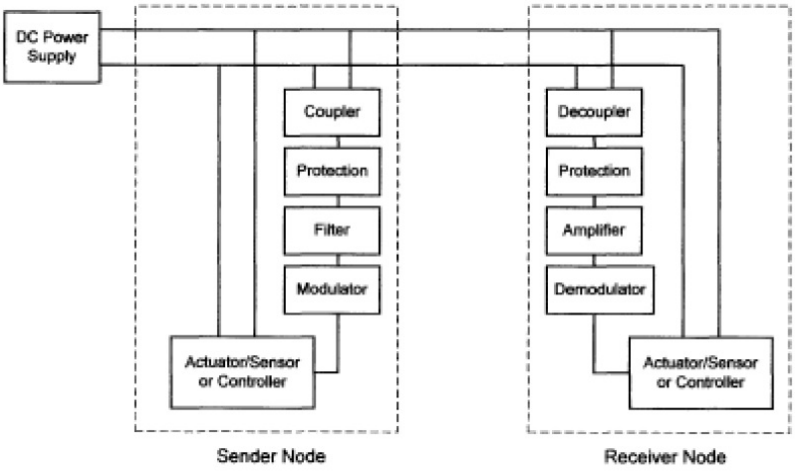


Figura 43: Esquema del funcionamiento de PowerLine

Como ya se ha comentado, cada nodo debe tener su propio módem. Estos módems deben tener como componentes fundamentales moduladores, demoduladores, amplificadores y filtros. Además se deben incluir acopladores y desacopladores, así como elementos de protección. Los circuitos típicos de acoplo y desacoplo utilizados en sistema de powerline communication AC no se pueden emplear en este sistema ya que a altas frecuencias los transformadores que contienen presentan gran cantidad de efectos parásitos y de resonancia, alejándose enormemente de su comportamiento en condiciones ideales. Estos efectos provocan atenuaciones en la señal de datos, que serán más grandes a medida que incrementamos los nodos del sistema. Este efecto se conoce como pérdidas de inserción.

Para solucionar las atenuaciones provocadas por los circuitos de acoplo y desacoplo se propone utilizar la teoría de los transformadores de transmisión lineal

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

(en inglés transmission line transformer, TLT). Un sistema TLT básico transmite la energía a la carga utilizando el modo de transmisión lineal transversa. Son mucho menos susceptibles a las pérdidas de inserción, ya que por su estructura la corriente siempre se entrega a la carga.

Solucionado el problema de las pérdidas de inserción, hay que asegurar que la corriente fluye desde el módem hasta la parte receptora, y no hacia la fuente de alimentación. Para ello se utiliza una configuración de TLT llamada Guanella TLT. Los Guanella TLT pueden transformar la impedancia, haciendo que la impedancia de la fuente de alimentación sea mucho mayor que la de los receptores.

El diseño de los Guanella TLT está regido por el comportamiento de los parásitos. Sintonizando estas corrientes parásitas junto con los condensadores se puede igualar la impedancia de las partes emisoras y receptoras maximizando así la transmisión de energía. No hay un modelo standard de diseño de la Guanella TLT, cada uno debe ser realizado cuidadosamente para cada aplicación en concreto. Aquí se ha partido del modelo de Guanella TLT Kuo, que tiene en cuenta las corrientes de magnetización.

Los valores de L_m y R_o son críticos para el funcionamiento del sistema, teniendo que ser cuidadosamente calculados en función de las cargas y del número de nodos con los que vaya a contar el sistema. A continuación se muestra un esquema que muestra un sistema con un nodo emisor y n nodos receptores (ver Figura 44).

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

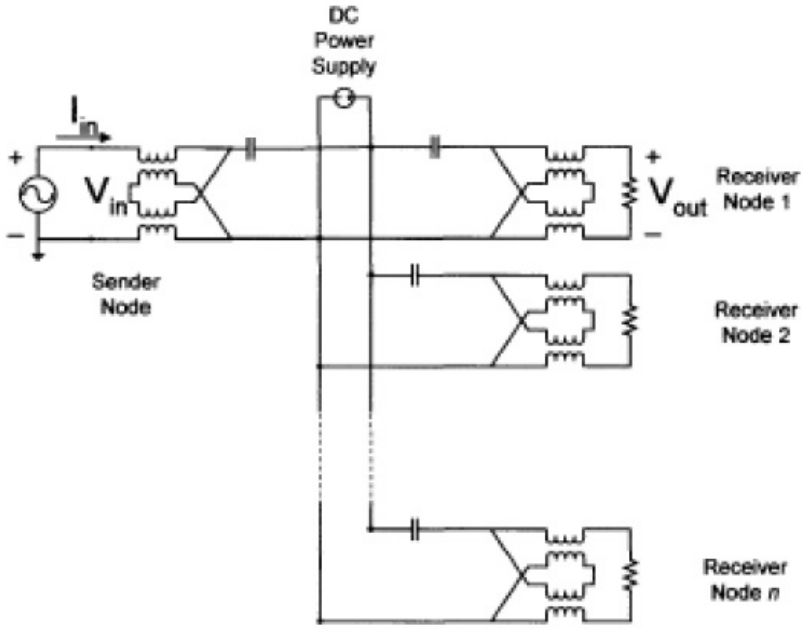


Figura 44: Esquema del funcionamiento Guanella TLT

Para comprobar los resultados, se llevó a cabo una prueba con un sistema que simulaba a un robot humanoide. Este sistema constaba de una fuente de 12 VDC, 30 nodos actuadores (con sus correspondientes módems) y cable coaxial de 75ohm.

Una vez analizado se puede llegar a la conclusión de que no sería posible trasladar este sistema a nuestro proyecto debido al carácter abierto de la arquitectura modular del mismo. Se necesitaría un diseño específico para cada uno de las configuraciones que quedamos construir con nuestro robot, lo cual no es viable. Y aún sin tener en cuenta lo anterior, la elevada complejidad y dificultad del diseño haría preferible buscar otras alternativas menos complejas a nuestro problema

5.3. BIAS-T

Bias-T o bias tee son circuitos diseñados para inyectar corrientes o voltajes DC en circuitos RF.

Un circuito RF es un circuito que opera con frecuencias en el espectro de las radiofrecuencias (de 3 kHz a 300 GHz). Son empleados es dispositivos que requieren transmitir o recibir ondas de radio. Los circuitos bias-T son utilizados

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

en los circuitos RF para alimentar las antenas y amplificadores necesarios en la transmisión.

Se denominan bias-T debido a su forma de "T", y se venden ya encapsulados de manera comercial. Debido a sus aplicaciones, están preparados para usar cable coaxial.

La estructura del circuito es sencilla (ver Figura 45), se coloca una inductancia en serie con la fuente de DC y se coloca un condensador en paralelo a la señal de datos. La bobina impedirá que la corriente DC fluya a través de la señal de datos y el condensador hará que la señal de datos no fluya a través de la fuente DC.

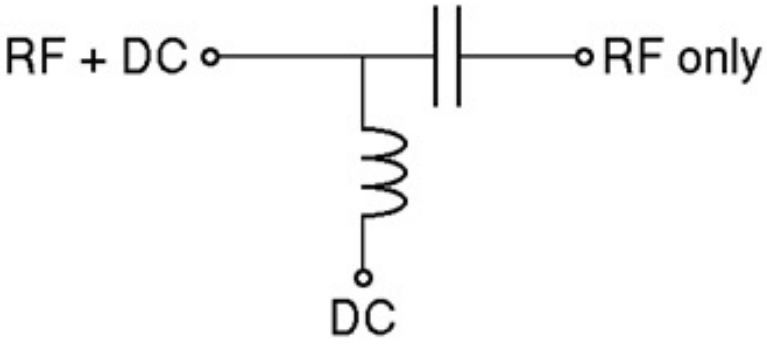


Figura 45: Esquema del funcionamiento Bias T

La señal resultante contiene información contiene la alimentación e información de la señal de datos. Sin embargo, esta señal no se puede leer directamente. En la resultante, la señal de datos, que tiene forma de pulso cuadrado, aparecerá en forma de picos. Por esto, es necesario un software que sea capaz de leer esta nueva señal de datos.

Otro de los problemas que presenta el circuito bias-T es que es muy susceptible al ruido, por lo que es recomendable modular la señal de datos antes de enviarla. Lo más habitual en estas aplicaciones es la modulación por desplazamiento de frecuencia (FSK).

La modulación FSK es una técnica de modulación para la transmisión digital de información, donde la señal moduladora solo varía entre dos valores de frecuencia para transmitir un 0 o un 1 (ver Figura 46).

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

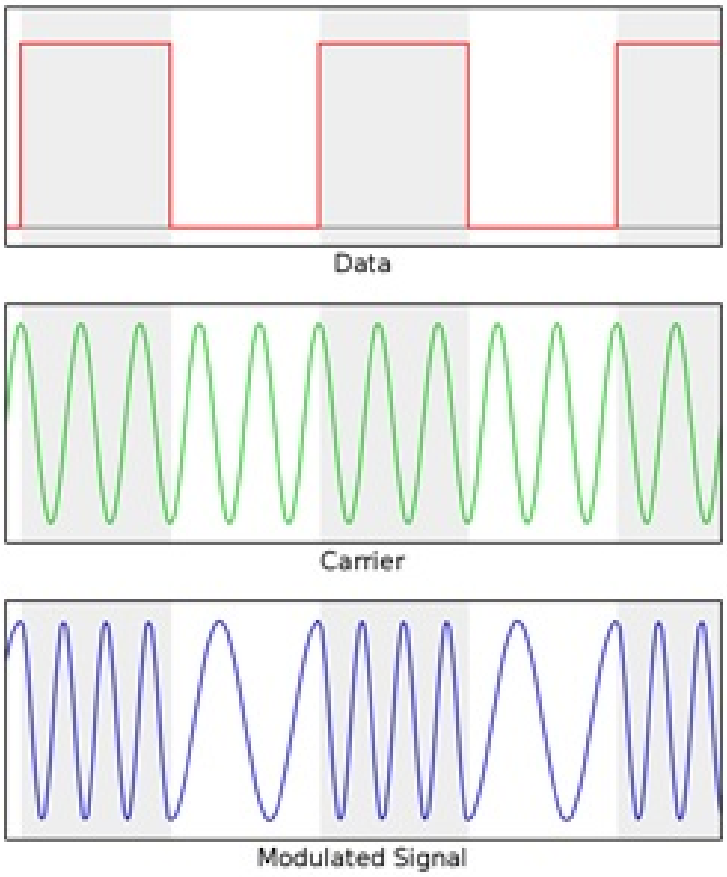


Figura 46: Funcionamiento Bias T

Una vez analizado el funcionamiento de los bias-T se propone el análisis de otro circuito con unos principios de funcionamiento similares (Figura 47)

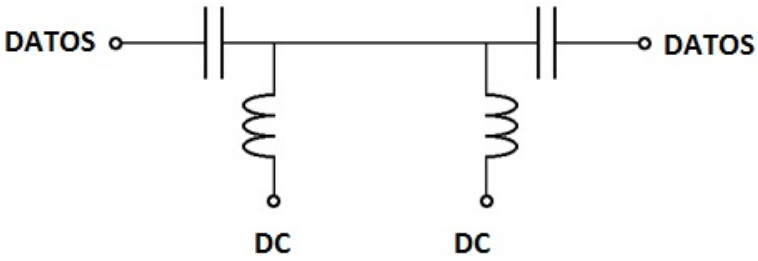


Figura 47: Esquema prueba realizada

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

En la primera parte del circuito las señal de datos se acoplaría a la alimentación, como ya hemos visto en el bias-T, y en la segunda la señal se desacoplaría separando ambas señales nuevamente. Se representa el sistema en multisim (Figura 48).

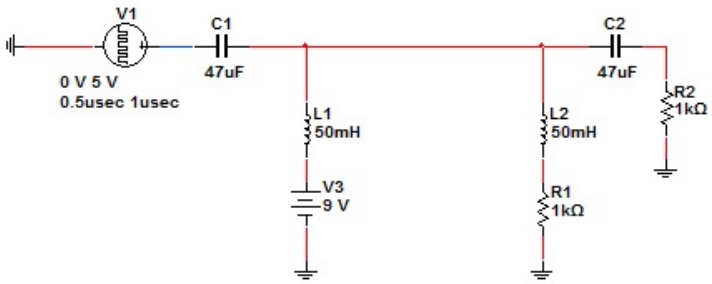


Figura 48: Esquema realizado en Multisim para este proyecto

Se utiliza una señal cuadrada de 100 kHz y 5V de amplitud, una fuente DC de 9V, dos condensadores y dos bobinas. Las resistencias representan las impedancias de los terminales que se conectarían a la salida del circuito. Para los condensadores se ha utilizado un valor de 47uF y para las inductancias un valor de 50mH. Estos valores se han determinado a través de numerosas simulaciones, eligiendo los valores que nos dan una respuesta más adecuada. También se ha comprobado que tener valores iguales par a par es lo mejor para dar uniformidad a la respuesta. Se analiza la respuesta con un osciloscopio (ver Figura 49). La señal de alimentación no nos da el valor de voltaje correcto, oscilando en torno al original. La línea azul represente la señal de alimentación original y la roja la alimentación tras el circuito

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

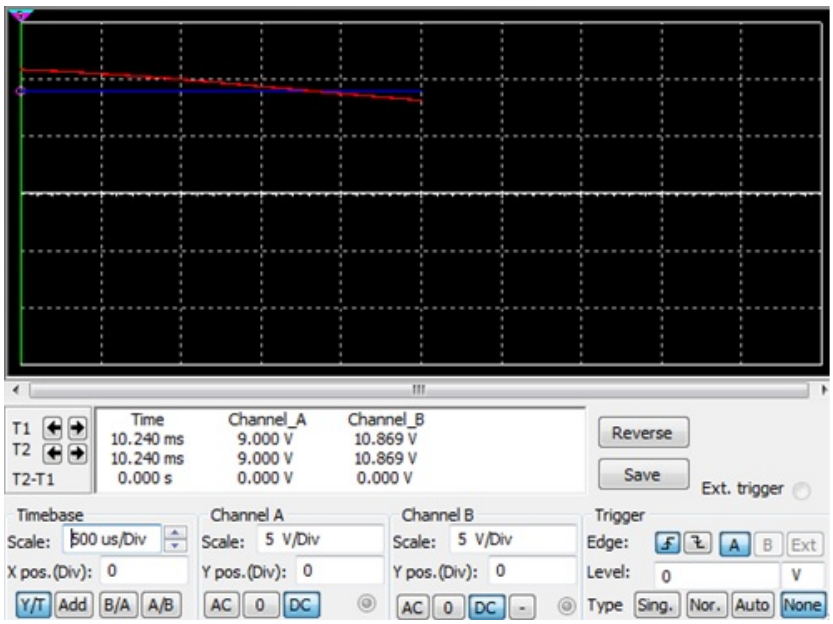


Figura 49: Simulación realizado en Multisim para este proyecto

En cuanto a la señal de datos (ver Figura 50) tenemos que la forma en la señal original y en la salida es igual, pero los valores de voltaje tampoco son correctos, oscilando entorno a los originales de la misma manera que en la señal de alimentación.

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa
Operativo Plurirregional de Crecimiento Inteligente 2014-2020

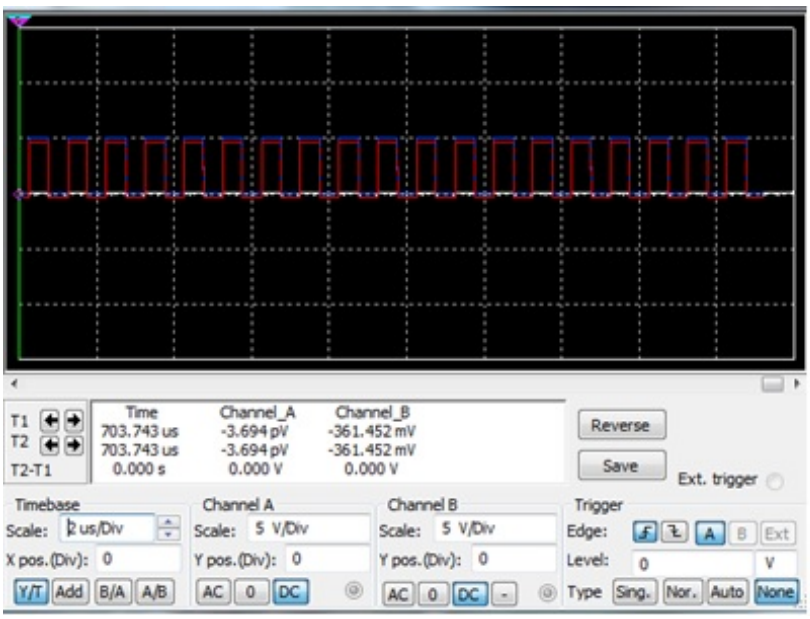


Figura 50: Simulación realizado en Multisim para este proyecto

Si dejamos que evoluciones temporalmente se puede ver la oscilación en los valores de voltaje.Para solucionar este problema se podría poner un regulador de voltaje a la salida del circuito. La solución a la oscilación en la señal de datos ya es más compleja. Si conectamos más de un nodo a la salida del circuito (recordemos que en nuestro robot modular tendríamos una entrada y un número variable de salidas) la respuesta no varía mucho. Por esto, esta sería una posible solución a implementar, siempre recordando que el ruido afecta notablemente a la respuesta, por lo que probablemente se necesitase una modulación previa de la señal.

5.4. 1-WIRE

1-Wire es un protocolo de comunicaciones diseñado por Dallas Semiconductor Corp. Proporciona transmisión de datos y alimentación en una misma señal. Es decir, nos permite realizar montajes con tan solo dos cables, uno para alimentación y datos y otro para la tierra. Las redes basadas en el protocolo 1-Wire (denominadas redes MicroLAN) siguen el modelo de comunicación maestro-esclavo. El dispositivo que actúa como maestro puede comunicarse con uno o varios dispositivos esclavos a través de una única línea. La topología de esta red es tipo bus.

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

Como maestro se puede usar cualquier PC o microcontrolador. Sin embargo, y aquí es donde se encuentra la principal restricción de este sistema, es que los esclavos solo pueden ser dispositivos diseñados específicamente para este tipo de redes. Los dispositivos 1-Wire más habituales son sensores digitales de bajo coste empleados para las medidas meteorológicas, como por ejemplo el sensor de temperatura DS18S20. Por esto, uno de los usos más frecuentes del 1-Wire es montar redes de sensores. Otro de los inconvenientes del 1-Wire es que su velocidad de transmisión, de hasta 16,3 kbps, es bastante baja. Otros protocolos de comunicación, como por ejemplo el I2C, ofrecen velocidades de 100 kbps en su modo standard. Sin embargo, se trata de una tecnología muy extendida. Para arduino existen multitud de librerías y ejemplos que permiten trabajar con este tipo de redes con relativa facilidad. Existen dos modos de alimentación:

- Modo de alimentación parásita (parasite power mode). En este modo solo necesitamos dos cables (Datos/alimentación y GND). Es necesario conectar una resistencia de pull-up de 4,7 kohm al bus. Cada dispositivo dispone de un condensador (de unos 800pF) que se carga cuando la línea se encuentra en estado alto. De esta manera, los dispositivos permanecen alimentados cuando la línea está en estado bajo.

- Modo normal. Necesita de alimentación externa. Utiliza tres cables, alimentación, datos y GND. También es necesario conectar una resistencia de pull-up de 4,7 kohm al bus.

5.5. DIGITAL COMMAND CONTROL

Digital Command Control o DCC es un sistema utilizado para operar líneas de modelismo ferroviario. A pesar de su aplicación, puede resultar interesante analizar sus principios de funcionamiento. Una estación de control DCC modula el voltaje de los raíles (que esencialmente se trata de dos cables) para codificar información de manera digital a la vez que proporciona alimentación.

Esto se consigue con una señal DC bipolar. La estación alterna rápidamente entre el valor de voltaje negativo (-Vcc) y positivo (+Vcc) consiguiendo así una corriente alterna con forma de onda cuadrado. Para enviar un 1 se envía un pulso corto (58us) y para enviar un 0 se usa un pulso más largo (100us)

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

Cada elemento en el circuito incorpora en su entrada un rectificador de voltaje lo que hace posible alimentar a cada componente con un voltaje +Vcc.

Cada elemento también consta de un decodificador DCC al que le corresponde un número de dirección único. De esta manera, el maestro (la estación) puede enviar información a cada uno de los esclavos (trenes) de manera individual transmitiendo a través de la vía (bus). Las instrucciones del maestro tienen la siguiente estructura de la Figura 51.

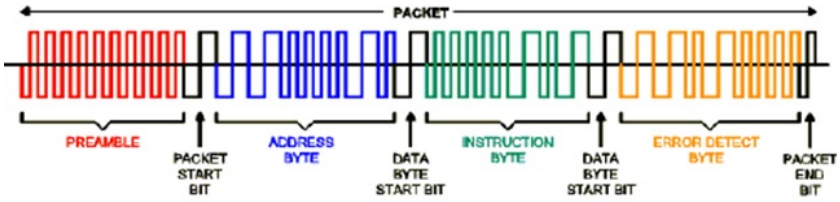


Figura 51: Protocolo DCC

De esta manera se puede montar una red maestro-esclavo con multitud de esclavos con tipología en bus con tan solo dos cables (uno para el voltaje negativo y otro para el negativo).

5.6. FILTRO PASO ALTO/PASO BAJO ENTRE TERMINALES DE ALIMENTACIÓN Y DATOS

Con este montaje (ver Figura 52) proponemos una alternativa distinta a la de unificar alimentación y datos en una misma línea. Esta solución consiste en colocar un filtro entre los terminales de alimentación y datos de manera que se obtenga una referencia de voltaje que pueda utilizarse como tierra en los dispositivos que se vayamos a comunicar. De esta manera, solo sería necesario desplegar dos cables para alimentación y voltaje ahorrándonos el cable para GND.

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

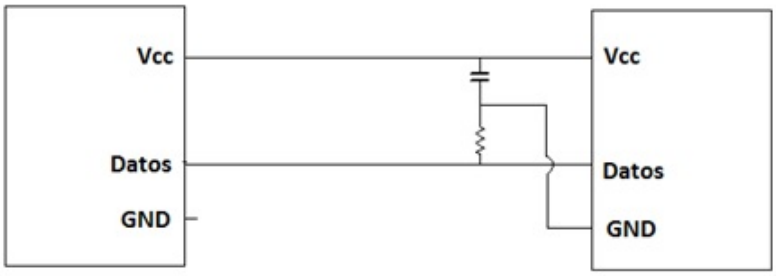


Figura 52: FILTRO PASO ALTO/PASO BAJO ENTRE TERMINALES DE ALIMENTACIÓN Y DATO

El filtro consiste simplemente en un condensador y una resistencia de un determinado valor. Desde el punto de vista del terminal Vcc se trataría de un filtro paso bajo mientras que para el terminal de datos se trata de un filtro paso alto.

Se realizan simulaciones mediante el software Multisim para comprobar la validez de la solución (ver Figura 53).

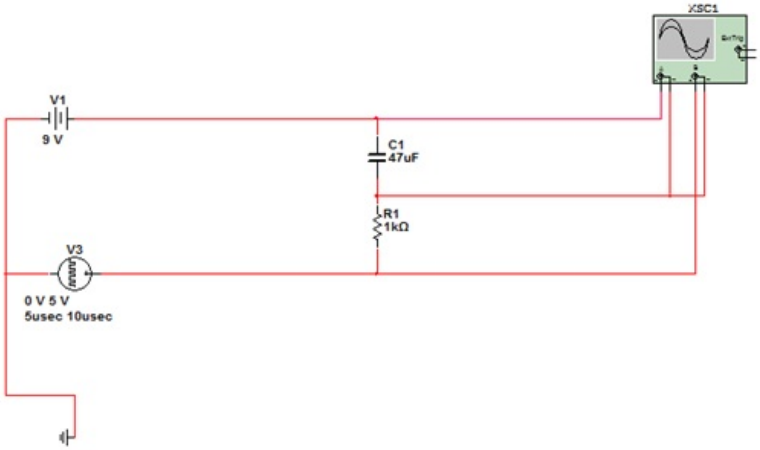


Figura 53: Esquema Filtro en Multisim

Para representar la señal de datos se ha usado una fuente que suministra un pulso cuadrado con un periodo de 10 us y una amplitud de 5V. Para seleccionar la frecuencia de trabajo se ha tomado como referencia la velocidad de 100 kbps del protocolo de comunicación I2C. Como señal de alimentación se emplea una fuente de corriente continua de 9V. En la parte derecha se ha colocado un osciloscopio

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

para representar las señales resultantes. Para elegir los valores de la resistencia y del condensador se efectúan pruebas con distintos valores usando como referencia la frecuencia de trabajo de la señal de datos y la constante de tiempo del filtro t. Los resultados aparecen en las Figuras 54, 55, 56.

Figura 54: Resultado simulacion Filtro 1 en Multisim

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

Figura 55: Resultado simulacion Filtro 2 en Multisim

Figura 56: Resultado simulacion Filtro 3 en Multisim

Se aprecia que las señales siguen teniendo la forma adecuada, pero los niveles de voltaje han disminuido en ambas señales, alcanzando voltajes negativos en la

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa
Operativo Plurirregional de Crecimiento Inteligente 2014-2020

señal de datos. Tras realizar varias simulaciones se determina que el decremento de voltaje se produce en la misma cantidad en las dos señales.

Una vez analizado el funcionamiento del circuito podemos llegar a la conclusión de que la solución propuesta no es adecuada ya que:

- Los valores de R y C dependen de la frecuencia con la que estemos trabajando
- La respuesta no es del todo estable
- Una vez estabilizada, la respuesta reduce los niveles de voltaje.

6. Anexo II. Demostrador ROS en EDISON con Robot Renacuajo

6.1. Introducción

Las necesidades del proyecto requieren la utilización de ROS, para ello, será necesario el uso de un sistema operativo en el que poder instalarlo. Aunque existen otras opciones (sin soporte, nada estables) el sistema operativo a utilizar será una distribución de Linux. Un ejemplo de placas de control que soportan ROS puede ser: Raspberri Pi, BeagleBone Black e Intel Edison. En este caso, se utilizará la Intel Edison, que por defecto tiene un software basado en Linux (Yocto). Sin embargo, no es trivial instalar y trabajar con ROS sobre Yocto, la alternativa más utilizada en la red es cambiar Yocto por Ubilinux (una variación de Debian). Ubilinux permite al programador realizar cambios más profundos sobre la Intel de una manera relativamente sencilla, sin perder las capacidades de Yocto.

Una vez configurado ROS en la IntelEdison se comprueba en el apartado 6.4 su utilidad para controlar el Robot Renacuajo de BQ como demostrador de su aplicabilidad.

6.2. Instalación de Ubilinux en la Edison

6.2.1. Introducción

La distribución Ubilinux fue desarrollada por la empresa EmutexLabs, está basada en Debian “Wheezy”. En un principio, no fue desarrollada pensando en la

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

Intel Edison, sin embargo, actualmente, la propia página de la empresa ofrece una imagen del firmware así como una guía de instalación en la Edison.

En este documento se van a seguir los pasos ofrecidos en la página, completando el proceso con algunos pasos que en ésta no aparecen y son necesarios.

6.2.2. Instalación de Ubilinux

En primer lugar, se descarga la imagen de Ubilinux para la intel EDISON de la página de EmutexLabs (desarrollador). A continuación, comienza el proceso de instalación. Será necesario realizar unos pasos previos (para eliminar el contenido de la Intel). Para ello, se conecta la placa al ordenador de trabajo y se corrobora que el dispositivo se monta en la ruta:

```
/media/>>(nombre de usuario)<</Edison
```

.

Desde la terminal:

```
$ cd /media/>>username<< Move to the directory
$ ls -lag Edison       Show files mounted in the Edison
$ rm -rf Edison/*      Erase visible files
$ ls -lag Edison        Verify the files have been removed
```

Instalación de paquetes necesarios.

```
$ sudo apt-get install dfu-util   Driver to connect devices over USB
$ sudo apt-get install screen   Needed to serial communication
```

Finalmente, se puede cargar el firmware de Ubilinux en la placa. Para ello en dos terminales se siguen los siguientes pasos:

1. Se descomprime el archivo que contiene la imagen. En la carpeta “toFlash” aparece el archivo llamado “flashall.sh” (o “flashall.bat”).

2. Se retira la alimentación de la placa (por completo, incluyendo el USB de alimentación) pero dejando el microUSB de la comunicación serie.

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

3. En una de las terminales, se debe abrir la comunicación serie con la Edison, para ello, escribir: ***sudo screen /dev/ttyUSB0 115200***. Con esta línea de comandos no sólo abrimos una comunicación en el puerto en el que está montada la Intel, también modificamos la velocidad a 115200 bauds. Esta terminal no es estrictamente necesaria para la instalación, pero nos permite hacer un mejor seguimiento de la misma. Además, una vez instalado el software en la placa, en esta terminal se podrá ver perfectamente que el proceso ha finalizado.

4. Se ejecuta el archivo "flashall.sh" en la otra terminal (desde el directorio: /.../toFlash, escribir: ***sudo ./flashall.sh***).

5. Se alimenta la placa. Veremos que el proceso comenzará. Ahora se debe esperar hasta que la instalación termine por completo. Cuando esto ocurra, la placa se reiniciará. **6.** Los paquetes de la Wi-Fi están instalados, pero es necesario realizar los siguientes pasos para activar la red.

PASO 1. Necesario estar conectado vía Puerto Serie (sudo screen /dev/ttyUSB0 115200). Podemos conectarnos utilizando ssh (***sudo ssh edison@192.168.2.15*** utilizando el puerto USB).

PASO 2. Modificar el archivo interfaces que se encuentra en la ruta: ***/etc/network/***. Para abrirlo y editarlo existen varias opciones (vim, nano,...). Se explica el primero, ya que es menos intuitivo y necesita de mayor ayuda. En la terminal, escribir: ***vim /etc/network/interfaces***. Pulsar la letra E para editar. Se abrirá el archivo (dentro de la terminal) para editarlo pulsar la letra i y comenzar a escribir/borrar como en un archivo word.

Las líneas que empiezan con almohadilla (#) son comentarios. Es necesario descomentar la línea en la que pone "auto wlan0", comentar "auto usb0" y cambiar en WPA el usuario y la contraseña (sin entrecomillar). Una vez se ha terminado de editar se le da a la tecla esc y para guardar y salir ZZ (mayúsculas, pulsando shift).

PASO 3. Desactivar la red y activarla para que los cambios surtan efecto (ifdown wlan0 y ifup wlan0 respectivamente). Esto debería conectar la placa a la red

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

elegida pero en nuestro (red eduroam) caso no es así. Existen formas de activar una red por terminal cogiendo el certificado CA (modificando el archivo: ***/etc/wpa_supplicant/wpa_supplicant.conf***). Aún no se ha logrado, pero se está trabajando en ello.

6.2.3. Instalación de ROS (Indigo)

De entre todas las versiones de ROS se ha elegido instalar **Indigo**, ya que es la más avanzada de las versiones estables. La instalación de ROS sobre Ubilinux es similar a la instalación sobre Raspbian, tan sólo unos pocos cambios las hacen diferentes. Para realizar la instalación (ésta o cualquier otra) es necesario ser **superusuario (root)**.

Paso 1. Cambios con respecto a Raspberry Pi. Editar el archivo: /etc/apt/
sources.list añadiendo la línea: deb http://http.debian.net/debian wheezy-backports main e instala la librería 'liblz4' con las líneas de comandos:

```
$ apt-get update
$ apt-get install liblz4-dev
```

Paso 2. Prerrequisitos. Introducir las siguientes líneas por la ventana de comandos:

```
$ sh -c 'echo "deb http://packages.ros.org/ros/ubuntu wheezy
main" > /etc/apt/sources.list.d/ros-latest.list'
$ wget https://raw.githubusercontent.com/ros/rosdistro/master/
ros.key -O - | apt-key add -
$ apt-get update
$ apt-get upgrade
$ apt-get install python-setuptools python-pip python-yaml python
-argparse python-distribute python-docutils python-dateutil python
-setuptools python-six
$ pip install rosdep rosinstall\_generator wstool rosinstall
$ rosdep init
```


Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

```
$ rosdep update
$ apt-get install liblz4-dev
```

Paso 3. Más prerrequisitos. Estas librerias también son necesarias incluir para una correcta instalación de ROS.

```
$ apt-get install python-empy
$ apt-get install python-nose
$ apt-get install libboost-filesystem1.49-dev
$ apt-get install libboost-program-options1.49-dev
$ apt-get install libboost-regex1.49-dev
$ apt-get install libbz2-dev
$ apt-get install libboost-signals1.49-dev
```

Paso 4. INSTALACIÓN.

```
$ mkdir ~/ros\_catkin\_ws
$ cd ~/ros\_catkin\_ws
$ rosinstall\_generator ros\_comm --rosdistro indigo --deps --wet
-only --exclude roslisp --tar > indigo-ros\_comm-wet.rosinstall
$ wstool init src indigo-ros\_comm-wet.rosinstall
```

terminal outputs: update complete

La versión de Debian es 'wheezy', para ella es necesario instalar las librerías 'liblz4-dev' (instalada antes) y 'libconsole-bridge-dev'. Sin embargo, también se han instalado las demás librerías que vienen en la instalación para RaspberryPi (sin ellas, no se completaba la instalación de ROS).

```
$ mkdir ~/ros\_catkin\_ws/external\_src
$ apt-get install checkinstall cmake
$ sh -c 'echo "deb-src http://mirrordirector.raspbian.org/raspbian
/ testing main contrib non-free rpi" >> /etc/apt/sources.list'
$ apt-key adv --keyserver keyserver.ubuntu.com --recv-keys 9165938
D90FDDD2E
$ apt-get update
$ cd ~/ros\_catkin\_ws/external\_src
```


Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

```
$ apt-get build-dep console-bridge
$ apt-get source -b console-bridge
$ dpkg -i libconsole-bridge0.2\_
-bridge-dev\_*
```

Otras Librerías necesarias

urdfdom_headers

```
$ cd ~/ros\_catkin\_ws/external\_src
$ git clone https://github.com/ros/urdfdom\_headers.git
$ cd urdfdom\_headers
$ cmake .
$ sudo checkinstall make install
```

urdfdom

```
$ cd ~/ros\_catkin\_ws/external\_src
$ sudo apt-get install libboost-test-dev libtinyxml-dev
$ git clone https://github.com/ros/urdfdom.git
$ cd urdfdom
$ cmake .
$ sudo checkinstall make install
```

collada-dom

```
$ cd ~/ros\_catkin\_ws/external\_src
$ sudo apt-get install libboost-filesystem-dev libxml2-dev
$ wget http://downloads.sourceforge.net/project/collada-dom/
Collada\%20DOM/Collada\%20DOM\%202.4/collada-dom-2.4.0.tgz
$ tar -xzf collada-dom-2.4.0.tgz
$ cd collada-dom-2.4.0
$ cmake .
$ sudo checkinstall make install
```

```
$ cd ~/ros\_catkin\_ws
$ rosdep install --from-paths src --ignore-src --rosdistro indigo
 -y -r --os=debian:wheezy
```


Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

Finalmente, se instala ROS:

```
$ ./src/catkin/bin/catkin\_make\_isolated --install -DCMAKE\_
BUILD\_\\TYPE=Release --install-space /opt/ros/indigo
$ source /opt/ros/indigo/setup.bash    NECESARIO PARA UTILIZAR ROS
```

Comprobaciones de la instalación:

```
$ rosversion -d   (debe salir 'indigo')
$ roscore
```

6.3. Multiple machine

Para poder acceder a la Edison desde el PC es necesario realizar una serie de pasos tanto en el PC como en la Edison.

En el PC:

```
export ROS\_HOSTNAME=mi\_pc
export ROS\_MASTER\_URI=http://mi\_pc:11311}
```

Donde *mi_pc* es la dirección ip usada en la red.

En la Intel Edison:

```
export ROS\_HOSTNAME=mi\_edison
export ROS\_MASTER\_URI=http://mi\_pc:11311}
```

El tutorial empleado para comunicar la Edison con el PC es el siguiente: `http://nootrix.com/2012/06/ros-networking/`.

Además, en la wiki de ROs hay varios tutoriales más que son bastante interesantes: `http://wiki.ros.org/ROS/Tutorials/MultipleMachines` y http://wiki.ros.org/ROS/NetworkSetup.

6.4. Demostrador con el PrintBot Renacuajo

Se ha realizado un controlador basado ROS para la comunicacion y control del robot Printbot Renacuajo desde la Intel Edison. Los pasos para poder utilizar este controlador son los siguientes

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

6.4.1. Generación del Workspace

Para poder compilar el controlador en ROS para PrintBot Renacuajo es necesario crear un nuevo workspace, siguiendo los siguientes pasos:

```
$ source /opt/ros/indigo/setup.bash
$ mkdir -p ~/workspace/src
$ cd ~/workspace/src
$ catkin_init_workspace
$ cd ..
$ catkin_make
$ source devel/setup.bash
```

6.4.2. Instalación del paquete rosserial

Para poder comunicar la placa arduino con ROS a través del puerto serie es necesario emplear el paquete rosserial dentro del workspace creado en el apartado anterior. Lo primero es instalar las librerías en el sistema:

```
$ sudo apt-get install ros-indigo-rosserial-arduino
$ sudo apt-get install ros-indigo-rosserial
```

Una vez hecho esto, hay que añadir y compilar el paquete rosserial al workspace creado:

```
$ cd workspace/src
$ git clone https://github.com/ros-drivers/rosserial.git
$ cd workspace
$ catkin_make
$ catkin_make install
$ source workspace/install/setup.bash
```

Tras realizar estos pasos, debe haber aparecido el paquete rosserial dentro de la carpeta `src` del workspace.

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

6.4.3. Instalación de las librerías ros_lib

Para que Arduino pueda funcionar junto a ROS es necesario incluir una serie de librerías. Los pasos para incluir dichas librerías dentro del sketchbook de Arduino son los siguientes:

```
$ cd sketchbook/libraries
$ rm -rf ros_lib
$ rosrun rosserial_arduino make_libraries.py .
```

NOTA: la carpeta `sketchbook` aparece al instalar el IDE de Arduino en Linux.

Para estar seguros de que las librerías se han instalado correctamente, se puede comprobar que aparezca una nueva opción dentro del apartado `Examples` del IDE de Arduino.

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

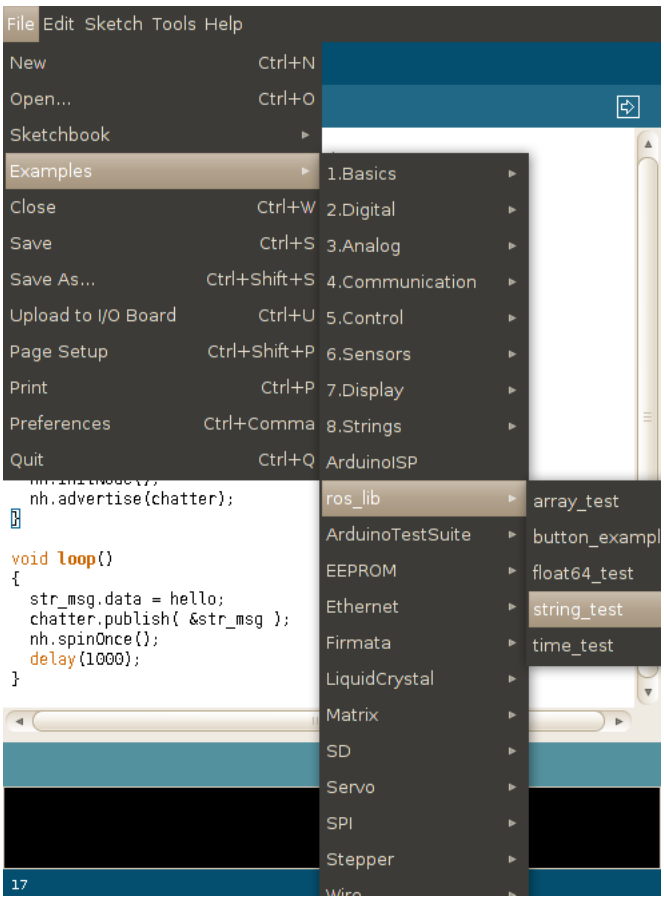


Figura 57: ros_lib en Arduino

6.4.4. Compilado del programa

Una vez instaladas las librerías necesarias, se puede completar la instalación del programa. Para ello, se copian los paquetes `renacuajo` y `renacuajo_control` en la carpeta `src` del workspace y se realizan los siguientes pasos en un terminal:

```
$ cd workspace
$ catkin_make
```

Se carga el firmware al Arduino:

```
$ catkin_make renacuajo_control_firmware_renacuajo-upload
```

Y tras confirmar la carga del programa, está listo para ser ejecutado.

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

6.5. Ejecución del Demostrador

Para ejecutar el programa demostrador simplemente hay que utilizar el siguiente comando en un terminal:

```
$ rosrun renacuajo_control renacuajo_control
```

Una vez iniciado el nodo, se podrá enviar instrucciones al robot. Nos pedirá dos argumentos de entrada:

- La función a realizar que puede ser: avanzar, retroceder, girar_derecha, girar_izquierda, seguir_linea y parar.
- La velocidad porcentual (0 - 100) a la que se moverán los servos realizando esa función (a excepción de la función parar que no requiere velocidad)

6.6. Simulación en Gazebo del Demostrador

Para simular el PrintBot Renacuajo en Gazebo hay que ejecutar el siguiente comando en un terminal:

```
$ roslaunch renacuajo_gazebo renacuajo.launch
```

Y se abrirá el simulador Gazebo con el modelo del PrintBot Renacuajo ya cargado.

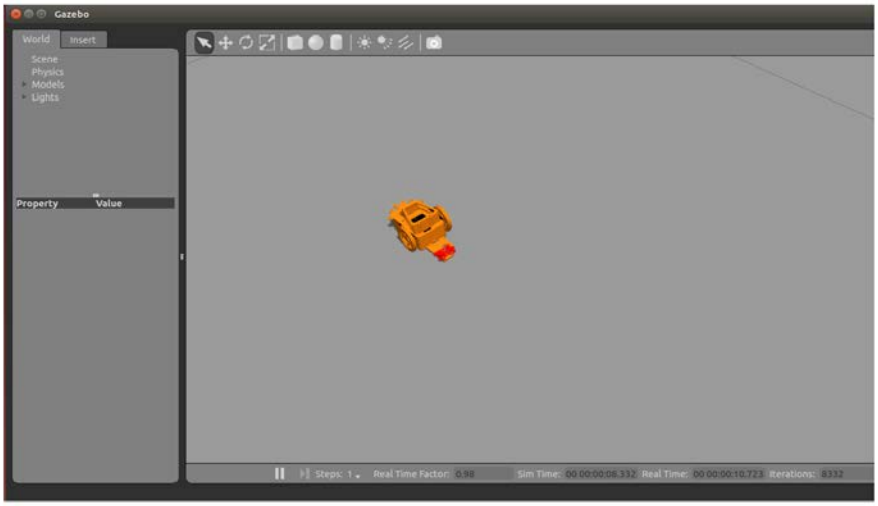


Figura 58: Simulación del PrintBot Renacuajo

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

7. Anexo III. Ejemplos de robots prediseñados

7.1. URDF

Unified Robot Description Format (URDF) es un formato estándar XML ROS para describir un modelo de un robot. Esto puede incluir la geometría física, las propiedades cinemáticas y dinámicas, e incluso los lugares y las propiedades de los sensores. Uno de los aspectos más fundamentales del URDF es un conjunto de enlaces (*links*) y articulaciones (*joints*) que se etiquetan de forma única y describen la relación de todas las partes del robot. URDF permite definir tanto la representación visual, como el modelo de colisiones del robot. La descripción típica tiene esta forma:

Listing 1: ejemplo.urdf

```
<robot name = "r2d2" >
    <link> ... </link>
    <link> ... </link>
    <link> ... </link>

    <joint> ... </joint>
    <joint> ... </joint>
    <joint> ... </joint>
</robot>
```

URDF permite definir distintos tipos de articulaciones como:

- **Revolute**: unión de tipo rotativo a lo largo de un eje con rangos limitados.
- **Continuous**: unión de tipo rotativo a lo largo de un eje sin rangos limitados.
- **Prismatic**: unión deslizante a lo largo de un eje con rangos limitados.
- **Fixed**: unión fija, no tiene grados de libertad.
- **Floating**: unión que permite el movimiento en los seis grados de libertad.
- **Planar**: unión que permite el movimiento en el plano perpendicular a un eje.

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

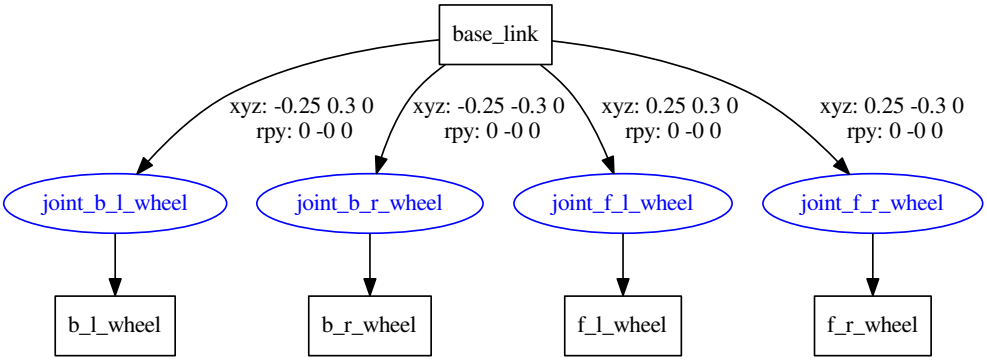


Figura 59: Ejemplo de representacion graphviz

El paquete de ROS URDF permite trabajar con este formato y tiene herramientas para comprobarlos y obtener un diagrama de graphviz entre otras cosas.

Para simplificar los archivos URDF existe un lenguaje llamado Xacro (XML Macros) que es un lenguaje de macros en XML, permite incrementar la modularidad y parametrización, reducir la redundancia y generar URDF a partir de archivos xacro. También permite ejecutar expresiones matemáticas y bloques condicionales. El lenguaje se basa en el uso de macros.

Listing 2: ejemplo xacro

```
<xacro:property name="pi" value="3.1415926535897931" />
<xacro:property name="the_radius" value="2.1" />
<xacro:property name="the_length" value="2.5 * pi" />
<geometry type="cylinder" radius="${the_radius}" length="${
    the_length}" />
```

La transformación de Xacro a URDF se realiza a través del paquete xacro con un script de python con el siguiente comando:

```
rosrun xacro xacro.py -o model.urdf model.urdf.xacro
```

URDF también permite elementos de transmisión para describir la relación entre un actuador y una unión, un ejemplo de transmisión es:

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

Listing 3: ejemplo transmisión

```
<transmission name="simple_trans">
  <type>transmission_interface/SimpleTransmission</type>
  <joint name="foo_joint">
    <hardwareInterface>EffortJointInterface</hardwareInterface>
  </joint>
  <actuator name="foo_motor">
    <mechanicalReduction>50</mechanicalReduction>
    <hardwareInterface>EffortJointInterface</hardwareInterface>
  </actuator>
</transmission>
```

7.2. SDF

SDF es un formato XML que describe objetos y entornos para simuladores de robots, también permite la visualización y control. Originalmente fue desarrollado como parte de Gazebo y fue diseñado para aplicaciones robóticas. Se pueden describir todos los aspectos de un robot usando SDF, ya que el robot es un chasis simple con ruedas o un humanoide. Además de los atributos cinemáticos y dinámicos, sensores, propiedades de la superficie, texturas, fricción de las uniones, y más propiedades que pueden definir a un robot. Estas características le permiten utilizar SDF tanto para la simulación, visualización, planificación de movimiento y control del robot.

Listing 4: ejemplo transmisión

```
    <?xml version='1.0'?>
      <sdf version='1.4'>
        <model name="my_robot">
        <static>true</static>
          <link name='chassis'>
            <pose>0 0 .1 0 0 0</pose>
            <collision name='collision'>
              <geometry>...</geometry>
            </collision>
            <visual name='visual'>
              <geometry>...</geometry>
            </visual>
          </link>
```


Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

```
37        </model>
38     </sdf>
```

7.3. Rviz

Rviz es una herramienta de visualización 3D de ROS que además de elementos pasivos permite simular actuadores y sensores. Rviz permite cargar el formato URDF directamente pero no se pueden cargar archivos en formato SDF, esto se puede solucionar con un paquete llamado **gazebo2rviz**. Un archivo con formato launch que permite cargar en rviz un fichero en URDF con xacro es el siguiente:

Listing 5: ejemplo launch rviz

```
39 <?xml version="1.0"?>
40 <launch>
41          <param name="robot_description"
42     command="\$(find xacro)/xacro.py '\$(find urdf_serp)/urdf/
           robot_movil/robot_ruedas.urdf.xacro'" />
43         <node name="joint_state_publisher" pkg="
               joint_state_publisher" type="joint_state_publisher" />
44         <param name="use_gui" value="true"/>
45         <node name="robot_state_publisher" pkg="
               robot_state_publisher" type="state_publisher" />
46         <node name="rviz" pkg="rviz" type="rviz" />
47 </launch>
```

La visualización en rviz puede verse en las figuras 60 y 61.

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

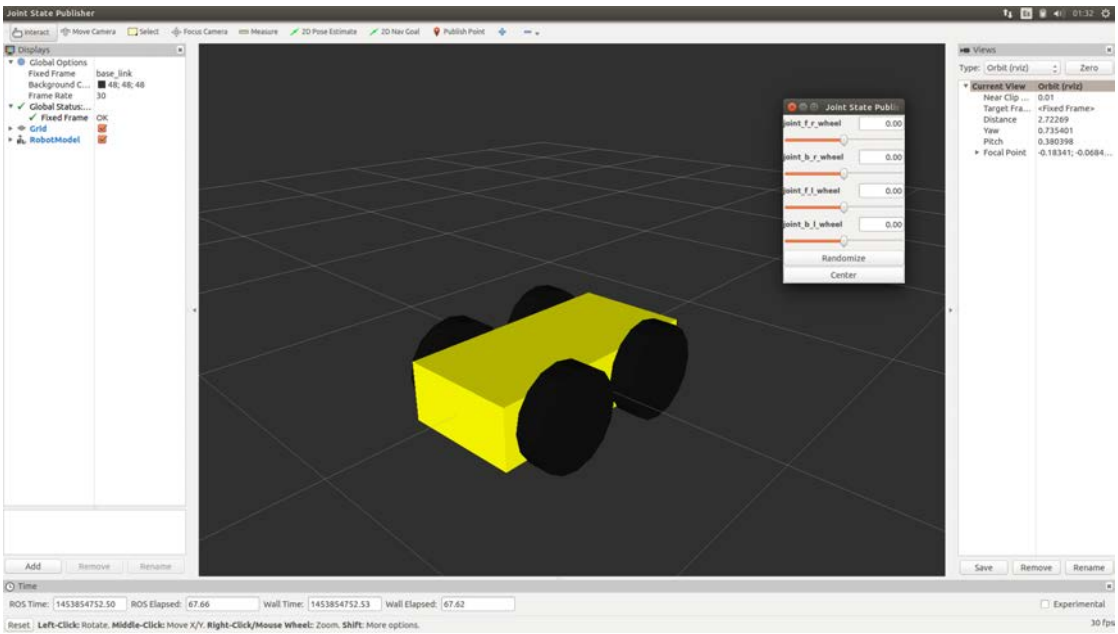


Figura 60: Captura de rviz

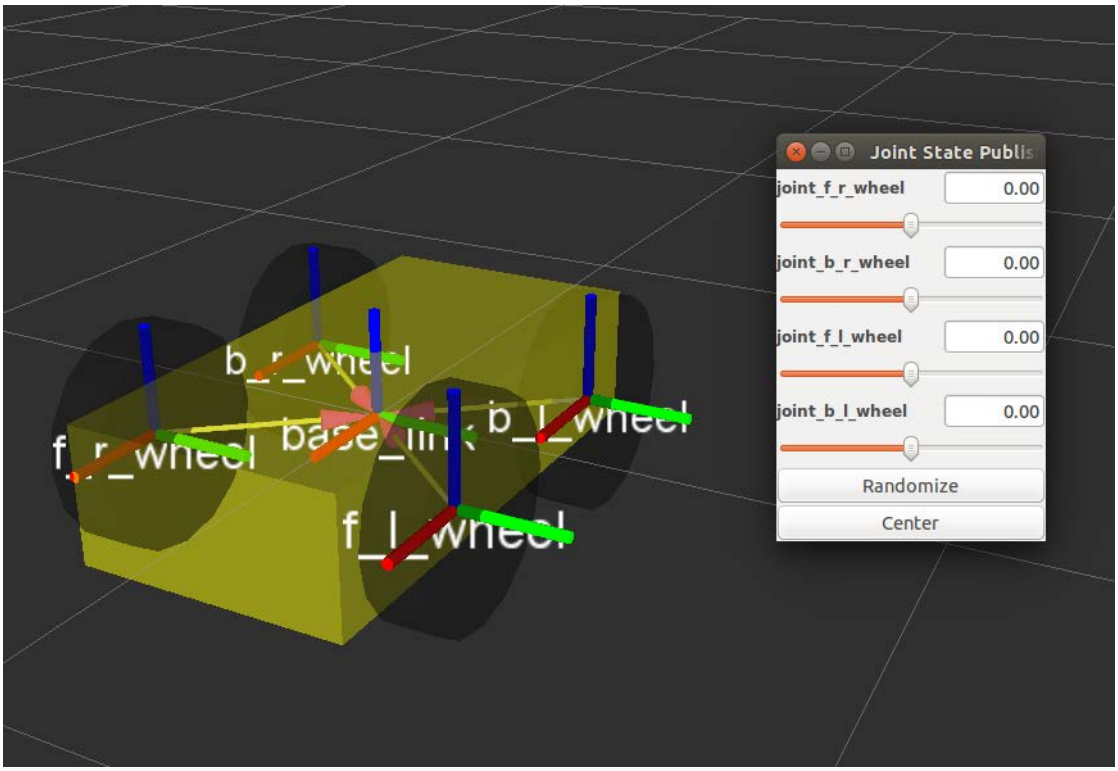


Figura 61: Visualización uniones rviz

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

7.4. Gazebo

Es un simulador que se puede integrar en ROS. Gazebo ofrece la posibilidad de simular con precisión y eficiencia poblaciones de robots en entornos interiores y exteriores complejos. Posee un robusto motor de física, gráficos de alta calidad, y las interfaces gráficas y de programación son buenas. Además Gazebo es una herramienta open source con una gran comunidad trabajando con ella. La principal ventaja de Gazebo sobre rviz es el motor de física, pero también tiene una serie de plugins (que pueden ser programados) que añaden funcionalidades como cámara, laser, acelerómetro, ... También tiene controladores para robots, por ejemplo de tres/cuatro apoyos y locomoción diferencial (*differential drive*) como el Renacuajo de BQ, o de cuatro ruedas con dirección deslizante (*skid-steer drive*).
Un ejemplo del archivo launch para cargar un URDF a Gazebo es el siguiente:

Listing 6: ejemplo launch gazebo

```
<?xml version="1.0"?>
<launch>
    <arg name="model"/>
    <arg name="gui" default="False" />
   <include file="\$(find gazebo_ros)/launch/empty_world.launch">
     <arg name="world_name" value="\$(find urdf_serp)/worlds/
         gazebo_world.world"/>
<node name="spawn_urdf" pkg="gazebo_ros" type="spawn_model" args="
     -file \$(find urdf_serp)/urdf/assem_servo3.urdf.xacro -urdf -x
     0 -y 0 -z 0.2 -model servo1" />
</launch>
```

En la Figura 62 puede verse un ejemplo usando la cámara en Gazebo.
En la Figura 63 puede verse una comparación entre rviz y Gazebo.

7.5. Simulaciones

A continuación se muestran algunos de los modelos desarrollados y que servirán como base a los robots predefinidos de la arquitectura modular. En primer lugar se muestra el modelo de una articulación rotacional basada en un servo, y en base a dicha estructura se generan los siguientes modelos.

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

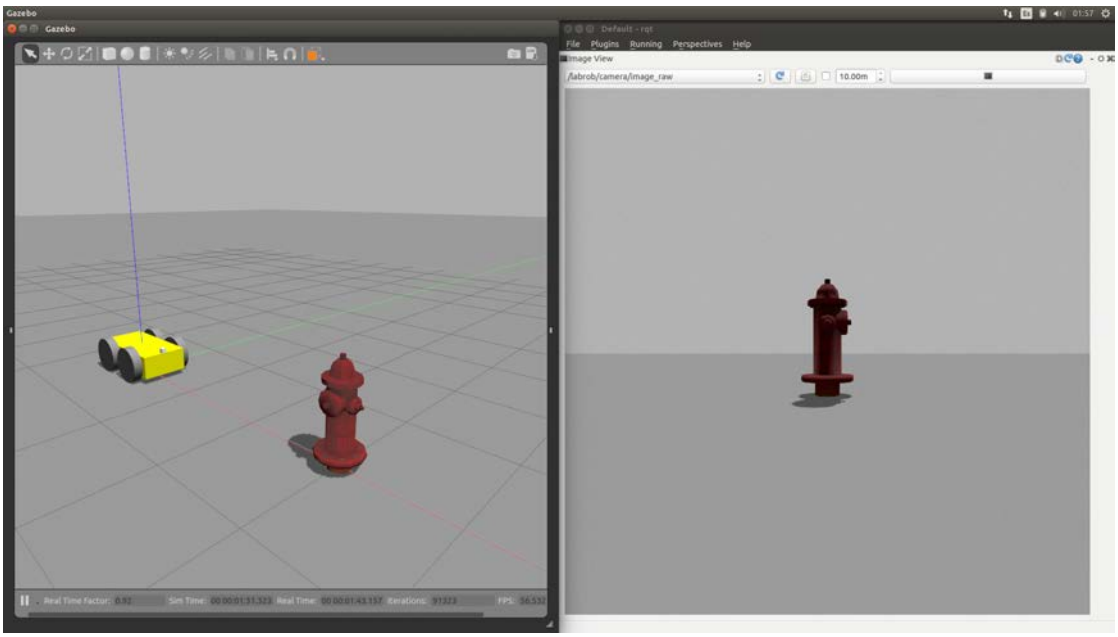


Figura 62: Cámara en Gazebo

7.5.1. Servo

El servo simulado consta de tres piezas, dos de ellas prismáticas y una cilíndrica. Primero se genera el URDF y puede ser importado desde rviz o Gazebo. Se han generado dos URDF, uno manualmente y otro a través de un programa de CAD que hace mas sencilla la creación del fichero. Ambos se comparan en la figura 64.

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

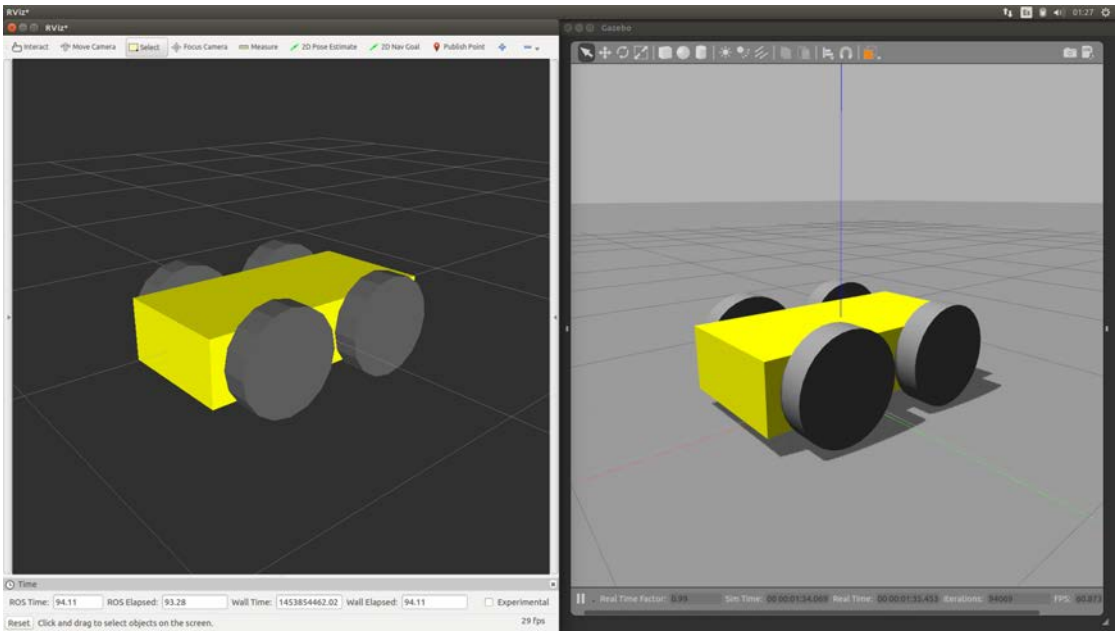
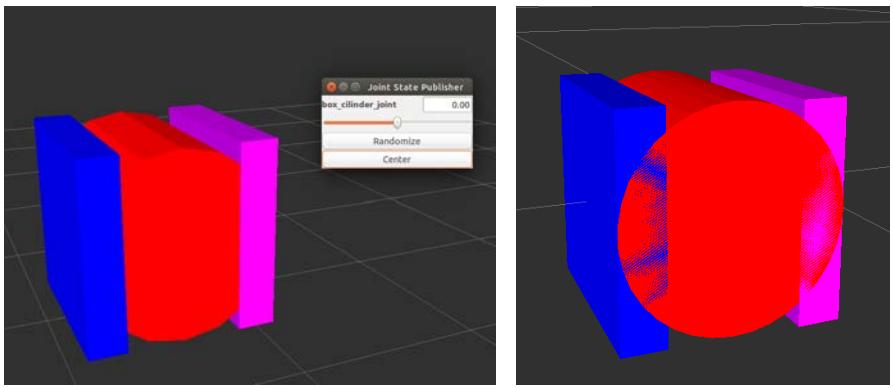


Figura 63: Comparación entre rviz y Gazebo



(a) Servo manual (b) Servo Cad

Figura 64: Servo en rviz

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa
Operativo Plurirregional de Crecimiento Inteligente 2014-2020

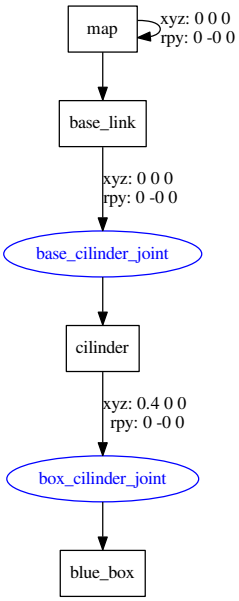


Figura 65: Graphiz del servo

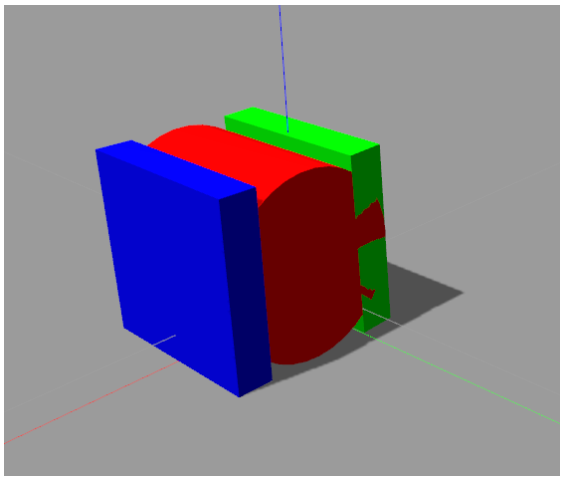
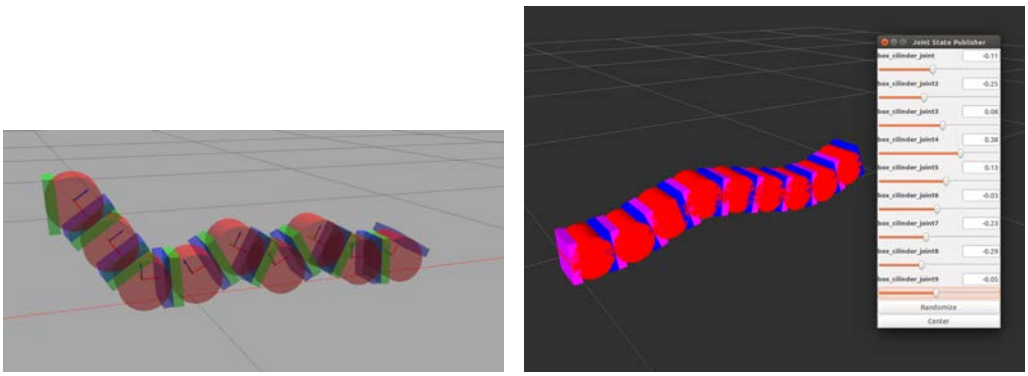


Figura 66: Servo en Gazebo

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

7.5.2. Serpiente

Tras tener el modelo en URDF del servo de forma correcta es posible crear una serpiente de forma sencilla uniendo los distintos servos mediante uniones fijas. En la figura 67 se muestra la representación de la serpiente en Gazebo y rviz respectivamente.



(a) Serpiente Gazebo (b) Serpiente rviz

Figura 67: Modelo de serpiente construido mediante elementos servo.

7.5.3. Zowi

Para generar los ficheros de descripción de este robot primero se ha realizado el diseño del mismo en un programa de CAD (Solid Works), para después generar automáticamente los ficheros. La representación de Zowi en rviz se muestra en la figura 68.

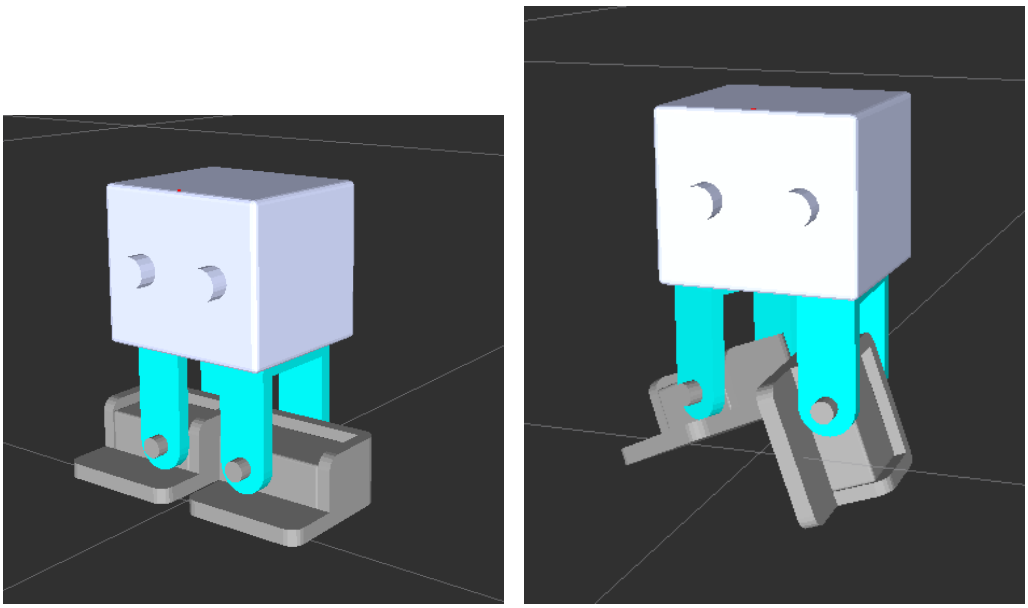
7.5.4. Manipulador

Este apartado muestra el modelo de un manipulador industrial con tres articulaciones de rotación modeladas mediante elementos de tipo servo. La representación en Gazebo se muestra en la Figura 69.

7.5.5. Robots móviles

En primer lugar se muestra un modelo de robot de locomoción diferencia (dos ruedas) con un apoyo adicional para tener estabilidad. Se representa en la figura 70.

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

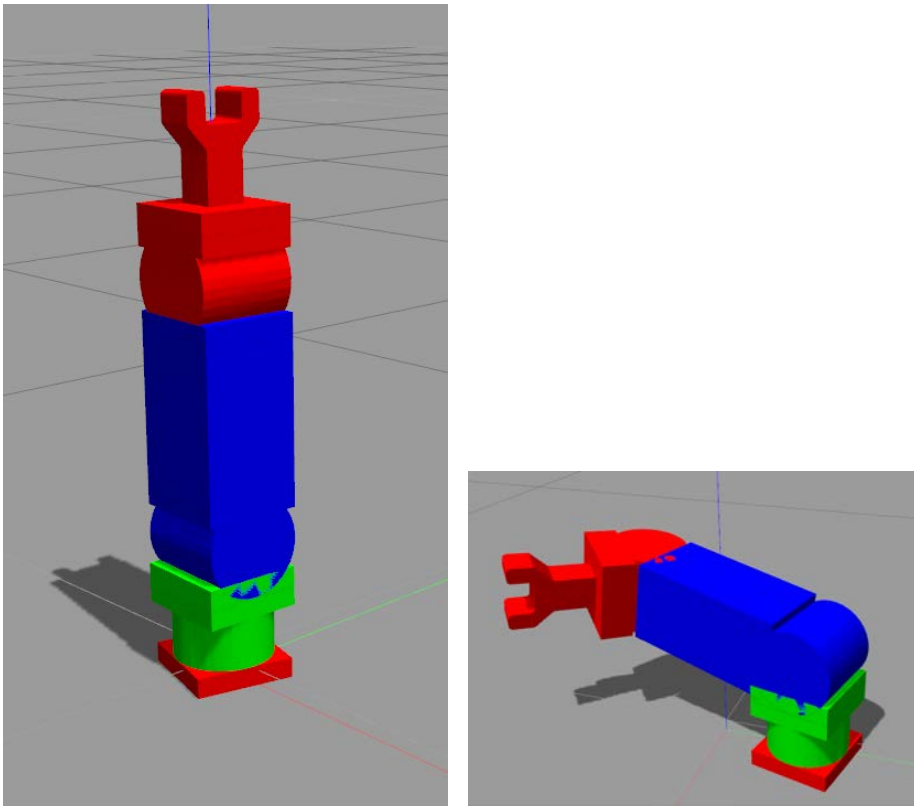


(a) Zowi en reposo (b) Zowi en movimiento

Figura 68: Visualización del modelo de Zowi en rviz.

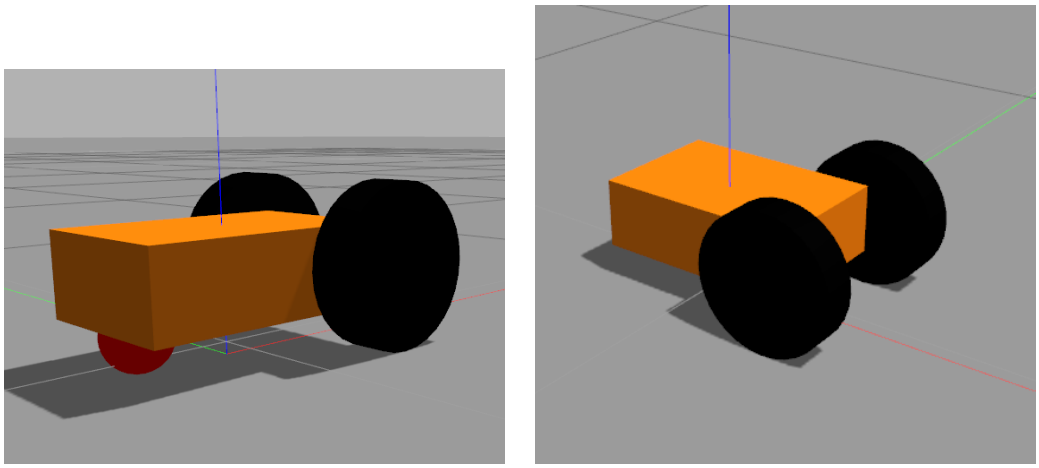
Por último, en la figura 71 se vuelve a mostrar el modelo generado en Gazebo para un robot móvil de cuatro ruedas y dirección deslizante.

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa
Operativo Plurirregional de Crecimiento Inteligente 2014-2020



(a) Manipulador Gazebo (b) Manipulador Gazebo

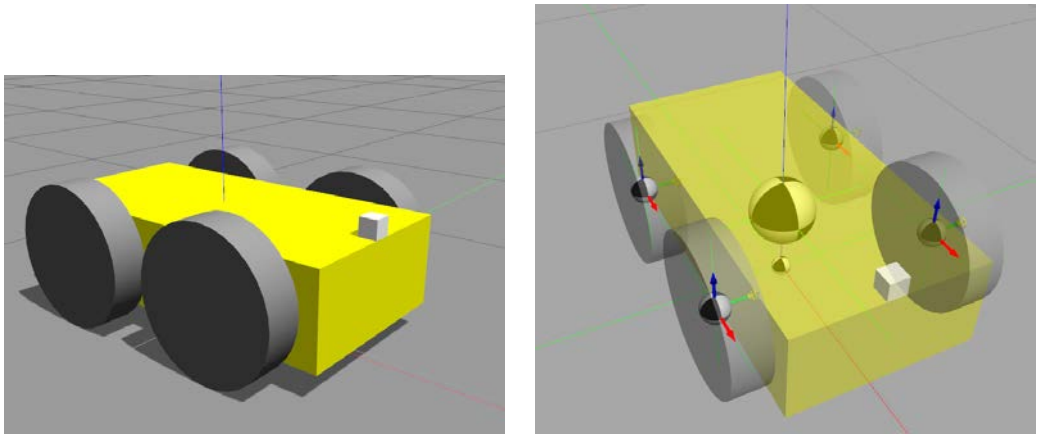
Figura 69: Modelo de manipulador de tres gdl en Gazebo.



(a) Vista 1 (b) Vista 2

Figura 70: Modelo de robot de locomoción diferencial.

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020



(a) Vista 1 (b) Vista 2

Figura 71: Modelo de robot de cuatro ruedas con dirección deslizante

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa
Operativo Plurirregional de Crecimiento Inteligente 2014-2020

Referencias

[1] Bitbloq. `http://bitbloq.bq.com/`. Último acceso: 2015-12-23.

[2] International collaborative robots workshop 2015. `www.robotics.org/icrw2015`. Último acceso: 2015-12-12.

[3] Moveit! `http://moveit.ros.org/`. Último acceso: 2015-12-23.

[4] My model robot. `http://mymodelrobot.appspot.com/`. Último acceso: 2015-12-23.

[5] Robot operating system. `http://www.ros.org/`. Último acceso: 2015-12-23.

[6] Ieee standard ontologies for robotics and automation. *P1872/D3, August 2014*, pages 1–55, April 2015.

[7] M. Beetz, U. Klank, I. Kresse, A. Maldonado, L. Mosenlechner, D. Pangercic, T. Ruhr, and M. Tenorth. Robotic roommates making pancakes. In *Humanoid Robots (Humanoids), 2011 11th IEEE-RAS International Conference on*, pages 529–536, Oct 2011.

[8] K. Goris. *Autonomous Mobile Robot Mechanical Design*. PhD thesis, Vrije Universiteit Brussel, 2005.

[9] C. Schlenoff, E. Prestes, R. Madhavan, P. Goncalves, H. Li, S. Balakirsky, T. Kramer, and E. Miguelanez. An ieee standard ontology for robotics and automation. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 1337–1342, Oct 2012.

[10] M. Tenorth and M. Beetz. Knowrob: A knowledge processing infrastructure for cognition-enabled robots. *International Journal of Robotics Research*, 32(5):566–590, 2013.

[11] M. Tenorth, A. Perzylo, R. Lafrenz, and M. Beetz. Representation and exchange of knowledge about actions, objects, and environments in the roboearth framework. *Automation Science and Engineering, IEEE Transactions on*, 10(3):643–651, July 2013.

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020

[12] E. R. Wade. *DC powerline communication system using a transmission line transformer for high degree of freedom applications*. PhD thesis, Massachusetts Institute of Technology, 2004.

[13] M. Waibel, M. Beetz, J. Civera, R. D'Andrea, J. Elfring, D. Galvez-Lopez, K. Haussermann, R. Janssen, J. Montiel, A. Perzylo, B. Schiessle, M. Tenorth, O. Zweigle, and R. van de Molengraft. Roboearth. *Robotics Automation Magazine, IEEE*, 18(2):69–82, June 2011.

Proyecto Financiado por el Centro de Desarrollo Tecnológico Industrial (CDTI).
Expediente IDI-20150289
Cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER) a través del Programa Operativo Plurirregional de Crecimiento Inteligente 2014-2020