



Universidad
de Alcalá

ESCUELA POLITECNICA SUPERIOR

MÁSTER UNIVERSITARIO EN
INGENIERÍA DE TELECOMUNICACIÓN

TRABAJO DE FIN DE MÁSTER

**VEHÍCULO MÓVIL NO TRIPULADO PROVISTO
DE SENSORES Y ACTUADORES PARA
RECONOCER ENTORNOS**

**Daniel Zugasti Royuela
2015**

Quiero agradecer a todas las personas que de una forma u otra me han apoyado en la realización de mi trabajo fin de máster y han hecho que sea posible y se convierta en una realidad.

En primer lugar, quiero agradecer a mi madre, Montserrat, la educación y el cariño que me han dado durante toda mi vida, y que me ha dado la oportunidad de formarme académica y personalmente y que me ha permitido labrarme un futuro prometedor. A mi hermana, Sara, que me ha apoyado en todo momento demostrándome su confianza. También quiero agradecer a mi compañera inseparable, Almudena, su especial apoyo y complicidad demostrada desde siempre sin la cual esto no hubiera sido posible. No puedo dejar en el olvido a mi abuela Jesusa, que siempre ha estado animándome, a su manera, a terminar todo aquello que se empieza. También quiero dedicar unas palabras a mi abuelo Pablo que, aunque ya no se encuentra entre nosotros, siempre ha sido y será para mí un ejemplo a seguir.

También quiero agradecer a mi tutor, Enrique Alexandre, el haberme ayudado a alcanzar de forma satisfactoria y eficiente la consecución de este trabajo fin de carrera. No solo quiero agradecerle el apoyo técnico, sino también el anímico.

No quisiera olvidarme de todos mis compañeros y amigos que, aunque sigan pasando los años, hacen ver que las cosas pueden seguir igual que el primer día.

Por último, a toda mi familia: por tener paciencia infinita, por estar apoyando de forma incondicional haciendo que sea capaz de creer en mí mismo, por enseñarme que un tropiezo no es más que el principio de un nuevo intento y que la lucha, el esfuerzo y una actitud positiva es algo fundamental para seguir adelante. Por hacer que sea lo que hoy soy.

A TODOS, MUCHAS GRACIAS

RESUMEN

El principal objetivo de este proyecto es proporcionar una plataforma móvil que pueda ser controlada de forma remota y que permita integrar la sensorización y actuación sobre dispositivos y equipamiento heterogéneo en un entorno inteligente.

Las directrices a seguir son que la plataforma sea modular y ampliable, de bajo coste y bajo consumo.

La aplicación de esta plataforma puede ser multipropósito: explorar entornos hostiles, reconocimiento de terrenos poco accesibles o con condiciones adversas, realizar mantenimientos en áreas rurales o de difícil acceso, sistemas de seguridad, etc.

PALABRAS CLAVE

Sensor, actuador, raspberry, entorno inteligente, robot, rover.

ABSTRACT

This project aims to provide a mobile platform that can be remotely controlled to collate the sensing and acting on heterogeneous devices and equipment in an intelligent environment.

The aims of modularity, expandability, low cost and low power consumption are sought

The implementation of this platform can be multipurpose: explore hostile environments, recognition of inaccessible terrain or adverse conditions, perform maintenance in rural areas or difficult to access, security systems, etc.

KEYWORDS

Sensor, actuator, raspberry, intelligent environment, robot, rover.

ÍNDICE DE CONTENIDOS.

CAPÍTULO 1: INTRODUCCIÓN Y OBJETIVOS.	9
CAPITULO 2: PLATAFORMA HARDWARE	10
RASPBERRY PI	10
ARDUINO	12
OTRAS ALTERNATIVAS	13
ELECCIÓN HW	14
CAPÍTULO 3: PLATAFORMA SOFTWARE	16
SISTEMA OPERATIVO	16
BUS DE COMUNICACIONES	16
LENGUAJES DE PROGRAMACIÓN	17
CAPÍTULO 4: ROBÓTICA	18
SEGÚN SU CRONOLOGÍA	18
1.ª GENERACIÓN.	18
2.ª GENERACIÓN.	18
3.ª GENERACIÓN.	18
4.ª GENERACIÓN.	18
SEGÚN SU ESTRUCTURA	19
POLIARTICULADOS	19
MÓVILES	19
ANDROIDES	20
ZOOMÓRFICOS	20
HÍBRIDOS	21
CAPÍTULO 5: SOLUCIÓN TÉCNICA	22
CONFIGURACIÓN DE LA PLATAFORMA	23
CONFIGURACIÓN DE E/S	23
CONFIGURACIÓN DE PERIFÉRICOS	25
BLOQUE DE SENSORIZACIÓN	26
BLOQUE DE MOVIMIENTO	30
BLOQUE DE MANDO Y CONTROL	32
MONTAJE DEL ROBOT	35
CAPÍTULO 6: CONCLUSIONES Y FUTURAS LÍNEAS	38
BIBLIOGRAFÍA	41

PLIEGO DE CONDICIONES.....	42
CONDICIONES DE MATERIALES Y EQUIPOS.....	42
HARDWARE UTILIZADO: REQUISITOS MÍNIMOS	42
SOFTWARE UTILIZADO.....	44
CONDICIONES DE EJECUCIÓN.....	44
PRESUPUESTO	45

ÍNDICE DE FIGURAS.

Figura 1 Evolución de la robótica.....	22
Figura 2 Puertos GPIO.....	24
Figura 3 HC-SR04 Funcionamiento.....	27
Figura 4 MJPG-Streamer.....	29
Figura 5 L298N Esquema	30
Figura 6 L298N Alimentación	31
Figura 7 Interconexión de bloques	32
Figura 8 Menú	33
Figura 9 Ejecución de comandos	33
Figura 10 Montaje Completo	36
Figura 11 Chasis del robot.....	37

ÍNDICE DE TABLAS.

Tabla 1 Tabla de conversión GPIO	25
Tabla 2 Presupuesto de material.....	45
Tabla 3 Presupuesto de mano de obra	46
Tabla 4 Presupuesto de OCDs	46
Tabla 5 Presupuesto de costes.....	47
Tabla 6 Presupuesto de ventas	47

CAPÍTULO 1: INTRODUCCIÓN Y OBJETIVOS.

El principal objetivo de este proyecto es realizar un robot controlado remotamente y que proporciona una plataforma que permita integrar diferentes sensores y actuadores que podrán ser consultados remotamente. De esta forma podremos caracterizar un entorno a distancia..

Las directrices a seguir son que la plataforma sea modular y ampliable, de bajo coste y bajo consumo.

Como ya se explicara detalladamente a lo largo de la memoria, la aplicación de esta plataforma puede ser multipropósito: explorar entornos hostiles, reconocimiento de terrenos poco accesibles o con condiciones adversas, realizar mantenimientos en áreas rurales o de difícil acceso, sistemas de seguridad, etc.

Este documento está subdividido en diferentes secciones de cara facilitar al lector la comprensión del proyecto.

Primero se describirán algunas de las alternativas disponibles en el mercado tanto software como hardware, que se han barajado para la ejecución de la plataforma, indicando sus principales características y reforzando aquellas que hayan sido determinantes para su posterior elección.

Posteriormente se introducirán conceptos relacionados con la robótica

Una vez descritos explicaremos con todo detalle la solución implementada ilustrando los conceptos con diferentes diagramas para una mejor comprensión.

Finalmente se expondrán las conclusiones y los posibles futuros pasos para una posible evolución de la plataforma o distintas aplicaciones de la misma.

CAPITULO 2: PLATAFORMA HARDWARE

En primer lugar, nos centraremos en las diferentes posibilidades que existen en el mercado para albergar la plataforma sobre la que desarrollamos el proyecto.

Para cumplir uno de los objetivos del proyecto -recordemos que debía ser una plataforma de bajo coste- nos centraremos en diferentes alternativas de microordenadores y microcontroladores que están teniendo mucha repercusión por diferentes motivos que explicaremos más adelante.

RASPBERRY PI

En primer lugar nos centraremos en la Raspberry Pi. Se trata de un «ordenador de tamaño de tarjeta de crédito que se conecta a su televisor y un teclado». Es una placa que soporta varios componentes necesarios en un ordenador común. «Es un pequeño ordenador capaz, que puede ser utilizado por muchas de las cosas que su PC de escritorio hace, como hojas de cálculo, procesadores de texto y juegos. También reproduce vídeo de alta definición», apuntan en la página web del producto.

Este proyecto fue ideado en 2006 pero no fue lanzado al mercado febrero de 2012. Ha sido desarrollado por un grupo de la Universidad de Cambridge y su misión es fomentar la enseñanza de las ciencias de la computación los niños. De hecho, en enero de este año Google donó más de 15.000 Raspberry Pi para colegios en Reino Unido.

La placa, aunque es más pequeña que una tarjeta de crédito, tiene varios puertos y entradas: dos USB, Ethernet y salida HDMI entre otros. Estos puertos permiten conectar el miniordenador a otros dispositivos, teclados, ratones y pantallas.

También posee un System on Chip que contiene un procesador (CPU) ARM que corre a 700 Mhz, un procesador gráfico (GPU) VideoCore IV y hasta 512 Mb de

memoria RAM. Es posible instalar sistemas operativos libres a través de una tarjeta SD.

La fundación da soporte para las descargas de las distribuciones para arquitectura ARM, Raspbian (derivada de Debian), RISC OS, Arch Linux ARM (derivado de Arch Linux) y Pidora (derivado de Fedora) y promueve principalmente el aprendizaje del lenguaje de programación Python y otros lenguajes como Tiny BASIC, 12 C y Perl.

Los ordenadores que tenemos están orientados a tareas informáticas o de ocio, pero no vienen preparados con conexiones que posibiliten “pequeños proyectos de hardware” o con herramientas para aprender a programar o un lenguaje de programación. Este es el nicho que cubre (perfectamente) la Raspberry Pi. Tiene unas dimensiones de placa de 8.5 por 5.3 cm.

Todo ello equivale en la práctica a un ordenador con unas capacidades gráficas similares a la XBOX de Microsoft y con la posibilidad de reproducir vídeo en 1080p. En la placa nos encontramos además con una salida de vídeo y audio a través de un conector HDMI, con lo que conseguiremos conectar la tarjeta tanto a televisores como a monitores que cuenten con dicha conexión.

En cuanto a vídeo se refiere, también cuenta con una salida de vídeo compuesto y una salida de audio a través de un minijack. Posee una conexión Ethernet 10/100 y, si bien es cierto que podría echarse en falta una conexión Wi-Fi, gracias a los dos puertos USB incluidos podremos suplir dicha carencia con un adaptador Wi-Fi USB de terceros si lo necesitamos.

Los puertos tienen una limitación de corriente, por lo que si queremos conectar discos duros u otro dispositivos tendréis que pensar en hacerlo a través de un hub USB con alimentación.

Para terminar la Raspberry Pi cuenta con una serie de conexiones o puertos de entrada y salida de propósito general o GPIO, que nos permiten hacer todo tipo de

interfaces hardware, como el control de otros dispositivos, encendido y apagado de luces, proyectos de domótica, etc.

ARDUINO

Arduino es una plataforma de hardware libre, basada en una placa con un microcontrolador y un entorno de desarrollo, diseñada para facilitar el uso de la electrónica en proyectos multidisciplinarios.

El hardware consiste en una placa con un microcontrolador Atmel AVR y puertos de entrada/salida. Los microcontroladores más usados son el Atmega168, Atmega328, Atmega1280, ATmega8 por su sencillez y bajo coste que permiten el desarrollo de múltiples diseños. Por otro lado el software consiste en un entorno de desarrollo que implementa el lenguaje de programación Processing/Wiring y el cargador de arranque que es ejecutado en la placa.

Arduino se puede utilizar para desarrollar objetos interactivos autónomos o puede ser conectado a software. El entorno de desarrollo integrado libre se puede descargar gratuitamente.

Arduino puede tomar información del entorno a través de sus entradas analógicas y digitales, puede controlar luces, motores y otros actuadores. El microcontrolador en la placa Arduino se programa mediante el lenguaje de programación Arduino (basado en Wiring) y el entorno de desarrollo Arduino (basado en Processing). Los proyectos hechos con Arduino pueden ejecutarse sin necesidad de conectar a un computador.

Sin duda, este pequeño dispositivo nos ofrece una variedad de interesantes opciones inversamente proporcional a su reducido tamaño. Sólo hace falta un poco de imaginación, algunos conocimientos de electrónica y programación y lanzarse a la aventura.

OTRAS ALTERNATIVAS

Aunque menos conocidos, existen otros competidores a las anteriores plataformas. Hay otras placas de desarrollo que podemos encontrar en el mercado. Tenemos otras posibles candidatas que, animados por el gran éxito de la Raspberry Pi, están haciendo aparición tanto comercialmente como buscando financiación en sitios como Kickstarter.

Si bien para desarrollar proyectos de desarrollo hardware las placas Arduino son una buena opción, si consideramos además la faceta de la Raspberry Pi como posible centro multimedia, los principales competidores son los mini pc con sistema operativo Android.

Si buscas alguna alternativa antes deberías considerar la gran ventaja que tienen frente al resto de dispositivos, que no es otra que la enorme cantidad de desarrollos y proyectos de la comunidad. Podemos optar por una placa con mejores características de hardware (conectividad, mejor procesador, memoria, etc.), pero puede que al final resulte que la distribución de Linux o aplicaciones que queramos usar no estén disponibles.

La Cubieboard, una interesante tarjeta de desarrollo con mejores posibilidades que la Raspberry Pi. Una de las opciones más interesantes es la Cubieboard con un procesador alrededor de un 40% (Arm 8) más potente que la Raspberry Pi, 1 GB de RAM y 4 GB de almacenamiento, así como un puerto SATA y numerosos GPIO (I2C, SPI, RGB/LVDS, CSI/TS, FM-IN, ADC, CVBS, VGA, SPDIF-OUT, R-TP...).

Hackberry viene a mejorar las características de la Raspberry Pi, sobre todo por contar con Wi-Fi y capacidades de hardware similares a las de la Cubieboard (con un procesador a 1.2 GHZ en lugar de a 1 como en la Cubieboard), pero sin el interfaz SATA.

Hay otros competidores, como la Gooseberry, Via APC, y seguro que aparecen muchos más en un futuro no demasiado lejano.

Además existen los llamados Mini PC, dispositivos con el tamaño de un pendrive pero que cuentan con un corazón de Android, de acuerdo con las posibilidades de hardware de cada uno. Dichos dispositivos cuentan con una salida HDMI para conectarse a nuestros televisores o monitores.

Así, hay Mini PCs como el Droid Stick T10, con Android 4.1, un procesador de doble núcleo Rockchip RK3066 a 1,6 GHz, una GPU Mali400, 1Gb de RAM y 8Gb de ROM, conexión por Wi-Fi 802.11b/g/n, y potencia suficiente para reproducir vídeo en Full HD. También cuenta con un puerto USB 2.0 y otro para su alimentación (Mini USB), así como una ranura Micro-SD y un puerto USB On The Go.

Sin embargo, el nicho de estos Mini PC es hacerse un hueco en el salón de aquellas casas que no tienen un Smart TV. La idea de tenerlo todo integrado, pequeño tamaño, conexión por HDMI a la televisión y, sobre todo, un sistema operativo familiar y conocido como es Android puede ser un gran atractivo para decantarnos por ellos.

En su contra he de decir que son sistemas más cerrados y es más complicado modificar el S.O. para aprovechar su hardware

ELECCIÓN HW

Después de conocer el estado del arte de algunos de los dispositivos que existen en el mercado y haber valorado las fortalezas y debilidades de cada dispositivo (flexibilidad, soporte, prestaciones, etc.) se ha seleccionado la Raspberry como parte de la plataforma para el proyecto. Dispone de la capacidad de procesamiento necesaria. Alberga un S.O. completo que permite la configuración de servicios como un servidor web, programación en Python o Shell script, etc. Tiene conectividad de red sin accesorios adicionales. Tiene un mínimo consumo eléctrico.

Dispone de un conjunto de puertos de E/S para poder interactuar con los diferentes sensores y actuadores del entorno y comunicarse con otras placas.

Puesto que la Raspberry Pi no dispone de protección ante picos de corriente, se ha optado por incorporar a la plataforma diferentes placas de Arduino de cara a independizar la placa eléctricamente de los diferentes sensores/actuadores y así evitar que corrientes de retorno no deseadas puedan dañar el elemento que aporta la inteligencia y el control al robot. En resumen, estos dos tipos de elemento cubren todos los requisitos necesarios para implementar la plataforma.

CAPÍTULO 3: PLATAFORMA SOFTWARE

En este apartado, puesto que aumenta el número de opciones existentes, vamos a exponer directamente la solución elegida y el motivo.

SISTEMA OPERATIVO

El sistema operativo elegido es Raspbian, que es una distribución GNU/Linux basada en Debian optimizada para el Hardware de Raspberry muy estable y que usa un escritorio ligero LXDE y contiene herramientas de desarrollo.

BUS DE COMUNICACIONES

Puesto que disponemos de varios elementos en la plataforma que queremos comandar o controlar de forma centralizada y necesitamos que se traspase información de unos a otros, se ha optado por unir tanto la Raspberry como los Arduinos mediante un Bus I²C. La elección se ha basado en que este bus es un bus de comunicaciones serie muy extendido y es soportado de forma nativa por ambos. La velocidad de transmisión no es muy elevada pero se ha considerado suficiente para la información que se deben intercambiar: datos de los sensores, comandos, etc.

La principal característica de I²C es que utiliza dos líneas para transmitir la información: una para los datos y otra para la señal de reloj. También es necesaria una tercera línea, pero esta sólo es la referencia (masa). Como suelen comunicarse circuitos en una misma placa que comparten una misma masa esta tercera línea no suele ser necesaria.

Las líneas se llaman:

- SDA: datos
- SCL: reloj
- GND: tierra

Los dispositivos conectados al bus I²C tienen una dirección única para cada uno. También pueden ser maestros o esclavos. El dispositivo maestro inicia la

transferencia de datos y además genera la señal de reloj, pero no es necesario que el maestro sea siempre el mismo dispositivo, esta característica se la pueden ir pasando los dispositivos que tengan esa capacidad. Esta característica hace que al bus I²C se le denomine bus multimaestro.

LENGUAJES DE PROGRAMACIÓN

Python es un lenguaje de programación interpretado cuya filosofía hace hincapié en una sintaxis que favorezca un código legible.

Se trata de un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, usa tipado dinámico y es multiplataforma.

Debido a su facilidad de uso y versatilidad, se ha optado en utilizar este lenguaje para implementar el código principal que correrá sobre la Raspberry.

Además, se utilizará algunos Shell scripts para ejecutar ciertos comandos en la plataforma.

CAPÍTULO 4: ROBÓTICA

Una vez explicadas las diferentes opciones, tanto software como hardware que existen en el mercado, introduciremos unos conceptos que ayudarán a entender mejor la finalidad del proyecto.

La robótica es la rama de la tecnología que se dedica al diseño, construcción, operación, disposición estructural, manufactura y aplicación de los robots.

La robótica combina diversas disciplinas como son: la mecánica, la electrónica, la informática, la inteligencia artificial, la ingeniería de control y la física.

Los robots pueden ser clasificados de diferentes maneras atendiendo a diferentes criterios:

SEGÚN SU CRONOLOGÍA

La que a continuación se presenta es la clasificación más común:

1.ª GENERACIÓN.

Manipuladores. Son sistemas mecánicos multifuncionales con un sencillo sistema de control, bien manual, de secuencia fija o de secuencia variable.

2.ª GENERACIÓN.

Robots de aprendizaje. Repiten una secuencia de movimientos que ha sido ejecutada previamente por un operador humano. El modo de hacerlo es a través de un dispositivo mecánico. El operador realiza los movimientos requeridos mientras el robot le sigue y los memoriza.

3.ª GENERACIÓN.

Robots con control sensorizado. El controlador es una computadora que ejecuta las órdenes de un programa y las envía al manipulador para que realice los movimientos necesarios.

4.ª GENERACIÓN.

Robots inteligentes. Son similares a los anteriores, pero además poseen sensores que envían información a la computadora de control sobre el estado del proceso.

Esto permite una toma inteligente de decisiones y el control del proceso en tiempo real.

SEGÚN SU ESTRUCTURA

La estructura, es definida por el tipo de configuración general del Robot, puede ser metamórfica. El concepto de metamorfismo, de reciente aparición, se ha introducido para incrementar la flexibilidad funcional de un Robot a través del cambio de su configuración por el propio Robot. El metamorfismo admite diversos niveles, desde los más elementales (cambio de herramienta o de efecto terminal), hasta los más complejos como el cambio o alteración de algunos de sus elementos o subsistemas estructurales. Los dispositivos y mecanismos que pueden agruparse bajo la denominación genérica del Robot, tal como se ha indicado, son muy diversos y es por tanto difícil establecer una clasificación coherente de los mismos que resista un análisis crítico y riguroso. La subdivisión de los Robots, con base en su arquitectura, se hace en los siguientes grupos: poliarticulados, móviles, androides, zoomórficos e híbridos.

POLIARTICULADOS

En este grupo se encuentran los Robots de muy diversa forma y configuración, cuya característica común es la de ser básicamente sedentarios (aunque excepcionalmente pueden ser guiados para efectuar desplazamientos limitados) y estar estructurados para mover sus elementos terminales en un determinado espacio de trabajo según uno o más sistemas de coordenadas, y con un número limitado de grados de libertad. En este grupo, se encuentran los manipuladores, los Robots industriales, los Robots cartesianos y se emplean cuando es preciso abarcar una zona de trabajo relativamente amplia o alargada, actuar sobre objetos con un plano de simetría vertical o reducir el espacio ocupado en el suelo.

MÓVILES

Son Robots con gran capacidad de desplazamiento, basados en carros o plataformas y dotados de un sistema locomotor de tipo rodante. Siguen su camino por telemando o guiándose por la información recibida de su entorno a través de sus

sensores. Estos Robots aseguran el transporte de piezas de un punto a otro de una cadena de fabricación. Guiados mediante pistas materializadas a través de la radiación electromagnética de circuitos empotrados en el suelo, o a través de bandas detectadas fotoeléctricamente, pueden incluso llegar a sortear obstáculos y están dotados de un nivel relativamente elevado de inteligencia.

ANDROIDES

Son Robots que intentan reproducir total o parcialmente la forma y el comportamiento cinemático del ser humano. Actualmente, los androides son todavía dispositivos muy poco evolucionados y sin utilidad práctica, y destinados, fundamentalmente, al estudio y experimentación. Uno de los aspectos más complejos de estos Robots, y sobre el que se centra la mayoría de los trabajos, es el de la locomoción bípeda. En este caso, el principal problema es controlar dinámicamente y coordinadamente en el tiempo real el proceso y mantener simultáneamente el equilibrio del Robot.

ZOOMÓRFICOS

Los Robots zoomórficos, que considerados en sentido no restrictivo podrían incluir también a los androides, constituyen una clase caracterizada principalmente por sus sistemas de locomoción que imitan a los diversos seres vivos. A pesar de la disparidad morfológica de sus posibles sistemas de locomoción es conveniente agrupar a los Robots zoomórficos en dos categorías principales: caminadores y no caminadores. El grupo de los Robots zoomórficos no caminadores está muy poco evolucionado. Los experimentos efectuados en Japón basados en segmentos cilíndricos biselados acoplados axialmente entre sí y dotados de un movimiento relativo de rotación. Los Robots zoomórficos caminadores múltipedos son muy numerosos y están siendo objeto de experimentos en diversos laboratorios con vistas al desarrollo posterior de verdaderos vehículos terrenos, pilotados o autónomos, capaces de evolucionar en superficies muy accidentadas. Las aplicaciones de estos Robots serán interesantes en el campo de la exploración espacial y en el estudio de los volcanes.

HÍBRIDOS

Corresponden a aquellos de difícil clasificación, cuya estructura se sitúa en combinación con alguna de las anteriores ya expuestas, bien sea por conjunción o por yuxtaposición. Por ejemplo, un dispositivo segmentado articulado y con ruedas, es al mismo tiempo, uno de los atributos de los Robots móviles y de los Robots zoomórficos.

Para este proyecto, nos centraremos en un robot móvil de 4ª generación.

CAPÍTULO 5: SOLUCIÓN TÉCNICA

A continuación, pasaremos a describir la solución completa y sus características.

La plataforma que se va desarrollar pretende dotar a un robot móvil de inteligencia cubriendo los siguientes aspectos:

1. Telemando (ejecución de órdenes)
2. Telecontrol (control remoto)
3. Vigilancia (streaming de video)
4. Movimiento (motores DC)
5. Pan/Tilt para la cámara
6. Sensor de Ultrasonidos (medidor de distancias)
7. Sensor de Temperatura
8. Sensor de Humedad
9. Sensor de Luminosidad
10. Autonomía (Batería)

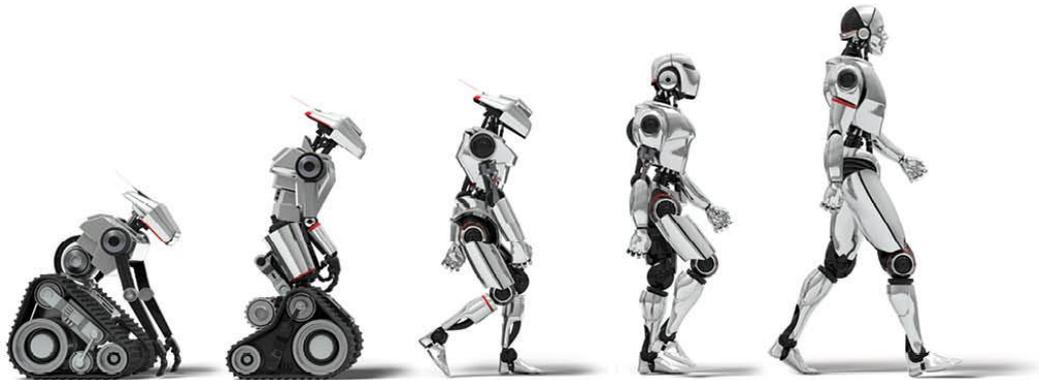


Figura 1 Evolución de la robótica

La solución será fácilmente extensible a otras necesidades en cualquier entorno. Para que sea lo más flexible posible, se configurara la plataforma para que sea accesible desde el exterior utilizando algún servicio de DNS dinámico si no disponemos de una IP fija. De este modo, podremos gestionarla remotamente.

Además, para dotarla de versatilidad, la configuraremos de modo que podamos realizar las consultas o ejecutar en remoto de varias formas. Para que pueda ser usada por el mayor número de usuarios posible, no es necesario preinstalar nada salvo un cliente SSH en caso de que no se disponga de él y se puede acceder desde cualquier dispositivo con conexión a internet: PC, portátil, tablet, móvil, etc.

Puesto que cada día el uso de dispositivos móviles está a la orden del día, se podría aprovechar las APIs de algunas de las aplicaciones de mensajería disponibles en el mercado para interactuar con la plataforma. Otra opción será desarrollar una aplicación propia para aprovechar al máximo las prestaciones. Esto será algo a explicar en el apartado de trabajos futuros.

CONFIGURACIÓN DE LA PLATAFORMA

Lo primero que debemos hacer es configurar la plataforma.

Para ello, nos descargaremos el S.O. Raspbian que hemos elegido y lo instalamos en la tarjeta SD siguiendo uno de los múltiples manuales que existen en internet. Puesto que vamos a permitir conexiones a nuestra Raspberry desde el exterior, es recomendable modificar el password por defecto de nuestro usuario.

No necesitamos de momento ninguna configuración especial del sistema operativo.

Para dotar de un poco de independencia a los cables a la plataforma, le vamos a configurar un dongle wifi. De este modo, no será necesario que la plataforma se encuentre cerca del router.

CONFIGURACIÓN DE E/S

Para que la Raspberry interactúe con el entorno, vamos a aprovechar los pines de entrada/salida de propósito general de los que dispone (GPIO). Como ya

comentamos en apartados anteriores, esta característica fue una de las decisivas a la hora de decantarnos por esta plataforma.

En la siguiente figura se puede ver la distribución de los pines para el modelo B:

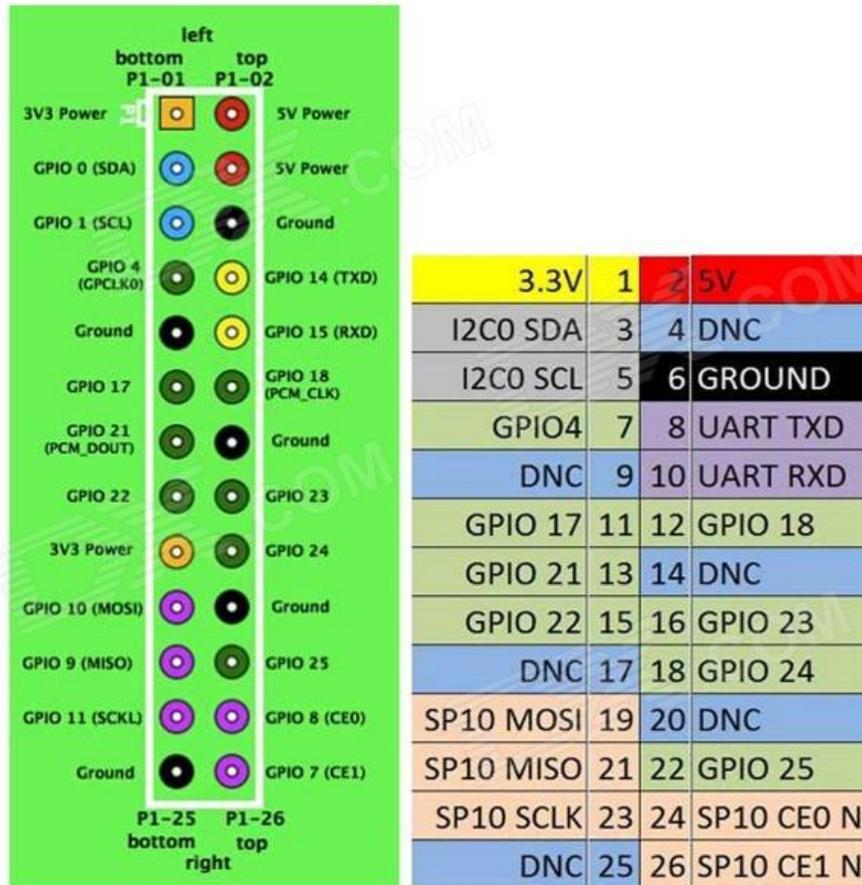


Figura 2 Puertos GPIO

Podemos apreciar un pin de 5v, otro de 3.3v, tierra (GND), seis de tipo DNC (Do Not Connect), ocho GPIO normales y otros especiales. Por ejemplo, entre los especiales nos encontramos los UART TXD y RXD para comunicaciones serie, la interfaz SDA, SCL, MOSI, MISO, SCLK, CE0, CE1. Como ya hemos comentado anteriormente, prestaremos especial atención a los pines de I2C puesto que los usaremos para conectar los diferentes periféricos.

Al estar trabajando sobre un sistema operativo UNIX donde la filosofía es que todo es un fichero, incluido el hardware, podremos acceder a los puertos GPIO directamente desde un terminal como si de un fichero de texto se tratase.

Sin embargo, vamos a aprovechar alguna de las librerías disponibles que nos facilitaran el acceso a los pines desde otros lenguajes de programación de alto nivel simplemente con importarlas:

- RPi.GPIO
- Wiring Pi.

En este caso, hemos seleccionado la primera. Hay que tener en cuenta que existen diferencias entre los nombrados de los pines según que librería se use (véase la imagen siguiente).

P1: The Main GPIO connector						
WiringPi Pin	BCM GPIO	Name	Header	Name	BCM GPIO	WiringPi Pin
		3.3v	1 2	5v		
8	Rv1:0 - Rv2:2	SDA	3 4	5v		
9	Rv1:1 - Rv2:3	SCL	5 6	0v		
7	4	GPIO7	7 8	TxD	14	15
		0v	9 10	RxD	15	16
0	17	GPIO0	11 12	GPIO1	18	1
2	Rv1:21 - Rv2:27	GPIO2	13 14	0v		
3	22	GPIO3	15 16	GPIO4	23	4
		3.3v	17 18	GPIO5	24	5
12	10	MOSI	19 20	0v		
13	9	MISO	21 22	GPIO6	25	6
14	11	SCLK	23 24	CE0	8	10
		0v	25 26	CE1	7	11
WiringPi Pin	BCM GPIO	Name	Header	Name	BCM GPIO	WiringPi Pin

Tabla 1 Tabla de conversión GPIO

CONFIGURACIÓN DE PERIFÉRICOS

Para cubrir todos los objetivos de la plataforma, se ha optado por separar en tres grandes bloques la configuración. Dicha distribución se ha hecho en función de los diferentes componentes HW usados. De esta forma, podemos hablar de:

- Bloque de Sensorización

- Bloque de Movimiento
- Bloque de Mando y Control

A continuación, pasaremos a detallar con más detalle cada uno de los bloques.

BLOQUE DE SENSORIZACIÓN

Este bloque será el encargado de gestionar todos los sensores involucrados en el reconocimiento del terreno.

Una placa arduino nano dedicada se encargará de controlar cada uno de los sensores para poder consultarlos remotamente.

Actualmente el robot dispone de las siguientes capacidades:

- Un sensor de luz con el que se encenderá automáticamente un LED cuando la luminosidad del entorno esté por debajo de un determinado umbral (configurable).
- Un sensor de humedad y temperatura que nos devolverá la temperatura del entorno en grados centígrados y la humedad relativa en tanto por ciento. Además, mediante una formula podemos convertir dichos grados a Fahrenheit y obtener la sensación térmica del entorno.

Estas capacidades pueden ser ampliadas añadiendo nuevos sensores. Habrá que tener en cuenta el número de pines disponibles en la placa. Esto será ampliamente explicado en el apartado de trabajos futuros.

El sensor de humedad y temperatura elegido ha sido un DHT11. Dispone de una salida digital calibrada. Su tecnología garantiza la alta fiabilidad y una excelente estabilidad a largo plazo. Es compatible con la tecnología Arduino, PIC, AVR, COP, DSP, STM32, etc.

- Compatible con sistemas electrónicos operando entre 3v-5v
- Corriente máxima de 2.5 mA cuando se realiza la conversión.
- Humedad relativa: 0-80% ($\pm 5\%$)
- Temperatura: 0-50°C ($\pm 2^\circ\text{C}$)
- Tiempo de respuesta: ≈ 10 segundos
- 3 pines de conexión (VCC, GND y DATA)

- No requiere componentes activos externos.

El sensor de luz que se va a utilizar es un LDR (Light Dependent Resistor o resistencia dependiente de la luz). Con esto pretendemos emular un sistema de regulación de luz donde se enciende un LED según va cayendo la noche compensando la deficiencia de luz.

Aunque eléctricamente está conectado a otra placa de Arduino, el robot también dispone de un sensor de ultrasonido. Dicho sensor será controlado remotamente desde la Raspberry. El sensor ultrasónico HC-SR04 es un sensor que, entre otras posibles aplicaciones, sirve para medir distancias. Funciona enviando un pulso de ultrasonidos (inaudible para el oído humano por su alta frecuencia) a través de uno de los cilindros que componen el sensor y esperando a que dicho sonido rebote sobre un objeto y vuelva. El retorno es captado por el otro cilindro del sensor

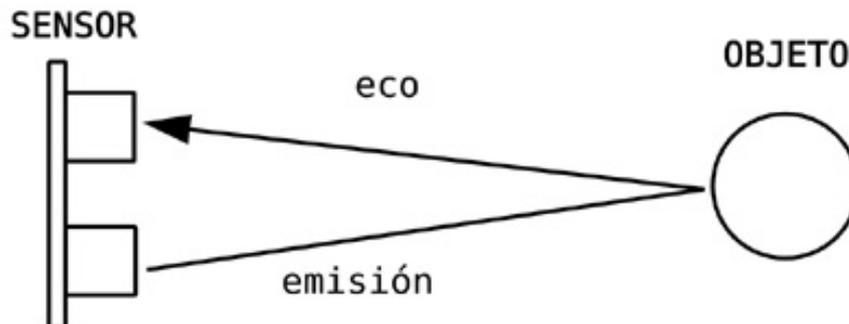


Figura 3 HC-SR04 Funcionamiento

Sabemos que la velocidad del sonido en el aire es 340 m/segundo así que calculando el tiempo transcurrido entre el envío del pulso y la recepción de la señal de retorno y luego aplicando una sencilla fórmula matemática obtenemos la distancia entre el sensor y el objeto que hay delante.

Este módulo tiene un área de trabajo de entre 3cm y 3m con una precisión de 3mm y se alimenta a 5V (15 mA).

Este sensor se podrá utilizar para un modo autónomo donde el robot vagabundee evitando chocar con los obstáculos que se encuentre.

Por último y no por ello menos importante, el robot dispone de una cámara para poder disponer de las imágenes que vea el robot durante su expedición. La cámara seleccionada es una cámara CSI que se conectará directamente a la Raspberry en un puerto específico para ella. La elección de este modelo se ha basado principalmente en que no ocupa un puerto USB ni requiere una alimentación adicional, lo cual aporta autonomía al Rover. Las características son bastante buenas para lo requerido en un principio: 5Mpx y distintos modos de video como 1080p30, 720p60 y VGA90.

Para realizar el streaming de video se ha utilizado un programa llamado MJPG-streamer. Este programa se encargará de servir las imágenes de la cámara vía HTTP para poder ser reproducida en cualquier navegador o incluso en cualquier reproductor de video en red. Para facilitar la habilitación/deshabilitación del servicio de streaming se ha creado un Shell script que arranca el servicio.

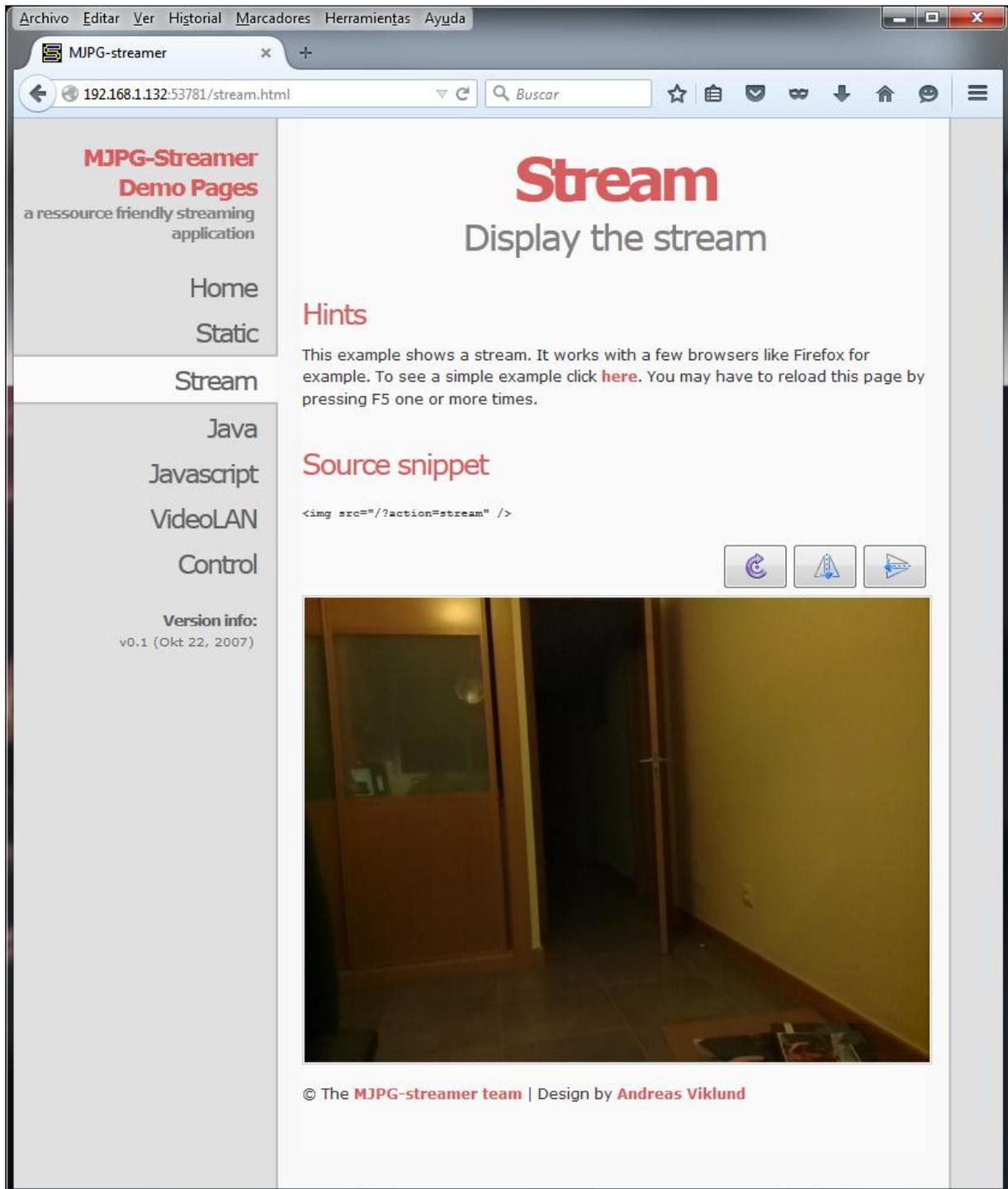


Figura 4 MJPG-Streamer

Este programa incluye un servidor web que permite compartir las imágenes obtenidas de la cámara de diferentes maneras: estáticas, en streaming, proporciona los códigos para embeberlos en una web mediante java, javascript, etcétera.

BLOQUE DE MOVIMIENTO

Este bloque será el encargado de mover el Rover por el entorno para realizar la exploración. Para este bloque se ha utilizado otra placa independiente de Arduino. Esta vez, el modelo elegido ha sido un Arduino UNO Rev3.

Como hemos comentado anteriormente, uno de los principales motivos por los que se ha utilizado un microcontrolador adicional a la Raspberry es debido a que de este modo independizamos eléctricamente la placa evitando que cualquier pico de corriente indeseada ponga en peligro la integridad de la placa. Esto es especialmente importante en este caso, puesto que para alimentar los motores de continua que moverán el robot, se necesita una tensión de al menos 6V y los pines de E/S de la RPi no lo soportaría.

Aunque se explicará posteriormente, para mover el chasis del robot se utilizarán dos motores de continua. Para controlar dichos motores se hará uso de un puente en H (L298N)

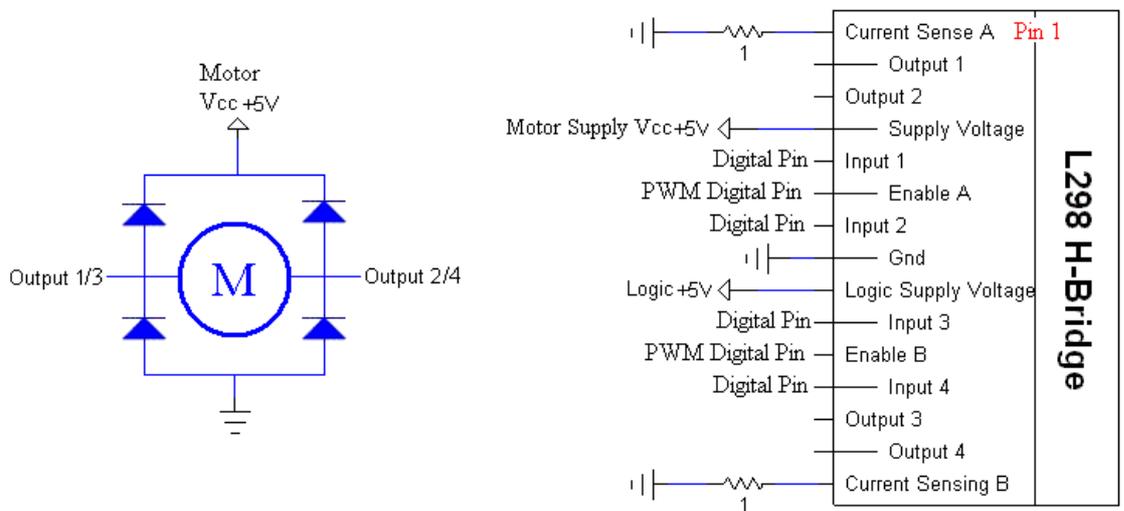


Figura 5 L298N Esquema

Este módulo basado en el chip L298N permite controlar dos motores de corriente continua o un motor paso a paso bipolar de hasta 2 amperios.

El módulo cuenta con todos los componentes necesarios para funcionar sin necesidad de elementos adicionales, entre ellos diodos de protección y un regulador

LM7805 que suministra 5V a la parte lógica del integrado L298N. Cuenta con jumpers de selección para habilitar cada una de las salidas del módulo (A y B). La salida A está conformada por OUT1 y OUT2 y la salida B por OUT3 y OUT4. Los pines de habilitación son ENA y ENB respectivamente.

Cuando el jumper de selección de 5V se encuentra activo, el módulo permite una alimentación de entre 6V a 12V DC. Como el regulador se encuentra activo, el pin marcado como +5V tendrá un voltaje de 5V DC. Este voltaje se puede usar para alimentar la parte de control del módulo (Arduino), pero recomendamos que el consumo no sea mayor a 500 mA.

Cuando el jumper de selección de 5V se encuentra inactivo, el módulo permite una alimentación de entre 12V a 35V DC. Como el regulador no está funcionando, tendremos que conectar el pin de +5V a una tensión de 5V para alimentar la parte lógica del L298N. Usualmente esta tensión es la misma de la parte de control, ya sea un microcontrolador o Arduino.

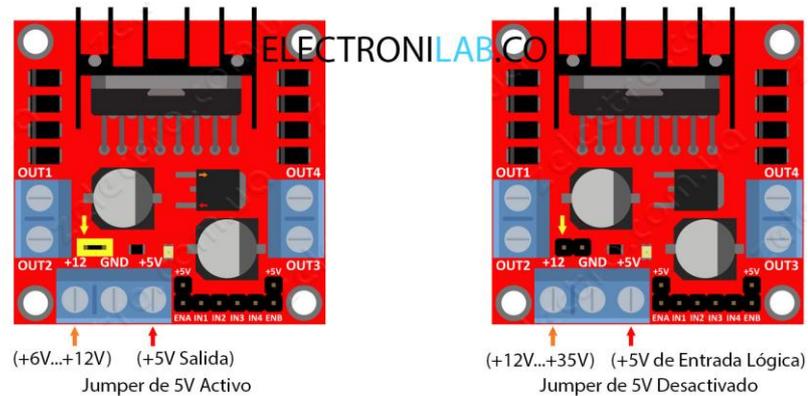


Figura 6 L298N Alimentación

Para hacer que el robot avance a distintas velocidades, los pines de enable se conectarán a los pines PWM del Arduino y controlaremos la velocidad variando el ancho del pulso por código. Combinando los pines de salida 1/2 y 3/4 conseguiremos que el cada motor gire en un sentido o en otro permitiendo que avance, gire a la derecha, gire a la izquierda o vaya marcha atrás.

Además de los motores de avance del robot, este módulo controla también los dos servos encargados de realizar las funciones de pan/tilt con las que moveremos una plataforma con cuatro grados de libertad donde estarán situados la cámara y el sensor de ultrasonido.

BLOQUE DE MANDO Y CONTROL

Este bloque será el que aporta la inteligencia al robot. Será el que controle remotamente el movimiento del robot así como el envío de las diferentes órdenes para solicitar la información que han podido recabar los diferentes sensores. También encenderá y apagará el streaming del video y controlará los movimientos de los servos para orientar la cámara y el sensor de ultrasonido.

Para todo ello se ha necesitado interconectar todos los bloques de la siguiente forma:

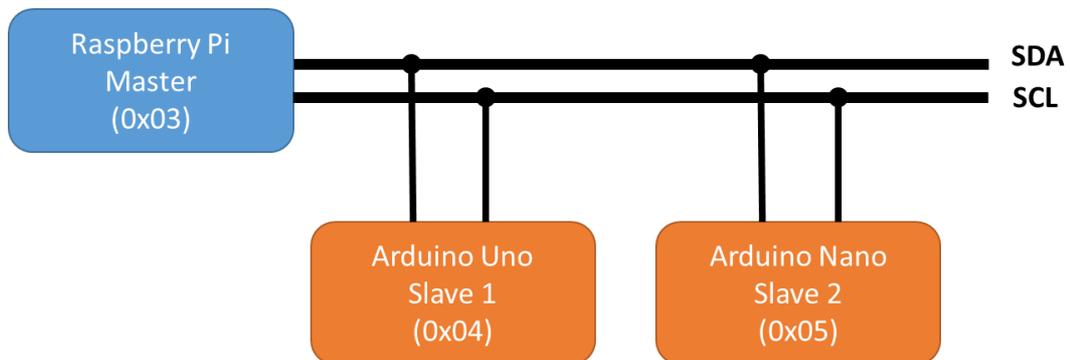


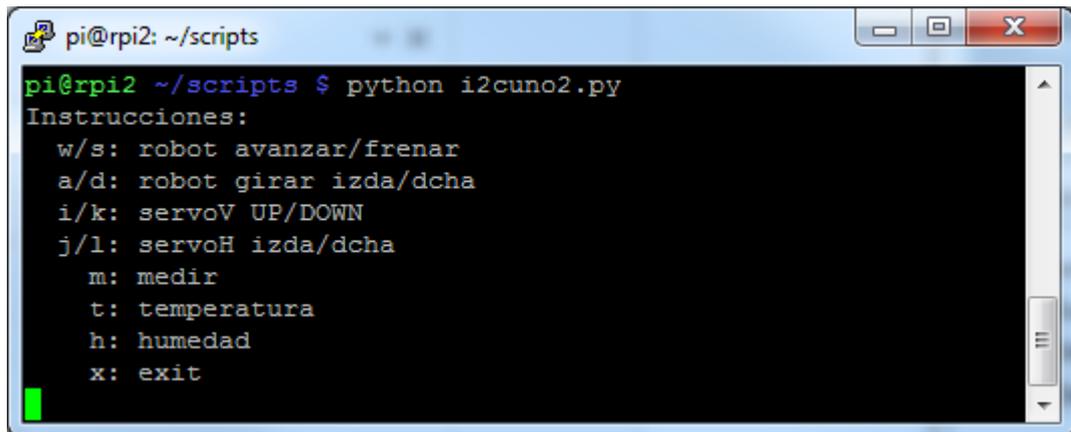
Figura 7 Interconexión de bloques

Además, se ha creado un protocolo de comunicación sobre I2C en el que se ha definido la forma en la que enviaremos la información a los distintos periféricos.

En la raspberry se ha creado un script en Python que dispone de un menú desde el que se podrá controlar el movimiento del robot, el movimiento de los servos para el pan/tilt con el que moveremos la cámara y el sensor de ultrasonido.

También podremos solicitar la información de los diferentes sensores que tiene incorporados.

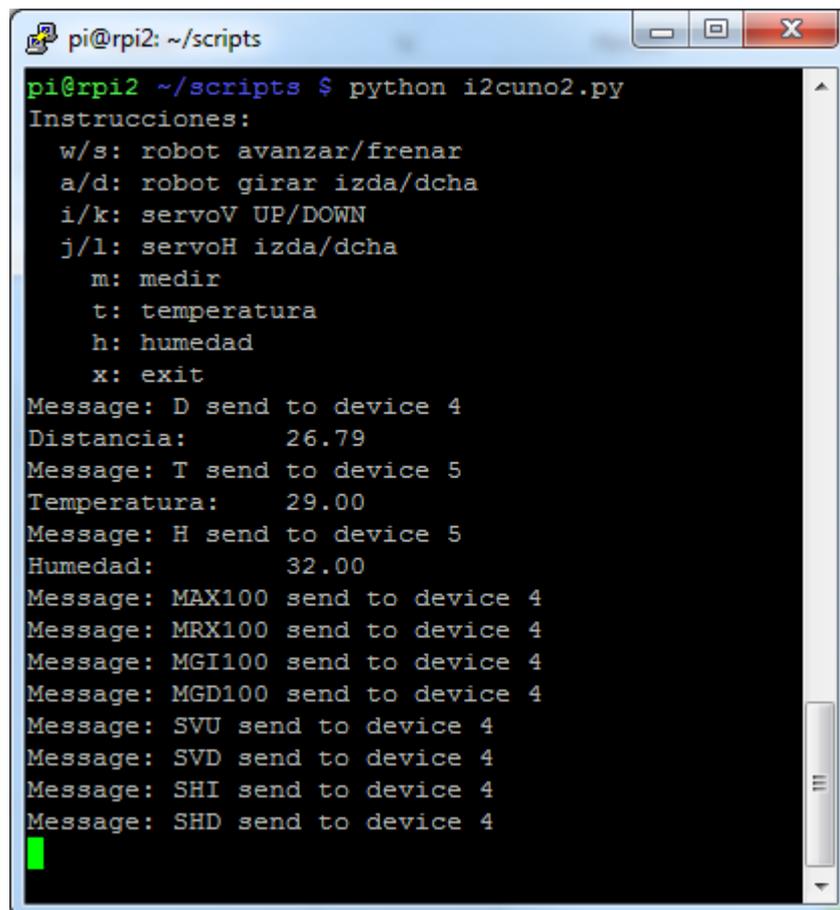
A continuación se muestra el menú:



```
pi@rpi2: ~/scripts
python i2cuno2.py
Instrucciones:
w/s: robot avanzar/frenar
a/d: robot girar izda/dcha
i/k: servoV UP/DOWN
j/l: servoH izda/dcha
m: medir
t: temperatura
h: humedad
x: exit
```

Figura 8 Menú

En la siguiente imagen se puede apreciar la salida después de haber ejecutado algunos comandos:



```
pi@rpi2: ~/scripts
python i2cuno2.py
Instrucciones:
w/s: robot avanzar/frenar
a/d: robot girar izda/dcha
i/k: servoV UP/DOWN
j/l: servoH izda/dcha
m: medir
t: temperatura
h: humedad
x: exit
Message: D send to device 4
Distancia:      26.79
Message: T send to device 5
Temperatura:    29.00
Message: H send to device 5
Humedad:       32.00
Message: MAX100 send to device 4
Message: MRX100 send to device 4
Message: MGI100 send to device 4
Message: MGD100 send to device 4
Message: SVU send to device 4
Message: SVD send to device 4
Message: SHI send to device 4
Message: SHD send to device 4
```

Figura 9 Ejecución de comandos

En este ejemplo, se han pulsado todas las opciones una vez:

- Primero se ha medido la distancia en cm desde el robot al obstáculo más cercano. En este caso, 26,79cm.
- Luego se ha consultado la temperatura que en este caso es de 29°C
- También se ha medido la humedad relativa de la habitación donde se encuentra el robot que es del 32%.
- Los siguientes 4 comandos han sido avanza/retrocede/gira izquierda/gira derecha
- Los últimos 4 comandos han sido para mover los servos. Hay dos servos: uno que controla el movimiento vertical y otro el horizontal. Como se puede intuir por los comandos, la secuencia ha sido: vertical arriba, vertical abajo, horizontal izquierda y horizontal derecha.

Merece la pena mencionar que, como se puede apreciar en la salida del script, cuando se solicita información del sensor de temperatura o humedad, se envía el dispositivo 5 (dirección 0x05 del bus I2C correspondiente con el Arduino Nano).

Para el resto de órdenes, la Raspberry envía los comandos pertinentes al dispositivo 4 (dirección 0x04 correspondiente con el Arduino Uno) que es donde están conectados los servos, el puente en H que gestiona los motores de continua y el sensor de ultrasonidos para medir las distancias.

También se puede intuir por la salida del programa el protocolo que se ha implementado para gobernar las placas Arduino desde la Raspberry. Consiste en enviar un string por el bus a la dirección del dispositivo correspondiente que se despertará cuando llegue un dato para él. Una vez recibe el primer carácter, entra en juego un parser que, en función de los siguientes caracteres, ejecutará la función correspondiente.

Por ejemplo, cuando el dispositivo 4 (bloque de movimiento) reciba el comando MAX100, entenderá que:

- M es que va a utilizar los motores
- A es que quiere avanzar (por lo que moverá hacia delante tanto el motor izquierdo como el derecho)

- X es un carácter de relleno para que siempre se envíen comandos del mismo tamaño.
- 100 Es el Duty Cycle del PWM, en este caso es del 100% que es que avance a la máxima potencia.

En caso de que la Raspberry envíe el comando SVU, querrá decir que:

- S indica que vamos a controlar los Servos
- V hace referencia al servo que controla el movimiento vertical
- U indica que queremos moverlo hacia arriba.

Como se puede apreciar, se ha definido un protocolo que permite incrementar el número de órdenes y de los parámetros en función de las necesidades permitiendo la ampliación de ordenes si se incluyesen nuevos sensores o placas..

Cuando enviamos una orden para consultar el valor de un sensor, lo que hace la placa de Arduino es contestar a esa orden con los valores numéricos convertidos a caracteres para que puedan ser enviados por el bus I2C y recibidos y mostrados por la Raspberry

Como se ha mencionado anteriormente, la Raspberry se ha configurado con para que tenga conexión a Internet. Además, se ha configurado un servicio de DNS Dinámico que permite acceder a la plataforma de forma remota con un nombre predeterminado. De esta forma, dotamos al robot de una conectividad tal que puede ser accedido desde cualquier dispositivo que tenga una conexión a Internet. Es decir, el robot podrá ser controlado de un PC, móvil o Tablet que tenga instalado un cliente SSH para establecer una conexión con ella.

MONTAJE DEL ROBOT

En la siguiente figura se puede apreciar el conexionado de varios de los componentes que componen el robot.

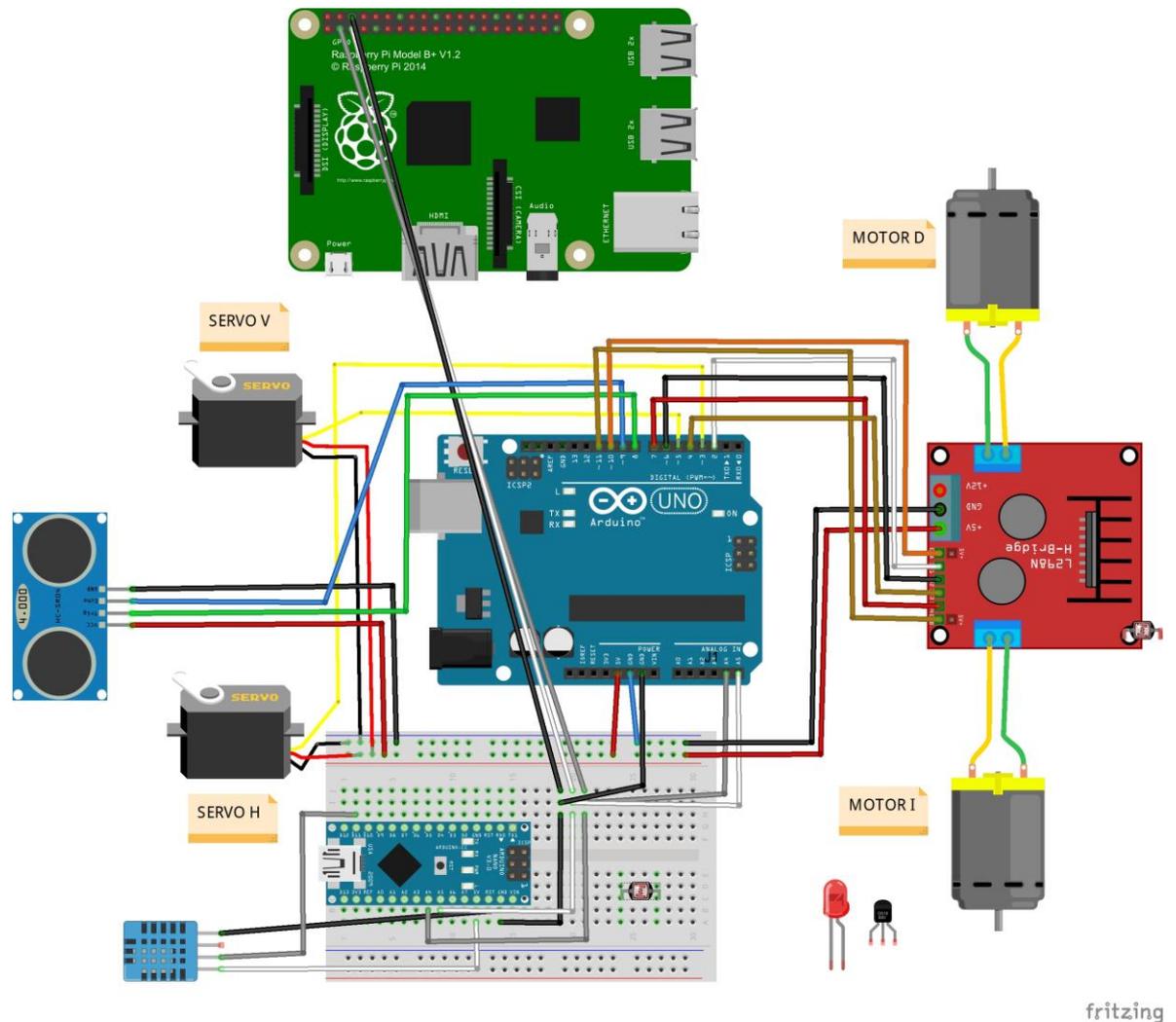


Figura 10 Montaje Completo

Las únicas restricciones de montaje relacionadas con los pines a usar, son las siguientes:

- Las entradas de EN (enablers) del puente en H que gestionan el Duty Cycle de los motores de continua deben estar conectadas a las entradas digitales de Arduino que pueden proporcionar una señal PWM con la que, variando el ancho del pulso por software, conseguiremos que los motores giren a diferente velocidad.
- Lo mismo ocurre con la conexión de los servos.
- Las dos conexiones de datos y reloj del bus I2C deben hacerse a los pines dedicados para ello: A4 y A5 de los Arduino y GPIO2 y GPIO3 de la Raspberry-

El resto de conexiones (sensores, ultrasonidos, etc) podrán conectarse a cualquier entrada/salida digital de las placas Arduino que este libres.

Todo esto se montará sobre el siguiente chasis:

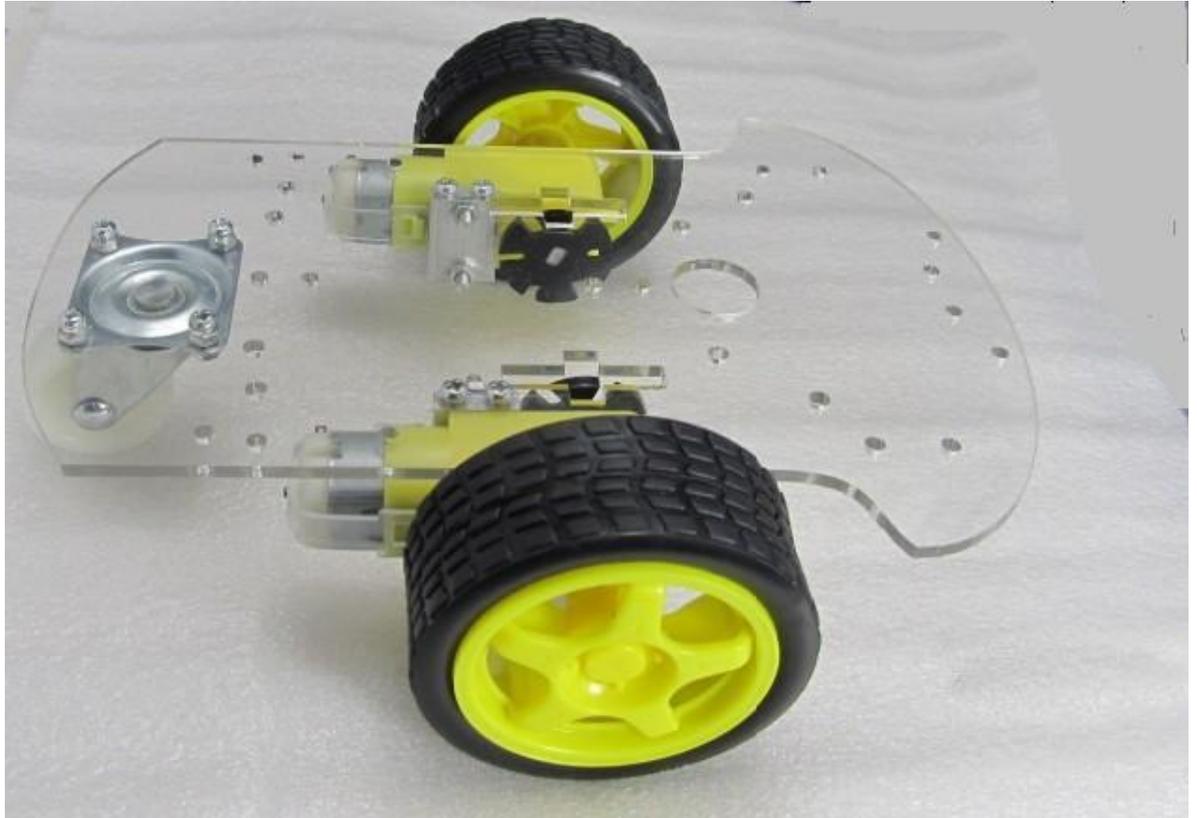


Figura 11 Chasis del robot

Para alimentar los diferentes elementos que conforman el Rover, se ha optado por la siguiente configuración:

- La Raspberry será alimentada por una batería externa USB portátil de 10.000 mAh que proporciona el amperaje y autonomía suficiente para mantener la Raspberry encendida durante mucho tiempo.
- El puente en H será alimentado por 4 pilas AAA de 1,5V. Al ser alimentado por más de 5V podemos utilizar la salida de dicho integrado que proporciona +5V.
- Ambas placas de Arduino (sensores y movimiento) se alimentarán con la salida de +5 que proporciona el LM298N.

CAPÍTULO 6: CONCLUSIONES Y FUTURAS LÍNEAS

Como se ha ido explicando a lo largo de la memoria, se ha construido un robot que es capaz, de forma remota, de ser controlado y comandado para dirigirlo y utilizar los sensores de los que está provisto a distancia.

Este comportamiento puede ser muy útil en determinados entornos en los que la presencia humana es difícil, bien sea por peligros o por accesibilidad. Lo único que requiere es que el robot y la persona que lo comande estén conectados de alguna forma.

En nuestro caso, la conexión la hemos configurado, no solo para estar accesibles dentro de una LAN, sino también a través de un acceso a Internet para que pueda ser utilizado a través de cualquier dispositivo y desde prácticamente cualquier lugar del mundo.

Como se definía al principio del documento, se han mantenido los principios de:

- Modularidad. Se han creado bloques diferenciados de funcionalidades que están aislados entre si.
- Escalabilidad. La configuración es tal, que permite añadir tantos módulos nuevos como sean necesarios según el propósito que se persiga.
- Bajo coste. Se ha utilizado hardware libre como son las placas de Arduino y Raspberry, que además tienen una gran comunidad detrás.

Los sensores que pueden integrarse en esta plataforma son prácticamente todos los que existen en el mercado, dependiendo del objetivo que se quiera cubrir. Se han realizado pruebas con módulos de transmisión de RF que permitirían comunicación autónoma con otros dispositivos, con sensores de proximidad de RFID para identificación y acceso restringido, sensores de movimiento para detección de intrusos, pantallas LCDs y sensores de humedad de tierra entre otros. También se pueden controlar relés para activar o desactivar diferentes dispositivos de forma remota.

También se podría incorporar al robot un dispositivo de posicionamiento GPS para tener localizado el robot o marcar los datos recopilados con coordenadas GPS.

Si se desea también se puede añadir sensores de infrarrojos o láser y un acelerómetro con giroscopio para enriquecer la navegación y plantear un guiado del vehículo de forma no tripulada y autónoma. Esto tiene muchas aplicaciones como el reconocimiento de terrenos donde el robot puede navegar persiguiendo unos objetivos (o vagabundear) implementando alguno de la gran variedad de algoritmos de planificación existentes o que sean desarrollados expresamente para él. Se ha hecho una prueba con un modo autónomo de navegación gestionado únicamente con el sensor de ultrasonidos que hacía que el robot vagabundease evitando los obstáculos que se encontrase en su camino. Puesto que finalmente, no era el objetivo del proyecto, no se ha mantenido en la versión final.

Al disponer una cámara HD, el robot es capaz de capturar imágenes y videos. Esta capacidad, abre las puertas a un mundo en auge y con infinitud de posibilidades como es el tratamiento de imágenes. Se pueden implementar detectores, clasificadores y realizar tracking de objetos de una forma relativamente sencilla. La Raspberry dispone de librerías OpenCV compiladas para su arquitectura. Estas librerías son de SW libre y muy potentes por lo que permiten crear, con relativa facilidad, algunos algoritmos de tratamiento de imágenes y/o video en tiempo real. La única limitación es que, puesto que el HW de Raspberry es limitado, el rendimiento de estas funciones está limitado. No es aconsejable capturar imágenes a gran resolución ni con una tasa alta de frames por segundo porque aumenta drásticamente el tiempo de procesado. Se han hecho pruebas de detección de señales con resultado satisfactorio pero no se han implementado en la versión final debido a que el tiempo de respuesta era demasiado alto para guiar el robot mediante este método.

Todos estos datos que proporcionan estos sensores son consultados en tiempo real en este proyecto. Sin embargo, también pueden ser almacenados en una Base de Datos dentro de la propia plataforma o ser enviados a un servidor remoto gracias a

su conectividad para luego ser explotados: página web de consulta, tablas, estadísticas, gráficas, etc.¹

Dependiendo del terreno por el que queramos que se desplace el robot, podría ser interesante que los motores tengan reductora o las ruedas sean de oruga.

Otro de los campos de mejoras podría ser la autonomía. Aunque para el proyecto se han utilizado baterías externas de USB y pilas, existen baterías LiPo especializadas para robots y coches de radiocontrol que son mucho más pequeñas y ligeras cubriendo este apartado con unas mejores condiciones.

Por último, no querría olvidar como posible línea de futuro que el robot, que hemos configurado como terrestre tenga la opción de convertirse en un DRON. Para poder hacer esto habría que añadirle ciertas capacidades como estabilizadores, acelerómetros, giroscopios, etc. Esta opción abre un mundo de posibilidades en áreas como el salvamento y socorrismo, seguridad, gestión de entornos con accesibilidad limitada. Se deja abierto al lector para que dé rienda suelta a su imaginación

¹ Para más información, véase mi Proyecto Fin de Carrera titulado PLATAFORMA DE INTEGRACIÓN DE SENSORIZACIÓN Y ACTUACIÓN SOBRE DISPOSITIVOS EN UN ENTORNO INTELIGENTE

BIBLIOGRAFÍA.

- Documentación Raspberry Pi: <http://www.raspberrypi.org/>
- Documentación Arduino: <https://www.arduino.cc/>
- Documentación Python: <https://www.python.org/>
- Documentación para desarrollo en Android: <http://developer.android.com/sdk>
- Datasheet sensor temperatura y humedad: <https://learn.adafruit.com/downloads/pdf/dht.pdf>
- Datasheet sensor ultrasonido: http://elec Freaks.com/estore/download/EF03085-HC-SR04_Ultrasonic_Module_User_Guide.pdf
- Datasheet sensor LDR: <http://kennarar.vma.is/thor/v2011/vgr402/ldr.pdf>
- Documentación I2C: <http://www.i2c-bus.org/>

PLIEGO DE CONDICIONES.

En esta parte del proyecto se incluyen las condiciones hardware y software bajo las que se ha desarrollado la plataforma.

CONDICIONES DE MATERIALES Y EQUIPOS

HARDWARE UTILIZADO: REQUISITOS MÍNIMOS

Se ha utilizado un ordenador portátil con las siguientes características:

- Processor: 2x Intel(R) Core(TM)2 Duo CPU T9400 @ 2.53GHz
- Memory: 2056MB (592MB used)
- Operating System: Ubuntu 14.04.1 LTS
- Hard disk: 227,1 Gb
- Conectividad de red: 10/100/1000 Ethernet (RJ-45) +
- Perifericals: keyboard, TouchPad, Generic Mouse

Se ha utilizado una Raspberry Pi modelo B con las siguientes características:

- SoC: Broadcom BCM2835 (CPU + GPU + DSP + SDRAM + puerto USB)
- CPU: ARM1176JZF-S a 700 MHz (familia ARM11)
- GPU: Broadcom VideoCore IV, OpenGL ES 2.0, -2 y VC-1 (con licencia), 1080p30 H.264/MPEG-4 AVC
- Memoria (SDRAM): 512 MB (compartidos con la GPU)
- Puertos USB 2.0: (vía hub USB integrado)
- Salidas de vídeo: Conector RCA (PAL y NTSC), HDMI (rev1.3 y 1.4), Interfaz DSI para panel LCD
- Salidas de áudio: Conector de 3.5 mm, HDMI
- Almacenamiento : SD / MMC / ranura para SDIO
- Conectividad de red: 10/100 Ethernet (RJ-45) via hub USB
- Periféricos de bajo nivel: 8 x GPIO, SPI, IC, UART
- Consumo energético: 700 mA, (3.5 W)
- Fuente de alimentación: 5 V vía Micro USB o GPIO header

- Dimensiones: 85.60mm x 53.98mm (3.370 x 2.125 inch)
- Sistema Operativo: GNU/Linux: Debian (Raspbian), Fedora (Pidora), Arch Linux (Arch Linux ARM), Slackware Linux, RISC OS

Se ha utilizado un Arduino UNO Rev3 con las siguientes características:

- Microcontrolador: ATmega328
- Voltaje: 5V
- Voltaje entrada (recomendado): 7-12V
- Voltaje entrada (limites): 6-20V
- Digital I/O Pins: 14 (de los cuales 6 son salida PWM)
- Entradas Analógicas: 6
- DC Current per I/O Pin: 40 mA
- DC Current parar 3.3V Pin: 50 mA
- Flash Memory: 32 KB (ATmega328) de los cuales 0.5 KB son utilizados para el arranque
- SRAM: 2 KB (ATmega328)
- EEPROM: 1 KB (ATmega328)
- Clock Speed: 16 MHz

Se ha utilizado un Arduino Nano con las siguientes características:

- Microcontrolador Atmel ATmega168 o ATmega328
- Operating Voltage (logic level) 5 V
- Input Voltage (recomendado) 7-12 V
- Input Voltage (limite) 6-20 V
- 14 Digital I/O Pins (de los cuales 6 proporcionan una salida PWM)
- 8 Analog Input Pins
- DC Current per I/O Pin: 40 mA
- Flash Memory 16 KB (ATmega168) o 32 KB (ATmega328) de los cuales 2 KB son usados por el bootloader
- SRAM1 KB (ATmega168) o 2 KB (ATmega328)
- EEPROM 512 bytes (ATmega168) o 1 KB (ATmega328)
- Clock Speed 16 MHz

- Dimensions 0.73" x 1.70"
- Length 45 mm
- Width 18 mm
- Weigth 5 g

Se han utilizado los siguientes perifericos para la plataforma:

- Almacenamiento: SD 4Gb SandDisk
- Conectividad WIFI: Adaptador USB Nano Inalámbrico N de 150MbpsTL-WN725N 802. 11 b/g/n
- Sensor de temperatura: DHT11
- Sensor de humedad: DHT11
- Sensor de movimiento: PIR Sensor 555-28027
- Sensor de luz: LDR
- Sensor de ultrasonidos: HC-SR04
- Webcam: Raspberry Pi Camera CSI

SOFTWARE UTILIZADO

- Vi: editor de texto
- Fritzing: software libre de documentación de prototipos hardware
- Libreoffice: suite ofimática libre
- Arduino IDE: programación y carga del código en las placas de Arduino
- Python 2.7: lenguaje de programación

CONDICIONES DE EJECUCIÓN

No es necesario que el usuario tenga conocimientos sobre estas plataformas salvo que se pretenda realizar modificaciones sobre la misma. Si este es el caso, el usuario deberá estar familiarizado con los entornos Linux y deberá tener unos conocimientos básicos de programación para modificar el software y conocimientos básicos de electrónica para la parte hardware.

PRESUPUESTO

En esta sección se detalla el presupuesto de ejecución del proyecto. Desglosaremos este presupuesto en distintas secciones para una mayor facilidad a la hora de comprender el importe final del proyecto.

Presupuesto de material (MAT)

En este apartado están recogidos los costes de material y licencias de software necesarios para la realización del trabajo que ya se indicaron previamente en el pliego de condiciones y que se han utilizado tanto como para el desarrollo de la plataforma como para la posterior redacción de la presente documentación.

Concepto	Ud	Precio
Ordenador portátil	1	350,00
Raspberry Pi Modelo B	1	35,00
Tarjeta SD 4GB	1	10,95
Adaptador wifi 802.11b/g/n	1	8,95
Arduino UNO Rev3	1	22,58
Arduino Nano	1	32,80
Sensor Temp/Hum DHT11	1	2,10
Cableado M/M	40	6,90
Protoboard	1	4,95
Sensor LDR	1	2,65
Sensor de ultrasonidos	1	3,49
Puente en H	1	6,49
Webcam	1	26,91
Kit pan/tilt	1	5,80
Chasis Robot	1	19,80
Licencia Fritzing	1	0,00
Licencia Raspbian	1	0,00
Licencia Ubuntu Desktop	1	0,00
Licencia LibreOffice	1	0,00
TOTAL		539,37

Tabla 2 Presupuesto de material

VEHÍCULO MÓVIL NO TRIPULADO PROVISTO DE SENSORES Y ACTUADORES PARA
RECONOCER ENTORNOS

Presupuesto de mano de obra (MO)

En este apartado están recogidos los costes de mano de obra necesaria para la realización del trabajo dividida por perfiles y tasas en función de las actividades que se realizan.

Perfil	Horas	Tasa (€/h)	Precio
Dirección de Proyecto	80	45	3.600,00
Ingeniería de Sistemas	160	35	5.600,00
Desarrollo	240	30	7.200,00
Integración	160	30	4.800,00
Soporte	160	20	3.200,00
Logística	40	20	800,00
TOTAL			21.600,00

Tabla 3 Presupuesto de mano de obra

Presupuesto de otros costes derivados (OCD)

En este apartado se recogen otros costes incurridos durante la ejecución del proyecto.

Concepto	Ud	Precio unitario	Precio Total
Desplazamientos	15	20	300,00
Logística (gastos envío, etc.)	5	10	50,00
Soporte (Impresión, encuadernación, etc.)	4	35	140,00
TOTAL			190,00

Tabla 4 Presupuesto de OCDs

En resumen, el presupuesto de costes quedaría de la siguiente forma:

VEHÍCULO MÓVIL NO TRIPULADO PROVISTO DE SENSORES Y ACTUADORES PARA
RECONOCER ENTORNOS

Concepto	Ud	Precio
Costes de MAT	1	539,37
Costes de MO	1	21.600,00
Costes de OCD	1	190,00
TOTAL		22.329,37

Tabla 5 Presupuesto de costes

Aplicando un margen de beneficio sobre los materiales y otros costes de un 15% y de un 30% sobre la mano de obra, el presupuesto de venta sería:

Concepto	Ud	Precio
Venta de MAT	1	620,28
Venta de MO	1	28.080,00
Venta de OCD	1	218,50
TOTAL		28.918,78

Tabla 6 Presupuesto de ventas

Si se quisiera plantear el proyecto como una producción en serie y no como un proyecto de I+D, podríamos repercutir los costes de desarrollo sobre un número determinado de unidades.

Por ejemplo, la partida de material de la plataforma propiamente dicha son aproximadamente 180€/ud (0,6% del presupuesto). Si se venden a un precio de 250€/ud, estaríamos amortizando 70€ de desarrollo en cada unidad por lo que si se venden 415 unidades, la inversión inicial en desarrollo estaría amortizada al 100%. Cualquier venta posterior de esta plataforma daría unos beneficios del 28% por unidad de la cual se podría desviar una parte a otro proceso de actualización y evolución de la plataforma para mantener competitivo el producto.