

Diseño y desarrollo de un prototipo robótico con orange PI

Design and development of a robotic prototype with orange PI

John Jairo Leiva Mármol
*Universidad Nacional Abierta y A
Distancia – UNAD*

Correo electrónico: john.leiva@unad.
edu.co

John Alejandro Valero Henao
Autor de correspondencia
*. Universidad Nacional Abierta y
A Distancia – UNAD*

Correo electrónico: javh55@gmail.
com

Información del artículo: recibido: 25 de Abril de 2017, aceptado: 22 de Agosto de 2017
<https://10.17081/invinno.5.2.2760>

Resumen

En este artículo se presenta el desarrollo de un robot móvil controlado vía web, diseñado para ejecutar tareas de monitoreo. El sistema es controlado mediante la tarjeta Orange Pi+ 2E y una tarjeta Arduino. El Robot está construido en acrílico, posee Una pantalla LCD touch para la comunicación con el usuario, 4 motores (para mover el vehículo), un controlador de carga, Sensores ultrasónicos para medir la distancia entre otros dispositivos de control. Se programó el sistema para que efectuará las tareas de manera óptima transmitiendo las imágenes de la cámara, la respuesta de los sensores y las órdenes de control del usuario cada 100 ms, la herramienta empleada es el framework ROS (Robot Operating System). En este caso se empleó el sistema operativo Linux server en la Orange Pi; que permite usar múltiples lenguajes de programación como lo son Java, PHP, Xajax y SH, en conjunto se ejecutan en un solo programa o interfaz de usuario que, a través de HTML5 y un navegador WEB permiten: captar video de la cámara, leer los comandos del usuario y hacer la petición a la Arduino para que dé la orden a los motores de efectuar el movimiento, leer el sensor y publicar en la pantalla LCD.

Palabras Claves:

Orange pi, Arduino mega, Sensores, Actuadores, robot, Seguridad, OpenCV, Open Source, Open Hardware.



Abstract

In this abstract is presented the development of a mobile robot controlled on line, designed to perform monitor tasks. The system is controlled through the Orange Pi+ 2E card and an Arduino card. The Robot is built in acrylic; it has an LCD touch screen for the communication with the user, 4 motors (in order to move the vehicle), a loading control, and ultrasonic sensors to measure the distance between other control devises.

Keywords: *The system was programmed to carry out the tasks optimally, transmitting the images to the camera, a response to sensors and user control orders given every 100 meters, the tool used is the framework ROS (Robot Operating System). In this case the operative system Linux server in the Orange Pi was used; that permits to use multiple program languages such as Java, PHP, Xajax y SH, as a whole they are executed in one inter phased user program that, through the HTML5 and a WEB navigator let: capture a video from the camera, read the user commands and make an application to Arduino so that it can give the order to start the motion, read the sensor and show the video on the LCD.*

to login in with the Google account and database. The application is addressed to the users of Yopal City, Casanare department, who take the urban transportation service, to facilitate the information corresponding to the routes the minibus rides.

Index Terms
- Orange
pi, Arduino
mega, Sensors,
Performers,
robot, Security,
Open CV, Open
Source, Open
Hardware.

La robótica es un área multidisciplinar, pues combina mecánica, informática, ingeniería electrónica e inteligencia artificial. Los sistemas usados en la robótica han pasado de ser una “caja negra” llena de compuertas y transistores desde la cual se controlaba todo, a lo que conocemos hoy: sistemas embebidos que usan pequeños mini computadores con capacidad suficiente para procesar y controlar todos los requerimientos de un robot.

Los sistemas embebidos han ganado espacios en diversas tecnologías, desde dispositivos móviles, televisores y automotores hasta robótica; la tendencia es incluirlos de forma masiva en dispositivos y sistemas de uso general y particular, aumentando de forma continua su potencia, mientras que el precio disminuye. Estos sistemas cumplen unas restricciones muy estrictas:

- De tiempo: a menudo deben reaccionar en un periodo de tiempo muy corto.
- Cada vez deben ser más ligeros, portables y baratos.
- El consumo de energía debe ser muy reducido.

En los últimos años su evolución ha sido todavía mayor, gracias a la creciente demanda de dispositivos móviles.

Las características más importantes de los sistemas embebidos son las siguientes:

- Tamaño reducido: 10 cm x 5 cm x 1 cm.
- Peso reducido: menos de 50 gramos.
- Requerimientos de energía reducidos: alimentación 5V, Consumo inferior a los 1 A (menos de 5W).
- Alta conectividad: conectividad por USB, Ethernet, Wifi, CAN, I2C, GPIO, etc.

Por todo ello, ofrecen una clara ventaja en el desarrollo de la robótica móvil, incluyendo robots terrestres, aéreos o acuáticos, donde los requisitos de tamaño y peso se ajustan a las necesidades y requerimientos del diseño.



Componentes del sistema

En esta sección se describen los principales componentes utilizados para armar la estructura del robot:

- **Arduino Mega:** la tarjeta de desarrollo Arduino Mega es una solución de control genérico, de bajo costo y de fácil implementación para manejar motores, sensores, pantallas, etc; gracias a la amplia gama de librerías desarrolladas [1].
- **Orange Pi:** la Orange Pi+ 2E es una mini computadora de tamaño reducido, posee conectividad HDMI, Ethernet, Wifi, 4 puertos USB, procesador Quad core de 1.5 Gigas. y es compatible con sistema operativo Linux [2].

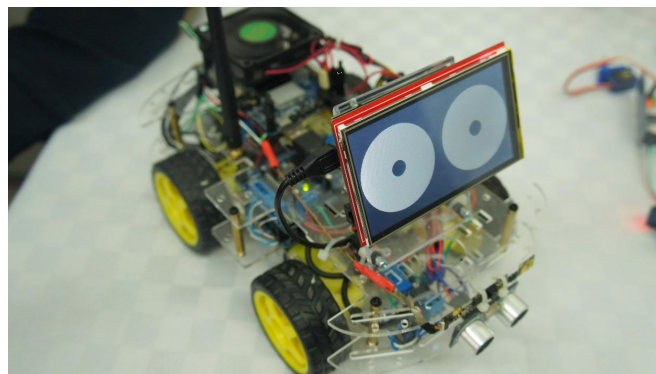


Figura 1.
*Parte frontal del
robot*

Estructura del sistema

Para la construcción del robot se utilizó el kit 4WD Smart Robot Car Chasis fabricado por Shenzhen CAIZHIXING Electronic Co; compuesto por una estructura acrílica y 4 motores con ruedas. En la Figura 2 se observa la estructura.

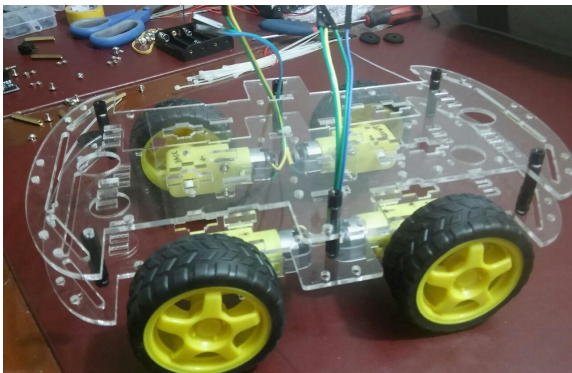


Figura 2.
*Estructura o chasis
del Robot*

El control de giro de cada motor se realiza mediante el Driver L293D, uno por cada dos motores. Para la comunicación con el usuario, se usa una pantalla LCD TFT Touch de 3.2 Pulgadas, un módulo cargador de baterías y sensores ultrasónicos para la detección de objetos. Todo conectado a la tarjeta Arduino Mega que finalmente se comunica vía GPIO al mini computador Orange Pi (ver Figura 3).

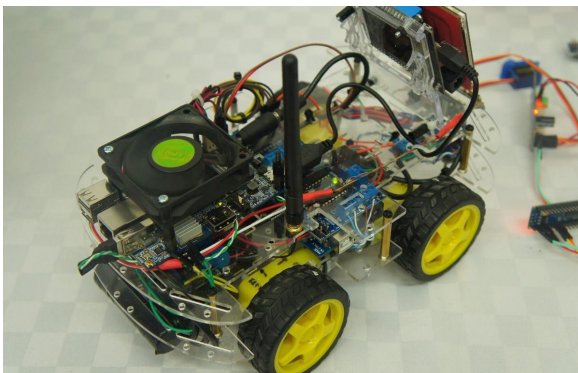


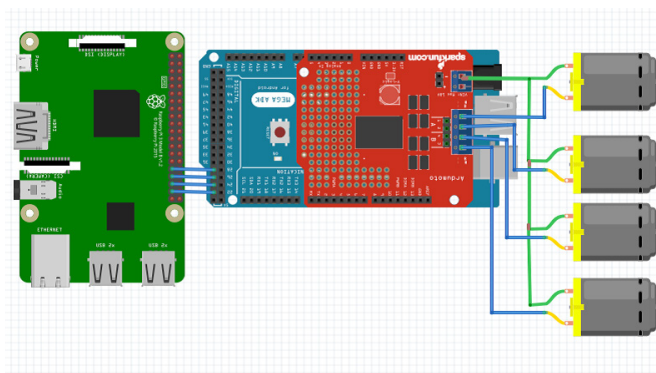
Figura 3.
*Parte posterior del
robot*

Para que el robot cumpla sus objetivos de movilidad, debe ser capaz de realizar las siguientes acciones: a) Desplazarse adelante y atrás; b) girar a la izquierda y a la derecha; c) no chocar con objetos. Estos movimientos se realizan a través del control de las 4 ruedas que posee el robot (Figura 4); en la cual se aprecian las conexiones lógicas entre la tarjeta Arduino y los motores. De esta forma, mediante comandos y posiciones lógicas, se logra el movimiento o desplazamiento deseado para el robot.

Para evitar los choques, se implementaron dos sensores ultrasónicos HC-SR04, ubicados en la parte delantera y trasera. Estos siempre censan la distancia de los objetos y cuando esta es inferior a 10 cm, el carro frena automáticamente, evitando así que choque; si el usuario insiste en ir al frente sin tener espacio suficiente, simplemente el carro no avanza, de igual forma sucede con la reversa.

Por último, para transmitir video en tiempo real, se emplea una cámara USB convencional compatible con Linux; usando la aplicación MOTION, las imágenes captadas por este dispositivo se convierten y se transmiten mediante el puerto 8888. Esta App está configurada de tal manera que permite ver el video en tiempo real, facilitando al usuario el control del robot sin problemas o latencias a la hora de observar el entorno donde está ubicado (ver Figura 5).

Figura 4.
Lógica de control de motores



Integración de los componentes

La Orange Pi procesa la información de la cámara y el Arduino Mega controla los actuadores. El usuario realiza el manejo del robot mediante el panel de control (ver en Figura 5 los botones en azul).

En el fondo de la aplicación se despliega en pantalla completa la imagen transmitida por la cámara, la actualización de la pantalla se realiza cada 100 msg. Este tiempo se definió con base en ensayos que no afectaron las órdenes enviadas por el usuario ni la visualización de la cámara.

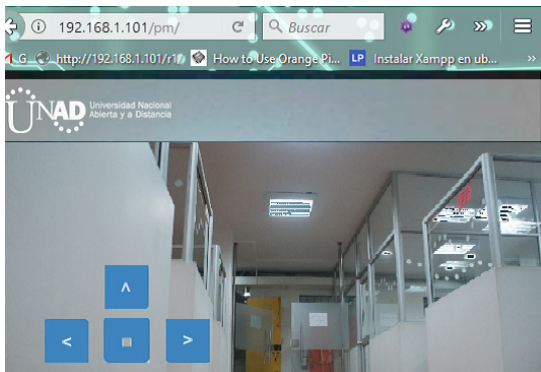


Figura 5.
Interfaz de usuario y control

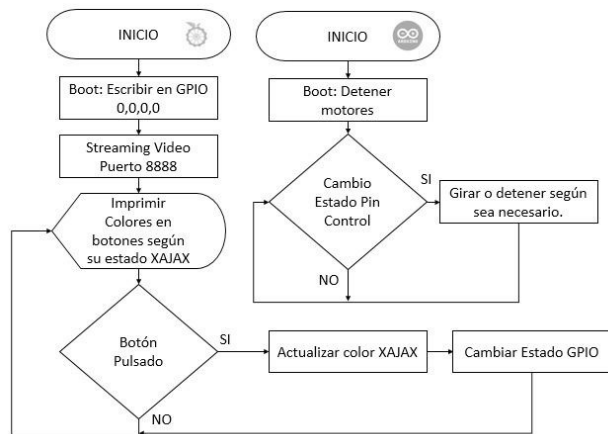
En el diagrama de flujo de la Figura 6, aparecen la Orange y el Arduino Mega trabajando al mismo tiempo. Aquí se aprecia el diagrama de flujo básico de cada placa. Básicamente, la Orange procesa video y ejecuta la aplicación de usuario en HTML5, y dependiendo del estado de cada botón, envía la orden al Arduino para que este se mueva según las disposiciones del usuario.

El Arduino se encuentra en modo esclavo y está siempre a la espera de ejecutar el comando enviado desde el mini ordenador. Este proceso es bidireccional y síncrono; y para la implementación de la comunicación, se



usó ROS (Robot Operating System) con la utilización de los servicios, que es un método de comunicación basado en el paradigma de request/reply con el concepto de maestro esclavo.

Figura 6.
Diagrama de flujo del robot



Experimentación

Los experimentos consistieron en la realización de pruebas de desempeño, en las que se observó el comportamiento global del robot con sus componentes integrados. La primera prueba consistió en la conducción del vehículo, para lo cual se intentó chocarlo contra objetos en el camino y se analizó su respuesta. La segunda prueba consistió en alejar el robot del Router inalámbrico y en analizar cómo afecta ello al control del mismo.

En la primera parte, el robot responde óptimamente, y luego de 10 intentos de choque se concluyó que la rutina de retroceso de 1 g es suficiente para frenarlo. En la segunda, la transmisión de imágenes se afectó seriamente cuando la distancia en línea de vista al Router supera los 50 metros. Para el caso, se usó un Router bac wifi 3G HUAWEI E5330 [6,7].

Consideraciones especiales

Es preciso ser cuidadosos en el uso de los motores, dado que el exceso de esfuerzo puede quemarlos con facilidad y afectar el sistema de baterías. La solución aplicada consistió en limitar el peso máximo que puede cargar la estructura a 300g. Por otro lado, se recomienda tener un canal con un ancho de banda de 5Mbps simétrico, y operar el robot en un radio de 35 m con línea de vista al Router. Esto con el fin de garantizar la transmisión de video y la operación.

Conclusiones

Gracias a la unión de estas dos tarjetas para desarrollo (Orange Pi+ 2E y Arduino Mega), se obtiene flexibilidad y adaptabilidad del sistema para adicionar sensores, actuadores y comunicaciones, sin perder la potencia de procesamiento de la Orange Pi. La Orange Pi de la visualización y control en tiempo real se realiza mediante interfaz WEB, apoyada con video en streaming de la cámara frontal. También se le otorga al Arduino la parte de control o carga como motores y sensores, facilitando la programación y escalabilidad de proyectos futuros.

El control del robot se realiza por interfaz WEB sin importar el sistema operativo del cliente. De este modo, puede controlarse desde IOS, Android, Windows o Linux y solo debe soportar HTML5; la pantalla o interfaz de usuario se adapta automáticamente a cada dispositivo. En la siguiente fase del proyecto se establecerá la seguridad de acceso a las páginas principales, creando una red con multitud de sensores NodeMCU o EPS, y se efectuará el censado de variables, condición del ambiente y demás factores físicos susceptibles a estudiar. La Figura 7 ilustra a grandes rasgos los dispositivos que conforman la siguiente fase [3-5].

Para finalizar, cabe mencionar que sus componentes son Open Source, Open Hardware y Open Code; lo que facilita la inclusión de este proyecto a nuevos sistemas y adaptabilidad o funciones.



Figura 6.
Próximo aplicativo del Robot



Referencias Bibliográficas

1. Arduino Inc, 2014. "Arduino Mega". Disponible en <https://store.arduino.cc/usa/arduino-mega-2560-rev3>
2. Orange Pi Inc, 2016. "Orange Pi+ 2E". Disponible en <http://www.orangepi.org/orangepiplus2/>
3. Gleason, K., "Tutorial Android WebRTC" [online], Enero 2016 Disponible en: <https://github.com/GleasonK/android-webrtc-tutorial/graphs/contributors>
4. Behnke, S., M. Schreiber, J. Stückler, R. Renner, H. Strasdat, 2006. See, walk, and kick: "humanoid robots start to play soccer", 6th IEEE-RAS International Conference on Humanoid Robots, 7 pp.
5. Behnke, S., M. Schreiber, J. Stückler, R. Renner, H. Strasdat, 2006. See, walk, and kick: humanoid robots start to play soccer, 6th IEEE-RAS International Conference on Humanoid Robots, 7 pp.
6. Huawei Inc, 2015. "Huawei E5330, modem-Router MiFi 3.5G WiFi". Disponible en <http://www.blauden.com/huawei-e5330-modemrouter-mifi-3-5g-wifi-portatil-libre>
7. E. Orozco Acosta, "Simulación en tiempo discreto de un proceso de abastecimiento de combustible como una herramienta de toma de decisiones: Caso estación de servicios en Barranquilla", Dictamen Libre, no. 1011, pp. 7-13, 2012.

Bibliografía de consulta

Honda Motor Co., "Honda unveils all-new ASIMO with significant advancements", [online], Junio 2014. Disponible en <http://world.honda.com/news/2011/c111108All-new-ASIMO/index.html>.

Itzees, Open CV About, Junio de 2014,[online], Disponible en <http://opencv.org/about.html>.

Open Source Robotics Foundation, ".About ROS", Febrero 2014, [online], Disponible en <http://www.ros.org/about-ros/>.

Valkov, E., 2013. Raspberry Pi camera with OpenCV. Disponible en <http://robidouille.wordpress.com/2013/10/19/raspberry-picamera-with-opencv/>.



Este artículo se cita

J. Leiva., y J. Valero, “Diseño y desarrollo de un prototipo robótico con orange PI”, Investigación e Innovación en Ingenierías, vol. 5, n°. 2, pp. 174-189., 2017