

**UNIVERSIDAD DE COSTA RICA
FACULTAD DE INGENIERÍA
ESCUELA DE CIENCIAS DE LA
COMPUTACIÓN E INFORMÁTICA**

CI2657- ROBÓTICA

Prof. Bach. Kryscia Daviana Ramírez Benavides

Tarea 3: Software y Hardware Q.bo

Elaborado por:

Julio Calderon B11226
julioalderonc@gmail.com

Gabriel Quesada B25275
gab.quesada56@gmail.com

Juan Carlos Solano B26441
yuk2794@gmail.com

Leonardo Villalobos B27219
artist_artistian@hotmail.com

5 de septiembre, 2016

Tabla de contenidos

Tabla de contenidos

1. Tema

2. Objetivos

3. Enunciado

4. Desarrollo

4.1 Software

4.1.1 Open Q.bo

4.1.2 Q.bo Apps

4.2 Hardware

4.2.1 Especificaciones de Hardware

4.2.2 Componentes del Q.bo

4.2.3 Q.boards

4.2.4 Control de Hardware

5. Referencias

1. Tema

Los Q.bot son robots enfocados en la interacción con el ambiente, conformados de un computador con varios sensores y actuadores los que permiten al robot reuperar información de su entorno y responder a este. El tema de este documento es el de entrar un poco más en detalle acerca del funcionamiento del robot tanto de un punto de vista físico, de la recolección de datos por medio del hardware que lo compone, así como de software y las acciones que se realizan con los datos de entrada.

2. Objetivos

- Familiarizarse y realizar diferentes actividades con el robot Q.bo.
- Describir las funciones de los componentes físicos del Q.bo
- Generar una visión más amplia del funcionamiento interno de los robot Q.bo
- Documentar los procesos internos de software así como los procesos físicos por los cuales el Q.bo realiza las acciones pedidas.

3. Enunciado

A cada grupo se le asignará un robot Q.bo. Luego cada grupo debe buscar información sobre el tema asignado y realizar lo siguiente:

Temas:

- a. OpenQbo (software).
- b. Q.bo (hardware).

Deben generar un archivo donde expliquen y resuman el tema. Además, deben programar el robot con una función asociada al tema. Luego deben realizar una presentación a la clase.

Algunos URL de interés:

- <http://thecorpora.com/blog/>
- <http://openqbo.org/wiki/doku.php>

4. Desarrollo

4.1 Software

4.1.1 Open Q.bo

Open Q.bo es una distribución de Linux basada en Ubuntu 11.10. Esta remueve programas nativos de Ubuntu y agrega características como la plataforma ROS, el programa de reconocimiento de voz Julius y el sistema de habla Festival.

En sí, la distribución no ofrece mucha diferencia de Ubuntu convencional. Quizá lo más característico de la distro sean los utilitarios para interactuar con el robot NAO. La distribución se puede descargar del repositorio de OpenQ.bo, y se encuentra actualmente en la versión beta 0.4.1. La principal utilidad de esta distribución viene de los paquetes: ROS, Julius y Festival.

ROS

Ros, o Robotic Operating System, es una plataforma para la escritura de software para robots e incluye una serie de bibliotecas, herramientas y convenciones. Su objetivo es simplificar la tarea construir un robot funcional complejo. Tiene un enfoque de desarrollo colaborativo, ya que espera que los distintos investigadores que utilizan ROS también contribuyan con sus hallazgos a la plataforma.

Ros cuenta con una serie de características robustas. La primera es una infraestructura de comunicación, ya sea mediante el paso de mensajes o llamados a función remotos (RPC). Esto permite comunicar todos los procesos del robot, y se le conoce como middleware. Otra funcionalidad son las funciones específicas de robot: una serie de bibliotecas útiles para cualquier robot. Estas incluyen, pero no se limitan a: estandarización de paso de mensajes, geometría robótica, descripción del robot, diagnóstico y navegación. ROS también posee una serie de herramientas útiles para el desarrollo de un robot. Entre estas se encuentran las herramientas de línea de comandos, un visualizador, y un desarrollador de interfaces.

Adicional a esto, ROS cuenta con una wiki para el aprendizaje de la plataforma y la solución de problemas. Se ofrecen diversos tutoriales para que el usuario nuevo en ROS pueda entender como funciona la plataforma; no obstante, se recomienda tener conocimientos previos en el uso de un sistema operativo basado en Unix. También tiene un sistema de preguntas y respuestas en caso de no encontrar soluciones en la wiki. Finalmente, ROS da la posibilidad de integrar con otras bibliotecas de robótica, como MoveIt o ROS-Industrial.

Julius

Julius es un programa de código abierto de reconocimiento de voz de alto rendimiento. Además de una serie de características sólidas, es ligero en memoria (32MB), es altamente configurable, y es independiente del lenguaje. El programa requiere un modelo de lenguaje entrenado para decodificar audio en ese lenguaje. Mientras que el proyecto original ofrece Inglés y Japonés, es posible crear un modelo de lenguaje a partir de una gramática y entrenamiento. Actualmente, la documentación de Julius se encuentra disponible únicamente en japonés.

Festival

Festival es un programa libre que funciona como conversor de texto a habla (TTS). Tiene la posibilidad de procesar texto en inglés y español. Festival puede ser invocado como API de un programa, desde la terminal, o como librerías de C++, Java y Emacs. El programa se encuentra bien documentado y lo mantiene la Universidad de Edimburgo.

Q.bo Apps es un repositorio de aplicaciones para ser utilizadas específicamente para el robot Q.bo. Estas están diseñadas principalmente en ROS, pero existen otras para Julius, Android y demás plataformas. Este es un repositorio abierto: cualquiera puede compartir sus aplicaciones desarrolladas para Q.bo. No todas las aplicaciones en esta página están documentadas, ya que son mantenidas por el usuario que la subió, por lo que puede resultar complicado su uso.

Existen numerosas aplicaciones en este repositorio que mejoran funciones existentes, agregan funcionalidad nueva y administran los recursos del robot Q.bo. Entre ellas se encuentran programas de reconocimiento facial y de objetos, un recopilador de información del sistema y un lector de gestos de mano, entre muchos otros.

Los Q.bo miden 45.6 cm de alto, 31.4 cm de ancho, 29.2 cm de fondo y pesan entre 9 y 11 kg aproximadamente. En la cabeza tienen 3 micrófonos que actúan como orejas, 2 cámaras y dos párpados mecánicos que actúan como ojos, 2 servos para movimiento de cabeza en 4 direcciones y 2 servos para cada movimiento de los ojos, 20 leds para la boca, 1 led para la nariz, 1 antena Wi-Fi para conexión y una Q.board2 como tarjeta controladora central de la cabeza.

En el cuerpo tienen 2 sensores ultrasónicos junto con dos opciones, 1 sensor infrarrojo para detectar el piso, 2 motores para movimiento de cuerpo capaces de llegar a 170 revoluciones por minuto, 2 llantas laterales y una llanta central frontal esférica, 2 parlantes para sonido y un controlador Mini-ITX Intel que puede ser un Atom en la versión Q.bo Lite o un Intel Core i3 en la versión Q.bo Pro.

Como componentes principales están los componentes plásticos, los componentes mecánicos, los componentes misceláneos, las tarjetas de control, los sensores, los componentes de la PC, los actuadores, los componentes de sonido y los componentes de energía.

- **Componentes plásticos:** piezas plásticas, los pines del motor, el adaptador XTION Pro y los tornillos.
- **Componentes mecánicos:** el chasis metálico, el adaptador para discos duros, el adaptador para montar la tarjeta madre, los engranajes, la rueda central, los tornillos, y las piezas mecánicas para el movimiento de los ojos.
- **Componentes misceláneos:** las instrucciones de montaje, la antena Wi-Fi y su cable, la pantalla LCD trasera y los cables generales.
- **Sensores:** las cámaras web, los buscadores de rango ultrasónico, el sensor de proximidad y el kit de sensores Asus (opcional).
- **Componentes de PC:** tarjeta madre, heatsink, procesador Atom, 2GB de memoria RAM, disco duro de 40 GB Intel y adaptador Wi-Fi 802.11n Intel.
- **Actuadores:** Servo GWS S125, Servo Dynamixel y motores EMG30.
- **Componentes de sonido:** parlante miniatura Monacor y micrófono Fonestar.

Los Q.bo tienen la capacidad de interactuar con sus alrededores, para lograr esto tienen tanto sensores como actuadores pero para ello el Q.bo necesita tarjetas llamadas Q.boards para poder controlar sus componentes.

El Q.bo contiene 5 tarjetas (Q.board1, Q.board2, Q.board3, Q.board4, Q.board5) para distintas funciones, que se presentarán a continuación:

Q.board 1

Es la tarjeta principal del Q.bo la cual tiene por funciones, controlar los dos motores de las ruedas, el amplificador de audio y el IC bus, esto por medio del microcontrolador ATmega1280 el cual es programable por medio del IDE de arduino.

El mismo microcontrolador AT mega 1280 es utilizado para controlar los motores de las ruedas, los cuales son dos motores DC (corriente directa) de 12 voltios a 2 amperios, y se recomienda el uso de los motores EMG30.

Por último la Q.board 1 se encarga de controlar el IC bus, el cual es utilizado para poder añadir periféricos externos al Q.bo, como sensores. Por ejemplo el sensor SRF10 el cual evita que el Q.bo choque contra paredes.

Ya que el microprocesador ATmega1280 está conectado a un convertidor Serial_to_USB es fácilmente conectable a un computador y ser programado por medio del IDE de arduino.

Q.board 2

La segunda tarjeta esta posicionada dentro de la cabeza del Q.bo y es compatible con la tarjeta Arduino Duemilanove. Así como contiene registros integrados con los que controla los led de la boca. De la tarjeta se conectan 3 microfones integrados y estos van redirigidos a la entrada de microfono de la PC del Q.bo, de esta manera la boca del Q.bo cambiará simultáneamente a cuando se de salida a un audio desde el PC del Q.bo.

Por ultimo esta tarjeta tambien se encarga de controlar tanto los servos del cuello del Q.bo como sus ojos.

Q.board 3

La Q.board 3 tiene por función controlar la carga de todo el sistema, para lograr el uso más eficiente de la energía. administrar, permitir las capacidades de carga de batería del Q.bo, el encendido del dispositivo y provee salidas para los leds del sistema, esto por medio de un microcontrolador basado en stm32f103 32bit Cortex-M3, el cual permite al Q.bo, si hay una fuente externa de poder conectada al mismo, redirigir el uso de energía a la fuente externa y simultáneamente cargar la batería del sistema, si no hay conector externo simplemente se utiliza la carga de la batería.

Esta tarjeta puede sacar un maximo de 150w y también permite el uso de un conector I2C lo cual permite a otros microcontroladores leer el nivel de batería y el estado de la energía. Es importante recalcar que aunque la tarjeta se reprogramable por medio de puerto serial no es recomendado por la naturaleza de la tarjeta.

Q.board 4

Permite el uso de tanto acelerómetro como giroscopio por medio de la interfaz IC2.

Q.board 5

Esta última tarjeta es la tarjeta de la matriz de leds que compone la boca del Q.bo ai como su nariz, esto por medio de 20 leds para la boca y un led tricolor que representa la nariz.

Qbo_arduqbo

Es un paquete de tipo ROS Package, creado por Miguel Angel Julian Aguilar, el cual sirve de driver para las placas controladoras del Qbo. Se necesita para controlar el movimiento y los sensores del Qbo siendo así indispensable para el uso del robot.

Este paquete es modular, esto significa que los sistemas que van a ser activados pueden ser configurados con algunos parámetros ROS. El nodo contiene los siguientes controladores:

- **Base Controller:** Este controlador activa el movimiento de la base del robot y el sistema de posicionamiento.
- **Battery Controller:** El controlador de batería activa el envío del nivel de batería. El (campo de nivel) nos da el nivel de batería en voltios, y el (campo de estado) nos da la información acerca del estatus del robot. El número de estado debe ser convertido en un número binario de 6 bits y cada bit tiene su significado.
- **IMU Controller:** Este controlador activa el envío de lecturas del sensor IMU. El sensor IMU recolecta la velocidad angular y lineal de los datos de la aceleración. Este está conectado a la tarjeta Q.board4 y tiene que ser conectado a la Q.board1.
- **Joint Controller:** El joint controller activa el movimiento de los servos del robot. Esto significa el cuello y los párpados del Qbo. Si una nueva orden es mandada antes de que la orden actual se termine, la orden actual se cancela dejando campo a la nueva lo cual nos permite hacer control reactivo. La posición meta es un número en radioan entre el valor máximo y el mínimo de este servomotor.

- **LCD Controller:** El control de LCD activa el envío de mensajes desde la PC hasta la pantalla LCD del Qbo.
- **Mics Controller:** Regula el nivel de ruido de los tres micrófonos del Qbo.
- **Mouth Controller:** La boca del Qbo es una matriz de 4 líneas con 5 LEDs cada una. Estos pueden ser controlados uno por uno. La boca es activada automáticamente con los speakers del Qbo.
- **Nose Controller:** La nariz del Qbo es un LED RGB. Este permite elegir el color de la luz LED.
- **Distance Sensors Controller:** El controlador de sensores de distancia maneja todos los sensores de distancia, infrarrojo y ultrasonido. Qbo tiene un sensor de distancia vertical infrarrojo en frente de él para chequear la distancia con el piso y así evitar saltar de la mesa por ejemplo, y dos sensores ultrasónicos horizontales, uno por cada rueda, para chequear las paredes.

Mensajes ROS

- **BatteryLevel:** Este está definido en el BatteryLevel.msg en la carpeta msg. Indica el nivel de la batería en voltios y el estado de la PC.
- **LCD:** Está definido en el archivo LCD.msg en el folder msg. Este provee la posibilidad de poner diferentes mensajes en la pantalla LCD del Qbo.
- **NoiseLevels:** Está definido en el archivo NoiseLevels.msg en el folder msg. Indica los niveles de ruido detectados en la salida de los micrófonos del Qboard2
- **Nose:** Está definido en el archivo Nose.msg en el folder msg. Provee la capacidad de cambiar el color de la nariz del Qbo. Solo se aceptan 3 diferentes colores.
- **Mouth:** Está definido en el archivo Mouth.msg en el folder msg. Provee la capacidad de cambiar la forma de la boca del Qbo.
- **Motor_state:** Está definido en el archivo motor_state.msg en el folder msg y contiene información útil sobre los servomotores.

5. Referencias

Wiki de OpenQ.bo http://openqbo.org/wiki/doku.php?id=software:openqbo_distro

Sitio web de ROS <http://www.ros.org/>

Sitio web de Julius <http://julius.osdn.jp/en/>

Sitio web de Festival <http://www.cstr.ed.ac.uk/projects/festival/>

Q.bo Apps en la wiki de Open Q.bo http://openqbo.org/wiki/doku.php?id=qbo_apps:start

Q.boards en la wiki de Open Q.bo <http://openqbo.org/wiki/doku.php?id=hardware:qboards>

Componentes Q.bo: <http://thecorpora.com/index.php/ecomponents>

Hardware

Control

Q.bo:

http://openqbo.org/wiki/doku.php?id=qbo_apps:ros_pack:qbo_arduqbo