

10053 MEJORANDO LAS POSIBILIDADES DE APRENDER A PROGRAMAR. AMPLIACIÓN DEL ROBOT EDUCATIVO MULTIPLO N6 MAX A FRANKESTITO

Rafael Zurita⁽¹⁾⁽²⁾, Juan de la Fuente⁽¹⁾⁽³⁾, Martín Bucarey⁽¹⁾⁽⁴⁾, Daiana Bonet⁽¹⁾⁽⁵⁾, Rodolfo Del Castillo⁽¹⁾⁽⁶⁾,
Guillermo Grosso⁽¹⁾⁽⁷⁾, Laura A. Cecchi⁽¹⁾⁽⁸⁾, Jorge Rodríguez⁽¹⁾⁽⁹⁾

⁽¹⁾ Grupo de Investigación en Lenguajes e Inteligencia
Artificial Facultad de Informática
Universidad Nacional del Comahue

⁽²⁾ rafa@fi.uncoma.edu.ar

⁽³⁾ juan.delafuente@fi.uncoma.edu.ar

⁽⁴⁾ martin.bucarey@fi.uncoma.edu.ar

⁽⁵⁾ daiana.bonet@fi.uncoma.edu.ar

⁽⁶⁾ rdc@fi.uncoma.edu.ar

⁽⁷⁾ guillermo.grosso@fi.uncoma.edu.ar

⁽⁸⁾ lcecchi@fi.uncoma.edu.ar

⁽⁹⁾ j.rodrig@fi.uncoma.edu.ar

Resumen: Se considera que la formación en conceptos fundamentales de Ciencias de la Computación es prioritaria para ampliar las posibilidades de comprender e intervenir mejor el mundo moderno.

La robótica educativa es un dispositivo metodológico ampliamente difundido como ambiente para la enseñanza de la disciplina a estudiantes sin formación previa en la temática.

Frankestito, es el robot educativo desarrollado por la Facultad de Informática con capacidad de visión y comunicación vía wireless.

La construcción de aprendizajes sobre programación implica, entre otras actividades, observar de comportamiento de las estructuras de programación durante la ejecución del programa.

El proceso de revisión del código como momento ligado a mejorar las posibilidades de aprender requiere implementar la interacción con los robots educativos durante la ejecución de programas.

Por otro lado, integrar funcionalidades de visión a los robots educativos amplía las posibilidades de abordar la enseñanza de aspectos más complejos.

En este trabajo se presentan las modificaciones realizadas al robot educativo Multiplo N6 Max con intención de dotarlo con las capacidades de visión y comunicación vía wireless presentes en Frankestito y un grupo de experiencias realizadas con el robot en actividades de Extensión Universitaria en las que participa la Facultad de Informática.

1. Introducción

La computación es uno de los campos disciplinares con mayor incidencia en el desarrollo de los sistemas productivos y la vida político social. Se considera a la formación en este campo como prioritaria para el desarrollo, ya que su enseñanza aporta conocimientos y herramientas que amplían las posibilidades de comprender e intervenir mejor el mundo moderno [11, 2, 10].

En este contexto crece el consenso acerca de integrar conceptos de las Ciencias de la Computación, en las propuestas de enseñanza en todos los niveles del sistema educativo, durante la escolaridad obligatoria, a fin de contribuir a mejorar las perspectivas profesionales y humanas de los estudiantes [1, 15].

La robótica educativa es un dispositivo metodológico ampliamente difundido y seleccionado con frecuencia como ambiente para enseñar Ciencias de la Computación a estudiantes sin formación previa en el campo disciplinar.

Desde esta perspectiva se sostiene que las experiencias de aprendizaje son más efectivas cuando los estudiantes manipulan activamente objetos tangibles utilizando lenguajes de programación para controlar estos dispositivos.

Los robots exponen y hacen tangibles algunos aspectos de la computación que frecuentemente se presentan como difíciles de percibir y comprender [14].

En esta dirección varias iniciativas internacionales se vienen desarrollando, como Scribbler [3], Lego MINDSTORM [5] y Finch Robot[9]. Sus esfuerzos generalmente se orientan al desarrollo de plataformas de robots educativos, equipados con luces, parlantes y con sensores IR y/o de sonido, y su locomoción está basada en ruedas. En cuanto a la forma de comunicación, la mayoría es cableada, algunos inalámbrica y muy pocos proveen WiFi.

En el ámbito nacional, RobotGroup ha desarrollado el sistema Multiplo N6 Max, una plataforma de robótica educativa con arquitectura de hardware y software Open Source. El sistema cuenta con dos entornos de programación, el robot percibe el medio a partir de un conjunto de sensores y la comunicación con el dispositivo en el que se desarrollan los programas es vía USB [8].

Frankestito, es el robot educativo Myro compatible de bajo costo desarrollado por la Facultad de Informática con capacidad de visión y comunicación vía wireless. Tanto el software como el hardware son publicados bajo licencias libres [12].

La enseñanza de la programación requiere analizar el flujo del programa y el comportamiento de las estructuras de los datos. En función de esto, los estudiantes identifican fortalezas y debilidades en sus construcciones previas, construyen nuevas hipótesis y diseñan e implementan nuevas aproximaciones a la solución del problema.

Una necesidad que se presenta al diseñar estrategias de enseñanza que se ajusten a la dinámica del aprendizaje por reconstrucción [13], es explotar la comunicación vía wireless, utilizando al robot como dispositivo de entrada/salida durante la ejecución, para mejorar la comprensión de la lógica del algoritmo y las características de las estructuras de programación.

A diferencia de las plataformas mencionadas anteriormente, Frankestito permite interactuar en tiempo de ejecución con el dispositivo del usuario y con otros dispositivos remotos (inclusive otros robots), recibiendo y enviando información.

En este trabajo se presentan las modificaciones realizadas al robot educativo Multiplo N6 Max con intención de dotarlo con las capacidades de visión y comunicación vía wireless presentes en Frankestito. Estas mejoras buscan contribuir a la producción de conocimiento científico en los campos del software embebido y de las plataformas de robótica educativa.

Por otro lado, se trabaja en la construcción de recursos teóricos y materiales que amplíen las posibilidades de concreción de experiencias educativas en el contexto de la enseñanza de las Ciencias de la Computación.

La estrategia discursiva para la presentación de este trabajo es la siguiente. En la sección 2 se describe la motivación de esta adaptación basada en la necesidad de interactividad y visión en los robots. En la sección 3, se describen las principales características de los sistemas Multiplo N6 Max y Frankestito y se presenta una comparación de las plataformas. La siguiente sección detalla las adecuaciones aplicadas para lograr la ampliación del robot y se explican las pruebas y experimentos llevados a cabo. En la sección 5 se presentan algunas experiencias del uso del robot adaptado. Finalmente, en la sección 6, presentamos las conclusiones y los trabajos futuros.

2. Ambiente de trabajo. Necesidad de interactividad y visión

Generalmente la construcción de aprendizajes sobre programación conlleva el desarrollo de productos de software que dan solución a problemas planteados. Avanzar sobre estos procesos implica la construcción progresiva de conceptos y prácticas sobre algoritmos y programación.

Los estudiantes buscan analizar y comprender el problema planteado, diseñar algoritmos que intentan resolverlos satisfactoriamente y construir programas que implementan las soluciones proyectadas.

Las soluciones parciales expresan el dominio sobre conocimiento a aprender que han logrado concretar en un momento específico. La evaluación acerca del grado en que el programa constituye una solución al problema planteado se ubica como dispositivo que explicita debilidades y confirma aprendizajes.

La resolución del conflicto establecido requiere observar el comportamiento de las estructuras de programación durante la ejecución del programa.

Esto habitualmente se realiza con herramientas de depuración integradas a los entornos de desarrollo, que permiten la ejecución paso a paso y la observación de los valores asignados a las variables.

El proceso de revisión del código como momento ligado a mejorar las posibilidades de aprender requiere implementar la interacción con los robots educativos durante la ejecución de programas.

Por otro lado, integrar funcionalidades de visión a los robots educativos amplía las condiciones de ubicarlos en el contexto de los entornos de programación descritos por Papert [13] como de “piso bajo”, fácil de iniciar, y “techo alto”, por la posibilidad de construir proyectos complejos.

En este sentido, ofrecer funcionalidades relacionadas a la visión propicia el trabajo sobre un rango más amplio de problemas como la detección, búsqueda y seguimiento de objetos, identificación de colores y mejora la interacción con el ambiente.

Estas características están presentes en Frankestito. Asimismo, durante la ejecución de programas el robot puede transmitir video en tiempo real vía HTTP e interactuar con dispositivos remotos.

Por esta razón, resulta necesario implementar modificaciones sobre la arquitectura y el software embebido del robot educativo Multiplo N6 Max, con intención de posibilitar este tipo de interacción y ofrecer funcionalidades de visión.

3. Descripción de las Plataformas

A continuación se describen las características principales de las arquitecturas de las dos plataformas y se realiza un análisis comparativo.

3.1. Sistema Multiplo N6 Max

El sistema Multiplo N6 Max [6] es una plataforma de robótica diseñada con fines educativos. El robot posee tracción a partir de dos ruedas motorizadas en forma independiente, y una tercera rueda libre. Esto posibilita movimientos hacia adelante, atrás y de giro a izquierda o derecha.

Como dispositivos de interacción con el ambiente integra un conjunto de sensores que le permiten detectar la proximidad de objetos, cambio de contraste entre superficies claras u oscuras y luminosidad. Cuenta además con un grupo LEDs que en conjunto con los pulsadores reset y run conforman la interfaz de usuario. La plataforma ofrece dos alternativas para programar el robot, DuinoPack y MiniBloq. DuinoPack C/C++ es un IDE Arduino modificado integrando características especialmente diseñadas para la programación de robots Multiplo N6 Max. La segunda alternativa, MiniBloq, es un entorno gráfico de programación por bloques para dispositivos Arduino, su principal objetivo es facilitar los procesos de enseñanza y aprendizaje de la programación en el contexto de la robótica educativa.

Los programas implementados en los ambientes DuinoPack o MiniBloq se compilan y luego se transfieren al robot a través de un puerto USB.

El principal componente de la arquitectura del robot Multiplo N6 Max es una placa DuinoBot 2.3 basada en el microprocesador Arduino compatible ATmega1284P que cuenta con un reloj 20 Mhz, PWN serial SPI, 128 Kb de memoria flash y 16kb de RAM. A esta placa se conectan 2 motorreductores planetarios 200 RPM, un juego de 3 sensores (ultrasonido, infrarojo y de control remoto), un portapilas AAA de tres elementos y un conector USB.

En la Fig. 1 se puede observar el esquema básico de la arquitectura del robot.

Tanto la arquitectura, el software, como el firmware del robot educativo Multiplo N6 Max se desarrollan en el contexto del Open Source y son compartidos bajo la licencia RobotGroup–Multiplo Pacifist License (RMPL). Ésta es básicamente una versión de licencia MIT ligeramente modificada, el principal ajuste consiste en la anexión de disposiciones que buscan impedir el uso con fines bélicos o relacionados con la pena de muerte.



Figura 1: Esquema de la arquitectura del sistema Multiplo N6 Max.

3.2. Sistema Frankestito

El software y hardware del robot Frankestito se distribuyen bajo licencias Open Source y Open Hardware respectivamente, lo que permite la distribución y modificación del diseño y del código fuente, con el solo requerimiento de que los productos derivados deben estar bajo una licencia compatible. Los fuentes pueden ser descargados de [4] y [7].

En la Fig. 2 se puede observar el esquema básico de la arquitectura del robot.

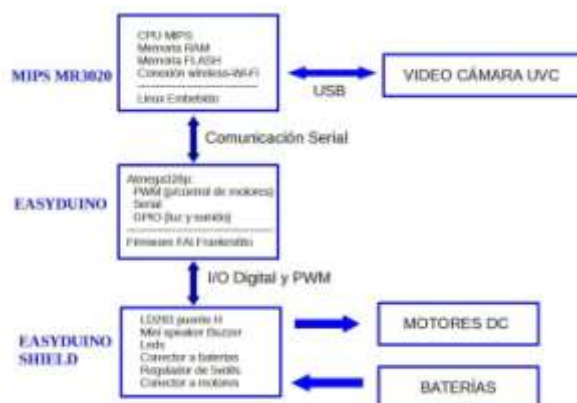


Figura 2: Esquema de la arquitectura del sistema Frankestito.

El principal componente del sistema es una placa controladora MIPS MR3020[7], de tamaño relativamente pequeño (PCB de 5,7cm 5,7cm), que consta de una CPU de 32-bits, una Ram de 32Mb, una memoria flash de 4 Mb, puertos de comunicación serial, USB Host y conexión inalámbrica mediante un canal en 2,4 Ghz. El propósito de esta

placa es actuar como interfaz entre el usuario (PC, teléfonos móviles) y el robot, ya que almacena el sistema operativo GNU Linux OpenWrt, provee la comunicación inalámbrica Wi-Fi y la comunicación con la placa **EasyDuino** [4] y controla la video cámara UVC tipo webcam.

La placa EasyDuino, es un clon Copyleft Hardware de **Arduino UNO** que cuenta con las mismas prestaciones que el original, a diferencia de la programación que se realiza mediante una conexión serie, ya que es la que utiliza MIPS MR3020 para la comunicación. Las principales características de la placa EasyDuino son que posee un microcontrolador AVR AT- MEGA328, 14 I/O digitales, 6 compuertas de entrada análogas con 10-bits, un reloj 20 Mhz, PWN serial SPI, 32 Kb de memoria flash y 2kb de RAM y un compilador GNU GCC.

Para la etapa de potencia, esto es el control de los motores DC, se ha diseñado y desarrollado un Shield para el EasyDuino que posee un chip **LD293** (puente H). En el Shield además, se instalaron tres LEDs y un mini speaker Buzzer que permiten realizar distintas señalizaciones de formas diferentes. La misma placa contienen un conector al pack de baterías que alimenta todo el sistema, un regulador usb de 5v y los conectores dos motores separados.

En cuanto a la arquitectura del software, En cuanto a la arquitectura del software, como se observa en la Fig. 3, está compuesta por cinco capas claramente definidas, distribuidas en los dos componentes necesarios para la utilización de la plataforma robótica educativa. Los componentes en recuadro Firmware AT-MEGA328 fueron desarrollados específicamente para Frankestito. Los otros componentes son piezas de software open source que fueron instaladas y en algunos casos, modificadas para lograr los objetivos de la plataforma robótica.

En el dispositivo remoto del usuario, es decir la PC, portátil o dispositivo móvil, se observan los siguientes niveles de software:

- Programas de usuario.
- Biblioteca de Myro y de Visión.
- Sistema Operativo con capacidad de comunicación TCP/IP.

En el robot se distinguen los siguientes niveles de software:

- Sistema Operativo Linux embebido, con drivers TCP/IP, wireless, USB, UVC y serial.
- Firmware para el ATMEGA328 de la placa EasyDuino que contiene los drivers serial, de motores, de los LEDs y del parlante. Asimismo, se implementó en este firmware, la capa de software que interpreta el protocolo de Myro, para procesar los paquetes provenientes de la biblioteca de Myro del computador remoto.

Frankestito tiene la capacidad de comunicarse de modo inalámbrico (Wi-Fi), a través de conexiones TCP/IP, lo que posibilita un mayor ancho de banda y mayor alcance. Estas características son necesarias para posibilitar la transmisión del streaming de visión y de los paquetes del protocolo Myro.

La aplicación de video stream del robot (mjpg-streamer) envía un streaming de video de cuatro FPS (frames per second - cuadros por segundo), con una resolución de 160x120 pixels para cada imagen. La biblioteca de visión, recibe estas imágenes, y detecta objetos por color generando una estadística de los elementos identificados. Luego, el programa de usuario puede llamar a funciones de consulta para identificar los tamaños de los objetos encontrados y su posición frente al robot. De esta manera, y

con precisión en pixels, un programa de usuario puede reconocer a “cuántos” pixels a la izquierda o derecha del centro del robot se encuentra un objeto de interés. Utilizando estas primitivas de reconocimiento en base a tamaño del objeto (el cual da una referencia de distancia), y posición (la cual da una referencia sobre la orientación horizontal con respecto a la visión del robot) se pueden implementar programas sencillos de “búsqueda” de objetos, interactuando con el mundo real.

3.3. Comparación

En la Tabla 1 se presenta una comparación de los dos sistemas a partir de un conjunto de características observadas.

4. Adaptación

A partir del análisis realizado sobre las arquitecturas hardware y software de ambas plataformas se realiza el proceso de adaptación del robot educativo Multiplo N6 Max.

4.1. Adecuaciones aplicadas a la Arquitectura del Hardware

Las arquitecturas del hardware presentes en ambos robots son similares, ya que los principales microcontroladores en ambos equipos son de la misma arquitectura (atmega) y los dispositivos de ENTRADA/SALIDA son los mismos. Por esta razón los cambios introducidos están centrados en ampliar las posibilidades ofrecidas por los robots Multiplo N6 Max proporcionando las habilidades de visión y comunicación inalámbrica WiFi.

Durante el proceso de selección del nuevo hardware se considera principalmente el nivel de compatibilidad con la arquitectura presente en el robot Multiplo N6 Max, la posibilidad de instalar el sistema operativo embebido OpenWrt y el peso, tamaño y costo adecuados a la construcción de un robot educativo de bajo costo.

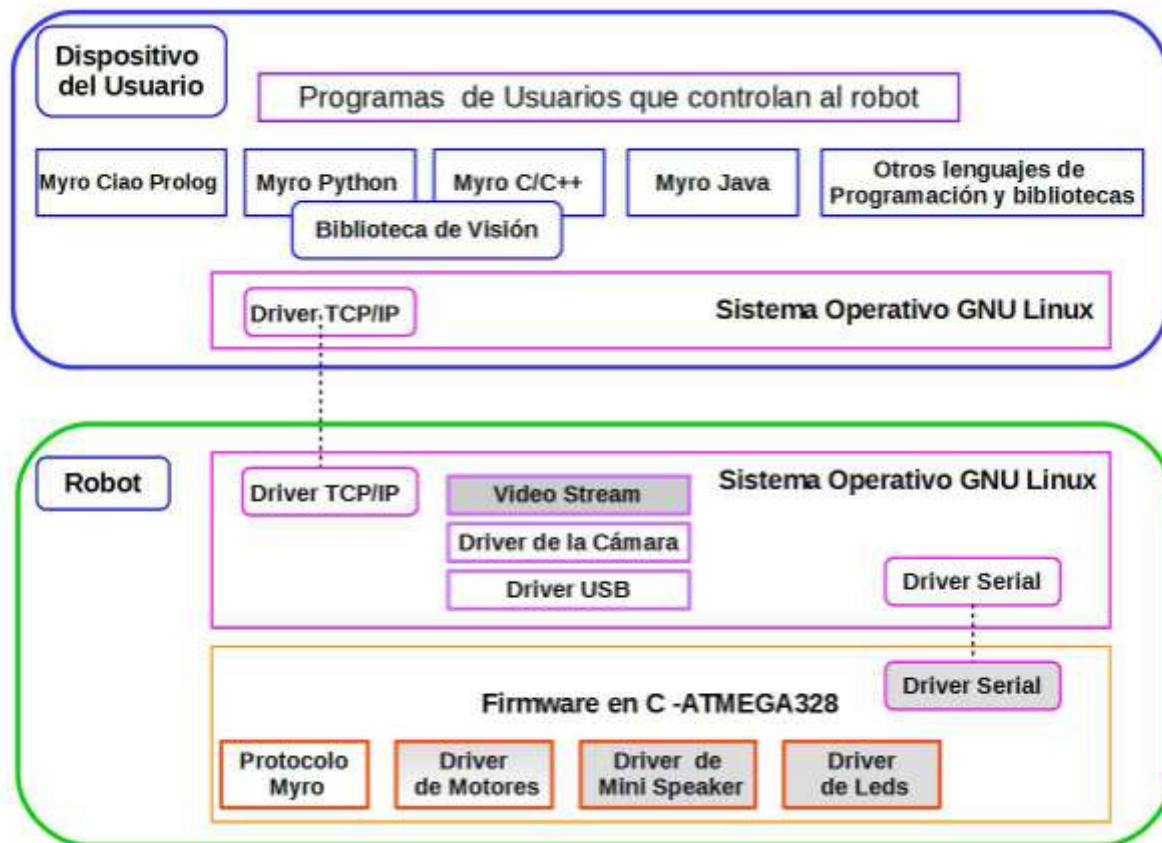


Figura 3: Esquema de la arquitectura del software del sistema.

La principal adecuación consiste en ensamblar al sistema un pequeño wireless router modelo TL-MR3040 que consta de un procesador Atheros AR7240, RAM de 32MiB y memoria Flash de 4MiB. Este dispositivo proporciona comunicación inalámbrica WiFi al robot a frecuencia de 2.4-2.4835GHz, controla una cámara de video UVC conectada al puerto USB y provee comunicación serial con la placa Duino-Bot v2.3 presente en la arquitectura del robot Multiplo N6 Max.

Su peso, dimensiones y autonomía resultan satisfactorias para la construcción de este tipo de robots. Cuenta con una batería interna de 2000mAh recargable vía mini-USB (5V) para periodos de 4 a 5 horas de uso, pesa 94 gr y su tamaño es 100 x 62 x 16mm.

La capacidad de visión se proporciona integrando una cámara de video UVC (USB video class), conectada al router. Este tipo de dispositivo, parte de la arquitectura Frankestito, es so- portado por OpenWrt[7] por medio de drivers del sistema y la aplicación mjpg-streamer, que permite transmitir video a través de conexiones TCP/IP. Más de 250 modelos de webcams comerciales de bajo costo son soportadas.

Las placas instaladas en los robots Multiplo N6 Max y Frankestito, DuinoBot v2.3 basada en el microprocesador ATmega1284P y EasyDuino basada en el microprocesador Atmega328, son Arduino compatibles y comparten las principales características, por lo que no fue requerida ninguna adecuación a nivel de hardware en este dispositivo. Se desarrolla un conversor de niveles de tensión, entre 3.3v y 5v, que actúa como interfaz serial para establecer comunicación serial entre wireless router y placa DuinoBot v2.3.

Tabla 1: Comparación

| Característica | Multiplo N6 Max | Frankestito |
|-------------------------------------|--|--|
| Arduino compatible | EasyDuino | Duino Bot |
| Motorización | Dos ruedas con motorización individual | Dos ruedas con motorización individual |
| Interfaz usuario | Basada en leds y buzzer | Basada en leds y buzzer |
| Fuente de Energía | Portapilas AAA de tres elementos | Pack de baterías interno |
| Sensores | IR y ultrasónico | No incluye |
| Cámara | No incluye | UVC ensamblable |
| Infraestructura de la Comunicación | Cableada vía USB | Wireless |
| Comunicación | Entrada(programa) | Entrada(instrucciones)/ Salida(vídeo) |
| Comunicación en Tiempo de Ejecución | No | Si |
| Lenguajes Soportados | DuinoPack-Minibloq | Python- C++-Java-Ciao Prolog |

4.2. Adecuaciones aplicadas a la Arquitectura del Software Embebido

Las arquitectura del software embebida en los robots Multiplo N6 Max es reemplazada completamente por las piezas de software desarrolladas en el contexto del proyecto Frakestito. Se introducen cambios centrados en acomodar la arquitectura a las características del nuevo hardware ensamblado. En la Fig. 3 se muestran en color gris los componentes que experimentan algún tipo de modificación.

En relación al sistema operativo embebido en el TL-MR3040, se instala la distribución GNU-Linux para dispositivos embebidos OpenWrt versión Chaos Calmer (15.05) [7]. Se configura, compila e instala para ser utilizado como interfaz de comunicaciones y para controlar la visión a través de mjpg-streamer.

Durante la etapa de configuración y compilación, basada en código fuente, de la distribución OpenWrt se configura el wireless y mensajes del kernel, se incorpora el paquete ser2net que proporciona una conexión serial al DuinoBot y se integran drivers UVC para controlar la cámara. Con el ambiente instalado, se configuran los drivers USB y UVC del sistema OpenWrt y el paquete ser2net.

La aplicación mjpg-streamer no se encuentra completamente desarrollada para la versión Chaos Calmer (15.05) embebida en TL- MR3040. Se realiza un proceso de cross - compilation de los fuentes del software para esta arquitectura de software y hardware en particular. Una vez completado satisfactoriamente este proceso se construye un script que inicia automáticamente la captura de imagen de la cámara y streaming de vídeo MJPEG para navegadores web a través de HTTP.

En relación al firmware embebido en el DuinoBot v2.3, sobre el firmware desarrollado para EasyDuino en el contexto del Proyecto Frankestito se implementan modificaciones para adecuarlo y portarlo completamente a la arquitectura DuinoBot v2.3.

Las diferencias en el hardware hacen necesaria la modificación de los drivers de motores y del puerto serial. Aunque EasyDuino y DuinoBot v2.3 son Arduino compatible, en Frankestito y Multiplo N6 Max se implementa la conexión de los distintos componentes en forma diferente. Se analizan esquemáticos y programas de ejemplo disponibles en el Proyecto Multiplo y a partir de esto se modifican los drivers desarrollados para Frankestito.

Durante la etapa de compilación del código fuente para obtener un código binario ejecutable en hardware del robot Multiplo N6 Max se utilizan como base las bibliotecas disponibles en Proyecto Multiplo para DuinoBot v2.3 [6]. En la etapa de transferencia se utiliza la aplicación Hiduploader para transferir la imagen binaria a DuinoBot v2.3. Este proceso está escasamente documentado para esta arquitectura en particular, por lo que se decide desarrollar un proceso de ingeniería inversa para identificar la forma en que el software Minibloq transfiere programas al robot Multiplo N6 Max.

En cuanto al software en el dispositivo remoto del usuario, es decir programas de usuario, entornos de programación, bibliotecas de Myro y de visión y sistema operativo con capacidad de comunicación TCP/IP no registran ninguna modificación. Desde el mismo dispositivo es posible programar tanto robots del Proyecto Frankestito como Multiplo N6 Max ampliado.

4.3. Testing

Para verificar las adaptaciones realizadas al robot Multiplo N6 Max se realizaron tests modulares y de integración. Los tests modulares permitieron verificar los diferentes componentes desarrollados o adaptados. Específicamente:

- Prueba de las modificaciones al firmware Frankestito para su ejecución en el robot Multiplo N6 Max.
- Verificación de la instalación del Linux embebido en el router mr3040.
- Prueba funcional del control de la cámara de vídeo a través de una conexión TCP/IP.
- Prueba individual de cada motor, para movimientos básicos (giro en ambos sentidos) a diferentes velocidades.

Una vez finalizados las pruebas modulares se realizó un test de integración, para verificar el correcto funcionamiento del robot Multiplo N6 Max adaptado. La prueba consistió en ejecutar los programas de tests utilizados con el robot Frankestito, desde una computadora portátil remota. Estos tests integradores permitieron verificar que los equipos portátiles ya preparados para controlar robots Frankestitos pueden controlar, sin ninguna modificación o configuración extra, los robots Multiplo N6 Max adaptados. Específicamente:

- Utilizando la biblioteca en Python MyRO se realizó movimientos del robots hacia todos los sentidos (utilizando los métodos de la biblioteca “atrás()”, “adelante()”).
- Se ejecutaron programas de tests de la visión, utilizando la biblioteca faivisión. Se distinguieron los colores uniformes en objetos y se identificaron sus ubicaciones.
- Se ejecutaron programas que realizan la tarea de encontrar objetos a través de la cámara, y llegar hasta los mismos, a fin de utilizar todas las características del robot Multiplo N6 Max adaptado. El robot pudo llegar hasta los objetos en un tiempo promedio a lo conseguido con los robots Frankestitos.

5. Experiencia

Desde hace más una década, la Facultad de Informática de la Universidad Nacional del Comahue viene desarrollando diversas iniciativas en el ámbito de la promoción, enseñanza y divulgación de las Ciencias de la Computación. Estas iniciativas se concretan en el contexto del vínculo establecido con organismos gubernamentales y escuelas secundarias y primarias de la región y se expresan en proyectos y actividades de investigación, extensión y actualización tecnológica.

En este sentido, se han desarrollado en 2016 dos proyecto de extensión universitaria de duración anual, que tienen entre sus objetivos la enseñanza de métodos y herramientas, que le permitan a los estudiantes de la escuela media, desarrollar una solución computacional a un problema de complejidad simple. Estos desarrollos se abordan utilizando robots como eje motivador.

Los estudiantes de la escuela media realizan en sus escuelas, en el marco de estas actividades, proyectos de software de complejidad incremental que permite al robot realizar alguna tarea final. Por ejemplo, el robot en su rol de jugador de fútbol ejecuta un penal.

Los desarrollos se verifican, sobre los robots adaptados Multiplo N6 Max, durante encuentros programados de los que participan docentes y estudiantes universitarios y de la escuela media.

En este sentido, se observa el comportamiento del robot y se determina el grado en que el programa soluciona el problema planteado. En caso de que los resultados no son los esperados, los estudiantes modifican el programa de modo de poder observar los valores de las estructuras de datos en tiempo de ejecución, detectan errores y revisan el código.

Estas ventajas también son aprovechadas por los docentes para marcar fortalezas en el código.

Asimismo, se ha participado como expositor en diversos eventos de divulgación tecnológica de la región, presentando a los robots Multiplo N6 Max adaptados, a fin de incentivar el interés en las escuelas y en la ciudadanía en los avances científicos y tecnológicos. En estos encuentros, el público interactuó con el robot en forma interactiva, hecho que despertó la curiosidad sobre el modo de comunicación. La transmisión de vídeo en tiempo real y la posibilidad de verlo a través de los

smartphones personales con sólo acceder a un navegador, fue un disparador para realizar consultas sobre su implementación y comprender algunos conceptos básicos de la comunicación.

6. Conclusiones y Trabajo Futuro

En este trabajo se presentó un conjunto de modificaciones realizadas sobre un grupo de ocho robots educativos Multiplo N6 Max a partir de la reutilización de la arquitectura de hardware y software presente en Frankestito.

Se logró instanciar el modelo teórico conceptual en otra arquitectura, lo que contribuye a demostrar las fortalezas del modelo. La documentación del proceso y el conocimiento producido en el ámbito de los sistemas embebidos abre la posibilidad a desarrollar futuras reutilizaciones en arquitecturas similares.

De esta forma se amplía la posibilidad de disponer de robots educativos de bajo costo, con las características presentes en la arquitectura Frankestito, a un número mayor de escuelas.

Se produjo conocimiento de carácter teórico y práctico en el contexto del software embebido y las plataformas de robótica educativa que contribuyen al desarrollo de ambos campos de conocimiento.

En el contexto del proyecto está previsto desarrollar nuevas líneas de acción en el marco de la Extensión Universitaria que utilicen como recurso el conjunto de robots modificados. Esto amplía las posibilidades de profundizar la concreción de experiencias educativas en el contexto de la promoción de la enseñanza de las Ciencias de la Computación en las escuelas primarias y secundarias de la región.

Referencias

- [1] Change the Equation Homepage. Último acceso Abril 2017, website <http://changetheequation.org>.
- [2] Consejo Federal de Educación - Resolución 263/15. Último acceso Abril 2017, website <http://www.me.gov.ar/consejo/resoluciones/res15/263-15.pdf>.
- [3] Institute for Personal Robots in Education Homepage. Último acceso Abril 2017, website <http://www.roboteducation.org/>.
- [4] Laboratorio de Tecnologías Informáticas. Homepage. Ultimo acceso Abril 2017, website <http://labti.fi.uncoma.edu.ar/trac/wiki/RoboTitos>.
- [5] Lego Mindstorms. Último acceso Abril 2017, website <http://www.lego.com/en-us/mindstorms>.
- [6] Multiplo —Open Source Robotics Building System.
- [7] OpenWrt Homepage. Último acceso Abril 2017, website <https://www.openwrt.org/>.
- [8] RobotGroup HomePage. Último acceso Abril 2017, website <http://www.robotgroup.com.ar>.
- [9] The Finch Robot. Último acceso Abril 2017, website <http://www.finchrobot.com/>.
- [10] The K–12 Computer Science Framework. The Computer Science Teachers Association,, New York, 2016.

- [11] Cameron Wilson and Leigh Ann Sudol and Chris Stephenson and Mark Stehlik. *Running on Empty: The Failure to Teach K–12 Computer Science in the Digital Age*. ACM and The Computer Science Teachers Association, 2010.
- [12] Eduardo Grosclaude, Rafael Zurita, José Riquelme, Rodolfo del Castillo, and Miriam Lechner. *Designing A Myro-Compatible Robot For Education As Copyleft Hardware*. In CACIC 2014, pages 372 – 382. UNLAM, 2014.
- [13] Seymour Papert. *Mindstorms: Children, computers, and powerful ideas*. Basic Books, Inc., 1980.
- [14] Seymour Papert and Idit Harel. *Constructionism*. Ablex Publishing Corporation, Norwood, NY, 1991.
- [15] Jim Stanton, Lynn Goldsmith, Richards Adrion, Sarah Dunton, Katie Hendrickson, Alan Peterfreund, Pat Yongpradit, Rebecca Zarch, and Jennifer Zinth. *State of the States Landscape Report: State- Level Policies Supporting Equitable K–12 Computer Science Education*. Education Development Center, 2017.