

EXPERIENCIAS EN EL DISEÑO DE ROBOTS EDUCACIONALES SUBMARINOS

Antonio Peñalver, Josep Marín, Javier Ponce, Raúl Marín, José V. Martí, Gustavo Casañ, Pedro Sanz
e-mail(penalvea,rmarin,vmarti,ncasan,sanzp@uji.es josep.marin,javier.ponce@laudebsv.com)

Resumen

El presente trabajo da a conocer las actividades preliminares realizadas en la Universidad Jaume I en el ámbito de la robótica educacional submarina y nuestros proyectos futuros. Se muestra la línea temporal seguida desde la primera experiencia desarrollada en el laboratorio, donde se mejoró el sistema básico MIT Sea Perch para su control remoto con el uso de un sistema Arduino, así como sus pruebas teleoperadas en una cuba de agua en el Interactive and Robotics Systems Lab (IRS Lab). El resultado conseguido con este sistema fue sin lugar a dudas la motivación del estudiante para desarrollar una tesis doctoral en el control de sistemas robóticos submarinos, la cual ha sido defendida recientemente. Con esta experiencia inicial, y dado el interés social por este campo, se preparó un taller de robots educativos submarinos con PVC y controladores Arduino programables a través de tabletas Android. El taller fue un éxito y fue la base para la preparación de un proyecto de investigación de una escuela en la First Lego League, obteniendo uno de los premios, y resultando extraordinariamente positiva la experiencia en cuanto a la motivación de los estudiantes se refiere. La última experiencia realizada hasta el momento es una extensión de la anterior, tomando como base la electrónica del robot submarino OpenRov, la cual ya dispone de un ordenador a bordo

Palabras clave: Robótica submarina, Educación, Arduino

1 INTRODUCCIÓN

En los últimos años, al mismo tiempo que alcanzaba cada vez más rincones de la sociedad, la robótica se ha vuelto habitual en todos los ámbitos educativos [2], desde la escuela a la universidad, demostrando su eficacia en la enseñanza STEM [3]. Sin embargo esta es una robótica basada en gran parte en simuladores [5], brazos robots y robots terrestres móviles sencillos [17] [6]. La existencia de robots aéreos y submarinos es casi desconocida en el ámbito educativo, tanto como

herramienta de aprendizaje de la ciencia y la tecnología como del aprendizaje de la propia robótica aérea y submarina. El presente trabajo da a conocer las actividades preliminares realizadas en la Universidad Jaume I en el ámbito de la robótica educacional submarina. En primer lugar se presenta un breve estado del arte. A continuación, se presentan las tres experiencias desarrolladas, tanto en sus aspectos técnicos como en la respuesta recibida por parte de los alumnos, para finalmente realizar unas breves conclusiones y comentar algunas de las líneas de trabajo futuras.

2 ESTADO DEL ARTE

Tal vez el robot submarino dedicado a propósitos educativos que tenga más influencia sea el MIT Sea Perch [11]. El Sea Perch es un ROV (remotely operated underwater vehicle, vehículo subacuático operado remotamente) fabricado con tuberías de PVC y otros materiales baratos. El Programa Sea Perch, creado en 2003, enseña a profesores a construirlo para que ellos a su vez puedan trabajar con sus estudiantes, fabricar los suyos propios y emplearlos en misiones de búsqueda en cuerpos de agua cercanos. Este robot permite que los alumnos aprendan las bases de la ciencia detrás de la construcción de submarinos y sus principios básicos de movimiento.

El centro Marine Advanced Technology Education [9] es una organización americana cuyas competiciones y material educativo han alcanzado rango internacional. De especial interés desde el aspecto educativo son sus competiciones ROV, creadas con el propósito de:

- Implicar a los estudiantes en STEM y exponerlos a carreras en ciencias y tecnología.
- Animar a los estudiantes a desarrollar y aplicar sus habilidades técnicas, de trabajo en equipo y de resolución de problemas.
- Proporcionar fondos, material y experiencia técnica para apoyar el aprendizaje de los estudiantes.
- Cubrir las necesidades de la industria.

La estructura de la competición está organizada en tareas realistas de creciente dificultad, dando la oportunidad a los estudiantes de incrementar sus conocimientos y habilidades.

La compañía OpenROV [12] ofrece una variedad de robots muy extendidos. Trident es un robot submatino de coste relativamente reducido y gran funcionalidad que ha sido empleado habitualmente en muchas universidades. Sin embargo desde el punto de vista educativo resulta mucho más interesante su serie de DIY ROVs, que permiten crear tus propios robots a partir de sus componentes y diseños.

3 PRIMER ROV EN PVC

Como parte de las actividades del IRS Lab y siguiendo las indicaciones del MIT Sea Perch, se diseñó y produjo un robot submarino con control a distancia. La estructura del submarino estaba hecha de PVC y polietileno, a la que se le adhirieron tres motores DC con el fin de permitir que el robot pudiese desplazarse en todas las direcciones dentro del agua. Aunque en el original solo se puede controlar el movimiento del submarino mediante un mando que está conectado por un cable al submarino y a una batería. Este mando consta de cuatro interruptores que utilizados correctamente permiten al submarino moverse en cualquier dirección. Como parte del proyecto se desarrolló una interfaz de usuario, la cual se comunica con un microcontrolador situado en una boya que también alberga una batería. La comunicación se realiza mediante una conexión TCP/IP. La boya (y el microcontrolador) se conecta con el submarino mediante un cable.

3.1 HARDWARE

Existen muchos microcontroladores y plataformas con microcontroladores disponibles para la computación física, como Parallax Basic Stamp, BX-24 de Netmedia, Phidgets y Handyboard del MIT. Todas estas herramientas organizan el complicado trabajo de programar un microcontrolador en paquetes fáciles de usar. Sin embargo el microcontrolador elegido fue Arduino UNO [1], junto con dos accesorios que le permiten obtener las funcionalidades necesarias para el proyecto: la placa Arduino Wiffy Shield, que dota al Arduino de capacidad para conectarse a una red de área local y transmitir y recibir información. Y la Arduino Motor Shield, que permite al microcontrolador manejar motores, tanto DC como servos y motores paso a paso. Se consideró que Arduino, además de simplificar el proceso de trabajar con microcontroladores, ofrece algunas ventajas respecto a

otros sistemas:

- Las placas Arduino son más asequibles que otras plataformas de microcontroladores.
- El software de Arduino funciona en los sistemas operativos Windows, Macintosh OSX y Linux.
- El entorno de programación de Arduino es fácil de usar para principiantes y lo suficientemente flexible para los usuarios avanzados.
- Pensando en los profesores, Arduino está basado en el entorno de programación de Processing con lo que el estudiante que aprenda a programar en este entorno se sentirá familiarizado con el entorno de desarrollo Arduino.
- El software Arduino esta publicado bajo una licencia libre y preparado para ser ampliado por programadores experimentados. El lenguaje puede ampliarse a través de librerías de C++, y si se está interesado en profundizar en los detalles técnicos, se puede dar el salto a la programación en el lenguaje AVR C en el que está basado. De igual modo se puede añadir directamente código en AVR C en tus programas si así lo deseas.
- Hardware ampliable y de código abierto. Está basado en los microcontroladores ATmega168, ATmega328 y ATmega1280 [10]. Los planos de los módulos están publicados bajo licencia Creative Commons, por lo que diseñadores de circuitos con experiencia pueden hacer su propia versión del módulo, ampliándolo u optimizándolo. Incluso usuarios relativamente inexpertos pueden construir la versión para placa de desarrollo para entender cómo funciona y ahorrar algo de dinero.

El Arduino UNO es una placa con microcontrolador basada en el ATmega328. Tiene 14 pines con entradas/salidas digitales (6 de las cuales pueden ser usadas como salidas PWM), 6 entradas analógicas, un cristal oscilador a 16Mhz, conexión USB, entrada de alimentación, una cabecera ISCP, y un botón de reset. Contiene todo lo necesario para utilizar el microcontrolador, simplemente se ha de conectar a un ordenador a través del cable USB o alimentarse con un transformador o una batería para empezar a trabajar con él.

El Arduino UNO facilita en varios aspectos la comunicación con el ordenador, otro Arduino o otros microcontroladores. El ATmega328 proporciona comunicación vía serie UART TTL (5V), disponible a través de los pines digitales 0(RX) y

1(TX). Un chip FTDI FT232RL integrado en la placa canaliza esta comunicación serie a través del USB y los drivers FTDI (incluidos en el software de Arduino) proporcionan un puerto serie virtual en el ordenador. El software incluye un monitor de puerto serie que permite enviar y recibir información textual de la placa Arduino. Los LEDs RX y TX de la placa parpadearan cuando se detecte comunicación transmitida a través del chip FTDI y la conexión USB (no parpadearan si se usa la comunicación serie a través de los pines 0 y 1). La longitud y amplitud máxima de la placa UNO es de 2.7 y 2.1 pulgadas respectivamente, con el conector USB y la conexión de alimentación sobresaliendo de estas dimensiones. Tres agujeros para fijación con tornillos permiten colocar la placa en superficies y cajas.

3.1.1 Arduino Wifly Shield

La Wifly Shield permite la conectividad Wifi a un Arduino de forma fácil y eficaz. Utiliza el módulo Wifi WiFly RN-131G de Roving Networks junto con un chip conversor de SPI/UART SC16IS750 que simplifica enormemente la tarea ya que se encarga de todo el protocolo y es capaz de conectarse a redes inalámbricas Wifi 802.11b/g. La alimentación es tomada del pin Vin del Arduino y regulada a 3,3V para alimentar el módulo Wifi. Para comunicarse con el módulo se utiliza el protocolo SPI utilizando los pines digitales 10 a 13 (CS, MOSI, MISO y SCLK). La Shield también incorpora una pequeña área de prototipado donde se pueden soldar algunos componentes.

3.1.2 Arduino Motor Shield

Arduino es un gran punto de partida para la electrónica, y con el Arduino Motor Shield puede ser una plataforma sencilla y agradable para la robótica y la mecatrónica. El Arduino Motor Shield dispone de:

- 2 conectores para controlar motores servos de 5V.
- Hasta 4 conectores para controlar motores DC bidireccionales, con una selección individual de velocidad de 8 bits (por tanto, aproximadamente un 0,5 por ciento de resolución).
- Proporciona 2 conectores para motores paso a paso (unipolar o bipolar) con bobina única, de doble bobina, intercalado o de micro-paso.
- También provee de 4 puentes H. El chip L293D ofrece 0,6A por puente (1,2A de pico) con la protección de apagado térmico, de 4,5V a 36V.

- Para facilitar el montaje, este accesorio dispone de terminales que son conectores grandes, para poder conectar fácilmente los cables, tanto de los motores como de la alimentación.

Por último, permite la utilización de una alimentación independiente para el Arduino Motor Shield y otra para la placa Arduino principal. Aunque también puede ser utilizada una única alimentación, bien sea conectada al Arduino Motor Shield, como a la placa principal.

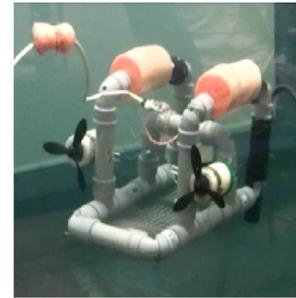


Figura 1: Submarino durante el experimento en la cuba.

3.1.3 Adición de la cámara

Se decidió añadir una cámara especialmente preparada para su uso dentro del agua al submarino, dado que no disponía de ningún método para medir el comportamiento del robot más allá de la vista del conductor. Las imágenes de la cámara proporcionarían la misma visión que si nos encontrásemos dentro del submarino (un ejemplo puede verse en la Figura 2). La cámara se colocó en posición vertical y con la lente mirando hacia abajo, junto al tubo de la estructura del submarino que se encuentra en la parte trasera y derecha, en posición también vertical. Una vez juntos se han adherido utilizando cinta americana, consiguiendo que la cámara quede totalmente sujeta al submarino. El resultado final se puede ver en la Figura 1.



Figura 2: Imagen obtenida por la cámara del submarino durante las pruebas.

Una vez que el submarino y la cámara están listos, se ha procedido a preparar el lugar donde se probará el experimento. El lugar escogido ha sido un estanque de agua con paredes de cristal, que permiten ver los movimientos del submarino al que se le ha incorporado una ánfora, para que mediante la cámara el usuario pudiera ver los objetos en el fondo del estanque. Finalmente, se ha procedido a añadir flotabilidad al submarino, para contrarrestar el exceso de peso que aporta la cámara.

3.2 SOFTWARE: ROS

Robot Operating System (ROS) [14] es un meta-sistema operativo para los robots, de código abierto. Ofrece los servicios que se esperarían de un sistema operativo como son la abstracción de hardware, control de los dispositivos a bajo nivel, implementación de las funcionalidades más comúnmente usadas, paso de mensajes entre procesos, y gestión de paquetes. También proporciona herramientas y librerías para la obtención, construcción, escritura y ejecución de código entre varios equipos. ROS implementa distintos estilos de comunicación, incluyendo comunicación síncrona de servicios al estilo RPC, transmisión asíncrona de datos a través de topics, y almacenamiento de datos en un servidor de parámetros. ROS no es un framework en tiempo real, a pesar de que es posible la integración de ROS con código de tiempo real.

El principal objetivo de ROS es apoyar la reutilización de código en la investigación y en la robótica. ROS es un marco de distribución de procesos, que permite acoplar en tiempo de ejecución, archivos ejecutables que han sido diseñados independientemente unos de los otros. Estos procesos se pueden agrupar en paquetes y pilas, que pueden ser fácilmente compartidas y distribuidas. ROS también es compatible con un sistema federado de repositorios de código que permiten una colaboración para una mejor distribución de código. Este diseño permite tomar decisiones independientes sobre el desarrollo y ejecución, pero pudiendo ser juntadas mediante la infraestructura de ROS. Para apoyar el objetivo principal de ROS, aparecen otros objetivos que ROS intenta cumplir:

- ROS está diseñado para ser lo más fino posible, para permitir que el código escrito en ROS pueda ser utilizado en frameworks de software de robots.
- El framework de ROS es fácil de implementar en cualquier lenguaje de programación moderno. Ya está implementado en Python, C++ y Lisp, y existen librerías experimentales en Java y Lua.
- ROS facilita el testeo mediante una orden interna llamada Rostest.
- ROS es apropiado para sistemas de largos tiempos de ejecución y para los procesos de desarrollo de gran tamaño.

En este proyecto, ROS se ha utilizado para poder comunicar la interfaz de usuario, con una aplicación que actuará como conectora entre dicha interfaz y el microcontrolador. Para ello, en primer lugar, se ha creado un paquete de ROS llamado mandos, mediante el comando:

```
roscreate-pkg mandos std_msgs roscpp
```

Además de crearlo, el comando le indica a ROS que éste paquete tiene dependencias con los mensajes de ROS y con C++. Una vez creado el paquete, se ha creado un tipo de mensaje llamado Movimiento, que tan solo consta de una parámetro llamado dirección, que es un número entero. Así pues se creará la interfaz como un cliente, que enviará mensajes de este tipo al servidor, que será el conector.

3.2.1 ¿QUÉ ES Qt?

Qt [15] es una amplia plataforma de desarrollo que incluye clases, librerías y herramientas para la producción de aplicaciones de interfaz gráfica en C++ que pueden operar en varias plataformas. Qt, incluye soporte de tecnologías como OpenGL, XML, Bases de Datos, programación para redes, internacionalización y mucho más.

Además Qt dispone de una amplia gama de herramientas que facilitan la creación de formularios, botones y ventanas de dialogo con el uso del ratón. Las aplicaciones creadas con Qt son muy elegantes, se ven y se operan mejor que las aplicaciones nativas. Qt dispone de tres grandes ventajas ante otras librerías de ventanas:

- Es completamente gratuito para aplicaciones de código abierto.
- Las herramientas, librerías y clases están disponibles para casi todas las plataformas Unix y sus derivados (como Linux, MacOS X, Solaris, etc) como también para la familia Windows, por lo que una aplicación puede ser compilada y utilizada en cualquier plataforma sin necesidad de cambiar el código y la aplicación se verá y actuará mejor que una aplicación nativa.
- Tiene una extensa librería con clases y herramientas para la creación de ricas aplicaciones. Estas librerías y clases están bien documentadas, son fáciles de usar y tienen

una gran herencia de programación orientada a objetos lo cual hace de la programación de interfaces gráficas una aventura placentera.

Tanto para el diseño como para la programación de la interfaz, se ha utilizado Qt Creator. Éste es un entorno de desarrollo integrado creado por Trolltech para el desarrollo de aplicaciones con las bibliotecas Qt.

3.2.2 INTERFAZ

Por lo que respecta al diseño de la interfaz, se ha intentado que la interfaz sea agradable, y fácil de utilizar. Por tanto se han utilizado dibujos para los botones, que describan claramente cual es su función. La interfaz está claramente separada en dos partes, como se puede ver en la figura 3. La parte izquierda se corresponde al control del motor del submarino situado en posición vertical. Y la parte de la derecha controla los otros dos motores. Debido a que la plataforma Qt funciona mediante eventos, es decir, la aplicación está dormida hasta que el usuario realiza alguna acción sobre la interfaz, se han necesitado botones para parar los motores, ya que no es posible indicar que el motor esté activo mientras esté pulsado un botón.

Las dos partes de la interfaz son totalmente independientes, de forma que puede estar pulsado un botón de la parte izquierda, y pulsar uno de la derecha sin que el anterior se desactive, o viceversa. Esto permite que el submarino pueda moverse en dos dimensiones simultáneamente, por ejemplo hacia arriba y hacia delante.

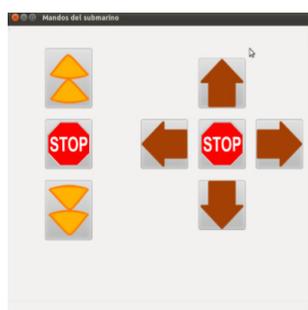


Figura 3: Interfaz con el robot.

Al pulsar cualquier botón de la interfaz, este se deshabilita y se habilita el botón que se encuentre deshabilitado de la parte dónde se encuentra el botón pulsado. Inicialmente se encuentran deshabilitados los dos botones de Stop. A continuación se van a detallar las funcionalidades que tienen cada uno de los botones:

- Parte izquierda de la interfaz. Doble flecha hacia arriba: Activa el motor en posición vertical, de tal forma que el submarino se mueva

hacia arriba. Doble flecha hacia abajo: Activa el motor en posición vertical, de tal forma que el submarino se mueva hacia abajo. Stop: Desactiva el motor en posición vertical.

- Parte derecha de la interfaz. Flecha hacia arriba: Activa los motores en posición horizontal para que el submarino se mueva hacia delante. Flecha hacia la derecha: Activa los motores en posición horizontal para que el submarino se mueva hacia la derecha. Flecha hacia abajo: Activa los motores en posición horizontal para que el submarino se mueva hacia detrás. Flecha hacia la izquierda: Activa los motores en posición horizontal para que el submarino se mueva hacia la izquierda. Stop: Desactiva los motores en posición horizontal.

Una vez diseñada la interfaz, se ha pasado a programarla utilizando C++ y Qt. También se ha utilizado ROS para poder comunicar la interfaz con otras aplicaciones.

3.2.3 Conexión interfaz-microcontrolador

También se ha programado una aplicación que es capaz de comunicarse tanto con la interfaz gráfica mediante ROS, como con el microcontrolador mediante una conexión TCP/IP. En esta sección se detalla el código del conector que realiza estas tareas.

3.2.4 Programación de la conexión

Lo que realiza el código de la conexión es, en primer lugar, inicializar ROS. A continuación se crea un socket que se comunicará con el microcontrolador, cuya dirección IP es 192.168.0.102, por el puerto 8080. Éste puerto había sido escogido anteriormente para realizar la comunicación entre los dos componentes. Una vez que el programa se puede comunicar con el microcontrolador, se crea un nodo de ROS, y un servidor en éste, el cual lee los mensajes de tipo Movimiento. También se le indica que para cada mensaje que lea, realice la función `enviar_movimiento`.

La función `enviar_movimiento`, recibe como parámetro un mensaje de tipo Movimiento, y la dirección donde escribir la respuesta a dicho mensaje. Dentro de la función, se lee el entero recibido mediante el mensaje. Dependiendo de dicho número, se enviará un carácter o otro al microprocesador, mediante el socket creado anteriormente, para que éste active o desactive los motores del submarino y éste se mueva en la dirección que el usuario desea. Por último, si el proceso se ha realizado correctamente, se contesta a la interfaz de usuario un 1.

3.3 Resultados obtenidos

Una vez preparados todos los componentes se procedió a la realización del experimento, permitiendo llegar a las siguientes conclusiones:

- Es sencillo utilizar el submarino gracias a la interfaz de usuario.
- La estructura del submarino permite la adición de nuevos componentes con facilidad y por tanto nuevas capacidades.
- Se puede tanto añadir como quitar flotabilidad al submarino fácilmente.
- Debido a la poca potencia de los motores, la adición de componentes pesados, puede impedir el correcto desplazamiento del submarino.

4 EXPERIENCIAS DOCENTES EN ESCUELAS

Tras el conocimiento adquirido gracias al trabajo explicado en la sección anterior, se procedió a validar su uso docente como actividad extraescolar en un Instituto cercano a la Universidad Jaume I, formando un equipo que participó de la competición First Lego League [7] los años 2015, 2016 y 2017. Este equipo fue entrenado por el Prof. Raúl Marín y los entrenadores auxiliares Josep Marín y Javier Ponce, los cuales participaron también en gran medida en el diseño de las actividades educativas, diseño y montaje de los robots. En el 2015 los alumnos se centraron en aspectos de programación y montaje del ROV, dado que el diseño y componentes ya estaban definidos.

En primera estancia ha sido de vital importancia la visita desarrollada al IRS Lab por los estudiantes, entendiendo el concepto de ROV, AUV, así como las tecnologías necesarias para poder sumergir motores y electrónica (ver Figura 4).

5 ROV MIT SEA PEARL EN PVC

En el año 2016 los estudiantes presentaron como trabajo de investigación un robot en PVC, teleoperado desde tierra por wifi con una controladora Arduino y una umbilical, e inspirado en la limpieza de los fondos marinos.

En primer lugar desarrollaron varios talleres de diseño e impresión 3D, presentando diversos diseños del robot y eligiendo ellos mismos el más interesante. También diseñaron algunas herramientas de limpieza, como ganchos, garras y palas (ver Figuras 5).



Figura 4: Visita de los estudiantes de instituto al laboratorio IRS de la UJI

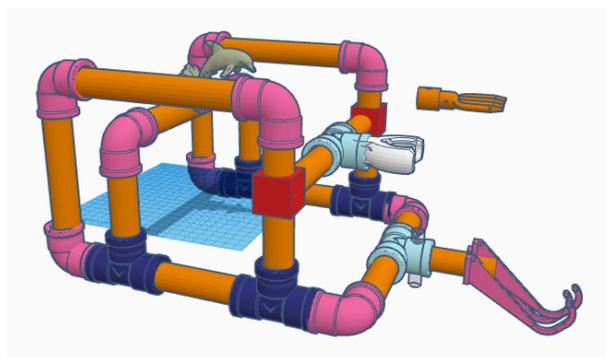


Figura 5: Uno de los diseños del robot submarino en PVC realizado por los estudiantes en el sistema Autodesk Tinkercad [18]

Una vez seleccionado el diseño se procedió a la compra del material en PVC, su montaje y pintura. La instalación de los motores fue realizada gracias a la Universidad de Girona (CIRS) [4] que nos mostró el proceso a seguir con cera de depilar. Los estudiantes disfrutaron en gran medida del montaje, programación de la electrónica Arduino (sistema Makeblock [8]), así como la interfaz de usuario Android (ver Figura 6).

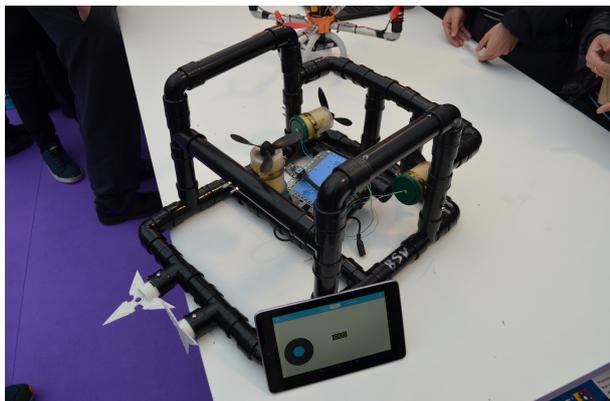


Figura 6: Robot Submarino en PVC construido por estudiantes de secundaria de Castellón

El momento más emocionante y motivador a nivel educacional fue la puesta en marcha del robot en la piscina, tras la instalación de la flotación y el trimado del vehículo (ver Figura 7).

El resultado fue la presentación del submarino en la First Lego League de la Comunidad Valenciana, causando una gran impresión al jurado, el cual otorgó al equipo uno de los premios.

6 MEJORA SISTEMA OPENROV

En el siguiente curso (2017) se decidió continuar con el estudio de los sistemas robóticos submarinos, pensando esta vez en la plataforma OpenRov [12], la cual dispone de electrónica embebida (i.e. ordenador, cámara, sensores y batería), gracias al uso de cilindros estancos. Los estudiantes disfrutaron de la mejora del robot con aluminio estructural, inspirado en su uso en entornos mas peligrosos como el de las aguas ya tratadas de una depuradora (ver Figuras 8 y 9). Dados los mayores problemas de trabajar con aluminio a los alumnos se les proporcionó un diseño y los componentes y realizaron el montaje. Su aprendizaje en esta ocasión se centró en la programación.

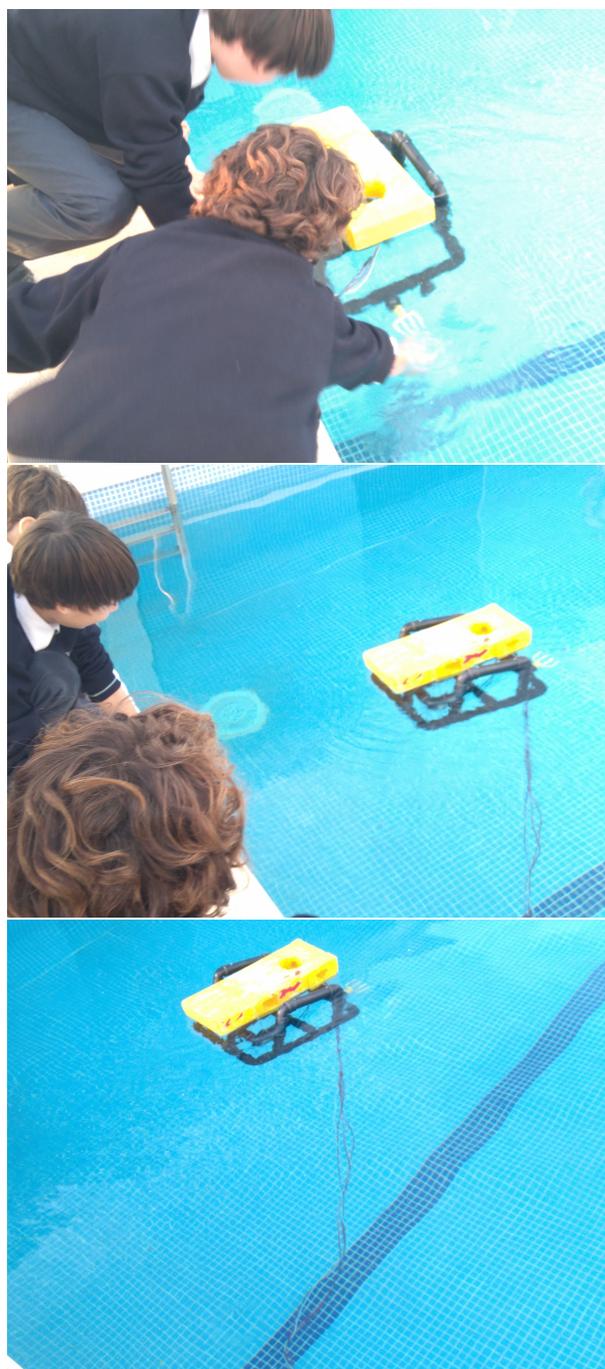


Figura 7: Robot en PVC en una piscina tras haber realizado la instalación de la espuma de flotación

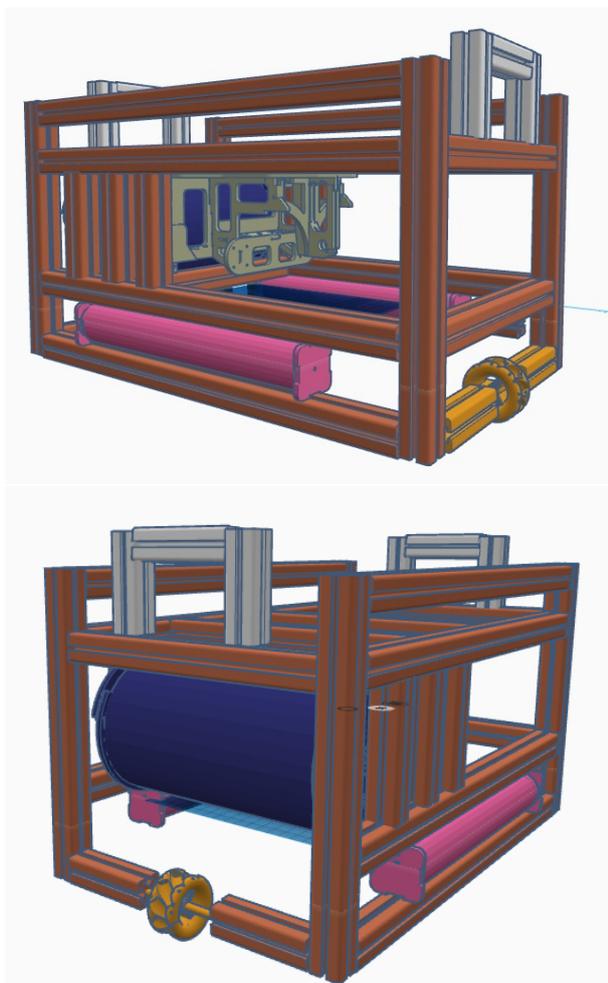


Figura 8: Diseño robot educacional con Tinkercad

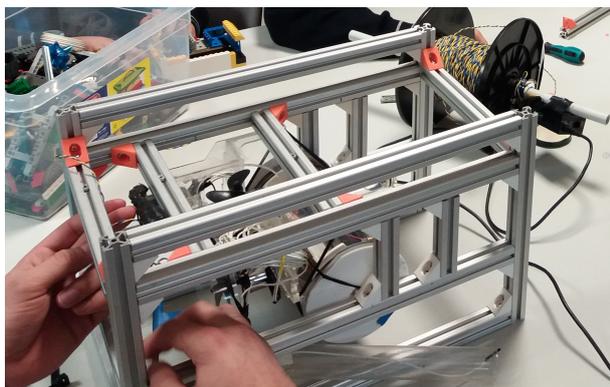


Figura 9: Construcción robot educacional con estructura de aluminio estructural y electrónica OpenROV con cámara y ordenador embebido

7 Conclusiones

Los campos de la robótica submarina y la robótica educativa están alcanzando grados de madurez que hacen factible que en pocos años exista una gran diversidad de robots submarinos educativos versátiles y de costo reducido.

Las experiencias del IRS Lab muestran como la creación de sistemas de enseñanza-aprendizaje de robótica submarina están listos para una gran expansión entre enseñanzas medias e incluso primarias.

En el futuro hay varios aspectos de la robótica submarina educativa que esperamos explorar:

- Reducción de tamaño de los sistemas manteniendo o incluso aumentando la funcionalidad, de forma que sean empleables incluso en recipientes tipo pecera, accesibles en centros de educación primaria e institutos.
- Incorporación de comunicaciones entre ROV para trabajar los aspectos de comunicación y coordinación entre robots.
- Creación de una red de centros de enseñanzas medias que permitan compartir recursos y aumentar la enseñanza de robótica submarina en niveles no universitarios.
- Integración del simulador UWSim [13] con Scratch [16], de forma que personas sin formación puedan comenzar a programar robots submarinos.

Agradecimientos

Este trabajo ha sido parcialmente financiado por el Ministerio de Economía, Industria y Competitividad de España con la subvención "DPI2017-86372-C3-1-R" (Proyecto COMOMUIS), por el Gobierno de la Comunidad Valenciana bajo las subvenciones "GVA-PROMETEO/2016/066" e "IDIFEDER/2018/013", y por la UJI bajo la subvención "UJI B2018-34" (Proyecto NEPTUNO).

English summary

EXPERIENCES IN THE DESIGN OF EDUCATIONAL UNDERWATER ROBOTS

Abstract

The present work discloses the preliminary activities carried out at the Jaume I University in the field of underwater educational robotics and our future projects. The

temporal line followed since the first experience in the laboratory, where the remote control of the MIT Sea Pearch basic system was improved with the use of an Arduino system, as well as its teleoperated tests in a water tank in the Interactive and Robotics Systems Lab (IRS Lab). The result achieved with this system was undoubtedly the motivation of the student to develop a doctoral thesis in the control of underwater robotic systems, which has been recently defended. With this initial experience, and given the social interest in this field, a workshop of underwater educational robots with PVC and programmable Arduino controllers through Android tablets was prepared. The workshop was a success and was the basis for the preparation of a research project of a school in the First Lego League, obtaining one of the prizes, and being extremely positive experience for the students, motivating them. The last experience done until now is an extension of the previous one, based on the electronics of the OpenRov submarine robot, which already has a computer on board.

Keywords: Underwater robotics, Education, Arduino.

Referencias

- [1] Arduino, <https://www.arduino.cc/>. Consultado 15 de mayo de 2019.
- [2] Barreto, F. y Benitti, V. (2012). Exploring the educational potential of robotics in schools: A systematic review. *Computers & Education*, vol. 58 (3), pp. 978–988.
- [3] Breiner, J. M., Harkness, S. S., Johnson, C. C. y Koehler, C. M. (2012). *What Is STEM? A Discussion About Conceptions of STEM in Education and Partnerships, School Science and Mathematics*, Wiley Online Library, issn 1949-8594, doi: 10.1111/j.1949-8594.2011.00109.x, vol. 112 (1), pp. 3-11.
- [4] CIRS Universitat de Girona, <http://cirs.udg.es>, consultado 4 de mayo de 2019.
- [5] D'Angelo, C., Rutstein, D., Harris, C., Bernard, R., Borokhovski, E. y Haertel, G. (2014). *Simulations for STEM Learning: Systematic Review and Meta-Analysis*. Menlo Park, CA: SRI International.
- [6] Dagdilelis, V., Sartatzemi, M. y Kagani, K. (2005) Teaching (with) robots in secondary schools: some new and not-so-new pedagogical problems. In: *Advanced Learning Technologies, 2005*. ICALT 2005. Fifth IEEE International Conference on, pp. 757-761.
- [7] First Lego League, <http://www.firstlegoleague.es>, consultado el 20 de mayo de 2019.
- [8] Makebot, <https://www.makeblock.com/> Consultado 2 de mayo de 2019.
- [9] Marine Advanced Technology Education (MATE), <https://www.marinetech.org/>, consultado 1 de mayo de 2019.
- [10] Microchip, <http://www.microchip.com>, consultado 30 de mayo de 2019.
- [11] MIT Sea Pearch, <https://seaperch.mit.edu/>, consultado 1 de mayo de 2019.
- [12] OpenROV, <https://www.openrov.com/>, consultado 2 de mayo de 2019.
- [13] Pérez, J., Fornas, D., Marín Prades, R. y Sanz, P. (2018). UWSim, an underwater robotic simulator on the cloud as educational tool. *RIAI - Revista Iberoamericana de Automática e Informática Industrial*, 15 (01), pp. 70-78.
- [14] Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R. y Ng, A.Y. (2009). Ros: an open-source robot operating system. In: *ICRA workshop on open source software*, vol. 3, p. 5. Kobe.
- [15] Qt, <http://www.qt.io>, consultado 25 de mayo de 2019.
- [16] Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B. y Kafai, Y. (2011). Scratch: Programming for all. *Communications of the ACM*, vol. 52 (11), pp. 60-67. DOI: 10.1145/1592761.1592779.
- [17] Schilling, K., Roth, H. y Rösch, O. J. (2002). Mobile mini-robots for engineering education. *Global Journal of Engineering Education*, vol. 6 (1), 79-84.

- [18] Tinkercad, <http://www.tinkercad.com>, consultado 18 de mayo de 2019.



© 2019 by the authors.
Submitted for possible
open access publication
under the terms and conditions of the Creative Commons Attribution CC BY-NC-SA 4.0 license (<https://creativecommons.org/licenses/by-nc-sa/4.0/deed.es>).