

# Coche autónomo esquivador de obstáculos

Alberto Acosta López, Francisco Javier López Franco  
Luis Fernando Peña Flores

**Resumen**—La implementación de un coche autónomo requiere que distintos componentes se comuniquen entre ellos, por lo que se utilizó una tarjeta Arduino Mega 2560 para la toma de decisiones a partir de las lecturas recibidas por los sensores. La comunicación entre la computadora y el coche se realizó mediante XBee y la ruta fue procesada en la computadora. En las conclusiones se ahonda en las limitaciones encontradas.

## 1. INTRODUCCIÓN

La capacidad de dispositivos autónomos móviles para tomar decisiones propias y no por el humano es un tema que adquiere cada día una mayor relevancia, ya que sus aplicaciones tanto en la industria como militar y comercial son benéficas para el ser humano. Es por ello que, para el desarrollo de dichos dispositivos, se requiere comprender el mecanismo detrás de esos mecanismos de sistemas autónomos. Debido a ello, este proyecto pretende desarrollar un sistema que permita a un coche ....

Para la construcción del coche, se trabaja con un microcontrolador Arduino para que controle la lógica del sistema para que, a partir de las lecturas que reciba de distintos sensores, controle los motores de cada una de las ruedas del coche. Mediante el uso de sensores optointerruptores se detecta la velocidad de cada rueda para que se ajuste la dirección que toma el coche y unos sensores para detectar el camino que debe seguir el coche. Dado que los datos se deben de transmitir de manera inalámbrica, se tuvo que incorporar el uso de XBee para que se transmita la información de lectura mediante un tópico de ROS, para visualizar mediante representaciones gráficas los diferentes indicadores del coche. [1] [2]

Este documento está estructurado de la siguiente manera: marco teórico, para describir y dar información de los componentes utilizados; desarrollo, para dar a comprender los pasos seguidos durante esta práctica; resultados, donde se exponen y presentan los objetivos alcanzados al finalizar el proyecto; conclusiones, donde se presentan las conclusiones individuales de los integrantes del equipo; rol o papel, donde se explica la división del trabajo entre los integrantes del equipo; y las fuentes consultadas.

## 2. MARCO TEÓRICO

El sensor *H21A2-1*, conocido también como sensor óptico de barrera, tiene un diodo infrarrojo como emisor y un fototransistor como receptor. Existe una separación de 3mm

aproximadamente entre ambos para que un objeto se pueda introducir y romper la barrera infrarroja. Su funcionamiento es similar al TCR5000, sin embargo, la principal diferencia radica en que éste sensor está diseñado para la identificación de objetos que atraviesen ciertas zonas. [3]

El módulo o chip XBee es capaz de brindar un medio inalámbrico para la interconexión y comunicación entre dispositivos. [4]

El motor de corriente continua es una máquina que convierte la energía eléctrica en mecánica mediante un movimiento rotatorio. Está compuesto por un estator, que da soporte mecánico al motor con un hueco en el centro y sus polos, que pueden ser imanes o devanados con hilo de cobre sobre el núcleo de hierro, y el rotor, que tiene una forma cilíndrica devanada y con núcleo, al cual le llega corriente mediante las escobillas. Su sentido de rotación depende del sentido relativo de las corrientes que circulan por los devanados, por lo que si se quiere invertir el sentido de rotación, se debe invertir el sentido de la corriente inducida. [5]

El puente H es un circuito electrónico que permite a un motor eléctrico de corriente continua girar en ambos sentidos para que se pueda avanzar o retroceder, según sea el caso deseado. Se compone de 4 interruptores, mismos que pueden ser mecánicos o mediante transistores para que se aplique tensión y los motores giren en un mismo sentido. Cuando se abren los interruptores y los otros 2 se cierran, se invierte el voltaje para que el motor gire en un sentido inverso. [6]

## 3. DESARROLLO

La implementación del proyecto final se dividió en distintas fases. La primera fase consistió en comprobar que los componentes del robot diferencial funcionaran adecuadamente. Es decir, se verificó que cada uno de los actuadores, los sensores, los XBee y el Arduino funcionaran de forma correcta. Posteriormente, se procedió a armar el robot correspondiente con los materiales proporcionados.

Una vez construido el soporte físico del robot, se procedió a conectar las salidas de los sensores H21A2-1 a una protoboard y el motor de corriente continua a un puente H construido en la misma protoboard. El sensor H21A2-1 se colocó sobre el engrane que movía el motor DC para detectar cada cambio realizado, de forma que 18 cambios significan una revolución. Una vez armados los circuitos

respectivos, se probó la funcionalidad de los sensores utilizando el siguiente código:

```
1 void setup() {
2   pinMode(2, INPUT);
3   Serial.begin(9600);
4 }
5 void loop() {
6   Serial.print(digitalRead(2));
7   delay(100);
8 }
```

Listing 1. Código utilizado para probar sensor H21a12-1

Para controlar la señal del motor de corriente continua, se utilizó la configuración de los pines de la figura 1. Se conectó el pin 16 al Arduino, así como los pines 3, 6, 11, 14 para hacer funcionar el motor y, finalmente, el pin 8 se conectó a una batería externa para suministrarle corriente que no fuera del Arduino. Para probar la funcionalidad del puente H, se utilizó el siguiente código:

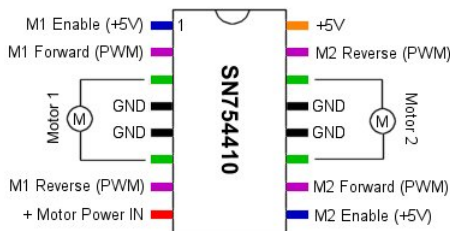


Figura 1. Diagrama de pines del puente H

```
1 void setup() {
2   pinMode(2, OUTPUT);
3   pinMode(3, OUTPUT);
4   pinMode(4, OUTPUT);
5   pinMode(5, OUTPUT);
6   Serial.begin(9600);
7 }
8 void loop() {
9   //Motor 1
10  digitalWrite(2, HIGH);
11  digitalWrite(3, LOW);
12  //Motor 2
13  digitalWrite(4, HIGH);
14  digitalWrite(5, LOW);
15  delay(100);
16 }
```

Listing 2. Código utilizado para probar puente H

Para la segunda fase del proyecto se configuraron los dispositivos de comunicación XBee para que pudieran transmitir y recibir entre ellos. Para dicho objetivo se consideraron los conocimientos y los resultados obtenidos en la práctica 3, que versaba acerca de los protocolos de comunicación. Al igual que en dicha práctica, se utilizó XCTU para la configuración de los parámetros apropiados en cada uno de ellos. Además, en esta fase, se realizó la conexión de los XBee con ROS para que se transmitir y recibir los tópicos correspondiente. Se utilizó un publisher para lograr el efecto de suscripción como si fuera un nodo más. Para probar la conexión entre los XBees y ROS, se utilizó el siguiente código:

```
1 #include <ros.h>
2 #include <std_msgs/String.h>
```

```
3 ros::NodeHandle nh;
4
5
6 std_msgs::String str_msg;
7 ros::Publisher chatter("chatter", &str_msg);
8
9 unsigned char hello[] = "Hello from Node 1!";
10
11 void setup()
12 {
13   nh.initNode();
14   nh.advertise(chatter);
15 }
16
17 void loop()
18 {
19   str_msg.data = hello;
20   chatter.publish(&str_msg);
21   nh.spinOnce();
22   delay(1000);
23 }
```

Listing 3. Código utilizado para probar la conexión entre XBee

A lo largo de la tercera fase del proyecto se implementaron el algoritmo de cálculo del controlador PID y el algoritmo de búsqueda heurística A\* que servirá para calcular la ruta óptima que recorrerá el robot. El PID se implementó como en el realizado durante la práctica 4, que consistía en realizar funciones de control sobre motores DC. Dado que el controlador PID utiliza ecuaciones integrodiferenciales se calculó, por simplicidad, sus equivalentes en sumas y en diferencias. El controlador PID se puede observar de manera gráfica en la figura 2.

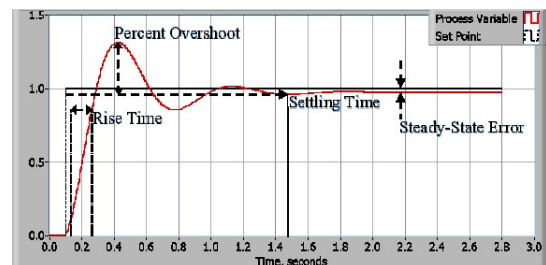


Figura 2. Gráfica del controlador PID

Para el algoritmo heurístico A\* se procedió utilizando los conocimientos aprendidos en el curso de Inteligencia Artificial. Las heurísticas utilizadas son las normas y las distancias euclidianas. Añadido a lo anterior, se calcula la velocidad que debe tener el robot en cada punto del trayecto. Y, posteriormente, los resultados se envían al robot a través de ROS. Finalmente, se utilizaron cálculos trigonométricos para eliminar rotaciones incongruentes a lo largo del camino.

## 4. RESULTADOS

Durante las distintas fases de la práctica se tuvieron tanto resultados positivos, que ayudaron en la finalización del proyecto, como resultados negativos, los cuales fueron resueltos a medida que avanzaba el proyecto.

Durante la primera fase del proyecto, la de construcción del robot, todo transcurrió de forma normal excepto por un par de componentes defectuosos. Dichos componentes fueron detectados después de la primera verificación de

correcto funcionamiento de componentes. Resulta que uno de los sensores leía datos que no correspondían a lo dado en realidad por el motor. Asimismo, se debió reorganizar el alambrado del puente H. Fuera de eso, esta fase se desarrolló esta fase de forma satisfactoria.

En la segunda fase del proyecto se tuvo que reconfigurar los XBee pues estos estaban con la configuración utilizada en la práctica 3. El contratiempo más importante fue el mal funcionamiento de los XBee. En múltiples ocasiones fallaba la escritura de información inicial. Además, una vez configurados los XBee, había interferencias con los XBee de los demás compañeros del laboratorio. Se podía escuchar mensajes enviados por otros equipos.

También hubo problemas con el código que permitía la ejecución de acciones una vez recibido información. Se recomienda al lector no olvidar verificar si la sentencia *Xbee.isAvailable()* regresa el resultado correcto durante la ejecución. También es importante mencionar que, durante la interconexión con ROS, el baudaje entre los XBee era erróneo. Para la configuración del XBee con ROS, ya no es necesario *Xbee.isAvailable()*. Las demás etapas de esta fase fueron desarrolladas sin mayores contratiempos.

Los inconvenientes de la tercer fase recayeron principalmente en cuestiones de corrección del código y ajusten en valores constantes de las funciones heurísticas. Asimismo, hubo problemas cuando la ruta propuesta se encontraba muy cerca de uno de los obstáculos en la simulación. Al estar cerca la ruta pasaba por encima del obstáculo. Para solucionar dicho problema se pensó en recalcular la ruta cada vez que el robot realizaba algún movimiento, pero ello resultaba en el comportamiento errático del algoritmo. Es decir, la ruta no se mantenía constante y en ocasiones era incongruente con el objetivo de llegada. Entonces se decidió calcular la ruta una única vez y que el robot la siguiera de inicio a fin. Aparte de lo ya mencionado, se procedió en esta fase sin más problemas.

## 5. CONCLUSIONES

### 5.1. Alberto Acosta

Los resultados de esta práctica fueron satisfactorios a pesar de que fue necesario mucho más tiempo del calculado para terminar. Al principio hubo complicaciones para comprender las conexiones físicas, sin embargo, a medida que avanzamos con su construcción y programación, fuimos comprendiendo cada parte necesaria. Al final, los problemas surgieron cuando intentamos hacer que todos los cálculos fueran realizados en ROS para que Arduino realizara los cálculos mínimos necesarios con el fin de que respondiera lo más rápido posible. Al final se logró el objetivo de calcular la ruta óptima que esquivase los obstáculos, el cálculo del cambio de ángulo para las llantas, aunque nos hubiera gustado que las conexiones entre XBee fueran más sencillas y sin tantas complicaciones.

### 5.2. Francisco López

A pesar de las dificultades con las que nos enfrentamos, al final logramos cumplir con el objetivo del proyecto. Al realizarlo, comprendimos verdaderamente el funcionamiento de una máquina autónoma, no solo viendo la parte del

software que hace que piense, sino también el hardware que permite la movilidad del vehículo. Pudimos poner a prueba todos los conocimientos que adquirimos a lo largo del semestre en la materia, y al mismo tiempo adquirir nuevos.

Algo con lo que nos podemos quedar tras la elaboración de esta práctica es que es complicado trabajar con tecnología inalámbrica, pues su inestabilidad nos causó múltiples inconvenientes que absorbieron la mayoría del tiempo. Por otro lado, ROS resultó ser muy útil para trabajar junto con el Arduino para poder transmitir mensajes. Sin duda esta herramienta nos será útil en el futuro.

### 5.3. Fernando Peña

A lo largo del proyecto final del laboratorio de Principios de Mecatrónica fui capaz de observar la unificación de la mayoría de los temas y los conceptos discutidos tanto en la clase de teoría como en las clases de laboratorio. El proyecto final me ayudó a comprender como se interconectan todos los experimentos implementados a lo largo de las prácticas. Conocimientos que van desde el armado de los componentes físicos del robot hasta la transmisión de datos entre los controladores XBee del mismo.

También pude dar un sentido práctico a los conocimientos no solo de la clase de teoría de Principios de Mecatrónica sino también a la de Redes de Computadora. En el caso de Principios el funcionamiento de los motores de DC aunado a la teoría de control del mismo fueron indispensables en el desarrollo del proyecto. Y, finalmente, para el caso de Redes, pude apreciar de forma práctica el protocolo ZigBee al utilizar XBees. En conclusión, esta ha sido una materia que permite la unificación de la parte práctica con la parte teórica en diversos ámbitos de la ingeniería.

## 6. ROL O PAPEL

1. **Alberto:** Revisar puntos a checar de la práctica acerca del lenguaje C++ y de ROS y realizar del reporte.
2. **Francisco:** Realizar el código en C++ para la funcionalidad de ROS.
3. **Fernando:** Realizar el reporte y apoyo en las conexiones de ROS desde distintas máquinas.

## 7. FUENTES CONSULTADAS

### REFERENCIAS

- [1] Arduino. Introduction to arduino. [Online]. Available: <https://www.arduino.cc/en/Guide/Introduction>
- [2] O. source Robotics Foundation. Ros introducion. [Online]. Available: <http://wiki.ros.org/ROS/Introduction>
- [3] F. Semiconductor. H21a1, h21a2 and h21a3 phototransistor optical interrupter switches. [Online]. Available: <http://www.robotstorehk.com/H21A1.pdf>
- [4] Arduino. Xbee shield. [Online]. Available: <https://www.arduino.cc/en/Main/ArduinoXbeeShield>
- [5] S. Herman, *Industrial Motor Control*, illustrated ed. delmar Cengage Learning, 2009.
- [6] F. Mecafenix. Puente h para control de motores. [Online]. Available: <http://www.ingmecafenix.com/electronica/puente-h-control-motores/>