**UNIVERSIDAD DE CONCEPCIÓN**

FACULTAD DE INGENIERÍA

DEPARTAMENTO DE INGENIERÍA ELÉCTRICA



**549305-1 – Taller de aplicación TIC ll**

Profesor Vincenzo Caro Fuentes

**Proyecto final**

**Vehículo “anti-fuego”**

Bastián Baeza Retamal

Pablo Olguin Molina

**Introducción**

Arduino es una plataforma de hardware open source basada en una placa con un microcontrolador y IDE amigable, que permite la programación de dicho microcontrolador, su diseño flexible y accesible facilita la creación y diseño de prototipo para proyectos de electrónica. El gran alcance que ha logrado Arduino se debe principalmente a sus precios, diferentes modelos existentes y su capacidad para integrarse con otros dispositivos y plataformas. En el ámbito de las TIC, Arduino permite el desarrollo de una amplia variedad de aplicaciones, desde la automatización del hogar y la robótica hasta la creación de dispositivos IoT, sistemas de monitoreo en tiempo real y soluciones educativas interactivas.

Para el Proyecto final del ramo T.I.C II, se nos encargó crear un dispositivo/aplicación para Arduino original y creativa, para ello tuvimos a disposición todas las herramientas prácticas entregadas durante el curso, nuestro kit de sensores y herramientas especificadas proporcionadas por el profesor.

Con lo anterior en cuenta, nuestro proyecto consiste en un vehículo “anti- fuego” el cual posee dos motores (ruedas) y una rueda fija, que permite que las se pueda mover libremente, para esto utilizamos un puente H, el cual es el encargado de controlar y redirigir la potencia entregada por nuestra fuente de energía (baterías) hacia los motores. Para apagar el fuego, optamos por un sistema consistente por una bomba y un servo motor, los cuales se complementan de tal manera que, la bomba impulsa el agua a través de una manguera acoplada al motor, y este, permite que el agua haga un barrido en la zona deseada.

**Tabla de Contenidos**

[Descripción de los implementos utilizados 1](#_Toc171200695)

[1.1 Sensores y dispositivos implementados 1](#_Toc171200696)

[Actividad n°1. Vehículo Anti-Fuego 2](#_Toc171200697)

[Ítem 1.1 2](#_Toc171200698)

[1.1 Esquema de circuitos 2](#_Toc171200699)

[1.2 Resultados y comentarios 3](#_Toc171200700)

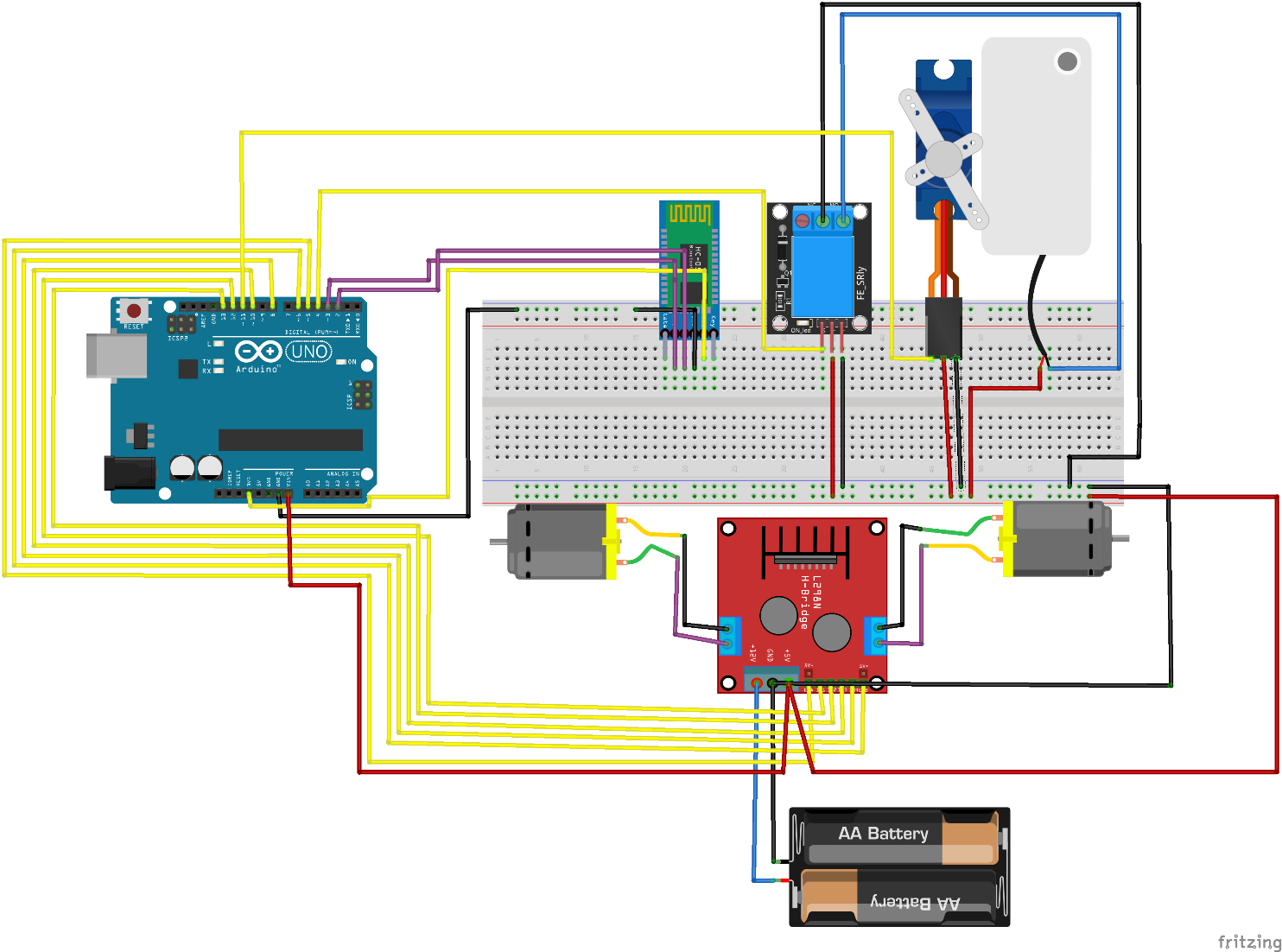
## Descripción de los implementos utilizados

### Sensores y dispositivos implementados

* Servomotor SG90
* KY-019 Relay 5V
* Puente-H L298N
* DC motor 5V x2
* Battery pack 11.1V
* Bluetooth HC-05
* Arduino Uno Rev 3
* Water Pump

# Vehículo Anti-Fuego

### 1.1 Esquema de circuitos

 Fig 1.1 Esquema en fritzing

Diagrama

Descripción generada automáticamente

Fig 1.2 Diagrama funcional del proyecto

### 1.2 Resultados y comentarios

El modo de uso de nuestro automóvil es el siguiente:  
Primeramente se debe energizar el circuito (conectando nuestro pack de baterías), así, se encienden el Arduino, el receptor bluetooth y el servo motor (girando en bucle desde 45° a 90°) de manera simultánea, en este momento, el usuario deberá conectarse al módulo bluetooth con su teléfono y manejar el carro con la aplicación Arduino RC, es en este momento el usuario puede mover el auto a la posición deseada, para luego presionar el botón X y disparar el chorro de agua al objetivo.

Como idea original, nuestro proyecto buscaba crear un vehículo capaz de detectar objetos delante de él (con un sensor de ultrasonido que hacia un paneo, gracias a un servomotor) y capaz de dispararle agua a un objetivó, pero encontramos dificultades cuando quisimos implementar ambas características, la implementación de una interfaz que desplegara la información recopilada (a tiempo real) por el sensor de ultrasonido requería el uso de un cable conectado a un computador, este hecho no nos permitiría hacer el vehículo Wireless, que era una de las claves de nuestro proyecto, por esto mismo, decidimos descartar la idea del radar y solo implementar la bomba de agua. El sistema de agua se instaló dentro del mismo carro, la bomba se colocó dentro del contenedor y de este salen los cables (conexiones) y una manguera, que es la salida del agua.

El final de la manguera, que es la salida, está acoplado al servomotor, que está constantemente cambiando de estado, entre los 45° y los 90°, para lograr hacer un paneo del área objetivo y apagar de manera más efectiva el fuego.

Una de las complicaciones que encontramos, fue que, al momento de dirigir el auto, a veces cuando presionábamos un botón, y el vehículo se quedaba atascado en la última instrucción que recibió. Las veces que nos pasó, paso con cualquier instrucción, ya sea hacia atrás, izquierda, derecha o hacia adelante, e incluso, a veces simplemente se volvía loco y empezaba a moverse sin rumbo, y la única manera de detenerlo era desconectándole la batería. Creemos que esto es por la alta latencia con la que recibe y envía datos el sensor HC-05.

La otra gran complicación que tuvimos fue el contenedor para el agua, lo único que pudimos conseguir fue una botella de colonia, que pegamos con silicona caliente a la base del vehículo. El problema de esto es que, si el vehículo llega volcarse, cosa a la que es muy propenso, debido al problema descrito en el párrafo anterior, el agua se puede derramar y estropear toda la electrónica. Finalmente, decidimos cerrar el contenedor de agua y dejar solo unas salidas para el tubo y el cable, y una entrada con una tapa (plastilina) para poder introducir el agua.

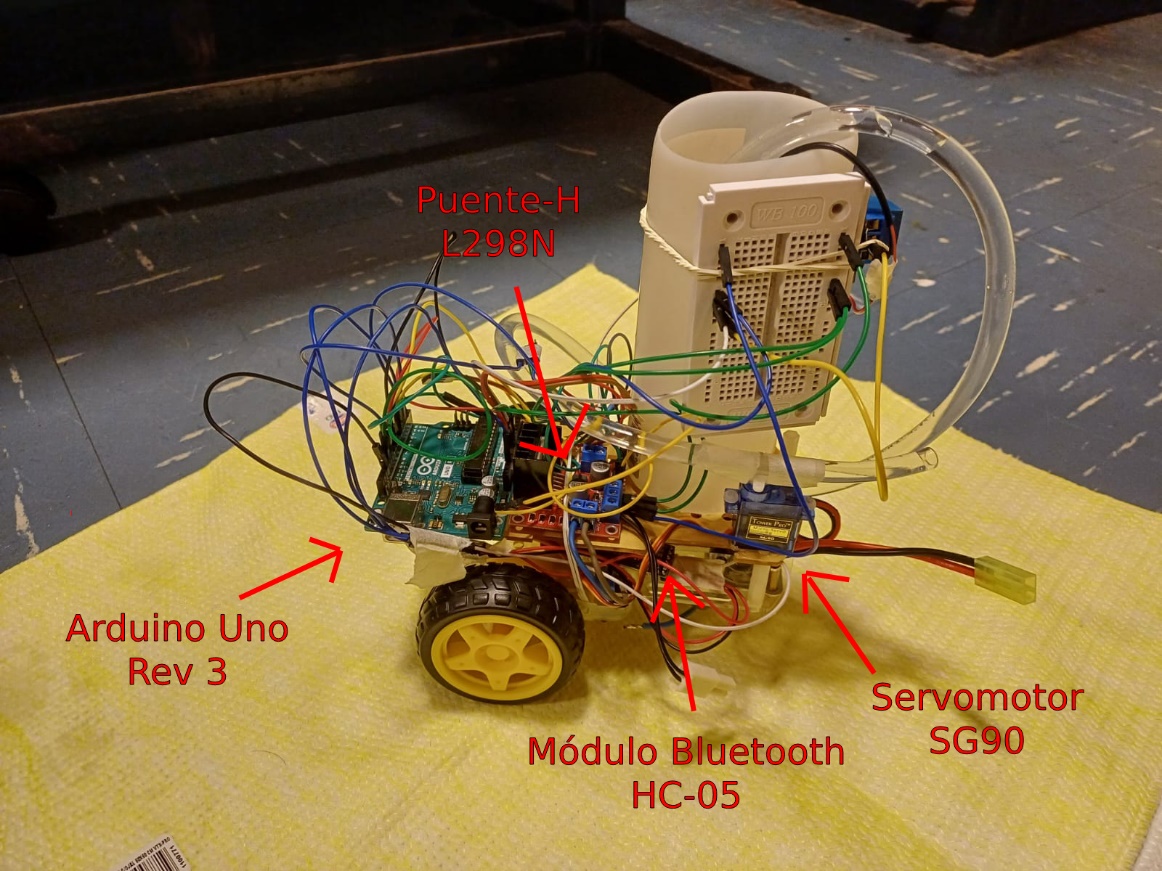


Fig 1.3 Partes rotuladas del prototipo (1)

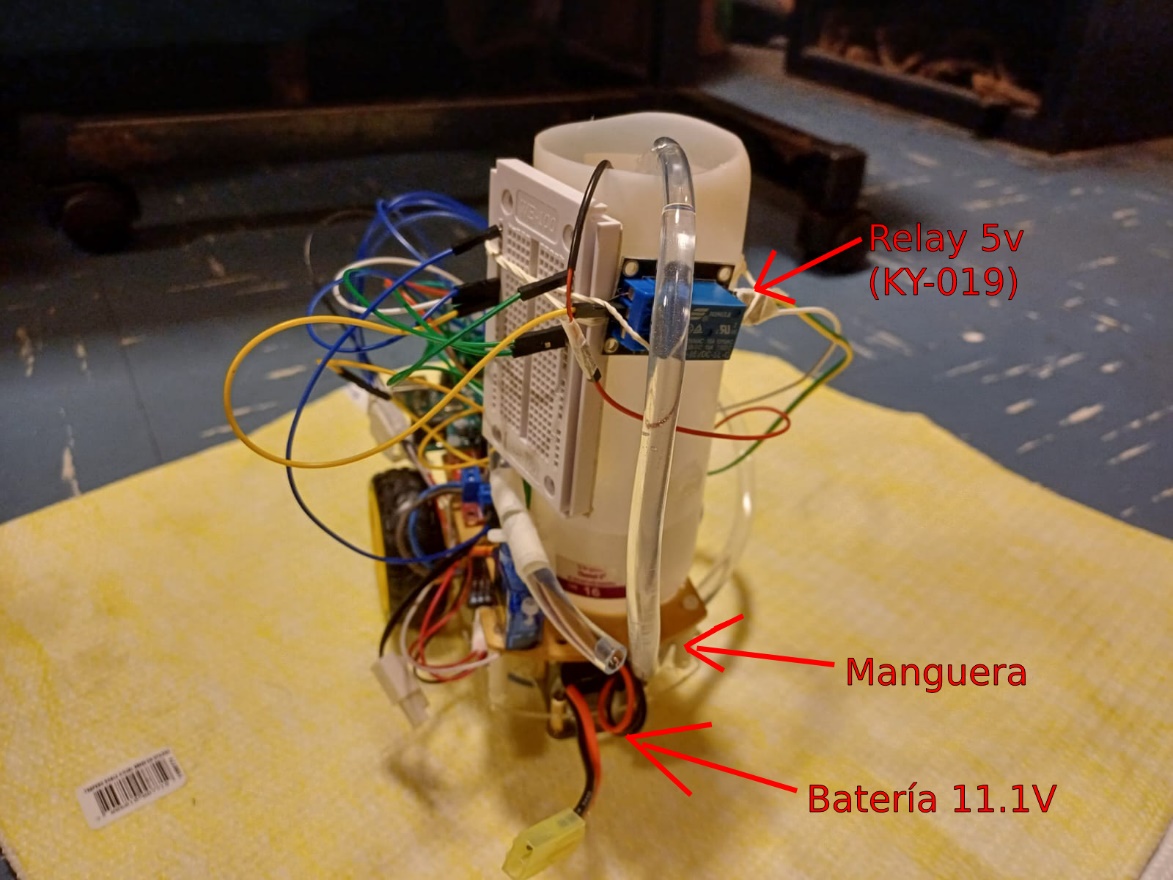


Fig 1.4 Partes rotuladas del prototipo (2)

#### Análisis del Código del proyecto

Al inicio del código, comenzamos incluyendo las librerías necesarias, definimos los pines y declaramos las variables a utilizar.

|  |
| --- |
| // Incluimos librerías  #include <SoftwareSerial.h>  #include <Servo.h>  Servo motor;  int posicion = 0;  // Instanciamos objeto SoftwareSerial para controlar el bluetooth  SoftwareSerial hc06(2, 3);  // Definición pines EnA y EnB para el control de la velocidad  int VelocidadMotor1 = 5;  int VelocidadMotor2 = 6;  // Definición de los pines de control de giro de los motores In1, In2, In3 e In4 , los In son los pines del puente H el cosito del voltaje  int Motor1A = 13;  int Motor1B = 12;  int Motor2C = 8;  int Motor2D = 10;  // Definición de la velocidad inicial de los motores  int speedMotor = 80;  int motorpump = 4;  // Variable para capturar el comando que llega desde la app, esto estaba en internet  String cmd = "";  // declarar el servo para que funciono  Servo servoMotor; |

En la sección void setup, iniciamos el monitor serial e inicializamos los pines, ademas inicializamos la comunicación serial entre el Arduino y el módulo Bluetooth HC-05.

|  |
| --- |
| void setup() {  Serial.begin(9600);  hc06.begin(9600);  motor.attach(11);  pinMode(Motor1A, OUTPUT);  pinMode(Motor1B, OUTPUT);  pinMode(Motor2C, OUTPUT);  pinMode(Motor2D, OUTPUT);  pinMode(VelocidadMotor1, OUTPUT);  pinMode(VelocidadMotor2, OUTPUT);  pinMode(motorpump, OUTPUT);  analogWrite(VelocidadMotor1, speedMotor);  analogWrite(VelocidadMotor2, speedMotor);  servoMotor.attach(9);  } |

* En la sección void loop comenzamos con el movimiento del servo motor, que mueve la salida de agua del carro continuamente.

|  |
| --- |
| void loop() {  motor.write(90);  delay(500);  motor.write(135);  delay(500);  motor.write(90);  delay(500); |

Luego, se inicia un ciclo while que solo se activa si el módulo Bluetooth está recibiendo datos, dentro de este ciclo hay una serie de condicionales if y elif que llaman a ciertas funciones dependiendo de la señal que se reciba, estas permiten que el auto se mueva.

|  |
| --- |
| // Leemos los datos recibidos  while (hc06.available() > 0) {  cmd += (char)hc06.read();  }  // Programamos cada una de las acciones a realizar según el comando recibido  if (cmd != "") {  cmd = cmd[0];  if (cmd == "S") {  stopCar();  } else if (cmd == "F") {  moveForwardCar();  } else if (cmd == "B") {  moveBackwardsCar();  } else if (cmd == "L") {  turnLeftCar();  } else if (cmd == "R") {  turnRightCar();  } else if (cmd == "G") {  moveForwardLeft();  } else if (cmd == "I") {  moveForwardRight();  } else if (cmd == "H") {  moveBackwardsLeft();  } else if (cmd == "J") {  moveBackwardsRight();  } |

Dentro del ciclo while también se llaman a funciones que modifican la velocidad de los motores, debido a dificultades técnicas, la implementación de esta característica no fue exitosa, por lo que configuramos el auto para que utilizara la velocidad máxima en todo momento.

|  |
| --- |
| else if (cmd == "0") {  speedMotor = 80;  analogWrite(VelocidadMotor1, speedMotor);  analogWrite(VelocidadMotor2, speedMotor);  } else if (cmd == "1") {  speedMotor = 90;  analogWrite(VelocidadMotor1, speedMotor);  analogWrite(VelocidadMotor2, speedMotor);  } else if (cmd == "2") {  speedMotor = 100;  analogWrite(VelocidadMotor1, speedMotor);  analogWrite(VelocidadMotor2, speedMotor);  } else if (cmd == "3") {  speedMotor = 110;  analogWrite(VelocidadMotor1, speedMotor);  analogWrite(VelocidadMotor2, speedMotor);  } else if (cmd == "4") {  speedMotor = 120;  analogWrite(VelocidadMotor1, speedMotor);  analogWrite(VelocidadMotor2, speedMotor);  } else if (cmd == "5") {  speedMotor = 130;  analogWrite(VelocidadMotor1, speedMotor);  analogWrite(VelocidadMotor2, speedMotor);  } else if (cmd == "6") {  speedMotor = 140;  analogWrite(VelocidadMotor1, speedMotor);  analogWrite(VelocidadMotor2, speedMotor);  } else if (cmd == "7") {  speedMotor = 150;  analogWrite(VelocidadMotor1, speedMotor);  analogWrite(VelocidadMotor2, speedMotor);  } else if (cmd == "8") {  speedMotor = 160;  analogWrite(VelocidadMotor1, speedMotor);  analogWrite(VelocidadMotor2, speedMotor);  } else if (cmd == "9") {  speedMotor = 170;  analogWrite(VelocidadMotor1, speedMotor);  analogWrite(VelocidadMotor2, speedMotor);  } else if (cmd == "q") {  speedMotor = 180;  analogWrite(VelocidadMotor1, speedMotor);  analogWrite(VelocidadMotor2, speedMotor);  } |

Aquí se define el código de tal manera que cuando se presiona el botón X en el joystick, la bomba expulse agua durante 5 segundos.

|  |
| --- |
| else if (cmd == "X") {  analogWrite(motorpump,255);  delay(5000);  analogWrite(motorpump,0);  delay(1000);  } |

Finalmente, definimos todas las funciones que controlan el carro, por ejemplo, la función stopCar apaga los motores del carro y lo deja inmóvil, por el otro lado, la función moveForwardCar hace que los 2 motores se muevan hacia adelante.

|  |
| --- |
| void stopCar() {  // Paramos el carrito  digitalWrite(Motor1A, LOW);  digitalWrite(Motor1B, LOW);  digitalWrite(Motor2C, LOW);  digitalWrite(Motor2D, LOW);  }  void turnRightCar() {  // Configuramos sentido de giro para girar a la derecha  digitalWrite(Motor1A, HIGH);  digitalWrite(Motor1B, LOW);  digitalWrite(Motor2C, LOW);  digitalWrite(Motor2D, LOW);  }  void turnLeftCar() {  // Configuramos sentido de giro para girar a la izquierda  digitalWrite(Motor1A, LOW);  digitalWrite(Motor1B, LOW);  digitalWrite(Motor2C, LOW);  digitalWrite(Motor2D, HIGH);  }  void moveForwardCar() {  // Configuramos sentido de giro para avanzar  digitalWrite(Motor1A, HIGH);  digitalWrite(Motor1B, LOW);  digitalWrite(Motor2C, LOW);  digitalWrite(Motor2D, HIGH);  }  void moveBackwardsCar() {  // Configuramos sentido de giro para retroceder  digitalWrite(Motor1A, LOW);  digitalWrite(Motor1B, HIGH);  digitalWrite(Motor2C, HIGH);  digitalWrite(Motor2D, LOW);  }  void moveForwardLeft() {  // Giramos a la izquierda mientras avanza  analogWrite(VelocidadMotor2, speedMotor + 60);    digitalWrite(Motor1A, HIGH);  digitalWrite(Motor1B, LOW);  digitalWrite(Motor2C, LOW);  digitalWrite(Motor2D, HIGH);  delay(20);  analogWrite(VelocidadMotor2, speedMotor);  }  void moveForwardRight() {  // Giramos a la derecha mientras avanza  analogWrite(VelocidadMotor1, speedMotor + 60);    digitalWrite(Motor1A, HIGH);  digitalWrite(Motor1B, LOW);  digitalWrite(Motor2C, LOW);  digitalWrite(Motor2D, HIGH);    delay(20);  analogWrite(VelocidadMotor1, speedMotor);  }  void moveBackwardsLeft() {  // Giramos a la izquierda mientras retrocede  analogWrite(VelocidadMotor2, speedMotor + 60);    digitalWrite(Motor1A, LOW);  digitalWrite(Motor1B, HIGH);  digitalWrite(Motor2C, HIGH);  digitalWrite(Motor2D, LOW);  delay(20);  analogWrite(VelocidadMotor2, speedMotor);  }  void moveBackwardsRight() {  // Giramos a la derecha mientras retrocede  analogWrite(VelocidadMotor1, speedMotor + 60);    digitalWrite(Motor1A, LOW);  digitalWrite(Motor1B, HIGH);  digitalWrite(Motor2C, HIGH);  digitalWrite(Motor2D, LOW);  delay(20);  analogWrite(VelocidadMotor1, speedMotor);  } |

Conclusión

Durante el desarrollo del "Vehículo Anti-Fuego" aplicamos todas las técnicas y conocimientos adquiridos en el curso T.I.C II. A través de este proyecto, logramos integrar con éxito varios componentes para crear un dispositivo funcional capaz de extinguir incendios a pequeña escala. A pesar de enfrentar dificultades, como la latencia en la transmisión de señales desde el celular y la estabilidad del contenedor de agua, pudimos implementar soluciones efectivas que cumplieron con nuestro objetivo.

En el futuro, creemos que este tipo de tecnología puede ser particularmente útil en el contexto de seguridad y prevención de incendios, especialmente en entornos donde el acceso humano es difícil o peligroso. Para mejorar su aplicación, el chasis del vehículo debería ser rediseñado con materiales más sólidos y capaces de soportar y disipar altas temperaturas así también en vez de tener 2 ruedas debería tener cuatro ruedas lo cual aumentaría la estabilidad notoriamente. Con estas mejoras, este proyecto tiene el potencial de ser una herramienta valiosa para la lucha contra incendios en escenarios de riesgo elevado.