SEGUIDOR LEGO

[ROBOT] & [SCIENCE]

CHALLENGER

DESCRIPCIÓN GENERAL

Diseñar y desarrollar un robot completamente autónomo, que sea capaz de recorrer un circuito de carreras formado por una línea negra en un fondo blanco. Gana el robot que complete el circuito en el menor tiempo posible.

1. CARACTERÍSTICAS TÉCNICAS DEL ROBOT

1.1 Las dimensiones del robot no podrán exceder 22.5 cm de ancho x 25 cm de largo de base, la altura del robot no está limitada. El peso máximo no está limitado.

1.2 El accionamiento del robot se realizará de forma manual o radio controlada cuando se indique la salida. Los robots no pueden tener partes en movimiento (como las ruedas) antes de la señal de salida.

1.3 Si el accionamiento del robot se hace de forma inalámbrica, en este caso, el control de activación debe ser visible para el juez y este sistema sólo debe activar y desactivar el robot durante la competencia. Una vez activado el robot, este debe comportarse de forma completamente autónoma.

1.4 Limitaciones:

- El único sistema de comunicación permitido con el robot es el de encendido y apagado por vía inalámbrica.

- El robot debe comportarse de forma completamente autónoma durante todo el recorrido.

1.5 El robot deberá estar preparado para trabajar bajo condiciones de luz variadas.

1.6 Los robots deben constar de hardware y software diseñado por los participantes y/o tarjetas de desarrollo, el tipo de controlador del sistema es libre, se prohíbe un robot conformado únicamente con kits de desarrollo para robots.

1.7 Se permitirá realizar un cambio al hardware o el software en los robots por los competidores sólo al término de la primera ronda.

2. CARACTERÍSTICAS DEL ÁREA DE TRABAJO DEL ROBOT.

2.1 Las características de la pista se dará a conocer 5 días antes de la inauguración del evento a través de la página oficial del evento (https://artilrobotics.com/robot&science)

2.2 En el inicio y el fin del recorrido estarán indicadas con líneas perpendiculares a la derecha con respecto de la línea de trayectoria, siguiendo el sentido del recorrido.

2.3 Las características principales de la pista donde se realizará la competencia son las que se muestran a continuación:

- La pista será en madera para la categoría loop, velocidad con turbina y sin turbina.

- En la pista no habrá cruces de línea.

- La aproximación más cercana de la línea de curso a los bordes de la pista será de no menos de 15 cm, medidos desde el centro de la línea.

2.4 No se garantiza una iluminación especial.

2.5 Se recomienda a los participantes tener en cuenta cada aspecto de la pista para luego no tener ningún inconveniente, si resultase el caso de tener algún inconveniente dar aviso a los respectivos jueces antes de la participación.

3. HOMOLOGACIÓN.

3.1 Se verificará que se cumplan satisfactoriamente las especificaciones técnicas del robot, tales como las dimensiones y verificación que no sea un robot comercial.

3.2 Se realizará una vuelta de prueba sobre la pista, verificando con esto el correcto funcionamiento y las limitaciones que se mencionaron en el apartado de CARACTERÍSTICAS TÉCNICAS DEL ROBOT.

4. DESARROLLO DE LA COMPETENCIA.

4.1 Todos los seguidores de línea deberán tener sus baterías completamente cargadas antes de la competencia, no se permitirá la recarga de estas entre cada carrera.

4.2 Los robots serán recolectados antes de iniciar la ronda de eliminación, esto con el fin de evitar el cambio de hardware, software y recargada de baterías entre turnos.

4.3 Los seguidores de línea estarán situados y resguardados en el área de jueces. Los prototipos serán entregados a sus respectivos dueños al finalizar cada ronda.

4.4 La competencia inicia en el momento que el seguidor de línea cruce la línea de salida, en este momento se comenzará a tomar el tiempo de recorrido.

4.5 El tiempo de recorrido será detenido cuando el robot cruce la línea de meta, este tiempo será almacenado.

4.6 Cada robot tendrá un tiempo máximo de 3 minutos para finalizar totalmente la pista. Así mismo, tendrá 2 oportunidades para lograr el objetivo, en caso de finalizar en las dos ocasiones la pista, se almacenará el menor tiempo realizado por el robot.

4.7 El robot está obligado a permanecer dentro de la pista y seguir la trayectoria marcada durante toda la carrera. Si el vehículo se sale de la pista y vuelve de nuevo al mismo punto en la pista por sí mismo, puede continuar la carrera. Si el vehículo se sale de la pista completamente o permanece inmóvil durante 5 segundos, la carrera se dará por terminada inmediatamente.

4.8 El operador del robot no podrá tocar al vehículo mientras este se encuentre haciendo la trayectoria, en caso de que esto suceda, el robot será descalificado. Solo podrá tocarlo cuando inicie o termine el recorrido.

4.9 Si ninguno de los equipos puede completar la trayectoria, el ganador será determinado por la distancia recorrida en el menor tiempo.

4.10 Es OPCIÓN de la organización realizar la competencia en tres fases:

- Primera fase: a) La competencia se realizará primeramente de manera individual, es decir, un robot tendrá que realizar totalmente la trayectoria marcada, el tiempo mínimo de la mejor vuelta será almacenada.

b) Todos los robots participantes ejecutarán esta acción, con ello se determinarán las posiciones para realizar las eliminatorias. Si es necesario, se eliminarán para esta primera fase a los robots con un tiempo mayor tal que el número de participantes se reduzca a número par.

- Segunda fase: a) Para la segunda fase la competencia será en eliminación directa; es decir, Robot vs. Robot, el robot que haga el menor tiempo pasa a durante los procesos de homologaciones.

4.11 En caso de tener alguna duda sobre esta normativa, comunicarse con el comité organizador. “En caso de haber muchos competidores se hará un análisis para determinar si se va a realizar la competencia en estas dos fases, caso contrario se dará aviso en la página oficial del torneo cinco días antes del torneo.”

5. JUECES

5.1 La figura del juez o los jueces es importante en la competencia, él será el encargado de que las reglas y normas establecidas por el comité organizador en esta categoría sean cumplidas.

5.2 Los jueces para esta competencia serán designados por el comité organizador.

5.3 Los participantes pueden presentar sus objeciones al juez encargado de la categoría antes de que acabe la competencia.

5.4 En caso de duda en la aplicación de las normas, la última palabra la tiene siempre el juez o los jueces encargados.

5.5 En caso de existir una controversia ante la decisión del juez o los jueces, se puede presentar una inconformidad por escrito ante el Consejo de Jueces de “Robot & Science”, una vez terminada la competencia, se evaluaran los argumentos presentados y se tomará decisión al respecto. Esta decisión es inapelable.

Transitorios:

- De no contar con mínimo 3 robots o equipos participantes, la categoría será considerada únicamente como exhibición y solo se realizará la entrega de medallas.
- Los reglamentos están sujetos a modificaciones a fin de mejorar el desempeño de los participantes.
- Todos aquellos sucesos que no se contemplen dentro del presente reglamento, durante la competencia serán resueltos por el Comité Organizador en conjunto con los Jueces sin derecho de apelación.
- Las llaves de cada categoría serán presentadas posterior al cierre de inscripciones.
- Una vez realizada la inscripción del robot no se realizarán devoluciones de dinero
- Los distintivos de cada participante serán entregados durante el desarrollo del evento.

HAZ TU PROPIO CRONÓMETRO, ESTE SERÁ EL CRONÓMETRO OFICIAL DEL EVENTO ROBOT AND SCIENCE CHALLENGER:

ELEMENTOS:

1X ARDUINO UNO

1X LCD1602 LCD Keypad Shield Backlight azul

1x Sensor Proximidad Arduino 5v E18-d80nk

CODIGO:

```
#include <LiquidCrystal.h>

// LCD pin to Arduino
const int pin_RS = 8;
const int pin_EN = 9;
const int pin_d4 = 4;
const int pin_d5 = 5;
const int pin_d6 = 6;
const int pin_d7 = 7;
LiquidCrystal lcd(pin_RS, pin_EN, pin_d4, pin_d5, pin_d6, pin_d7);

const int sensorPin = 12;     // PIN del sensor
int sensorState = HIGH;       // Estado inicial del sensor
int previousState = HIGH;     // Estado previo del sensor
bool startFlag = false;       // Indicador para iniciar o pausar el cronómetro
unsigned long startTime = 0;  // Tiempo de inicio del cronómetro
unsigned long elapsedTime = 0; // Tiempo transcurrido desde el primer pulso del sensor
unsigned long pauseTime = 0;  // Tiempo de pausa del cronómetro
bool pauseFlag = false;       // Indicador para controlar la pausa del cronómetro

void setup() {
  pinMode(sensorPin, INPUT_PULLUP); // Configura el PIN del sensor como entrada con resistencia pull-up interna
  lcd.begin(16, 2);             // Inicializa la pantalla LCD de 16 columnas y 2 filas
  lcd.print("Cronometro Listo");
}

void loop() {
  sensorState = digitalRead(sensorPin); // Lee el estado actual del sensor

  // Comprueba si el estado del sensor ha cambiado
  if (sensorState != previousState) {
    // Si el estado del sensor cambió a LOW y no se ha iniciado el cronómetro, se inicia
```

```
    if (sensorState == LOW && !startFlag && !pauseFlag) {
      startFlag = true;    // Marca el inicio del cronómetro
      startTime = millis();  // Guarda el tiempo actual como tiempo de inicio
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Cronometro Iniciado");
     //delay(1000);  // Espera 1 segundo para mostrar el mensaje
   }
   // Si el estado del sensor cambió a LOW y el cronómetro ya se ha iniciado, se pausa si ha pasado al menos 1 segundo
    else if (sensorState == LOW && startFlag && millis() - startTime >= 1000) {
    if (!pauseFlag) {
       startFlag = false;  // Marca la pausa del cronómetro
       pauseTime = millis() + 10000;  // Guarda el tiempo actual + 10 segundos como tiempo de pausa
      pauseFlag = true;  // Marca la bandera de pausa activa
    lcd.clear();
    lcd.setCursor(0, 0);
                          lcd.print("Cronometro Pausado-WEROBOT");
    lcd.setCursor(0, 1);
    lcd.print("Time: ");
    lcd.print(elapsedTime / 1000);
    lcd.print("s ");
    lcd.print(elapsedTime % 1000);
    lcd.print("ms");
    lcd.print("    ");
    }
   }
   }
  // Si el cronómetro está en marcha, muestra el tiempo transcurrido en la pantalla LCD
  if (startFlag) {
   elapsedTime = millis() - startTime;
   lcd.setCursor(0, 1);
   lcd.print("Time: ");
   lcd.print(elapsedTime / 1000);
   lcd.print("s ");
   lcd.print(elapsedTime % 1000);
   lcd.print("ms");
   lcd.print("    ");
  }
   // Mientras se espera los 10 segundos de pausa, muestra los asteriscos (*) en la pantalla
  if (pauseFlag && millis() <= pauseTime) {
    unsigned long remainingTime = pauseTime - millis();
     int numAsterisks = map(remainingTime, 10000, 0, 0, 16);
   lcd.setCursor(0, 0);
   for (int i = 0; i < numAsterisks; i++) {
    lcd.print("*");
   }
   for (int i = numAsterisks; i < 16; i++) {
    lcd.print(" ");
   }
  }
    // Si ha pasado el tiempo de pausa, desactiva la bandera de pausa y permite reiniciar el cronómetro
  if (pauseFlag && millis() > pauseTime) {
   pauseFlag = false;
   lcd.clear();
```

```
  lcd.setCursor(0, 0);
  lcd.print("Cronometro Listo-WEROBOT");
 }


  previousState = sensorState;  // Guarda el
estado actual del sensor como estado previo
}
```