

# Robot Kiut (MiniSumo)

## Universidad Blas Pascal (Córdoba)



### Integrantes:

Ing. César Osimani (Docente)  
Emiliano Álvarez (Alumno)  
Cristian Novarino (Alumno)

### Datos técnicos del Robot:

- **Placa controladora:** Arduino UNO

- **Alimentación:** - Batería de 3 Celdas 11.1 Volts (opción 1)  
- 5 pilas AA (opción 2)

- **Control de motores DC:** L298

- **Sensores:** - 2 sensores de tacto: adelante y atrás  
- 3 sensores de línea infrarrojos: adelante izquierda, adelante derecha y atrás

- **Llave de alimentación**

- **Pulsador para arranque**

- **A continuación el programa en C++ para el microcontrolador del Arduino**

```
// Velocidades
#define VMAX 150
#define VMED 100
#define VMIN 70

// Pines de control para el motor 1
#define MOTOR1_CTL1 2 // I1
#define MOTOR1_CTL2 4 // I2
#define MOTOR1_PWM 11 // EA

// Pines de control para el motor 2
#define MOTOR2_CTL1 12 // I3
#define MOTOR2_CTL2 8 // I4
#define MOTOR2_PWM 5 // EB
```

```

// Macros utilizadas para indicar que el motor avanza o retrocede
#define MOTOR_DIR_FORWARD 0
#define MOTOR_DIR_BACKWARD 1

// Macro utilizada para indicar qué pin usamos para iniciar el robot
// Entonces, cuando el pin 13 se pone a 5v empieza a contar el retardo reglamentario antes de
empezar a pelear
#define PULSADOR_INICIO 13

// Macro que fija el retardo iniciar antes de empezar a pelear luego de presionar
// el pulsador de inicio que conectaremos al pin 13.
#define RETARDO_REGLAMENTARIO 2000

// Macros para indicar cada uno de los 3 sensores de borde. 2 adelante y uno atrás.
#define SENSOR_BORDE_DELANTERO_IZQUIERDO 7
#define SENSOR_BORDE_DELANTERO_DERECHO 3
#define SENSOR_BORDE_TRASERO 6
#define SENSOR_FRONTAL 9

// Usada como bandera para saber si está iniciada o no la pelea.
boolean juegoIniciado = false;

void setup()
{
    // Configuración de control para el motor 1
    pinMode(MOTOR1_CTL1,OUTPUT);
    pinMode(MOTOR1_CTL2,OUTPUT);
    pinMode(MOTOR1_PWM,OUTPUT);

    // Configuración de control para el motor 2
    pinMode(MOTOR2_CTL1,OUTPUT);
    pinMode(MOTOR2_CTL2,OUTPUT);
    pinMode(MOTOR2_PWM,OUTPUT);

    // Configuramos el pin 13 para leer cuándo se presiona el pulsador para iniciar la pelea
    pinMode(PULSADOR_INICIO, INPUT);

    // Configuramos los pines 10, 11 y 12 para conectar los sensores de borde.
    pinMode(SENSOR_BORDE_DELANTERO_IZQUIERDO, INPUT);
    pinMode(SENSOR_BORDE_DELANTERO_DERECHO, INPUT);
    pinMode(SENSOR_BORDE_TRASERO, INPUT);
    pinMode(SENSOR_FRONTAL, INPUT);
}

// Seteamos la velocidad de algún motor (entre 0 y 255)
void setSpeed(char motor_num, char motor_speed) {
    if (motor_num == 1) {
        analogWrite(MOTOR1_PWM, motor_speed);
    }
    else {
        analogWrite(MOTOR2_PWM, motor_speed);
    }
}

// Iniciamos algún motor en la dirección que indicamos aquí y con la velocidad que seteamos con
setSpeed previamente
void motorStart(char motor_num, byte direction) {
    char pin_ctl1;
    char pin_ctl2;

    if (motor_num == 1) {
        pin_ctl1 = MOTOR1_CTL1;
        pin_ctl2 = MOTOR1_CTL2;
    }
    else {
        pin_ctl1 = MOTOR2_CTL1;
        pin_ctl2 = MOTOR2_CTL2;
    }
}

```

```

switch (direction) {
  case MOTOR_DIR_FORWARD:
    {
      digitalWrite(pin_ctl1,LOW);
      digitalWrite(pin_ctl2,HIGH);
    }
    break;

  case MOTOR_DIR_BACKWARD:
    {
      digitalWrite(pin_ctl1,HIGH);
      digitalWrite(pin_ctl2,LOW);
    }
    break;
}
}

// Detenemos algún motor
void motorStop(char motor_num) {
  setSpeed(motor_num, 0);

  if (motor_num == 1) {
    digitalWrite(MOTOR1_CTL1, HIGH);
    digitalWrite(MOTOR1_CTL2, HIGH);
  }
  else {
    digitalWrite(MOTOR2_CTL1,HIGH);
    digitalWrite(MOTOR2_CTL2,HIGH);
  }
}

// Loop principal
void loop() {

  // Si el juego nunca se inició se ingresará a este if.
  if (juegoIniciado == false) {
    while (digitalRead(PULSADOR_INICIO) != HIGH); // Espera dentro de este bucle hasta que se
    presione el pulsador conectado al pin 13

    delay(RETARDO_REGLAMENTARIO); // Luego de salir del bucle espera los RETARDO_REGLAMENTARIO
    segundos para iniciar la pelea

    juegoIniciado = true; // Esto es para que no vuelva a entrar a este if si se inició el juego

    setSpeed(1, VMED);
    setSpeed(2, VMED);

    motorStart(1, MOTOR_DIR_FORWARD);
    motorStart(2, MOTOR_DIR_FORWARD);
  }

  if(digitalRead(SENSOR_FRONTAL) == HIGH){

    setSpeed(1, VMAX);
    setSpeed(2, VMAX);
    motorStart(1, MOTOR_DIR_FORWARD);
    motorStart(2, MOTOR_DIR_FORWARD);

    while( (digitalRead(SENSOR_FRONTAL) == HIGH) || (digitalRead(SENSOR_FRONTAL) == HIGH) );
  }

  if(digitalRead(SENSOR_BORDE_TRASERO) == HIGH){

    setSpeed(1,VMIN);
    setSpeed(2,VMED);
    motorStart(1, MOTOR_DIR_FORWARD);
    motorStart(2, MOTOR_DIR_FORWARD);

    while(digitalRead(SENSOR_BORDE_TRASERO) == HIGH);
  }
}

```

```
if(digitalRead(SENSOR_BORDE_DELANTERO_IZQUIERDO) == HIGH){  
  
    setSpeed(1,VMED);  
    setSpeed(2,VMIN);  
    motorStart(1, MOTOR_DIR_BACKWARD);  
    motorStart(2, MOTOR_DIR_BACKWARD);  
  
    while(digitalRead(SENSOR_BORDE_DELANTERO_IZQUIERDO) == HIGH);  
}  
  
if(digitalRead(SENSOR_BORDE_DELANTERO_DERECHO) == HIGH){  
  
    setSpeed(1,VMIN);  
    setSpeed(2,VMED);  
    motorStart(1, MOTOR_DIR_BACKWARD);  
    motorStart(2, MOTOR_DIR_BACKWARD);  
  
    while(digitalRead(SENSOR_BORDE_DELANTERO_DERECHO) == HIGH);  
}  
}
```