

ME4250-1

PROPUESTA DE ROBOT AUTO-BALANCÍN

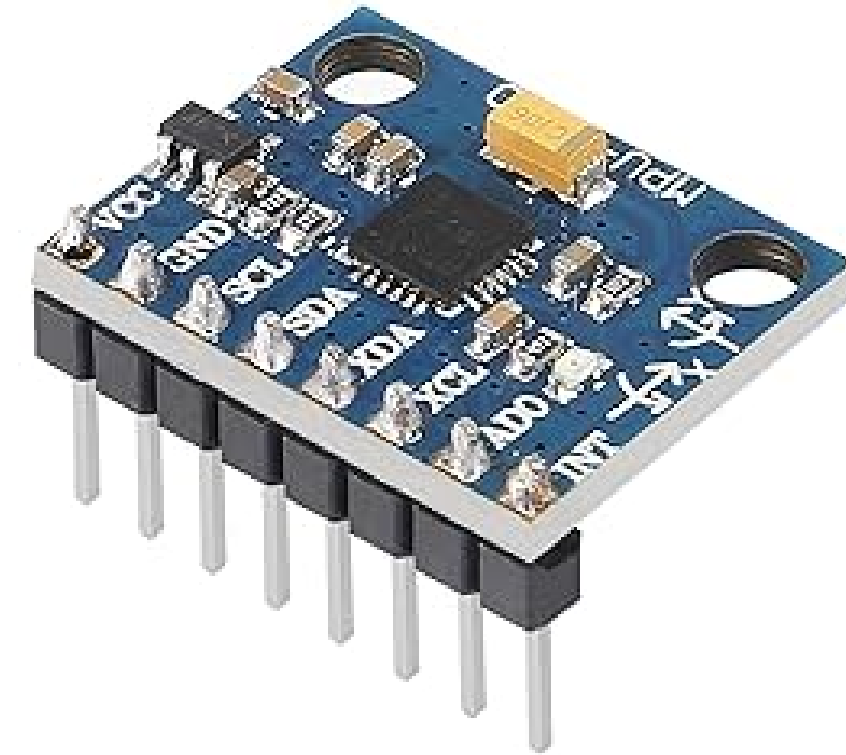


GRUPO 3

Integrantes:
Marina Olmedo.
Félicie Nguyen.
Joaquin Poblete Morales.
Enrique Rebolledo.
Pablo Varetto.

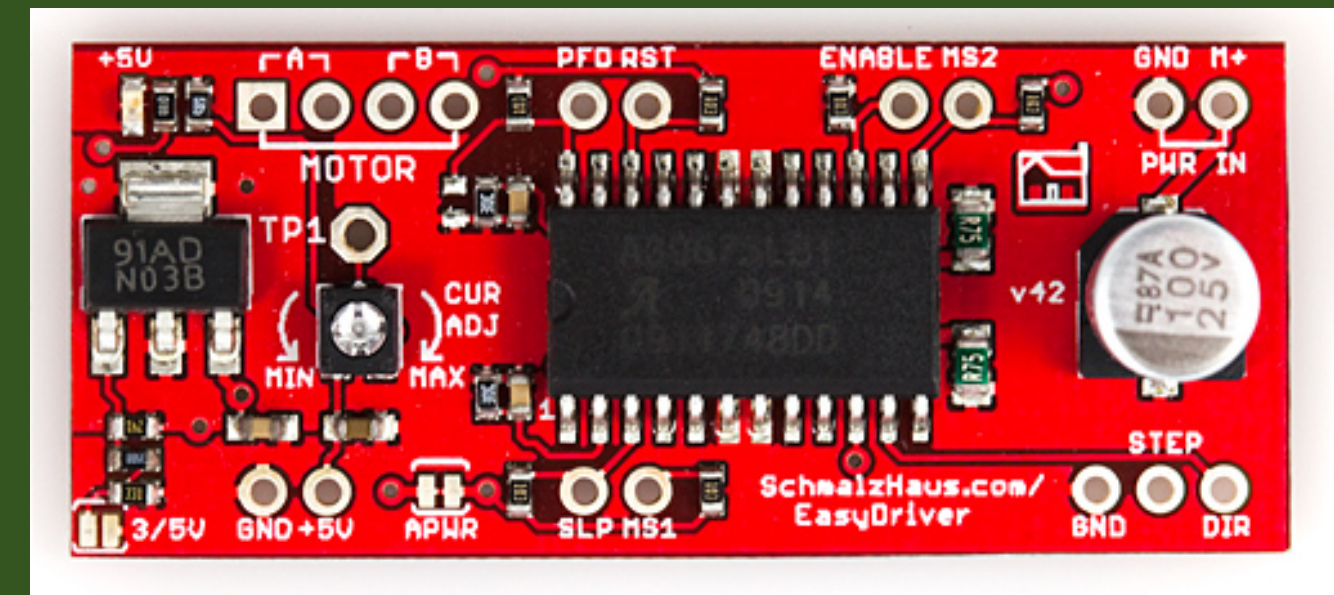
SENSOR (GY521)

- Acelerómetro + giroscopio.
- Puede medir temperatura, pero no es su función principal.
- Funciona con los 5V del Arduino Uno.
- Método más efectivo.

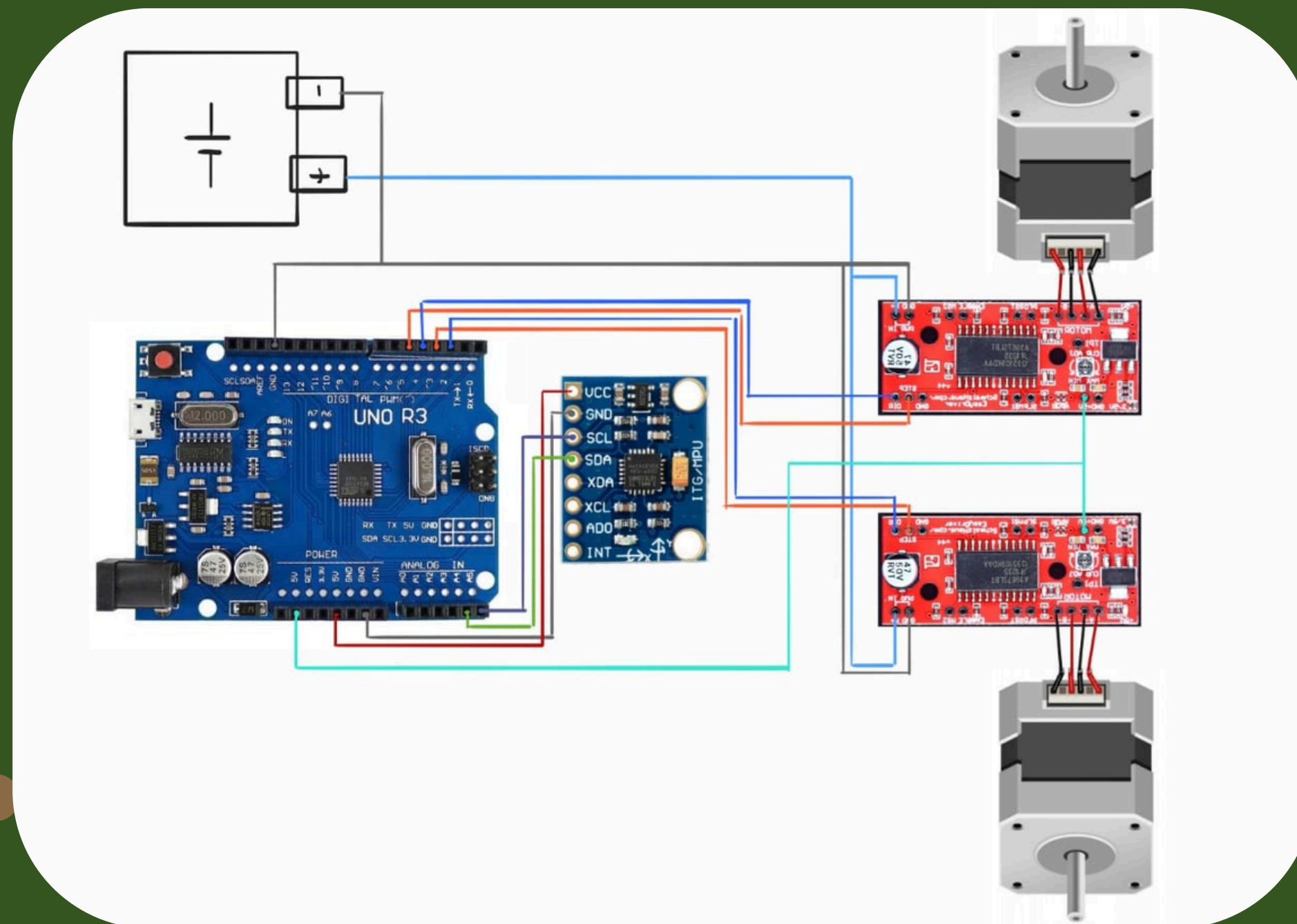


DRIVERS

- Easydriver A3967
- Compatibles con 5V.
- Corriente ajustable, ideal para motores paso a paso.
- Driver accesible, proporcionado por el equipo docente.

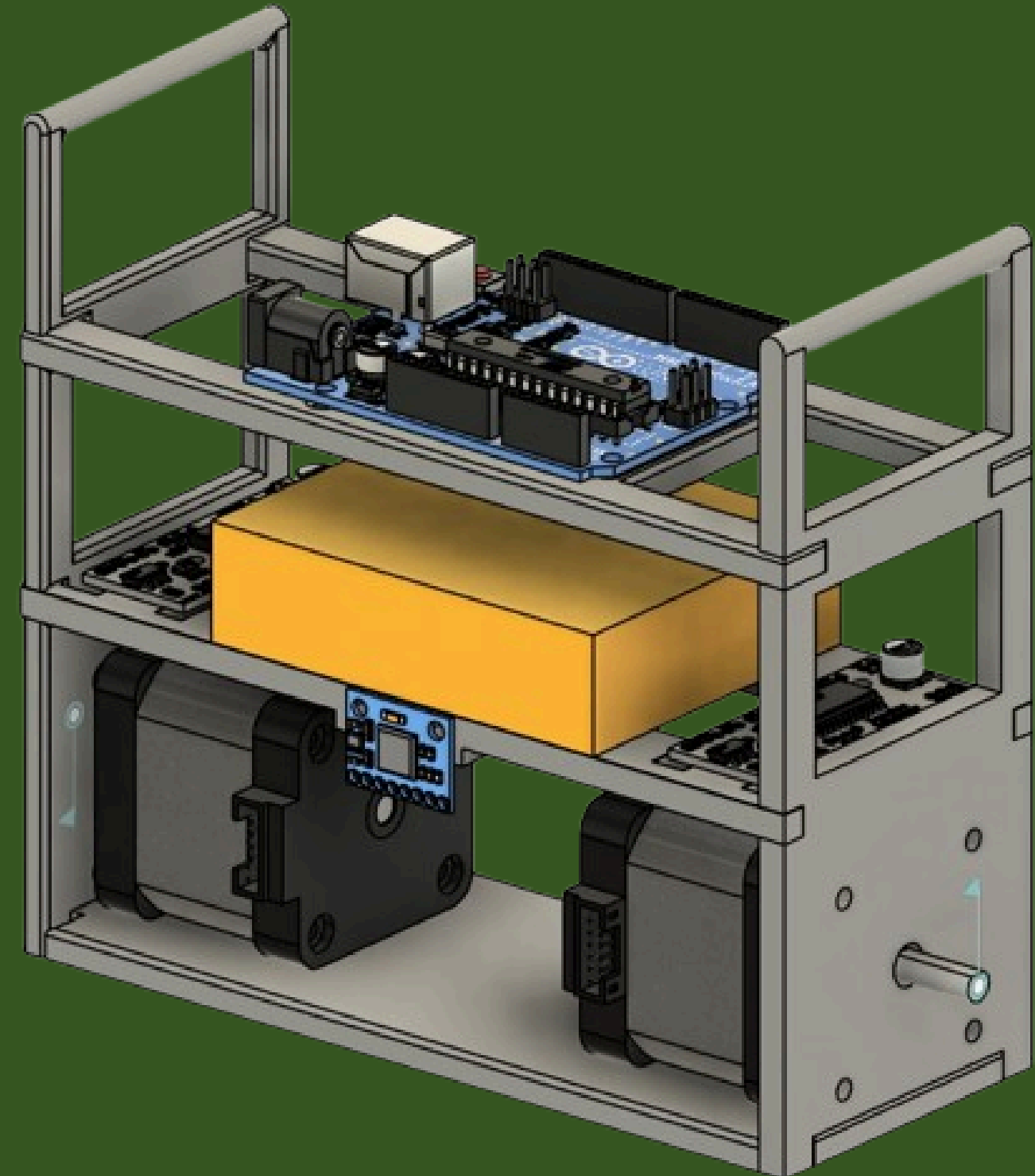


CONEXIONES PRELIMINARES



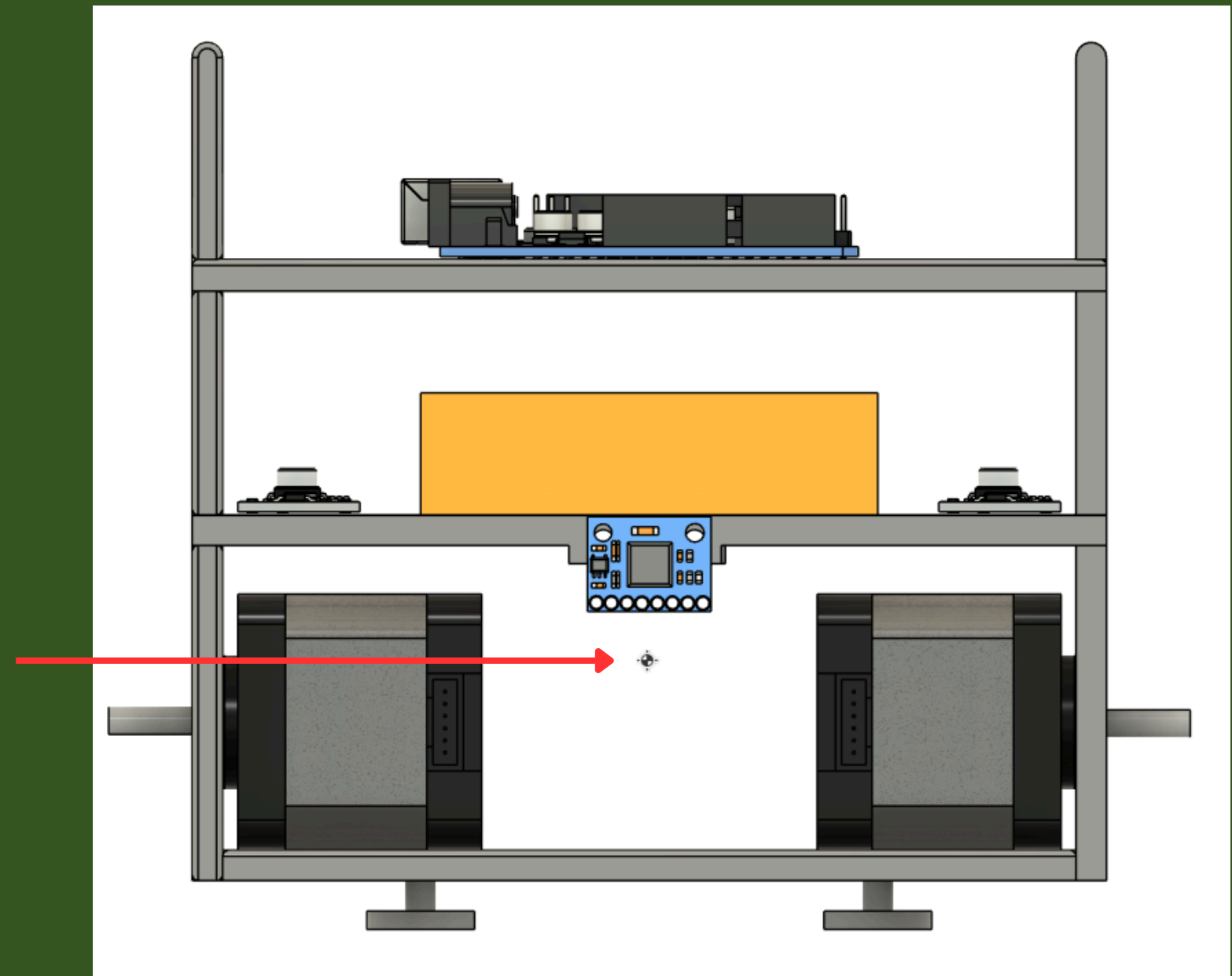
DISTRIBUCIÓN CARCASA

- Conexiones
- Drivers $\rightarrow 4.9 \times 2 \times (1 + 0.9)$ cm
- Sensor $\rightarrow 2.5 \times 2 \times (0.1 + 0.9)$ cm
- Motores
- Pilas $\rightarrow 2 \times 7.5 \times 6$ cm
- Interruptor $\rightarrow 1 \times 1.5 \times (1 + 0.7)$ cm



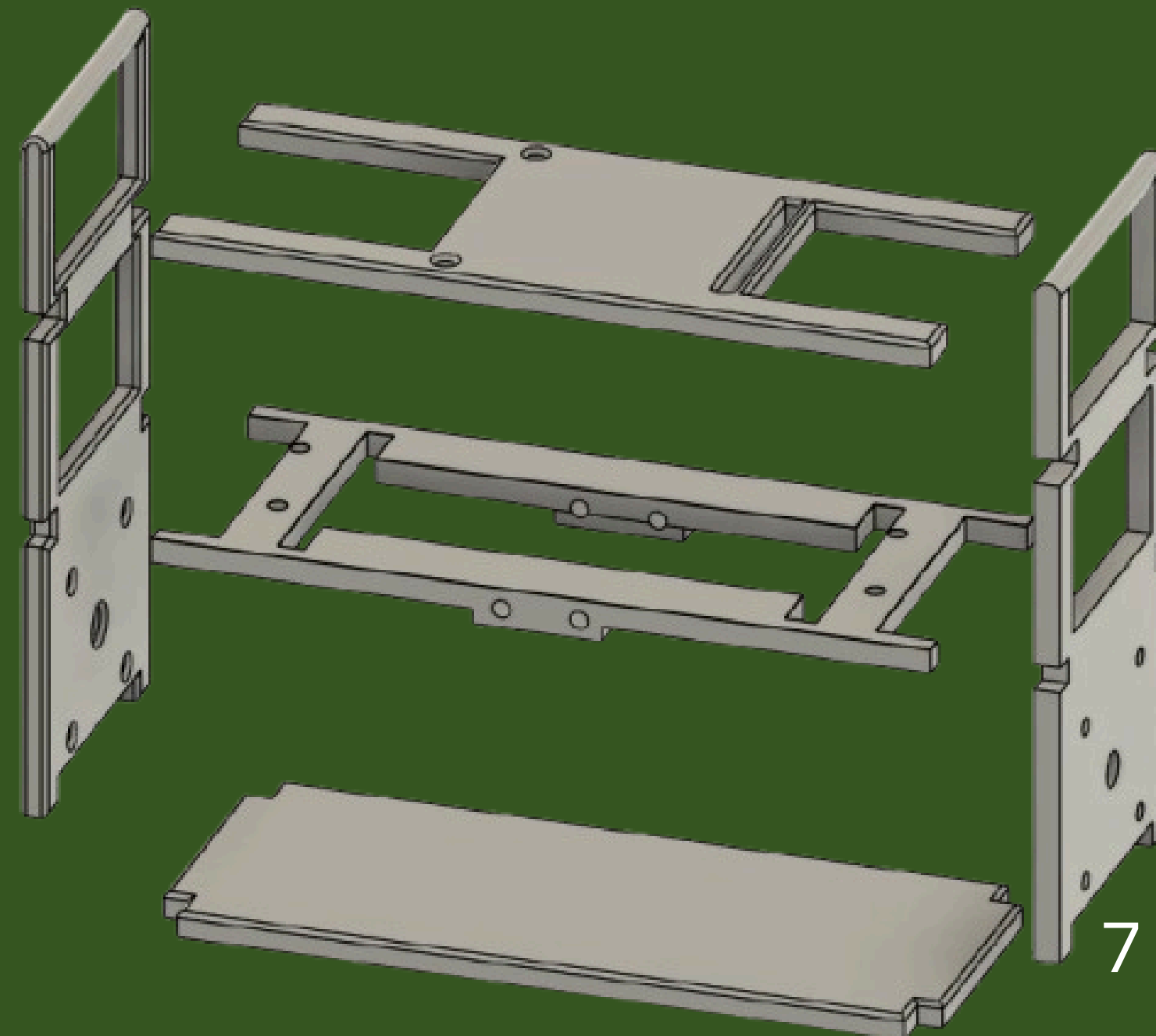
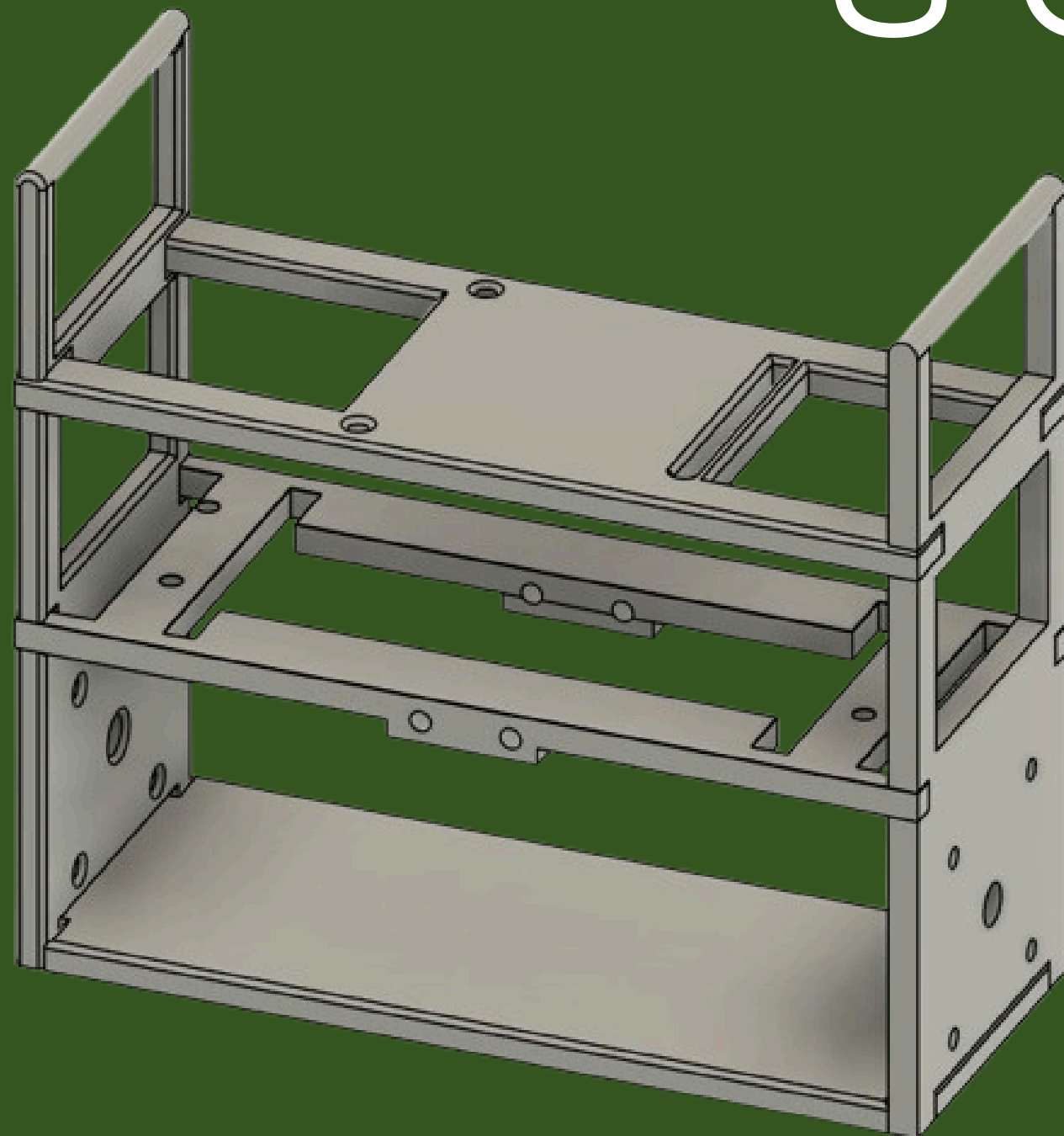
DISTRIBUCIÓN CARCASA

- Conexiones
- Drivers $\rightarrow 4.9 \times 2 \times (1 + 0.9)$ cm
- Sensor $\rightarrow 2.5 \times 2 \times (0.1 + 0.9)$ cm
- Motores
- Pilas $\rightarrow 2 \times 7.5 \times 6$ cm
- Interruptor $\rightarrow 1 \times 1.5 \times (1 + 0.7)$ cm



ANCLAJE SUPERIOR

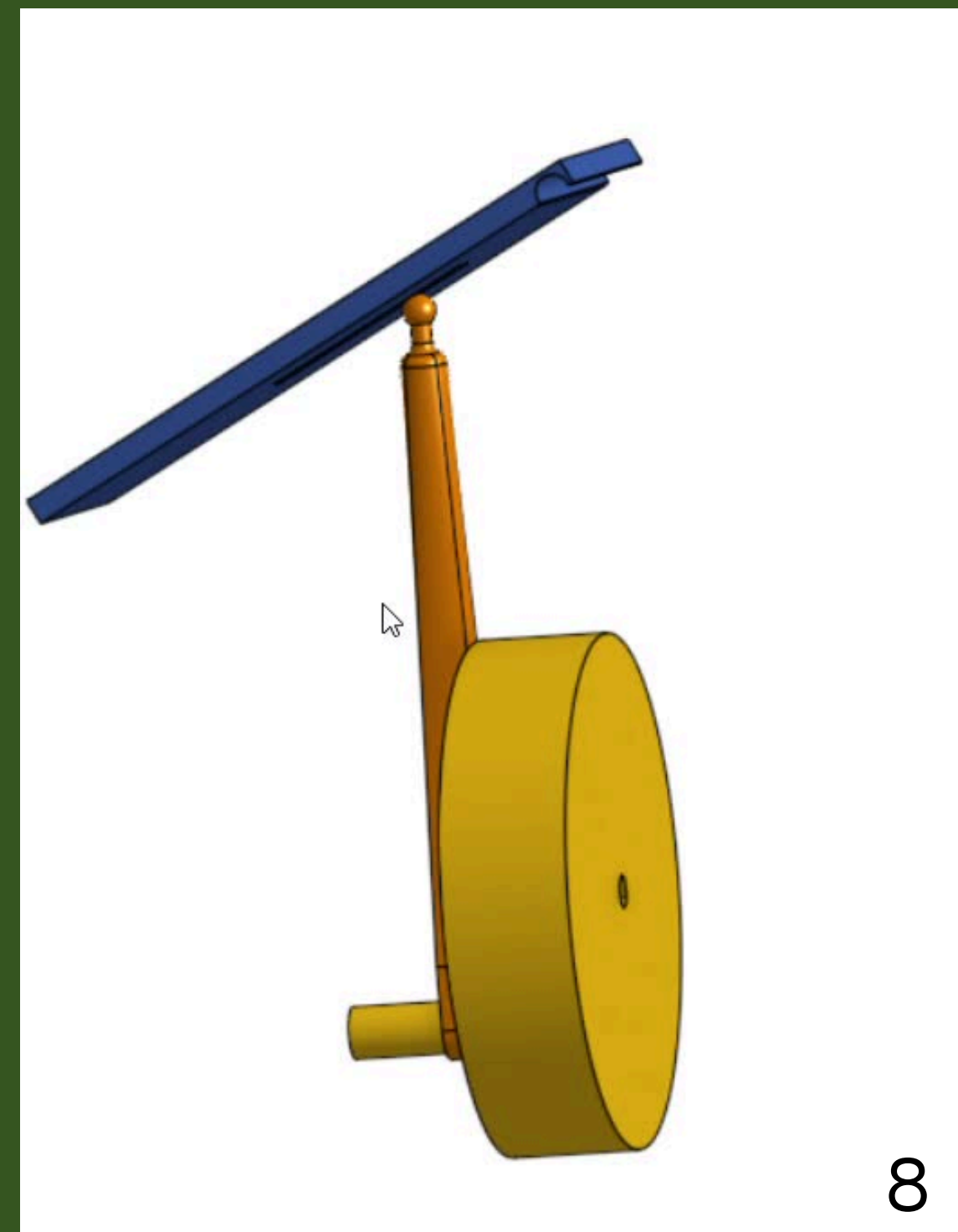
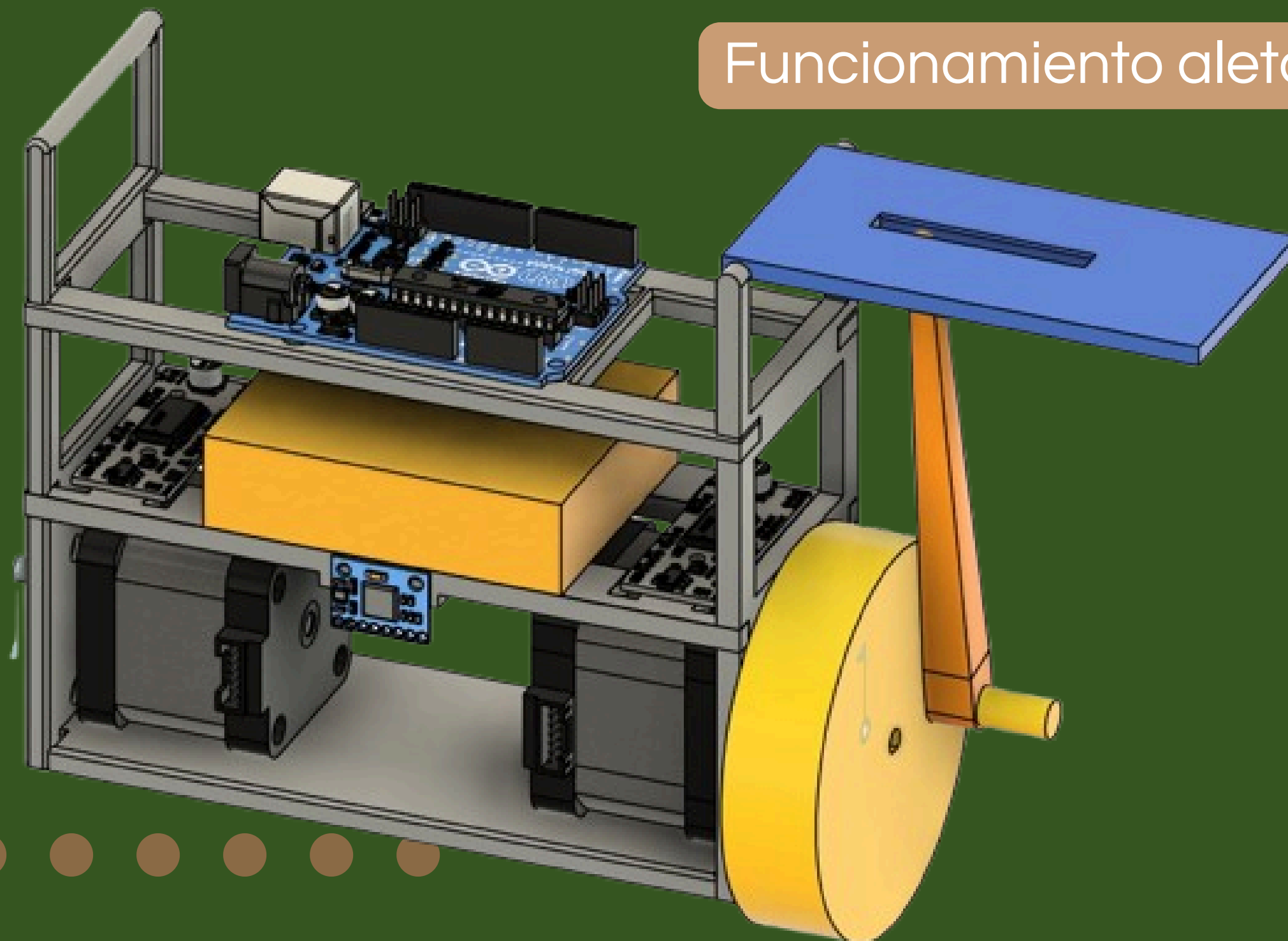
Modelo y materiales





ANCLAJE SUPERIOR

Funcionamiento aletas



CÓDIGO SENSOR

```
#include <Wire.h>
#define PI 3.14159265

const int MPU_ADDR = 0x68;

int16_t ax, ay, az;
int16_t gx, gy, gz;

int16_t gx_offset = 0, gy_offset = 0, gz_offset = 0;

float roll = 0, pitch = 0, yaw = 0;
unsigned long lastTime;
float dt;

void setup() {
  Wire.begin();
  Serial.begin(9600);

  Wire.beginTransmission(MPU_ADDR);
  Wire.write(0x6B); // Power management
  Wire.write(0);
  Wire.endTransmission(true);
}
```

Variables acelerómetro y giroscopio

- Ángulos:
- Roll: Eje x.
- Pitch: Eje y.
- Yaw: Eje z.
- Librería Wire realiza las conexiones.

Velocidad estándar para generar datos

CÓDIGO SENSOR

```
Serial.println("=== MPU-6050 Calibration ===");
Serial.println("Place the sensor still and press any key + ENTER to start...");
while (!Serial.available()); // Wait for user input
while (Serial.available()) Serial.read(); // Clear input buffer

calibrateGyro();
Serial.println("Calibration complete!");

lastTime = millis();
}

void loop() {
  readMPU6050();

  unsigned long now = millis();
  dt = (now - lastTime) / 1000.0;
  lastTime = now;

  float gyroZ = (gz - gz_offset) / 131.0;
  yaw += gyroZ * dt;

  // Convert int16_t to float
  float ax_f = (float)ax;
  float ay_f = (float)ay;
  float az_f = (float)az;

  // Avoid division by zero
  float denominator = sqrt(ay_f * ay_f + az_f * az_f);
  if (denominator < 0.0001) denominator = 0.0001;
```

atan2 calcula el ángulo en radianes

frecuencia de lectura de datos

131 implica +-250 radianes/segundo
medidos

leer los datos en bytes, salvo la
temperatura

```
// Compute angles in degrees
roll = atan2(ay_f, az_f) * 180.0 / PI;
pitch = atan2(-ax_f, denominator) * 180.0 / PI;

// Print angles
Serial.print("Roll: "); Serial.print(roll, 2);
Serial.print(" | Pitch: "); Serial.print(pitch, 2);
Serial.print(" | Yaw: "); Serial.println(yaw, 2);

delay(100);
}

void readMPU6050() {
  Wire.beginTransmission(MPU_ADDR);
  Wire.write(0x3B);
  Wire.endTransmission(false);
  Wire.requestFrom(MPU_ADDR, 14, true);

  ax = Wire.read() << 8 | Wire.read();
  ay = Wire.read() << 8 | Wire.read();
  az = Wire.read() << 8 | Wire.read();
  Wire.read(); Wire.read(); // Skip temperature
  gx = Wire.read() << 8 | Wire.read();
  gy = Wire.read() << 8 | Wire.read();
  gz = Wire.read() << 8 | Wire.read();
}
```

CÓDIGO SENSOR

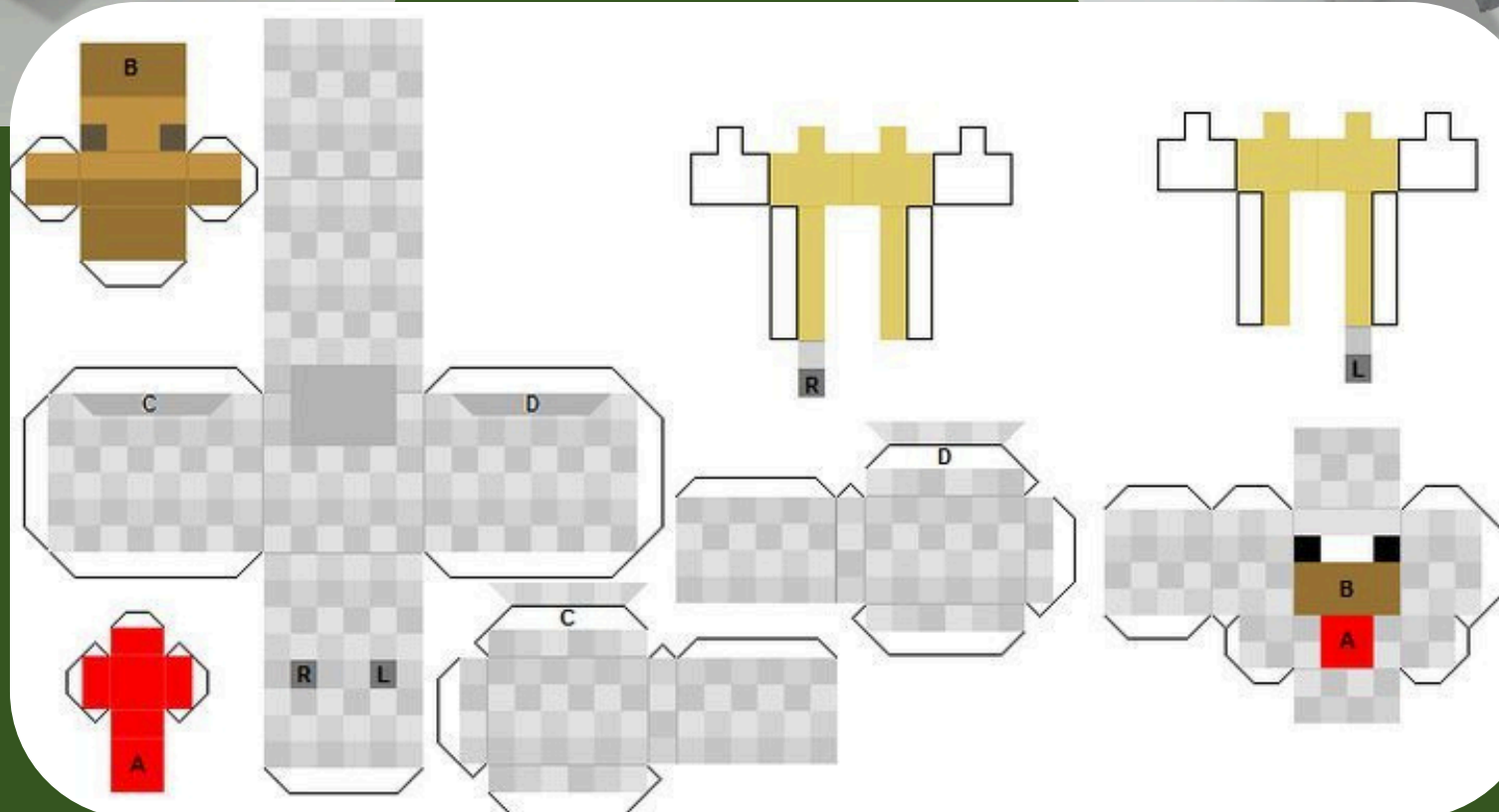
```
void calibrateGyro() {  
    long sumX = 0, sumY = 0, sumZ = 0;  
    const int samples = 100;  
  
    for (int i = 0; i < samples; i++) {  
        readMPU6050();  
        sumX += gx;  
        sumY += gy;  
        sumZ += gz;  
        delay(5);  
    }  
  
    gx_offset = sumX / samples;  
    gy_offset = sumY / samples;  
    gz_offset = sumZ / samples;  
}
```

- Offset: Desajuste.
- Calculan el offset del ángulo para sustraerlo.
- Samples son las muestras utilizadas para la calibración.



LO QUE QUEDA POR DELANTE

- Diseño final de las ruedas.
- Probar todo el sistema (unir la lectura del sensor al movimiento de los motores).
- PID.
- A ensamblar el prototipo final con la carcasa superior.



MUCHAS
GRACIAS

