

Diseño electrónico de un robot autónomo

Trabajo de Fin de Grado

Javier Macías Solá
Universidad de La Laguna - GIEIA

Tutor: Dr. Leopoldo Acosta Sánchez



Índice

1. Introducción.
 - a. Objetivos.
 - b. Punto de partida.
2. Estudio previo y bases de diseño.
 - a. Concepto básicos de robots móviles.
 - b. Cinemática.
3. Electrónica. Sensores y actuadores.
4. Implementaciones.
 - a. Resumen versión 1.
 - b. Versión 2: modificaciones.
 - i. Estructura.
 - ii. Software.
5. Conclusiones.





Introducción

Idea: navegación autónoma entre los puntos marcados en un mapa desde una interfaz.

1. Objetivos básicos
 - a. Control autónomo y manual
 - b. Evitar obstáculos
2. Punto de partida
 - a. Base cuadrada (50 x 70 cm)
 - b. Dos ruedas diferenciales y una loca





Estudio previo y bases de diseño

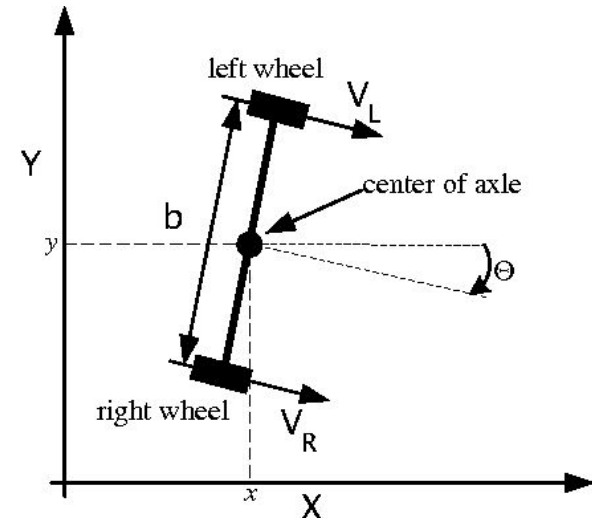


Estudio previo

Robot móvil: tres grados de libertad (x , y , θ) con respecto al mundo global.

Navegación: conocer la ubicación actual, idear, y seguir una ruta.

1. Localización: ¿dónde está el robot?
 - a. Orientación limitada al rango $[0, 360)$ grados.
 - b. Localización en grados decimales. Más cifras, más precisión y ruido.
2. Planificación: ¿ruta libre de obstáculos?



Fuente: Semantic Scholar



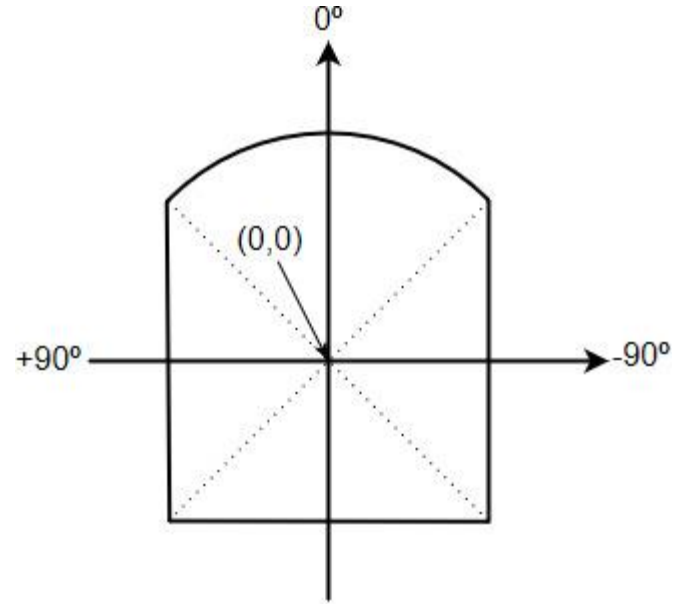
Estudio previo

Coordenadas locales:

1. Origen en el centro geométrico.
2. Cero grados frente del vehículo, eje y.
El signo del ángulo determina el lado.

¿Se puede ubicar un sensor?

Cinemática: depende de la velocidad y dirección de giro de las ruedas laterales.





Bases de diseño

1. Maniobrabilidad
2. Detectar y evitar obstáculos
3. Poco volumen (radio < 1 m), pasillos de 1.40 m
4. Bajo costo/modular
5. Centro de gravedad bajo, estabilidad
6. Espacio para futuras mejoras
7. Escalabilidad



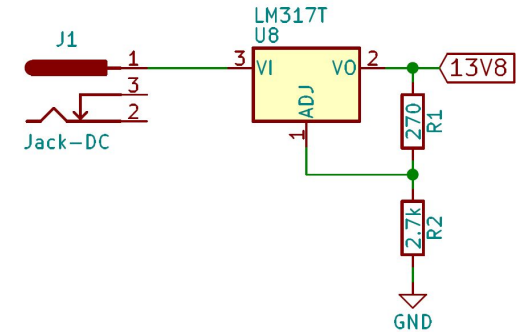
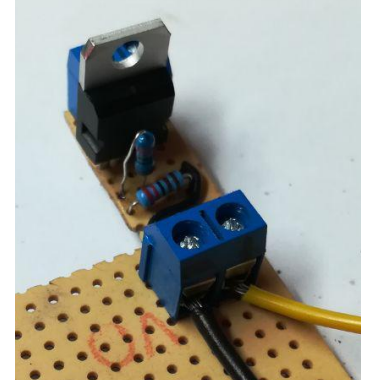
Electrónica

Sensores y actuadores



Electrónica general

1. Batería: aporta la energía al sistema.
 - a. Reciclada de una moto.
 - b. Química: plomo ácido.
 - c. Seis celdas, 2.1 voltios por celda, 12.6 V total (OCV).
 - d. Ventaja: voltaje seguro y típico. En OCV representa el nivel de carga.
2. Cargador: fuente que permite recargar la batería.
 - a. Invierte la reacción química.
 - b. Plomo ácido: voltaje de carga entre 2.3 y 2.4 V/celda.
 - c. Reutilizada de un equipo de 17 V, adaptar el voltaje.
 - d. Polarizar un regulador (LM317T) a 13.8 V.
 - i. Dropout voltage ≈ 3.2 V.
 - ii. Intensidad máxima: 2.2 A.





Electrónica general

- 3. Controlador - Arduino MEGA 2560
 - a. Alimentación a 12 V con regulador integrado.
 - b. 4x Puertos Serial.
 - c. 2x I2C.
 - d. 15x PWM.
 - e. 16x Entradas analógicas.
 - f. 15x I/O digitales.
 - g. 10 bits ADC.
 - h. 8 bits DAC.
 - i. C/C++

No aporta potencia, hasta 20 mA/pin.



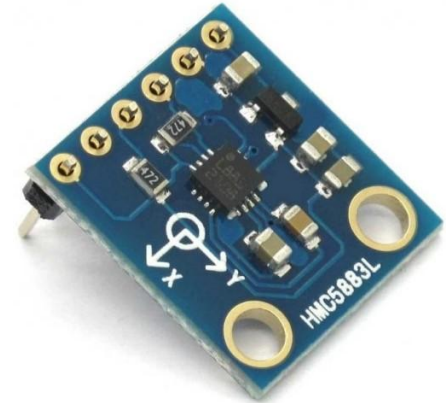
Fuente: Arduino.cc



Sensores

Permiten cerrar el lazo de realimentación del sistema.

1. Brújula: orientación del robot midiendo el campo magnético terrestre.
 - a. Orientación y ubicación críticas.
 - b. Separación de objetos metálicos (radio > 30 cm).
 - c. Librería: HMC5883_Simple.
 - d. Calibrar el módulo según la declinación magnética.
 - e. GY-271, magnetómetro HMC5883L
 - i. Alimentación a 5 V.
 - ii. Comunicación I2C.



Fuente: njuskalo.hr



Sensores

2. GPS: ubicación con respecto al mundo global.

- a. Latitud (y) y longitud (x).
- b. Librería: TinyGPS.
- c. GY-GPS6MV2.
 - i. Alimentación a 5 V.
 - ii. Comunicación Serial.
- d. Parsear sentencias típicas y estándar.
- e. Señales débiles.



Fuente: dx.com

Ejemplo: \$GPRMC,032606,A,3410.2358,N,11819.0865,W,0.0,207.2,180211,13.5,E,A*32



Sensores

2. GPS: ubicación con respecto al mundo global.

- a. Latitud (y) y longitud (x).
- b. Librería: TinyGPS.
- c. GY-GPS6MV2.
 - i. Alimentación a 5 V.
 - ii. Comunicación Serial.
- d. Parsear sentencias típicas y estándar.
- e. Señales débiles.



Fuente: dx.com

Ejemplo: \$GPRMC,032606,A,3410.2358,N,11819.0865,W,0.0,207.2,180211,13.5,E,A*32

↑
Latitud

↑
Longitud

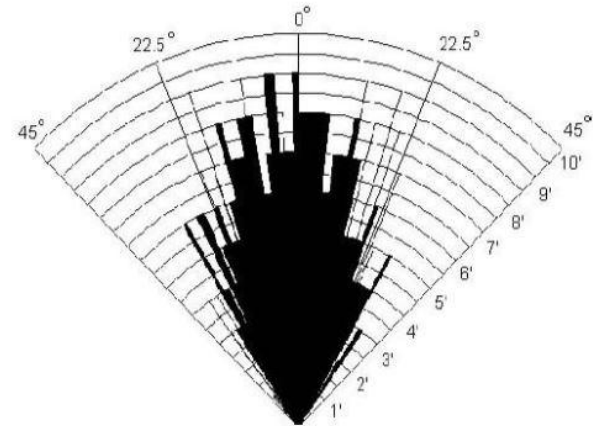
Sensores

3. Ultrasonidos: detectar obstáculos.

- a. Ventajas:
 - i. No hace falta librería. Función propia.
 - ii. Económicos.
- b. Desventajas:
 - i. Ondas de presión.
 - ii. Eco
 - iii. Escasa resolución espacial.
- c. HC-SR04:
 - i. Alimentación a 5 V.
 - ii. Comunicación por pines echo y trig.
 - iii. Apertura de 15 grados.
 - iv. Frecuencia máxima: 20 Hz



Fuente: amazon.co.uk



Fuente: [25]



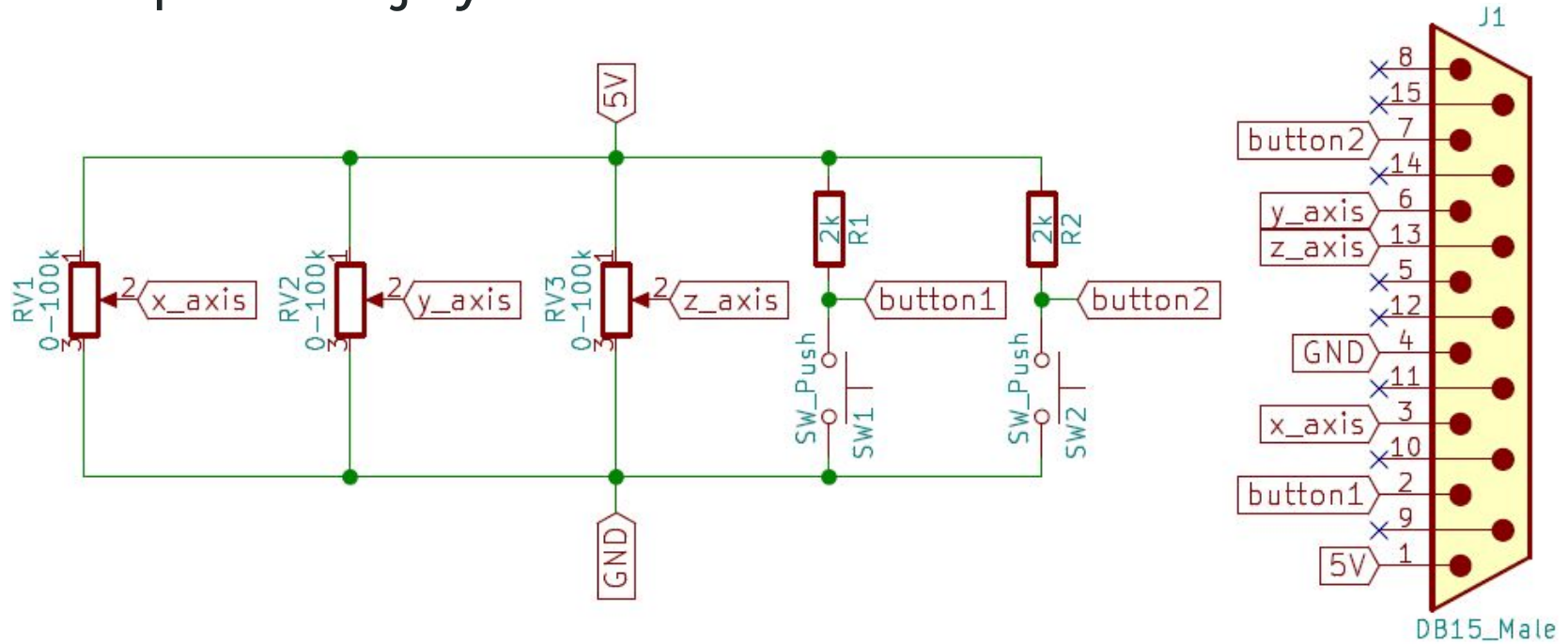
Sensores

- 4. Joystick: control manual del operario.
 - a. Manejo intuitivo.
 - b. Dos potenciómetros que marcan cada eje.
 - c. Adaptar Quickshot QS-203:
 - i. Potenciómetros polarizados a 5 V y lectura del valor analógico.
 - ii. Pull-up resistors para los botones.
 - iii. Trimado.
 - iv. Mantener el conector DA-15.





Adaptación joystick





Actuadores

1. Motores: cambian la pose del vehículo.
 - a. TGE589A. Motor de limpiaparabrisas.
 - i. DC, velocidad proporcional al voltaje.
 - ii. Tornillo sin fin, par alto (18.3 Nm).
 - iii. Consumo: 0.9 A con poca carga.
 - iv. Intensidad de cortocircuito: 12 A.
 - v. Velocidad angular: 66 rpm.
 - b. Desventajas: falta de documentación.
 - c. Reductora externa de 8/16.

Suponiendo correctos los datos anteriores:
hasta 176 kg con dos motores.



Fuente: rctankcombat.com

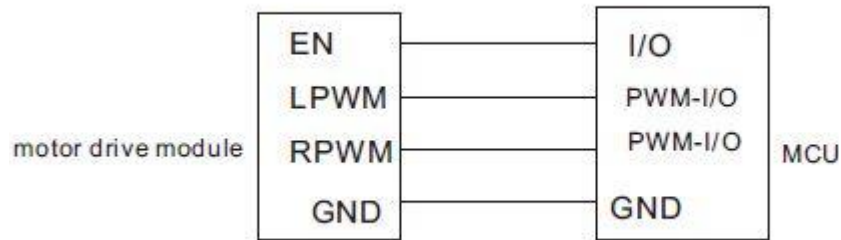


Problema: voltaje intermedio en un sistema digital

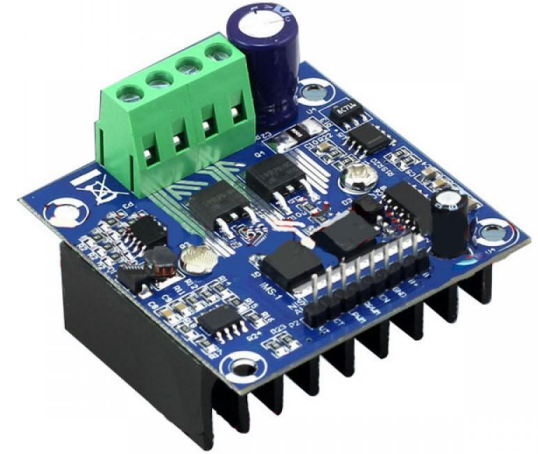
Solución: puente H excitado con PWM.

Módulo IMS-1

- a. Hasta 50 A.
- b. Velocidad de 8 bits (DAC Arduino).
- c. Librería: MOTOR.



Fuente: [10]



Fuente: thanksbuyer.com



Implementaciones



Resumen versión 1

Centrada en implementar los componentes.

a. Modo automático:

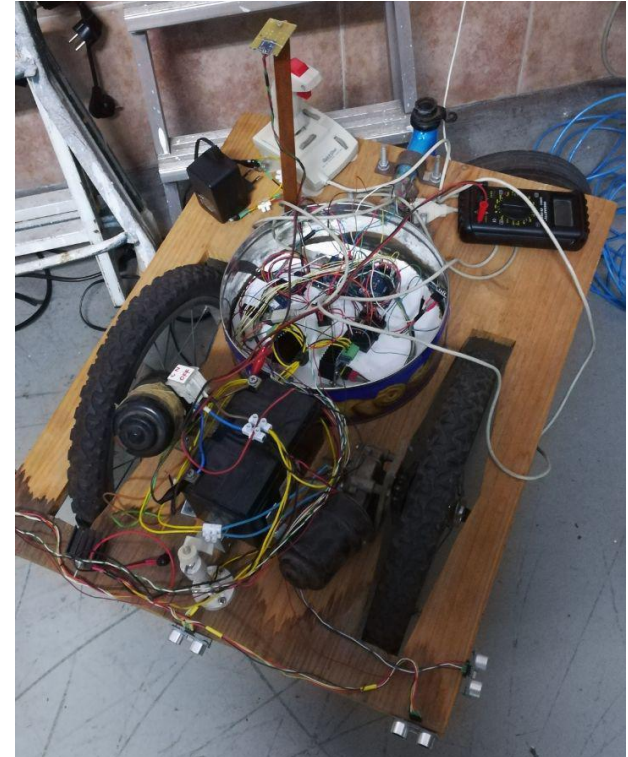
- i. Orientado a seguir un rumbo y evitar paredes.
- ii. Cuatro sensores ultrasonidos.
- iii. Controlador todo/nada.

b. Modo manual:

- i. Algoritmo: transformar coordenadas cartesianas a diamante y comandar los motores.

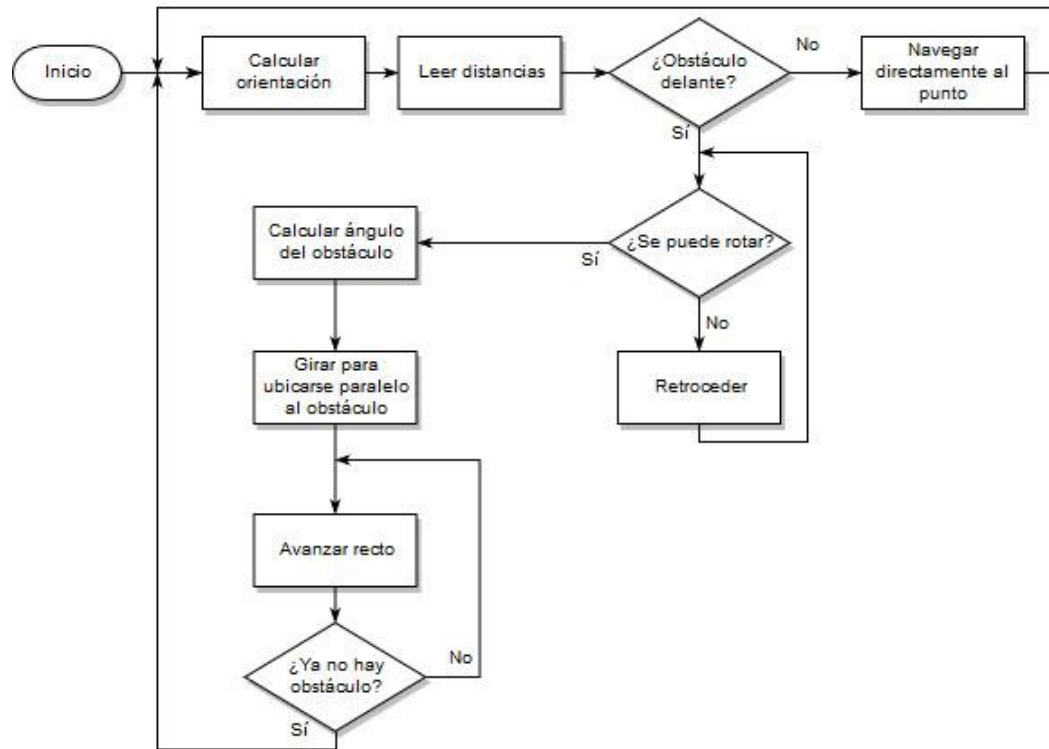
c. Problemas:

- i. Orientación y comunicación remota poco robusta.
- ii. Inercia. Centro de gravedad desplazado.



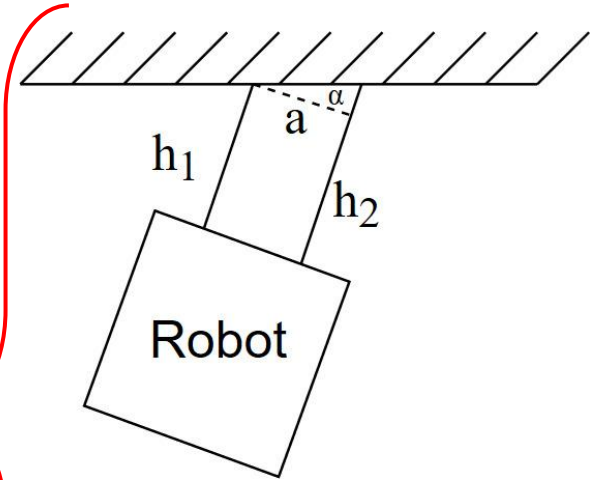
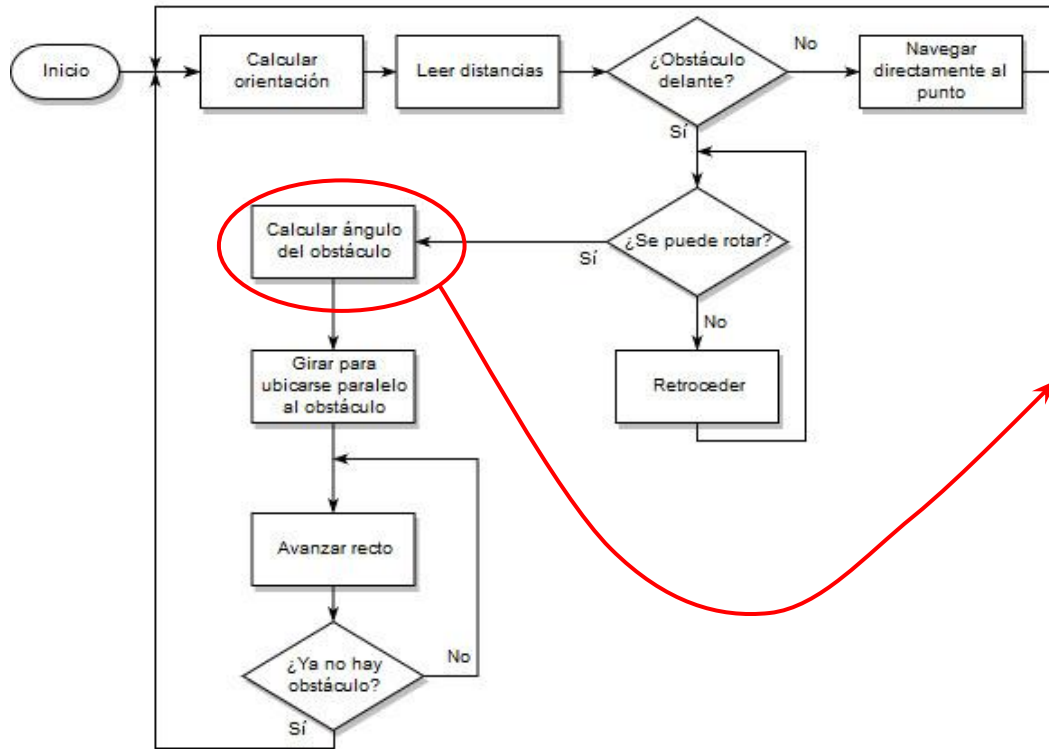


Resumen versión 1





Resumen versión 1



$$\alpha = \arctg\left(\frac{h_2 - h_1}{a}\right)$$



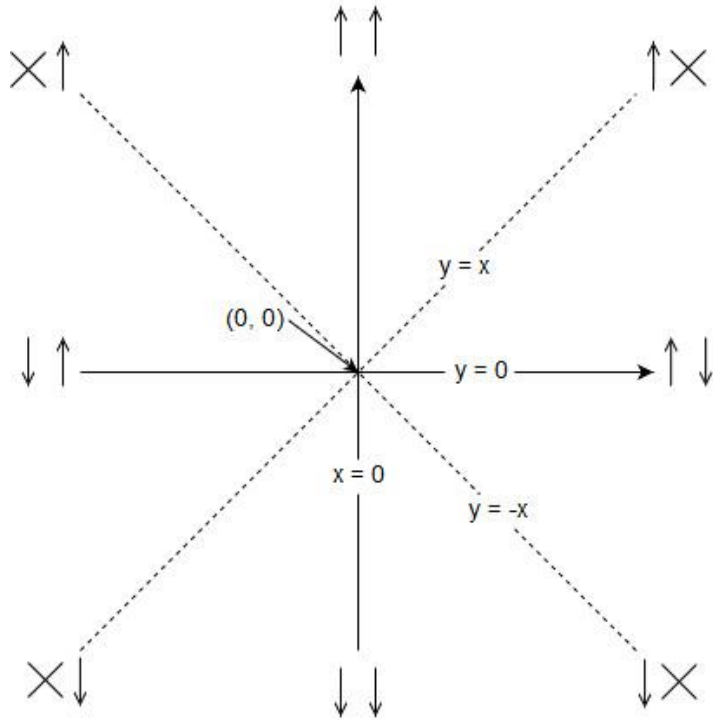
Resumen versión 1

Algoritmo modo manual: transformar coordenadas cartesianas a diamante [23].



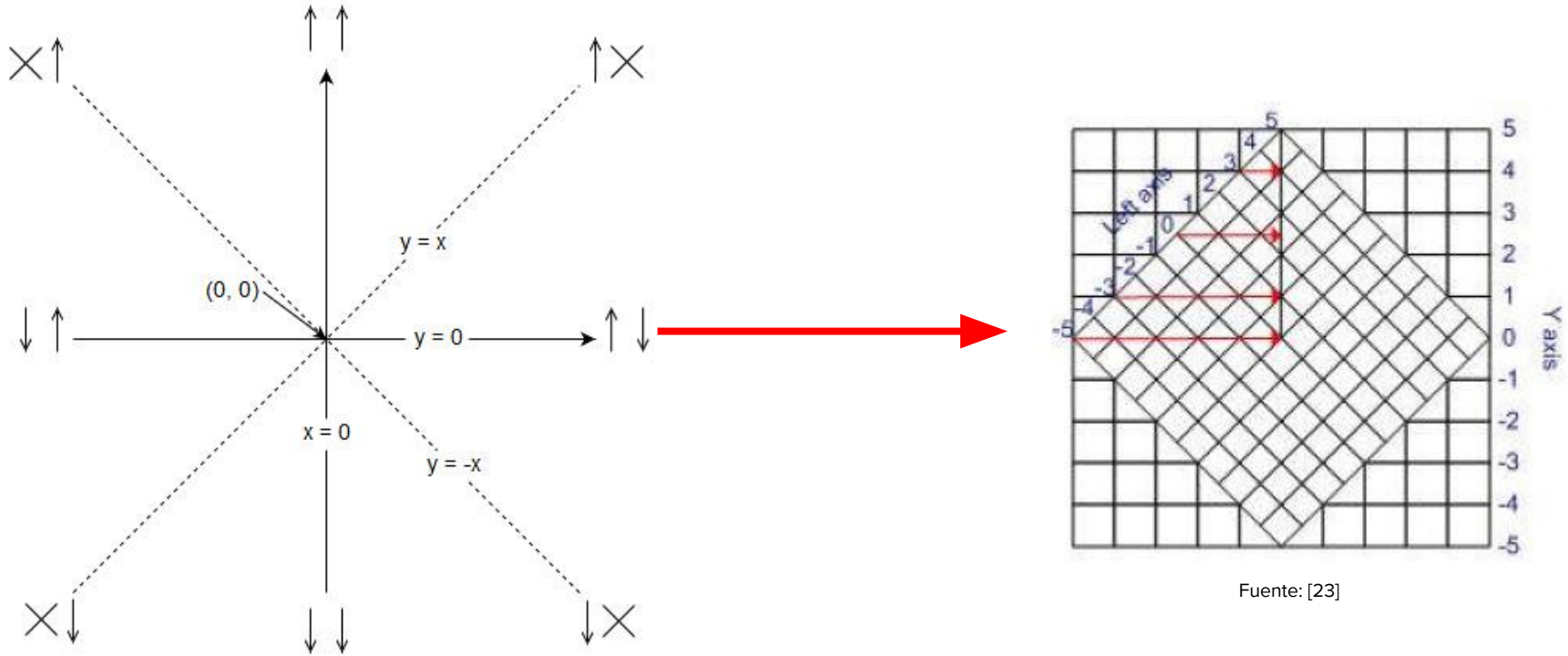


Resumen versión 1





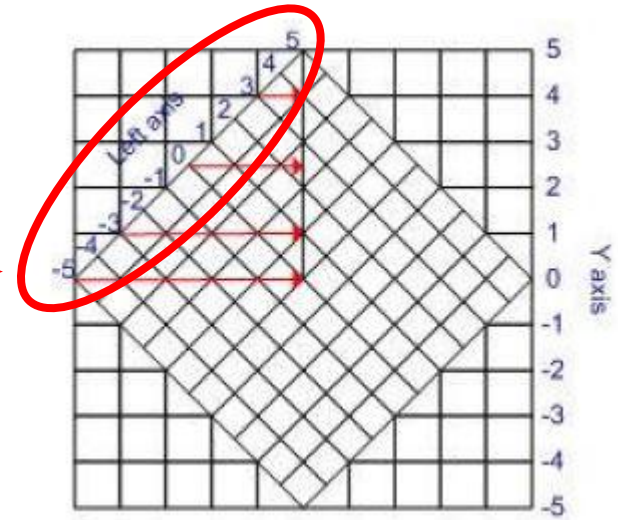
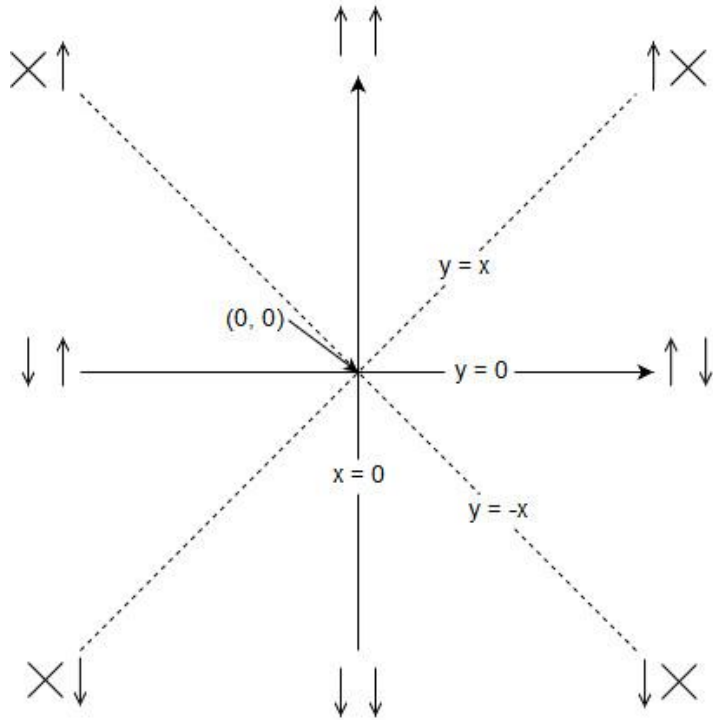
Resumen versión 1



Fuente: [23]



Resumen versión 1



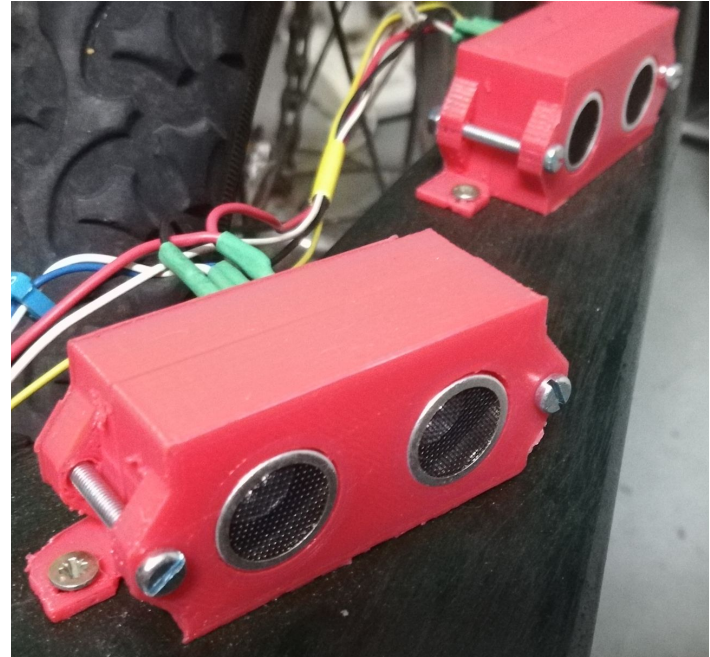
Fuente: [23]



Versión 2: modificaciones

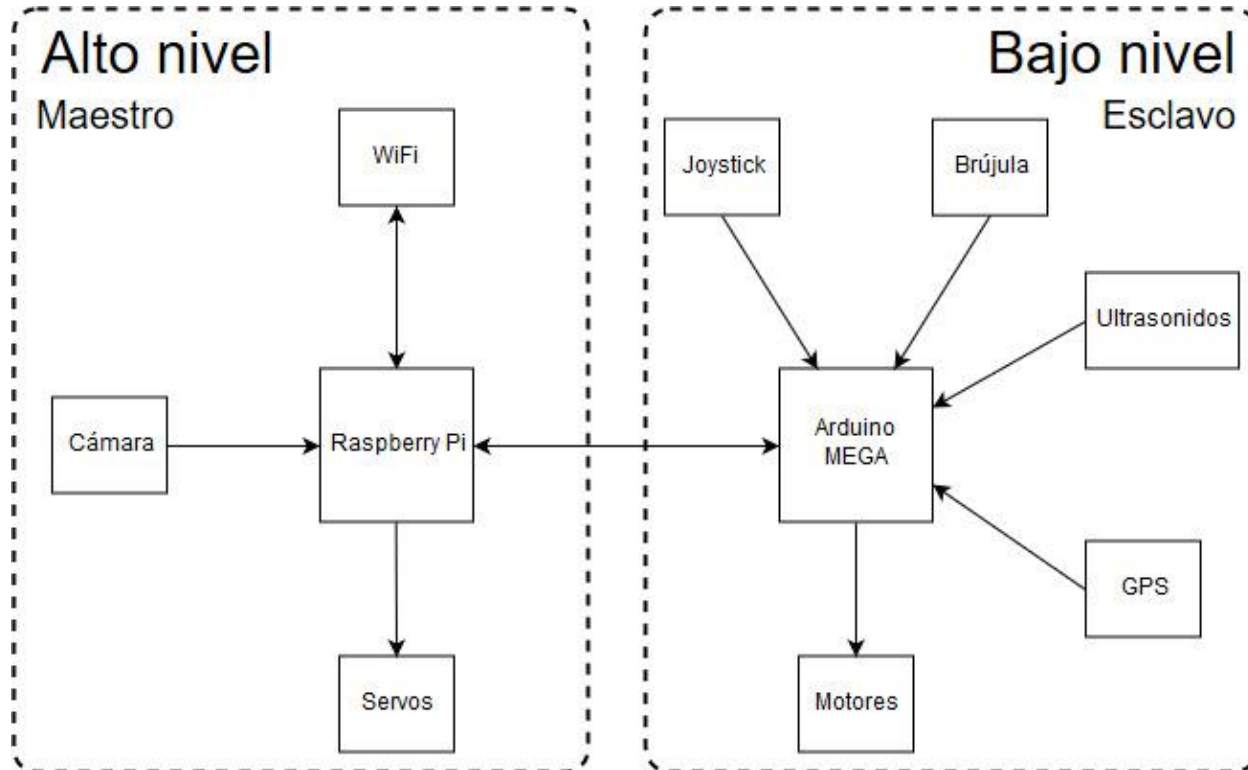
Implementar navegación y evitar obstáculos.

1. Electrónica jerárquica: multi-threading.
2. Centro de gravedad menos desplazado.
3. “The bubble rebound obstacle avoidance algorithm” [22].
4. Protecciones para sensores y nuevas ubicaciones.
5. Controlador PI.





Versión 2: electrónica





Versión 2: electrónica

1. Brújula.
2. Joystick.
3. Motores.
4. GPS.
5. Interruptor de tres estados.
6. Batería.
7. Interfaces de comunicación.
8. Resto de electrónica.
9. Array de ultrasonidos.





Versión 2: electrónica

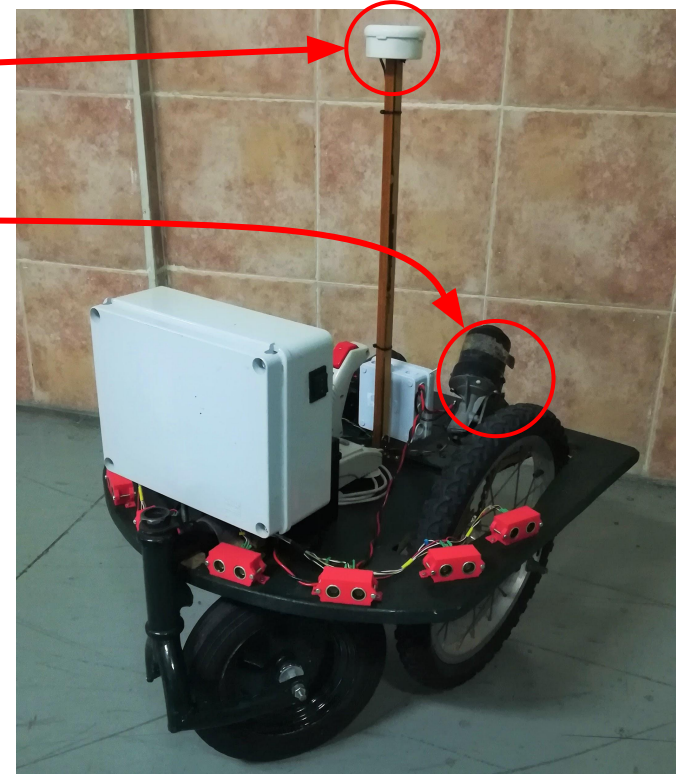
1. Brújula.
2. Joystick.
3. Motores.
4. GPS.
5. Interruptor de tres estados.
6. Batería.
7. Interfaces de comunicación.
8. Resto de electrónica.
9. Array de ultrasonidos.





Versión 2: electrónica

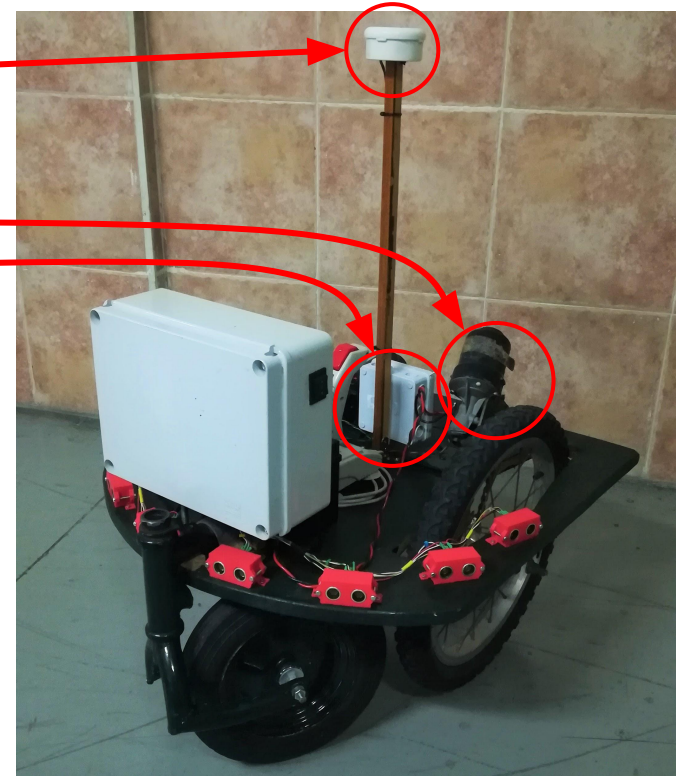
1. Brújula.
2. Joystick.
3. Motores.
4. GPS.
5. Interruptor de tres estados.
6. Batería.
7. Interfaces de comunicación.
8. Resto de electrónica.
9. Array de ultrasonidos.





Versión 2: electrónica

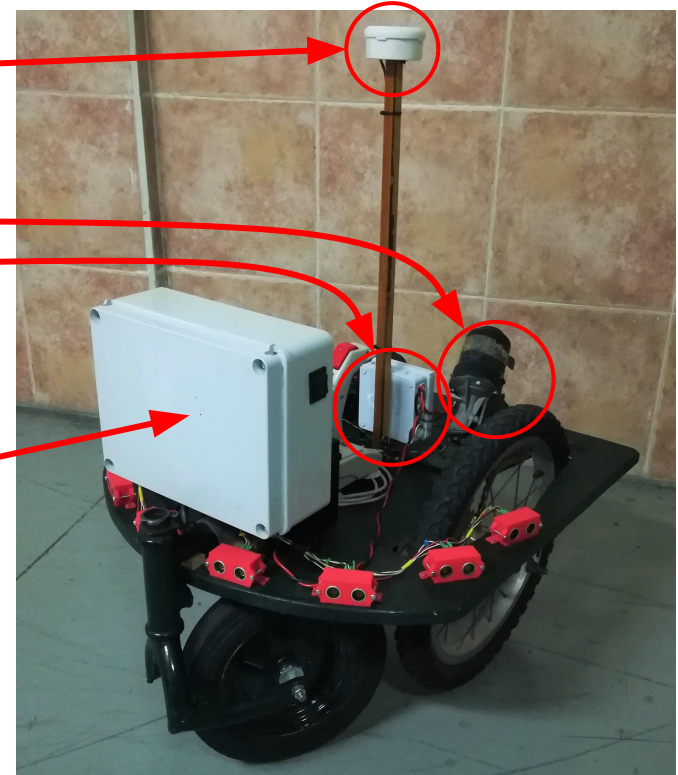
1. Brújula.
2. Joystick.
3. Motores.
4. GPS.
5. Interruptor de tres estados.
6. Batería.
7. Interfaces de comunicación.
8. Resto de electrónica.
9. Array de ultrasonidos.





Versión 2: electrónica

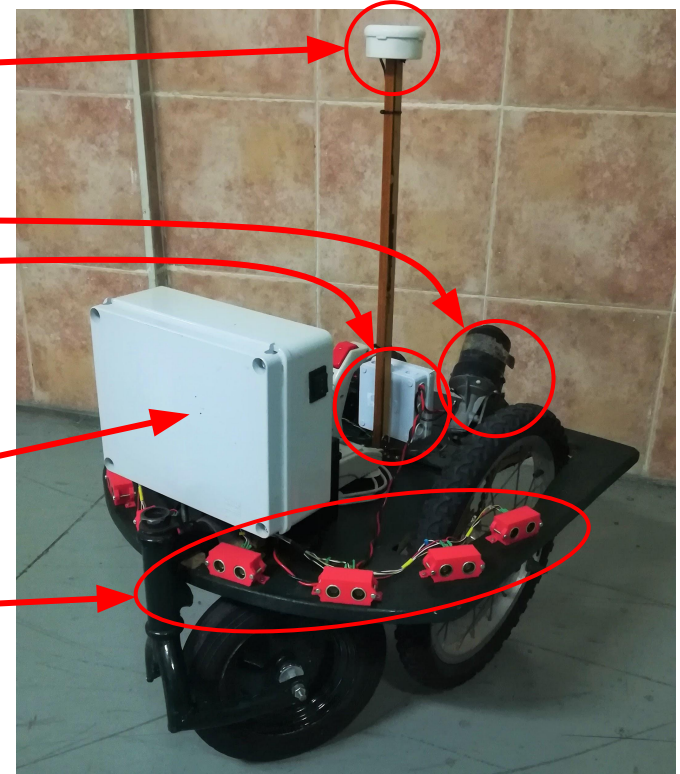
1. Brújula.
2. Joystick.
3. Motores.
4. GPS.
5. Interruptor de tres estados.
6. Batería.
7. Interfaces de comunicación.
8. Resto de electrónica.
9. Array de ultrasonidos.





Versión 2: electrónica

1. Brújula.
2. Joystick.
3. Motores.
4. GPS.
5. Interruptor de tres estados.
6. Batería.
7. Interfaces de comunicación.
8. Resto de electrónica.
9. Array de ultrasonidos.

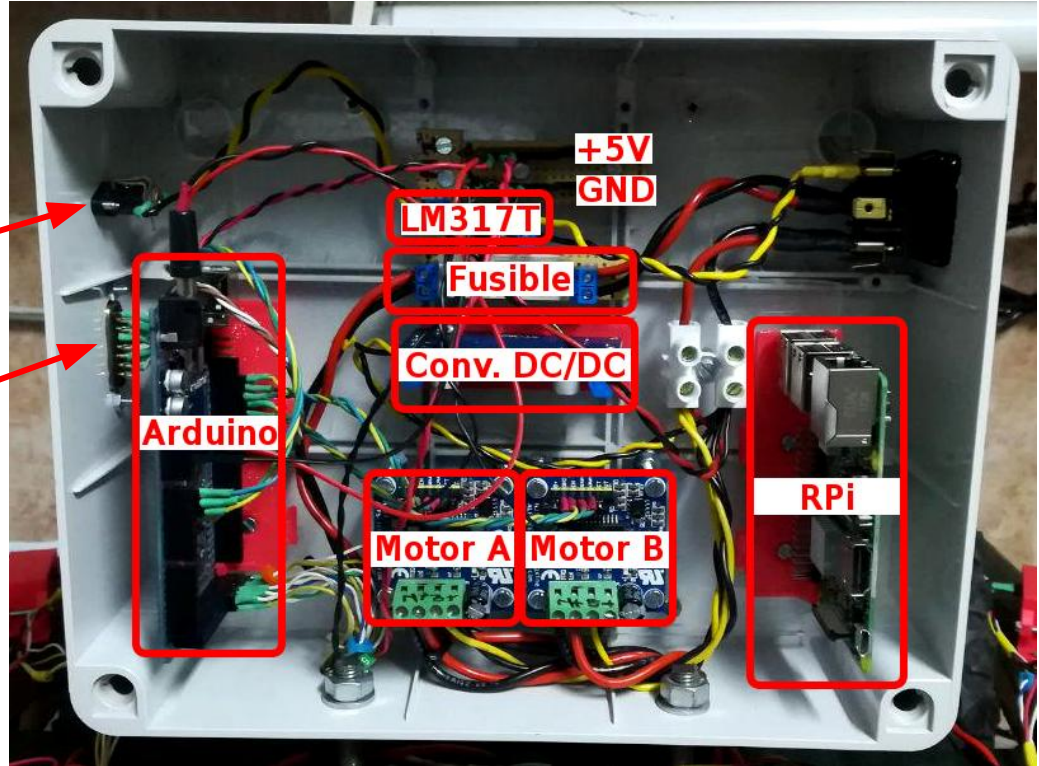




Versión 2: electrónica

DC jack cargador.

Conector DA-15 (joystick).





Versión 2: software. Raspberry Pi.

Gestionar la comunicación remota. Alto nivel.

1. AP para conexión SSH.
2. Tiempo de arranque lento.
3. GPIOs configurados en boot (bash).
4. Alias de Linux para cambiar el estado.
5. Comunicación Serial con Arduino (debug).



Fuente: raspberrypi.org



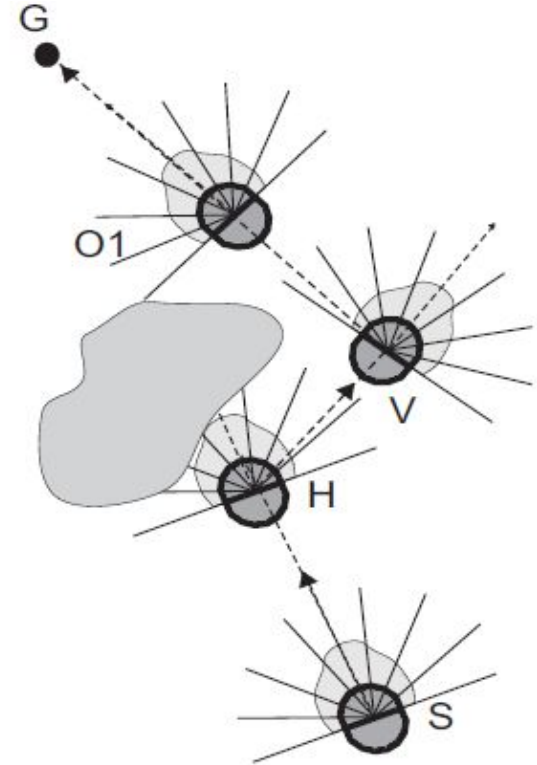
Versión 2: software. Arduino.

Gestionar el nivel más bajo de comunicación.

Tras confirmar que la Raspberry Pi está lista:

1. Navegar en campo abierto.
2. Evitar obstáculos (bubble rebound algo.)

	Original	Implementación
Lógica	Difusa	Booleana
Burbuja	Variable con la cinemática	Fija

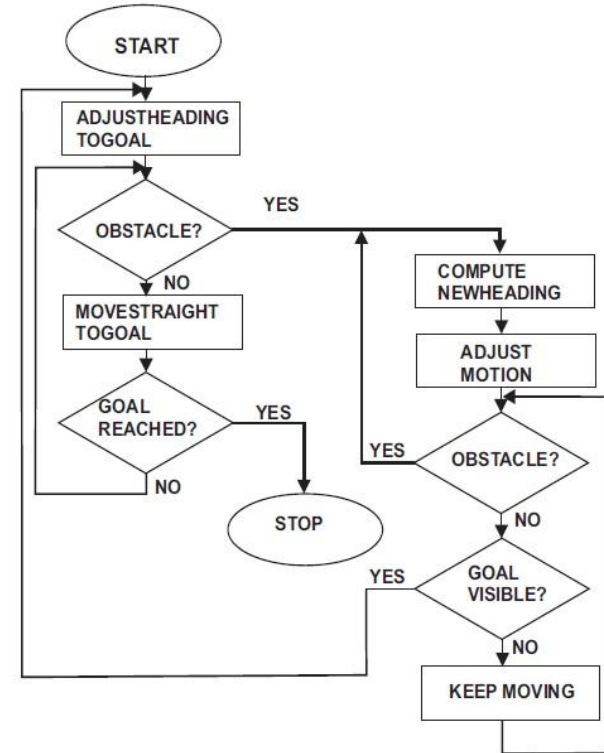




The Bubble Rebound algorithm. Resumen

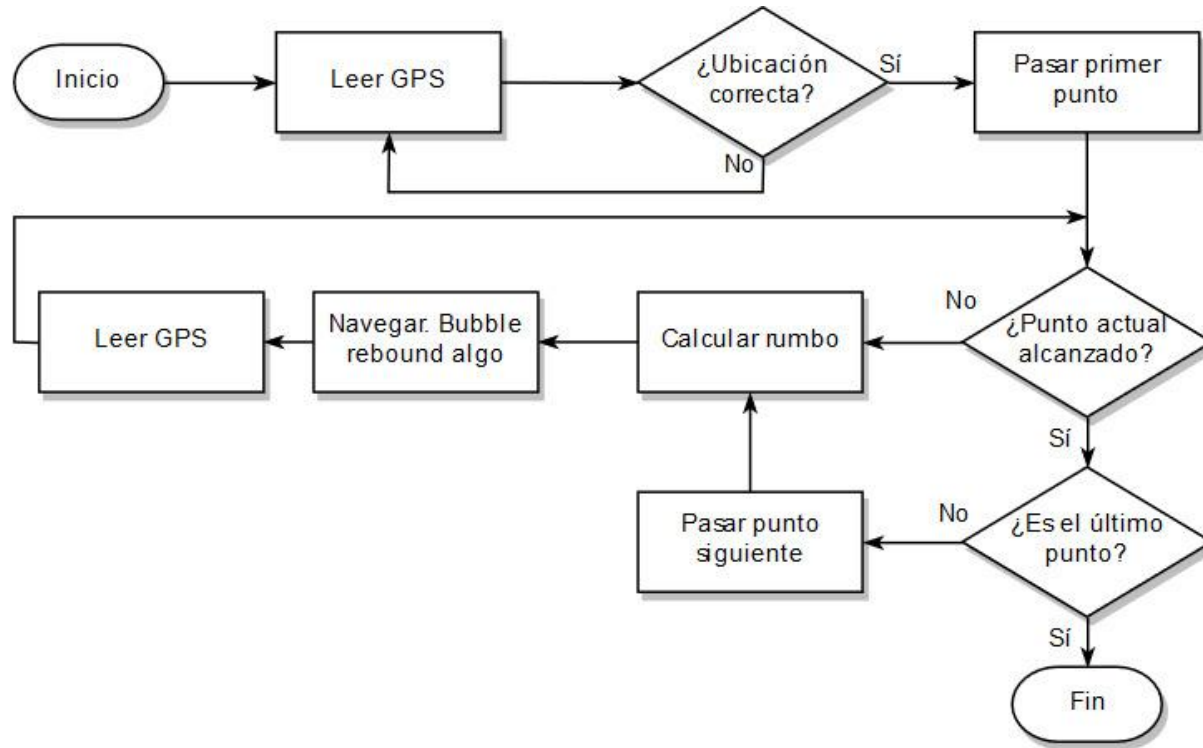
1. Ultrasonidos cubren 180 grados.
2. Si detecta que puede golpear, rebota hacia la menor densidad de obstáculos.
3. Navega recto hasta que el objetivo esté visible.
4. Si el objetivo está visible, navega directamente el punto.

$$\alpha_r = \frac{\sum_{i=-4}^4 \alpha_i D_i}{\sum_{i=-4}^4 D_i}$$





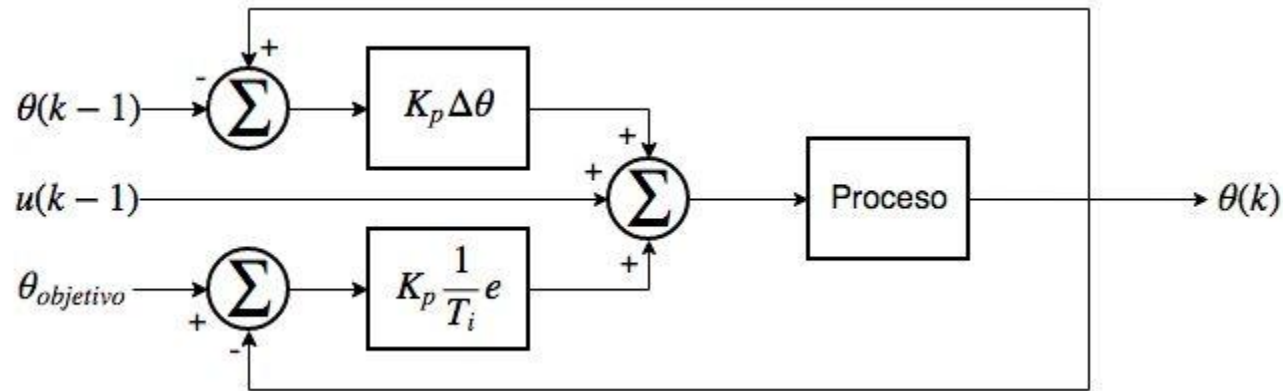
Versión 2: software. Arduino.





Versión 2: software. Arduino.

1. Rotar sobre sí mismo. Controlador PI.
 - a. Parte proporcional con el incremento de la variable de proceso [12, pp. 52-78].
 - b. Error sólo en la parte integral.
2. Mover hacia delante. Todo-nada calibrado para corregir la deriva.



Conclusions

Sensor location and filtering.
PI over bang-bang controller.
Hierarchical structure.





¡Gracias por su atención!