

Universidad de Castilla-La Mancha

Escuela Superior de Ingeniería Informática

Trabajo Fin de Grado

Grado en Ingeniería Informática

Tecnología Específica de

Xxxxxxxxx

Título Plantilla para la elaboración de la memoria del TFG

Autor

Mes, Año

|  |  |
| --- | --- |
|  |  |

Trabajo Fin de Grado

Grado en Ingeniería Informática

Tecnología Específica de

Ingeniería de Computadores

Plantilla para la elaboración de la memoria del TFG

|  |  |
| --- | --- |
| **Autor:** | Pedro López Marín |
| **Tutor:** | Tutor nombre |
| **Co-Tutor:** | Co-tutor nombre |

Mes, Año

Dedicado a mi familia y a todos aquellos……….

Declaración de Autoría

Yo, Pedro López Marín con DNI 49217546F, declaro que soy el único autor del trabajo fin de grado titulado “......” y que el citado trabajo no infringe las leyes en vigor sobre propiedad intelectual y que todo el material no original contenido en dicho trabajo está apropiadamente atribuido a sus legítimos autores.

Albacete, a.....

Fdo: ......

Resumen

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur, adipisci velit, sed quia non numquam eius modi tempora incidunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim ad minima veniam, quis nostrum exercitationem ullam corporis suscipit laboriosam, nisi ut aliquid ex ea commodi consequatur? Quis autem vel eum iure reprehenderit qui in ea voluptate velit esse quam nihil molestiae consequatur, vel illum qui dolorem eum fugiat quo voluptas nulla pariatur?

Agradecimientos

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur, adipisci velit, sed quia non numquam eius modi tempora incidunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim ad minima veniam, quis nostrum exercitationem ullam corporis suscipit laboriosam, nisi ut aliquid ex ea commodi consequatur? Quis autem vel eum iure reprehenderit qui in ea voluptate velit esse quam nihil molestiae consequatur, vel illum qui dolorem eum fugiat quo voluptas nulla pariatur?

Índice general

[Capítulo 1 Introducción 1](#_Toc41927807)

[1.1 Introducción 1](#_Toc41927808)

[1.2 Objetivos 2](#_Toc41927809)

[1.3 Estructura del proyecto 3](#_Toc41927810)

[Capítulo 2 Estado del Arte 5](#_Toc41927811)

[2.1 Introducción 5](#_Toc41927812)

[2.2 Listas y enumeraciones 5](#_Toc41927813)

[2.2.1 Listas 5](#_Toc41927814)

[2.2.2 Enumeraciones 5](#_Toc41927815)

[2.3 Tablas 6](#_Toc41927816)

[2.4 Figuras 6](#_Toc41927817)

[2.5 Código 7](#_Toc41927818)

[2.6 Bibliografía 7](#_Toc41927819)

[2.7 Conclusiones 7](#_Toc41927820)

[Capítulo 3 Metodología y Desarrollo 9](#_Toc41927821)

[3.1 Introducción 9](#_Toc41927822)

[3.2 Conclusiones 9](#_Toc41927823)

[Capítulo 4 Experimentos y Resultados 11](#_Toc41927824)

[4.1 Introducción 11](#_Toc41927825)

[4.2 Conclusiones 11](#_Toc41927826)

[Capítulo 5 Conclusiones y Trabajo Futuro 13](#_Toc41927827)

[5.1 Conclusiones 13](#_Toc41927828)

[5.2 Trabajo futuro 13](#_Toc41927829)

[Bibliografía 15](#_Toc41927830)

[Anexo I. Título del anexo 19](#_Toc41927831)

[I.1 Sección 19](#_Toc41927832)

[I.2 Sección 19](#_Toc41927833)

Índice de figuras

[**Figura 2.1.** Toda figura debe tener un pie descriptivo 7](#_Toc41491660)

Índice de tablas

[**Tabla 2.1.**  El título de la tabla se ha de colocar siempre encima de la tabla. Utilizar el estilo Titulo Tabla 6](#_Toc41491663)

[**Tabla 2.2.**  Utilizar el estilo Tabla con fila totales 6](#_Toc41491664)

# Introducción

## Introducción

### Importancia de la robótica en la educación

En la educación obligatoria, se plantea que las seis capacidades fundamentales que los estudiantes deben desarrollar durante esta etapa son: Resolución de problemas, Pensamiento crítico, Aprender a aprender, Trabajo colaborativo, Comunicación y Compromiso y responsabilidad. <https://www.argentina.gob.ar/sites/default/files/marco_nacional_de_integracion.pdf>

La introducción de la robótica en la escuela permite el trabajar las capacidades fundamentales de manera novedosa, proporcionando un enfoque práctico y lúdico para el aprendizaje en el que el alumno adquiere un papel más activo en la resolución de los problemas planteados, mediante el diseño y la experimentación, lo cuál se traduce en un aprendizaje de mayor calidad. Otra ventaja que hace que hace la robótica una excelente herramienta de aprendizaje es el feedback que obtienen del robot, el poder observar de forma física los resultados y de forma instantánea les permite una autoevaluación, poder medir su desempeño sin necesidad de que un adulto les diga si lo que han hecho está bien o mal, por esta misma razón cada alumno puede avanzar a su propio ritmo mientras se estimulan sus habilidades autodidactas. Estos resultados instantáneos también mejoran el autoestima y su afán de superación al ir resolviendo pequeños problemas poco a poco y ver como funcionan sus soluciones; al mismo esto también les ayuda a mejorar su tolerancia a la frustración.

Alfredo Pineda, profesor de informática de universidad y coordinador del curso de verano de la UPNA (Universidad Politécnica de Navarra) sobre el empleo de robots en los colegios, señala que “La robótica educativa ayuda a los alumnos a razonar; eso vale para Informática y para Filosofía”.

Muchos padres e incluso profesores pueden plantearse si introducir la robótica desde primaria es adelantar mucho, sin embargo, Alfredo Pineda, afirma todo lo contrario:

“No queremos que nuestros críos, o los adultos, sean robots, sino que sepan trabajar con ellos. Necesitamos movernos en la vida mediante botones, y hay que alimentar la curiosidad de los niños, que sepan qué funciona al apretar ese botón…”

<https://www.eldiario.es/navarra/ultimas-noticias/robotica-educativa-alumnos-informatica-filosofia_1_4697390.html>

La integración de la robótica en la educación puede tener muchos beneficios para los estudiantes, entre ellos:

* Aprendizaje práctico: la robótica brinda a los estudiantes una experiencia de aprendizaje práctico que les permite aplicar sus conocimientos en un entorno práctico. Esto puede hacer que el aprendizaje sea más atractivo y puede ayudar a los estudiantes a retener mejor la información.
* Resolución de problemas: la robótica requiere que los estudiantes piensen críticamente y resuelvan problemas, lo que puede ayudarlos a desarrollar importantes habilidades de resolución de problemas que son valiosas en muchos campos.
* Educación STEM: la robótica es un campo que combina conceptos de ciencia, tecnología, ingeniería y matemáticas (STEM). Al aprender sobre robótica, los estudiantes pueden obtener una comprensión más profunda de estos temas y cómo se relacionan entre sí.
* Oportunidades profesionales: la robótica es un campo emergente con muchas oportunidades laborales. Al aprender sobre robótica en la escuela, los estudiantes pueden comprender el campo y estar mejor preparados para seguir una carrera en él.
* Creatividad e innovación: la robótica anima a los estudiantes a pensar de forma creativa, a ser innovadores y a generar nuevas ideas.
* Colaboración: la robótica a menudo requiere trabajo en equipo y colaboración, lo que puede ayudar a los estudiantes a desarrollar importantes habilidades sociales y emocionales.
* Accesibilidad: la robótica también se puede utilizar para hacer que la educación sea más accesible para los estudiantes con discapacidades, brindándoles experiencias de aprendizaje interactivas y atractivas.

En general, la robótica tiene el potencial de inspirar a los estudiantes a aprender, a participar más en su educación, a pensar más críticamente y a estar más preparados para la futura fuerza laboral.

## Objetivos

El objetivo con este trabajo es la creación un coche programable similar al mBot y de bajo coste para que se pueda usar en la docencia con el objetivo de enseñar a programar haciendo que el dron funcione de manera autónoma.

La idea es que este dispositivo disponga de diversas funcionalidades las cuales puedan ser programadas por los usuarios. Para ello se pone a disposición este dispositivo con una serie de sensores y actuadores, los cuales serán detallados más adelante y una serie de librerías con funciones para trabajar con ellos. El objetivo de estas librerías de funciones es crear una API con funciones sencillas que abstraigan al usuario final de todos los aspectos técnicos y de electrónica. Como comentaba está destinado a un público inexperto y el objetivo es que empiecen por lo básico.

Este proyecto representa una oportunidad para aprender sobre la robótica y la programación de una forma práctica y divertida, ya que el mbot es una herramienta educativa muy versátil y accesible. Además, el éxito en el desarrollo de un coche programable similar al mbot podría tener un gran impacto en la educación y en la formación de futuros ingenieros y científicos en el campo de la robótica y la automatización.

## Estructura del proyecto

# Estado del Arte

## Introducción

En este apartado se detallarán algunos proyectos existentes los cuales han servido como referencia para la elaboración de este.

### Robots educativos

#### MBOT

Mbot es un robot educativo STEAM (Science, Technology, Engineering, Arts and Mathematics). El robot está recomendado para niños a partir de 8 años, el objetivo es que los niños aprendan programación, electrónica y robótica de forma práctica interactuando con el dispositivo a modo de juguete.

Este robot está diseñado por Makeblock, una empresa especializada en crear este tipo de kits de robótica. El dispositivo está compuesto por una serie de componentes modulares (\*detallados abajo\*) que se pueden ensamblar fácilmente y programar mediante una gran variedad de lenguajes de programación, entre los que se incluyen Python, C++ y Scratch. Resaltando este último ya que Makeblock ha desarrollado un software específico para programar el robot mediante Scratch.

Cabe destacar que es un proyecto open-source, lo que ha permitido que la comunidad cree más kits de expansión, para mejorar las capacidades de este /\*mencionar kits\*/. El kit básico tiene un precio de unos 90 euros y está compuesto por:

* **Partes mecánicas**. Son las piezas que permiten el movimiento y las que forman la estructura del robot. El robot está diseñado para que sea fácil de montar y las piezas mecánicas se pueden conectar mediante tornillos y conexiones a presión. En esta categoría se incluyen el chasis, las ruedas, la batería recargable y las demás piezas que forman el cuerpo
* **Electrónica**. Formada por la placa de control principal, los sensores y el resto de componentes electrónicos.

La placa de control principal está basada en Arduino UNO aunque con modificaciones realizadas por la gente de Makeblock para adaptarla a las necesidades de este kit, la placa ha sido rebautizada como mCore. Esta palca es la que posee la inteligencia del robot, es la encargada de controlar el movimiento y responder a las señales de los sensores.

Dispone de una serie de sensores que permiten la interacción con el entorno, entre los que destacan:

* + Un sensor de luminosidad.
  + Un sensor de ultrasonidos, para medir distancia a un obstáculo.
  + Receptor de Infrarrojos, para la comunicación con el mando.
  + Sensor sigue-líneas. Devuelve 0 si detecta línea (negra) y 1, si no (blanco de fondo).
  + Modulo bluetooth, permite conectarlo al smartphone, PC o tablet para controlarlo y programarlo.
  + Módulo de WiFi 2.4G, tiene el mismo objetivo que el módulo Bluetooth, pero este es útil cuando hay varios mBot juntos, para evitar interferencias.

También son necesarios una serie de actuadores, para permitir el movimiento y poder reaccionar a las señales de entrada de los sensores, estos actuadores son:

* Leds
* Buzzer, para emitir pitidos.
* Motores, permiten el movimiento.

Por último, mencionar los conectores para facilitar la conexión de otros componentes con la placa:

* Conectores RJ45. Similares a las habituales clavijas de teléfono, permiten conectar de forma sencilla componentes como el sensor de ultrasonidos o el sigue-lineas.
* Clavijas para los motores.
* **Software**. Como comentábamos anteriormente, se puede programar en varios lenguajes de programación, pero merece especial mención Scratch, ya que es el que principalmente va a ser empleado. Makeblock ha desarrollado un software llamado mBlock que permite precisamente esto, programar el robot en una versión de Scratch 2.0 modificada que incluye los bloques necesarios para dar instrucciones a nuestro robot.

//FOTOS

### AprilTag

Es un sistema fiduciario visual (Marcas de referencia) de filosofía open-source, creado por la Universidad de Michigan. Los ArpilTags, que es como se llaman estas marcas de referencia, son conceptualmente similares a los QR, ya que son un código de barras bidimensional con un patrón en blanco y negro único. Sin embargo, estos permiten codificar cargas útiles de datos más pequeñas de entre 4 y 12 bits, lo que permite una detección más robusta y mayor distancia de detección. Además, tienen la ventaja de estar diseñados para una alta precisión de localización, se puede calcular la posición precisa en 3D del TAG respecto a la cámara. Los marcadores AprilTag son resistentes a la oclusión parcial, la rotación y el escalado.

Los AprilTags se dividen en diferentes familias, con diferentes características para adaptarse a diferentes objetivos. Es importante entender la nomenclatura para poder diferenciar las familias. El primer número es el número de bits de datos, dicho de otro modo, el número de “bloques cambiables”, las familias con mayor número permiten generar más etiquetas diferentes de la misma familia. El segundo número indica la distancia de Hamming, el número mínimo de bits que se deben cambiar en el código de una etiqueta para alcanzar el código de otra etiqueta, las familias con mayor número en este caso permiten mayor tolerancia a fallos.

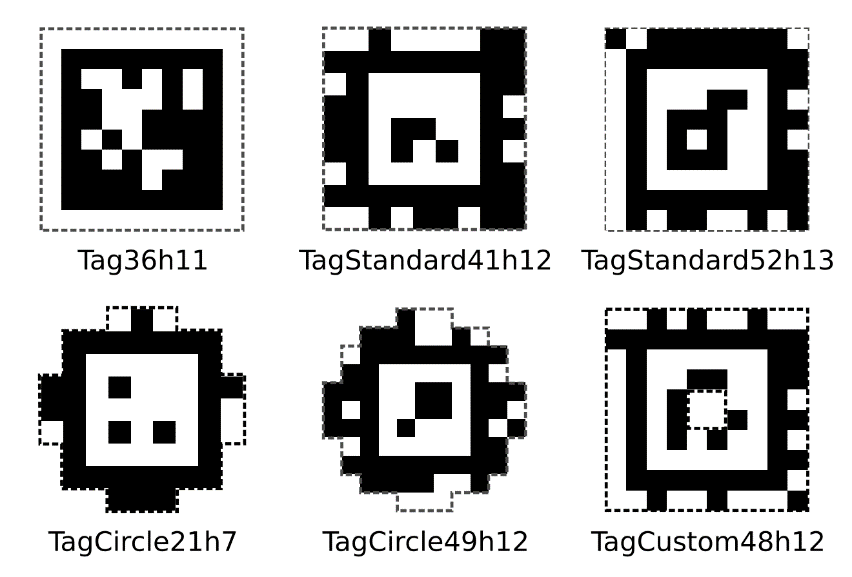


Ilustración 1. Familias de AprilTags

Por ejemplo para este caso se emplean las etiquetas de la familia 25h9, indica que son etiquetas de 25bits (matriz de 5x5) y con una distancia mínima de Hamming mínima de 9bits entre códigos. Esta familia permite crear hasta 35 etiquetas diferentes, la elección de esta familia esta principalmente condicionada por el hardware.

La biblioteca AprilTag proporciona una API en C/C++ y Python y sin dependencias externas para ser fácilmente integradas en otras aplicaciones, además están diseñadas para funcionar incluso en sistemas embebidos y en tiempo real. El sistema AprilTag puede funcionar con una amplia gama de cámaras, desde cámaras tradicionales hasta cámaras térmicas. Entre sus principales aplicaciones destacan la robótica, drones, realidad aumentada y calibrado de cámaras.

El algoritmo consta de nueve pasos:

1. Convierte la imagen a escala de grises de punto flotante (valores de píxel entre 0.0 y 1.0) y aplique un desenfoque Gaussiano.
2. Calcula el gradiente local (magnitud y dirección) en cada píxel.
3. Genera una lista de bordes, agrupando píxeles conectados con direcciones similares. Un borde está presente si la magnitud del gradiente para ambos píxeles está significativamente por encima de cero.
4. Crear grupos desde los bordes.
5. Pasar sobre los grupos, encajando líneas llamadas Segmentos.
6. Para cada segmento, busca segmentos que comiencen donde termina este segmento.
7. Buscar todos los segmentos conectados para encontrar bucles de longitud 4, llamados Quads. Cada quad representa el borde negro alrededor de una etiqueta candidata.
8. Decodificar los quads mirando los píxeles dentro del borde para ver si representan un código de etiqueta válido y generar una lista de TagDetections.
9. Buscar las TagDetections superpuestas y tome las mejores (la distancia de Hamming más baja o el perímetro más grande); desecha el resto.

El proceso del detector se puede optimizar mediante el uso de diferentes técnicas y algoritmos de procesamiento de imágenes, como el umbral, la detección de bordes y la coincidencia de patrones. La biblioteca también proporciona un conjunto de parámetros que se pueden ajustar para mejorar el rendimiento de detección en diferentes escenarios.

### Materiales usados

## Listas y enumeraciones

### Listas

Usar los estilos **Lista con viñetas** según el indentado necesario:

* Primer item: utilizar Lista con Viñetas

Continuar la descripción de item: Utilizar Lista

* Subitem: utilizar Lista con Viñetas 2

Continuar un subitem: utilizar Lista 2

* Subitem
* Sub-Subitem: utilizar Lista con Viñetas 3

Continuar un subitem: utilizar Lista 3

### Enumeraciones

Utilizar enumeraciones mediante los estilos **Lista con números**:

1. Enumeración: Lista con números
2. Enumeración 2: Lista con números 2
3. Enumeración: Lista con números 3

## Tablas

Se recomienda utilizar el mismo formato de tabla para todo el documento. En el documento se ofrecen dos diseños de tabla: Tabla con bandas (ver Tabla 2.1) y Tabla con fila totales (ver Tabla 2.2). La tabla siempre ha de aparecer referenciada en el texto utilizando Referencias Cruzadas. Esto permite que la numeración de las tablas se actualice automáticamente, por ejemplo, Tabla 2.1. Utilizar Referencias, Referencia cruzada, Tipo Tabla, Sólo Rótulo y Número para añadir la referencia a la tabla.

**Tabla 2.1.**  El título de la tabla se ha de colocar siempre encima de la tabla. Utilizar el estilo **Titulo Tabla** para formatear el encabezado de la tabla

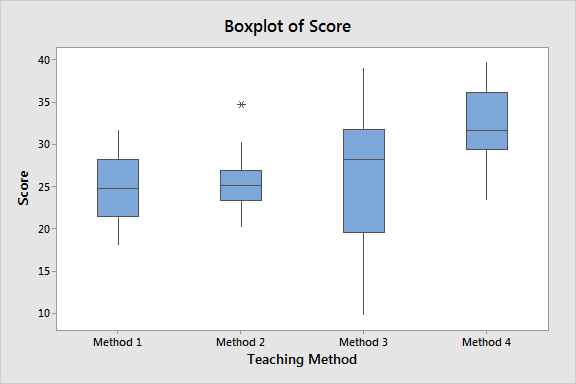
|  |  |  |
| --- | --- | --- |
| Columna 1 | Columna 2 | Columna 3 |
| Dato 1 | Dato 4 | Dato 6 |
| Dato 2 | Dato 5 | Dato 6 |
|  |  |  |
| Dato n | Dato n+1 | Dato n+2 |

**Tabla 2.2.**  Como se muestra la tabla con el diseño de tabla Tabla con fila totales

|  |  |  |
| --- | --- | --- |
|  | Columna 1 | Columna 2 |
| Fila 1 | Dato 4 | Dato 6 |
| Fila 2 | Dato 5 | Dato 6 |
|  |  |  |
| Fila n | Dato n | Dato n+1 |
| Total | Total 1 | Total 2 |

## Figuras

Al igual que ocurre con las tablas, toda figura tiene que estar referenciada en el texto, por ejemplo Figura 2.1. Utilizar Referencias, Referencia cruzada, Tipo Figura, Sólo Rótulo y Número. Seleccionar la imagen y aplicar el estilo Figura para ajustarla correctamente en el documento.



**Figura 2.1.** Toda figura debe tener un pie descriptivo

## Código

Para describir el código o pseudo-código en el documento utilizar el estilo Código:

**def** search (initial)

open = [initial]

**while** open

node = open.pop()

**if** test (node)

**return** node

**return** failure

**Código 2.1.** Código search

## Bibliografía

Se recomienda la utilización de un gestor de referencias como Mendeley o Endnote. Toda referencia que aparezca en la Sección de Bibliografía ha de aparecer en el texto. Un ejemplo de referencia en el texto sería (Babar, Kitchenham, Zhu, Gorton, & Jeffery, 2006).

Además, se recomienda utilizar estilos estándar para la bibliografía, por ejemplo el de American Psycological Association 6th Edition.

## Conclusiones

# Metodología y Desarrollo

## Introducción

## Materiales empleados

Por su similitud con el mBot, podemos dividir los componentes de este proyecto en las mismas categorías, de este modo también nos sirve para ver las diferencias existentes entre ambos:

### Partes mecánicas

El prototipo se ha construido a partir del kit *AptoFun 2WD Motor Smart Car Chassis for Arduino- with 2 Gear Motor and Battery Box*. Este kit incluye el chasis, dos motores y sus respetivas ruedas, una rueda universal para mantener el dispositivo en equilibrio, un zócalo para 4 pilas AA\* y sus conectores y toda la tornillería necesaria para armarlo.



Ilustración 2. Kit AptoFun, para parte mecánica

### Electrónica - Sensores

En este apartado se describirán los sensores empleados en el proyecto.

#### Sensor HC-SR04.

Es un sensor de ultrasonidos de bajo costo que se emplea para medir la distancia a un objeto, en un rango de 2-400cm con una precisión de ±3 mm. El módulo incorpora un emisor y un receptor piezoeléctricos y el circuito de control.

Un altavoz de color azul

Descripción generada automáticamente con confianza media

Ilustración 3. Sensor de ultrasonidos HC-SR04

Para este proyecto se han usado dos sensores de estos, uno para detectar obstáculos de la parte delantera y otro para la parte trasera. El modo de operación es:

1. Cuando le llega la señal por el pin TRIG (necesario mantener el nivel Alto de señal (5V) por al menos 10µs), el emisor lanza 8 pulsos de ultrasonidos (40KHz).
2. Las ondas viajan por el aire y rebotan al encontrar un objeto.
3. Estas ondas rebotadas son detectadas por el receptor y el in ECHO cambia a estado Alto (5V) por un periodo igual al que tardó la onda en emitirse y ser recibida (lo que se conoce como ECO).

Para calcular la distancia se usa la siguiente fórmula:

#### Tracking module.

También conocido como sensor sigue-líneas, este sensor no es otra cosa que un sensor de infrarrojos. La luz infrarroja es aquella que tiene una mayor longitud de onda o menor frecuencia que la que podemos ver los humanos.

Escala de tiempo

Descripción generada automáticamente con confianza baja

Ilustración 4. Espectro de luz visible

Está compuesto por un LED de luz infrarroja y un fotorreceptor colocados juntos, de forma que la luz emitida por el LED se refleja sobre superficies claras y llega al receptor, si no llega si es absorbida por superficies oscuras. Tiene una distancia de detección de entre 1 y 8 mm y la sensibilidad es ajustable mediante un potenciómetro.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Ilustración 5. Sensor sigue-líneas

La salida es digital, devuelve 0 si detecta negro (no recibe de vuelta los infrarrojos) y 1 en caso contrario. Este sensor se usa principalmente para construir robot sigue-líneas, estos robots siguen circuitos marcados por una línea negra sobre un fondo blanco o al revés, el modo de operación es colocar dos sensores paralelos y poder detectar si nos salimos por la izquierda o por la derecha.

Imagen que contiene Aplicación

Descripción generada automáticamente

Ilustración 6. Funcionamiento sensor sigue-lineas

<https://www.prometec.net/sensor-infrarrojo-siguelineas/>

#### Photoresistor module

Como su propio nombre indica, es un tipo de módulo que incorpora una fotorresistencia para detectar cambios en los niveles de luz. Una fotorresistencia o también conocida como resistencia dependiente de la luz (LDR), es un tipo de resistencia cuya resistencia cambia según la cantidad de luz a la que se expone. Cuando el nivel de luz aumenta, la resistencia del fotorresistor disminuye (hasta 500Ω) y cuando el nivel de luz disminuye, la resistencia del fotorresistor aumenta (hasta 50KΩ).

Diagrama, Esquemático, Gráfico de burbujas

Descripción generada automáticamente

Ilustración 7. Photoresistor module

Este dispositivo además del fotorresistor incorpora un circuito capaz de convertir estos cambios en la resistencia en un voltaje o corriente medible que puede ser leído por un microcontrolador u otro dispositivo electrónico para medir el nivel de luz.

Es importante tener en cuenta que la resistencia del fotorresistor se ve afectada por el tipo de material del que está hecho, la temperatura del entorno y la longitud de onda de la luz a la que está expuesto. Por lo tanto, la tensión o corriente de salida del módulo debe calibrarse para la aplicación y el entorno específicos en los que se utiliza.

#### Joystick

Este módulo se incluye en la categoría de sensores porque desde el punto de vista electrónico lo podemos ver como una combinación de potenciómetros, para controlar las dimensiones “X” (eje horizontal) e “Y” (eje vertical) y un botón, que equivale a la dimensión “Z” (profundidad).

Imagen de la pantalla de un celular

Descripción generada automáticamente con confianza media

Ilustración 8. Joystick module

El tipo de datos de los potenciómetros (dimensiones “X” e “Y”) son señales analógicas, tienen un rango de valor entre 0 y 4095, la posición central tiene aproximadamente un valor de 1960 (no siempre es así, en mi caso está entorno a 1872-1906); el del botón (dimensión “Z”), es digital. Es importante este detalle para poder operar con este módulo y conectarlo a los pines adecuados.

Imagen de la pantalla de un celular con letras

Descripción generada automáticamente con confianza media

Ilustración 9. Rangos del joystick

### Electrónica – Procesamiento

En este apartado se describen las placas empleadas para el procesamiento de las señales captadas por los sensores y para el control de las partes motrices.

#### ESP-WROOM-32 DevKit

Este es el “cerebro del coche”, es donde se procesarán todos los datos recibidos del resto de los sensores y de la cámara y desde dónde se controlarán las acciones.

Es una placa de desarrollo que incorpora el módulo ESP-WROOM-32 con el microcontrolador ESP32, hermano mayor del antiguo ESP8266. La función de la placa de desarrollo es albergar el microchip y la vez proporcionarnos las interfaces necesarias para interactuar con él, nos facilita acceso a sus pines (permite conectarle cables fácilmente y están serigrafiados) y en este caso también permite insertarla en una protoboard.

También proporciona componentes adicionales tales como un interfaz USB que sirve para programarlo y/o usarlo como puerto serie, los botones de RESET y BOOT y un LED integrado.

La placa posee un regulador de voltaje y puede ser alimentada por el USB, por los pines de 5v o 3.3v.

Respecto del módulo ESP-WROOM-32, es una PCB creado por Espressif, va dirigido a aplicaciones desde redes de sensores de baja potencia, hasta tareas más pesadas como codificación de voz, transmisión de música y decodificación de MP3. También incorpora WiFi, Bluetooth y Bluetooth LE, además de numerosos periféricos que van desde sensores táctiles capacitivos, sensores Hall, interfaz de tarjeta SD, Ethernet, SPI de alta velocidad, UART, I2S e I2C.

El núcleo de este módulo es el chip ESP32-D0WDQ6, el cuál posee dos núcleos que se pueden controlar individualmente y un coprocesador que puede ser usado para aplicaciones menos demandantes y ahorrar energía.

Esta PCB también incorpora una antena impresa tipo PCB para las conexiones WiFi y Bluetooth.

Imagen que contiene electrónica, circuito

Descripción generada automáticamente

Botones

Pines

Interfaz USB

ESP-WROOM-32con blindaje

Antena

LED integrado

<https://www.espressif.com/en/products/modules>

<https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf>

<https://programarfacil.com/esp8266/esp32/>

#### ESP32-CAM

Este módulo es el encargado de la parte de visión artificial, aquí se incluye la inteligencia de detección y decodificación de los TAGs.

Esta placa de desarrollo también incorpora conectividad WiFi y Bluetooth, aunque menos pines debido a su reducido tamaño. La principal diferencia es que este módulo está diseñado para integrarle un sensor de cámara, tiene soporte para sensores OV2640 y OV7670, en nuestro caso usamos el OV2640, de 2Megapixeles.

No posee interfaz USB, por lo que se debe programar usando el puerto serie (\*explicar).

En cuanto al chip usado, en este caso es un ESP32-S con antena WiFi

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

### Electrónica – Actuadores

En este apartado se describen los dispositivos empleados para el movimiento del robot.

#### L298N

Este módulo es el controlador de los motores, permite encender y controlar los dos motores desde el ESP32, variando tanto la dirección como la velocidad de giro. Por lo general los microcontroladores, como es el caso de nuestro ESP32 no disponen de potencia suficiente para mover los actuadores, su función es “ordenar” a los controladores que hagan este trabajo.

La corriente máxima que puede suministrar el L298N es 2A y una tensión de alimentación de 3V a 35V. /\*Poner que no puede entregar todo por las perdidas?\*/.

En cuanto a su funcionamiento, consta de dos puentes-H, uno para la salida A y otro para la salida B. Estos puentes son los encargados de permitir cambiar el sentido de la corriente para poder cambiar el sentido de giro.

<https://www.luisllamas.es/arduino-motor-corriente-continua-l298n/>

<https://naylampmechatronics.com/blog/11_tutorial-de-uso-del-modulo-l298n.html>

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Ilustración 10. Controlador L298N

En cuanto al funcionamiento del controlador, fijándonos en la imagen tenemos:

* Los pines IN1, IN2, controlan la dirección de giro del motor A. Los respectivos IN3, IN4, son para el motor B.
* Los pines ENA y ENB son para activar/desactivar la salida, podemos tenerlos conectados siempre mediante un jumper o como es nuestro caso controlarlos mediante una señal PWM para variar la velocidad de giro.
* En la parte inferior, también tenemos una bornera para la alimentación, cuyo comportamiento depende del “Regulador”:
  + Si el regulador esta activado (jumper cerrado), el pin 5v se comporta como una salida de 5v que podemos usar para alimentar otros dispositivos. En este caso se debe alimentar mediante el pin 12v con una tensión de 6-12v.
  + Si el regulador esta desactivado (jumper abierto), el pin 5v se comporta como una entrada a la que tendremos que proporcionar un voltaje de entre 4.5-5v.

En nuestro caso el regulador esta activado (jumper cerrado), se alimenta con dos baterías en serie para obtener 6v y el pin 5v se usa para alimentar al ES32 y el resto de los componentes.

* Por último, a ambos lados tenemos las borneras que suministran la salida a ambos motores.

#### Servomotor SG90

Es un pequeño actuador rotativo que permite un control preciso de la posición angular, en el caso del servomotor SG90 puede girar entre 0º y 180º.

Imagen que contiene interior

Descripción generada automáticamente

Ilustración 11. Servomotor SG90

Para funcionar necesita una señal de PWM, en ella el tiempo en HIGH es equivalente al ángulo (o posición) del servo. Estos valores van desde 0.5-1ms para 0º y de 2-2.4ms para 180º, el periodo de la señal debe ser cercano a 20ms.

Forma

Descripción generada automáticamente

Ilustración 12. PWM y servomotor

### Software

En este apartado se describe todo el software usado para implementar y cargar el código en los diferentes dispositivos programables, como pueden ser el ESP32 y el ESP32-cam; para diseñar el circuito electrónico y para diseñar el chasis.

#### Arduino IDE (Integrated Development Environment)

Es un entorno de desarrollo que ofrece un conjunto de herramientas software para permitir a los programadores desarrollar, depurar, cargar y ejecutar nuestros programas (llamados “sketches”) en placas Arduino.

Aunque el enfoque inicial fue para trabajar con placas Arduino, gracias a ser open-source y a la comunidad, podemos emplearlo para trabajar con otros tipos de placas como es en nuestro caso las ESP32 y ESP32-cam. Usa un lenguaje de programación de alto nivel propio, denominado Procesing, el cual está basado en C++, por lo que su sintaxis es muy similar.

Este entorno se puede ampliar mediante el uso de librerías, al igual que la mayoría de entornos de programación. Estas librerías brindan funcionalidades adicionales para usar en los Sketch, por ejemplo, para trabajar con hardware o manipular datos. Existen librerías oficiales como por ejemplo para los protocolos SPI, I2C y UART y luego podemos incluir bibliotecas de terceros o incluso diseñadas por nosotros mismos.

<https://arduino.cl/programacion/>

Captura de pantalla de computadora

Descripción generada automáticamente

Ilustración 13. IDE Arduino

#### Librerías usadas

Como comentaba en el apartado anterior, el IDE de Arduino nos permite incorporar librerías para ampliar la capacidades de este. Para este proyecto las librerías adicionales que se han usado son:

* Librerías de ESP32. Para poder usar las placas con el chip ESP32 en el IDE de Arduino es necesario añadir las URLs de estas placas para poder descargar el núcleo y las librerías necesarias de ESP32 para Arduino.
* Librerías de AprilTag. Existe la librería oficial creada por AprilRobotics, pero para nuestro caso usaremos una adaptación de esta para ser usada en un microcontrolador ESP32, esta adaptación ha sido diseñada por Satoshi Tanaka y la podemos descargar desde su repositorio de GitHub:

<https://github.com/stnk20/apriltag/tree/esp-idf>

* Librería de servomotor para ESP32. La librería oficial de Arduino (Servo.h), no es compatible con el esp32. Por suerte podemos encontrar una adaptación realizada por Kevin Harrington, que permite el control de hasta 16 servos en canales individuales y el control de Tone y analogWrite. También hace uso de los temporizadores de ESP32 PWM.
* Elegoo?

### Diseño electrónico.

Para la representación de las conexiones y la creación del diseño de la PCB, el software usado es Fritzing. La elección de este software es debido a que permite encontrar modelos de prácticamente cualquier componente para incorporar en nuestros bocetos, solo tenemos que descargar su archivo .fzpz e importarlo.

Imagen que contiene Escala de tiempo

Descripción generada automáticamente

Ilustración 14. Fritzing

### Diseño de chasis

## Desarrollo

### Comunicación ESP32 y ESP32-CAM

La interacción entre estos dos dispositivos consisten en que el ESP32-CAM, va capturando imágenes, las procesa y envía órdenes por el puerto serie al ESP32 para que este las transmita al resto de módulos.

El ESP32-CAM actúa como maestro, como comentaba anteriormente, va capturando imágenes, en caso de detectar AprilTAGs, los almacena en un array de tamaño dinámico que almacena estructuras con los parámetros extraídos de estos Tags como son su identificador y su posición 3D (x,y,z). Una vez hecha la imagen y la detección, podemos acceder al array para ver los tags detectados.

Para enviar órdenes al ESP32 se emplea el puerto serie, es un protocolo sencillo orientado a conexión, con mensaje de respuesta (ACK) y checksum para la integridad de los datos y con timeout configurable.

En primer lugar, se establece comunicación enviando el carácter ‘SYN’ al ESP32, este debe responder con un ‘ACK’, si el mensaje de respuesta llega correctamente en el tiempo indicado (timeout), se establece la comunicación.

Luego cada vez que vayamos a enviar una orden se comprueba que la conexión siga existiendo, si no es así se repite el proceso; si ya existe, se empieza enviando el carácter ‘STX’ (Start of TexT) para saber que lo que se va a enviar es una orden, esta orden se encapsula en una estructura predefinida en ambos dispositivos cuyos parámetros son un carácter para indicar el tipo de acción, un entero para indicar la velocidad (0-100) en caso de movimiento o la nota en caso del buzzer y un float para indicar el tiempo que dura la acción en segundos, la orden va seguida de checksum computado mediante el método XOR (\*incluir anexo\*) y del carácter ‘ETX’ (End of TexT), para indicar el fin de la orden.

Tras enviar la orden es ESP32-CAM, queda esperando el tiempo indicado en la orden (lo que durará la acción) a la respuesta del carácter ‘EOA’ (End Of Action) por parte del ESP32.

En caso de cualquier error ya sea por trama corrupta o tiempo de timeout excedido el protocolo es cortar la conexión con ‘EOT’ y volver a establecerse.

//DIAGRAMA DE SECUENCIA

La idea es que el procedimiento sea capturar una imagen y detectar los TAGs, luego hacer alguna acción en función de los tags detectados. En secciones posteriores se describen algunos ejemplos.

### Comunicación ESP32 – mando

Como mencionaba en anteriores secciones, el ESP32 incorpora conectividad bluetooth. El mando también esta conformado por otro ESP32.

### ESP32 y resto de componentes

## Conclusiones

Como se puede observar

# Experimentos y Resultados

## Introducción

### Obtener la distancia focal

### Mejorar calidad imagen

### Búsqueda de ArpilTAGs según orden dado

### Manejo mediante el mando BLE

## Conclusiones

# Conclusiones y Trabajo Futuro

## Conclusiones

## Trabajo futuro

Este Proyecto se da por completado, sin embargo, durante el desarrollo de este he ido detectando posibles cambios que se podrían aplicar en posteriores versiones y que por falta de tiempo o limitaciones técnicas derivadas de las capacidades del hardware o falta de colaboradores no se han podido llevar a cabo. Estas posibles modificaciones, si se implementan correctamente podrían aumentar considerablemente las capacidades de este robot, mejorando su desempeño e incluso añadiendo nuevas funcionalidades. Estas posibles mejoras son:

* Como el ESP32-CAM cuenta con conexión WiFi, esta se puede emplear para derivar el procesamiento de la imagen a un computador u otro microcontrolador y liberar de esta carga al ESP32-CAM, así de este modo su única función sería la de capturar las imágenes y enviarlas. Esto también nos permitiría procesar las imágenes en un dispositivo más potente como es el caso de un ordenador con tarjeta gráfica dedicada y añadir funciones como por ejemplo detección de objetos.
* Modificar los parámetros de la cámara para poder obtener una mayor distancia tasa de detección de los tags. La opción de sustituir el sensor queda descartada porque el módulo ESP32-CAM solo es compatible con el OV2640 (usado en el proyecto) y el OV7670, el cuál ofrece menos resolución, por tanto no mejoraríamos.

Siguiendo con este objetivo la mejor solución pasaría por sustituir el módulo por otro más potente, aunque esto encarecería los costes.

* Se puede usar la conectividad Bluetooth del ESP32, de la parte del coche para conectar con nuestros smartphone para poder controlarlo.

En cualquier caso este proyecto se ha desarrollado en diferentes librerías separadas según la naturaleza de cada componente y siempre buscando ser lo más modular y sencillo de entender posible, con el objetivo de que este proyecto sirva como base para el desarrollo de nuevos proyectos de la misma naturaleza.

Bibliografía

Alexa Internet Inc. (2014). The top 500 sites on the web. Retrieved February 11, 2014, from http://www.alexa.com/topsites

Anda, B., Sjøberg, D., & Jørgensen, M. (2001). Quality and Understandability of Use Case Models. In 15th European Conference on Object-Oriented Programming (ECOOP’01) (pp. 402–428). London, UK: Springer Berlin Heidelberg. https://doi.org/10.1007/3-540-45337-7\_21

Babar, M. A., Kitchenham, B. A., Zhu, L., Gorton, I., & Jeffery, R. (2006). An empirical study of groupware support for distributed software architecture evaluation process. Journal of Systems and Software, 79(7), 912–925. https://doi.org/10.1016/j.jss.2005.06.043

Basili, V. R., Caldiera, G., & Rombach, H. D. (1994). The Goal Question Metric Approach. In Encyclopedia of Software Engineering (Vol. 2, pp. 528–532). Wiley. Retrieved from http://wwwagse-old.informatik.uni-kl.de/pubs/repository/basili94b/encyclo.gqm.pdf

Basili, V. R., Shull, F., & Lanubile, F. (1999). Building Knowledge through Families of Experiments. IEEE Transactions on Software Engineering, 25(4), 456–473. https://doi.org/10.1023/A:1009742216007

Bereiter, C. (2002). Education and Mind in the Knowledge Age (1st ed.). Routledge. Retrieved from http://www.amazon.com/Education-Mind-Knowledge-Carl-Bereiter/dp/0805839437

Biostat Inc. (2006). Comprehensive Meta-Analysis. Retrieved April 17, 2013, from http://www.meta-analysis.com/

Booch, G., Rumbaugh, J., & Jacobson, I. (2005). The Unified Modeling Language User Guide (2nd ed.). Addison-Wesley Professional.

Canfora, G., Cimitile, A., Garcia, F., Piattini, M., & Visaggio, C. A. (2006). Evaluating advantages of test driven development. In 2006 ACM/IEEE international symposium on International symposium on empirical software engineering (ISESE’06) (p. 364). Rio de Janeiro, Brazil: ACM Press. https://doi.org/10.1145/1159733.1159788

Castro, J., Kolp, M., & Mylopoulos, J. (2001). A requirements-driven development methodology. In 13th Int. Conf. On Advanced Information Systems Engineering (CAiSE’01) (pp. 108–123). London, UK: Springer-Verlag. https://doi.org/10.1007/3-540-45341-5\_8

Celko, J., Davis, J. S., & Mitchell, J. (1983). A demonstration of three requirements language systems. ACM SIGPLAN Notices, 18(1), 9–14. https://doi.org/10.1145/948093.948094

Cockburn, A. (2000). Writing Effective Use Cases (1st ed.). Addison-Wesley Professional.

Cruz-Lemus, J. A., Genero, M., Caivano, D., Abrahão, S., Insfrán, E., & Carsí, J. A. (2011). Assessing the influence of stereotypes on the comprehension of UML sequence diagrams: A family of experiments. Information and Software Technology, 53(12), 1391–1403. https://doi.org/10.1016/j.infsof.2011.07.002

Cruz-Lemus, J. A., Genero, M., Manso, M. E., Morasca, S., & Piattini, M. (2009). Assessing the understandability of UML statechart diagrams with composite states—A family of empirical studies. Empirical Software Engineering, 14(6), 685–719. https://doi.org/10.1007/s10664-009-9106-z

Cruz-Lemus, J. A., Maes, A., Genero, M., Poels, G., & Piattini, M. (2010). The impact of structural complexity on the understandability of UML statechart diagrams. Information Sciences, 180(11), 2209–2220. <https://doi.org/10.1016/j.ins.2010.01.026>.

Cysneiros, L. M., & Yu, E. S.-K. (2004). Non-Functional Requirements Elicitation. In J. C. S. do Prado Leite & J. H. Doorn (Eds.), Perspectives on Software Requirements (pp. 115–138). Springer US. https://doi.org/10.1007/978-1-4615-0465-8\_6

Damian, D. (2001). An empirical study of requirements engineering in distributed software projects: is distance negotiation more effective? In 8th Asia-Pacific Software Engineering Conference (APSEC’01) (pp. 149–152). Macao, China: IEEE Comput. Soc. https://doi.org/10.1109/APSEC.2001.991471

De Lucia, A., Fasano, F., Oliveto, R., & Tortora, G. (2006). Can Information Retrieval Techniques Effectively Support Traceability Link Recovery? In 14th IEEE International Conference on Program Comprehension (ICPC’06) (pp. 307–316). Athens, Greece: IEEE. https://doi.org/10.1109/ICPC.2006.15

Dieste, O., Fernández, E., García Martínez, R., & Juristo, N. (2011). Comparative analysis of meta-analysis methods: when to use which? In 15th International Conference on Evaluation & Assessment in Software Engineering (EASE’11) (pp. 36–45). Durham, UK: IET.

Dourish, P., & Bellotti, V. (1992). Awareness and coordination in shared workspaces. In ACM conference on Computer-supported cooperative work (CSCW’92) (pp. 107–114). Toronto, Canada: ACM Press. https://doi.org/10.1145/143457.143468

1. Título del anexo
   1. Conceptos técnicos
      1. Señal PWM

PWM son las siglas de Pulse Width Modulation (Moduclación por Ancho de Pulso, en español). Para transmitir cualquier señal, ya sea analógica o digital, se debe modular para que se pueda transmitir sin perder señal o sufrir distorsión por las interferencias.

PWM es una técnica que permite “emular” señales analógicas mediante una señal digital. Esta técnica permite modificar el ciclo de trabajo (duty cycle) de una señal, mientras se mantiene constante el periodo. El ciclo de trabajo de una señal periódica, es el tiempo que la señal se mantiene en estado alto en relación al periodo y se expresa en porcentaje.

Básicamente, consiste en generar pulsos positivos que se repiten de manera constante activando una salida digital durante un tiempo y apagándola durante el resto. El promedio de esta tensión de salida, a lo largo del tiempo, será igual al valor analógico deseado. El voltaje promedio se calcula con:

Normalmente es 0v y será 5v o 3.3v.

* + 1. Puente – H

A modo de resumen un puente-H podemos decir que es un circuito electrónico que permite activar los motores eléctricos, hacer que giren hacia un sentido u otro y al mismo tiempo permite controlar variables como, velocidad y torque de estos.

La construcción de un puente-H consiste en 4 transistores conectados entre VCC y GND, actuando sobre estos 4 transistores podemos variar el sentido de la corriente que lo atraviesa. El termino de “Puente-H” viene de la representación gráfica del circuito.

* 1. Programación de la ESP32-CAM

En esta sección se describen los requisitos previos y los pasos necesarios para poder cargar nuestros programas en la ESP32.

* + 1. Requisitos previos
* Tener actualizada la librería de ESP32.
* Disponer de un adaptador UART a USB o un programador específico, esto es debido a que este módulo no dispone de puerto USB, debemos de programarlo usando el interfaz serie (pines Rx yTx).
* Si usamos una distribución de Linux, tendremos que instalar, si no lo está, el modulo de comunicación serial de Python. Es tan sencillo como usar este comando: pip3 install pyserial.
  + 1. Posibles complicaciones
* Si usamos Windows11 es posible tener problemas con los drivers del adaptador serie. Si accedemos al administrador de dispositivos, aparece cualquiera de estos dos mensajes:

PL2303TA DO NOT SUPPORT WINDOWS 11 OR LATER PLEASE CONTACT YOUR SUPPLIER

THIS IS NOT PROLIFIC PL2303. PLEASE CONTACT YOUR SUPPLIER

La mejor solución en mi caso fue usar una máquina virtual de Linux.

* Si necesitamos activar el PSRAM, tenemos que cambiar la placa a “ESP32 Dev Module”. Vamos a Tools->PSRAM->Enable. Volvemos a cambiar a “Ai Thinker”.
* //Incluir la dirección de la libreia actualizada
  + 1. Cargar programa en ESP32-CAM

1. En el IDE de Arduino debemos seleccionar la placa: “Ai Thinker ESP32”

Imagen que contiene Texto

Descripción generada automáticamente

1. Para conectarlo al PC mediante algún tipo de adaptador, las conexiones de los pines de Rx y Tx deben ir cruzadas con los del adaptador, para poder comunicarse. En cuanto a la alimentación es recomendable conectarla al pin de 5v, con el de 3.3v no suele funcionar correctamente.

Para poder programarla, además deberemos de puentear los pines GPIO 0 y GND. Para usarla luego, es quitar el puente y reiniciar con el botón de RESET.

//Incluir diagrama de conexión//