## Diseño electrónico del robot.

### Alimentación del robot.

#### Módulo de Alimentación.

La alimentación del robot, en general, se realiza mediante el módulo de alimentación, que es un módulo que contiene una batería LiPo de 7.4 V (dos celdas de 3.7 V). Este módulo realiza el suministro de corriente a todo el circuito tal y como se muestra en la siguiente imagen:

|  |
| --- |
|  |
| Esquema eléctrico de la alimentación del robot. |

#### Módulos principales.

Tanto el módulo de control como el de comunicación contienen una placa MCU y además un regulador de tensión (placa 5V) que convierte la tensión principal de 7.4 V a 5 V para poder alimentar de esta forma la placa controladora.

#### Módulos secundarios.

La alimentación de los módulos de sensor, y cualquier módulo secundario, se realiza mediante una derivación en el módulo de comunicación de la conversión de 7.4 V a 5 V para poder alimentar hasta tres componentes con una alimentación de 5 V. La idea principal es poder alimentar sensores, motores DC y servomotores.

|  |
| --- |
|  |
| Alimentación de los módulos conectados al módulo de comunicación. |

#### Módulo de locomoción.

El módulo de locomoción contiene un controlador de motores DC, modelo L9110, alimentado con 5 V de señal de entrada, dada a través de una placa 5V.

|  |
| --- |
|  |
| Controlador L9110. |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | **Característica** | **Valor** | |  | 2.5 V – 12 V | |  | 0 – 2 µA | |  | 200 µA – 500 µA | |  | 750 mA – 850 mA | |  | 1.5 A – 2 A | |
| Características técnicas del controlador L9110. |

#### Consumo de corriente de los módulos.

El módulo de control y el módulo de comunicación consumen lo mismo ya que están formados de los mismos componentes: placa NodeMCU y placa 5V. En la placa NodeMCU se tiene un consumo nominal de 80 mA, pero en transmisión de datos se alcanzan los máximos de 170 mA.

El módulo de locomoción atendiendo a las características técnicas puede tener un pico de hasta 2 A.

Para poder garantizar una duración amplia de la batería, permitiendo una mayor autonomía del robot y dado que se ha pensado el robot para tener una alimentación general global en todo el robot, con lo cual si se van añadiendo módulos el consumo aumentaría, de forma que con vistas a una ampliación del robot mediante diferentes módulos la batería usada es una batería Li-Po de capacidad 3000 mAh.

### Conector.

El conector es la parte del módulo que permite la conexión entre dos módulos y se encarga de transmitir las señales eléctricas entre ellos. Se busca utilizar como conector algo similar a lo siguiente:

|  |  |
| --- | --- |
|  |  |
| Ejemplo de conector magnético. Imagen obtenida de https://es.aliexpress.com | |

Estos conectores están formados por:

1. Carcasa del conector.
2. Conjunto de pines Pogo-Pin.
3. Uno o más imanes que permiten una conexión magnética entre conectores.

La idea principal de usar este tipo de conexión se debe a la intención de que la conexión entre los módulos se haga sin tener que encajar pines eléctricos ni tener que usar enchufes.

La parte electrónica del conector son los pines conductores tipo Pogo-pin.

#### Pines conductores tipo Pogo-Pin.

|  |
| --- |
|  |
| Conductor Pogo-Pin de dos pines. |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | **Característica** | **Valor** | | **Datos Generales** | | | Tiempo de funcionamiento | 100.000 usos | | **Datos Eléctricos** | | | Voltaje máximo | 500 V | | Corriente nominal | 1 A – 5 A | | **Datos Térmicos** | | | Temperatura nominal de funcionamiento |  | | Temperatura máxima |  | | Temperatura máxima (plástico) |  | |
| Características técnicas del pin tipo Pogo-Pin usado en el proyecto. |

La cantidad de pines a utilizar depende de los datos que se quieran transmitir, en este caso, el mínimo es de dos conductores () ya que se busca transmitir la señal de alimentación principalmente, pero se ha optado por tener cuatro pines para poder transmitir señales digitales de control para los sensores y/o motores.

Por tanto se dispone de dos señales digitales (DIO) que se pueden configurar como: In-In, Out-Out o In-Out, aumentando la versatilidad del conector.

En el conector diseñado hay dos tipos de disposición de los pines, según sea conductor M o conductor H (usando como referencia física la parte superior del conductor), como se muestra en la siguiente imagen:

|  |
| --- |
|  |
| Disposición de los pines en los dos tipos de conectores. |

### Conexiones digitales del robot.

#### Módulo de control.

El módulo de control, en el proyecto, tiene un lado como si fuera el módulo de comunicación para conectar el módulo de locomoción. Por tanto tiene dos pines digitales, configurados como salidas (Out-Out) ya que su función es activar los motores DC conectados en el módulo de locomoción.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | **Función** | **Pin NodeMCU (GPIO)** | | Activación motor 1 | D1 (5) | | Activación motor 2 | D3 (0) | | Dirección de giro del motor | D2 (4) | |
| Distribución de los pines digitales en el módulo de control. |

En un modelo teórico del proyecto, este módulo no dispone de pines digitales conectados ya que solo tendría conectados módulos de comunicación, hasta un máximo de 3 módulos, pero por simplificación del robot y por un modelado 3D más uniforme el módulo de control ejerce como el módulo de comunicación que se comunica con el módulo de locomoción.

#### Módulo de comunicación.

El módulo de comunicación, en el proyecto, tiene dos sensores de proximidad por ultrasonidos conectados por lo que tiene dos pares de pines digitales configurados como entrada-salida (In-Out) ya que su función es enviar una señal de activación del sensor y recibir la señal captada por el sensor.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | **Función** | **Pin NodeMCU (GPIO)** | | Sensor1 input | D1 (5) | | Sensor1 output | D2 (4) | | Sensor2 input | D7 (13) | | Sensor2 output | D8 (15) | |
| Distribución de los pines digitales en el módulo de comunicación. |

|  |
| --- |
|  |
| Distribución de los pines digitales entre módulo principal y módulo secundario. |

En un modelo teórico del proyecto, este módulo dispone de tres pares de pines digitales para poder conectar módulos de elementos (de sensor, motor DC o servomotor), hasta un máximo de 3 módulos, configurados según el elemento final conectado. Este módulo sería el único que tiene contacto con los elementos.

#### Módulo de locomoción.

El módulo de locomoción y su conexión con el módulo de control dispone de un pin digital extra, esto se debe a que para controlar un motor se necesitan dos pines digitales: pin de activación y pin de dirección. En el proyecto, el módulo de locomoción tiene dos motores DC, ya que facilitaba de esta forma el montaje del robot y tenía una mejor estabilidad de las piezas del módulo. Por ello, se ha dispuesto de un pin digital extra en el conector del módulo para que controle la dirección de giro de los motores, de manera simultánea, y la señal de activación de los motores vendrá dada por los dos pines digitales que normalmente tiene el conector.

### Microcontrolador del robot.

Para controlar el robot se busca un microcontrolador con la capacidad de conexión inalámbrica vía Wi-Fi, ya que se busca que los módulos se comuniquen de esta forma inalámbrica. Por ello se ha tenido en cuenta el microcontrolador ESP8266 debido a que es fácil de encontrar en el mercado con un precio aceptable, tiene muchas versiones que se adaptan a diferentes situaciones y tamaños. Además, como característica principal que ha llevado a su elección es la compatibilidad con Arduino lo que permite una mayor facilidad de programación y compatibilidad de componentes. Entre las versiones de este microcontrolador que se pueden encontrar más fácilmente en el mercado, ya que hay muchas descatalogadas, son: la versión principal del microcontrolador o ESP-01 y la versión más moderna en forma de placa PCB NodeMCU (versión V3).

#### Versiones del ESP8266.

ESP-01.

|  |  |
| --- | --- |
|  |  |
| |  |  |  | | --- | --- | --- | | **Nº** | **PIN** | **Función** | | 1 | GND | GND | | 2 | GPIO 2 | IO digital | | 3 | GPIO 0 | IO digital | | 4 | RX (GPIO 3) | UART0 data received / IO digital | | 5 | TX (GPIO 1) | UART0 data send / IO digital | | 6 | CH\_PD | Chip enable | | 7 | RST | Reset de la placa | | 8 | Vcc | Alimentación a 3.3 V | | |
| Pines de ESP-01. | |

Características de la placa:

* 1 pin de alimentación a 3.3 V.
* 1 pin GND.
* 4 pines digitales para uso general (GPIO).
* Memoria SPI flash externa de 1 MB.

NodeMCU V3.

|  |
| --- |
|  |
| Pines de NodeMCU 1.0 versión LoLin. |

Características de la placa:

* 1 pin de alimentación a 5 V [VIN] con una regulación a 3.3 V para alimentar el microprocesador con un regulador AMS1117 (Tanto la versión Wemos como la versión Lolin usan el mismo regulador). El pin de alimentación a 5 V está conectado a un puerto µUSB para poder alimentarlo a través de este puerto.
* 1 pin destinado a salida de tensión a 5 V [VU] a través de la alimentación de la placa.
* 3 pines de salida de tensión a 3.3 V.
* 4 pines GND.
* 13 pines digitales para uso general (GPIO).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | **MCU Lolin** | **ESP8266EX (GPIO)** | | D0 | 16 | | D1 | 5 | | D2 | 4 | | D3 | 0 | | D4 | 2 | | D5 | 14 | | D6 | 12 | | D7 | 13 | | D8 | 15 | | D9(RX) | 3 | | D10(TX) | 1 | | D11(S2) | 9 | | D12(S3) | 10 | |
| Equivalencias de entradas digitales entre ESP8266 y NodeMCU. |

* 1 pin de entrada analógica.
* Memoria SPI flash externa de 4 MB.

#### Comparación entre ESP-01 y NodeMCU V3.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Modelo** | **Pines GPIO** | **Vcc (V)** | **Dimensiones (mm)** | **Memoria Flash (MB)** | | ESP 01 | 4 | 3.3 | 24.8x14.3 | 1 | | NodeMCU LoLin | 13 | 5 | 30x58 | 4 | |
| Tabla comparativa ESP-01 y NodeMCU V3 Lolin. |

La mayor ventaja de la placa ESP-01 está en su tamaño, fácilmente adaptable a cualquier forma, permitiendo reducir los tamaños de las piezas. Se dispone de cuatro pines digitales libres, que podrían aumentarse mediante placas especiales externas pero complicaría el circuito y aumentaría el tamaño del conjunto. Para realizar una conexión mediante USB se necesita acoplar el USB de manera externa, lo que haría aumentar su tamaño, dificultaría la localización de la pieza y complica el circuito añadiendo cableado. Además, si se acopla el USB externo se necesitaría acoplar el regulador de voltaje de forma externa a este USB para conseguir los 3.3 V de alimentación necesaria para la placa ESP-01 y también se pierde dos de los pines digitales libres que deberían actuar de RX y TX de la comunicación USB.

La placa MCU dispone de una mayor cantidad de pines digitales libres, además de disponer de una alimentación a 5 V, incorporando la conexión por µUSB lo que permite una conexión al PC más sencilla y rápida. Su tamaño es mayor que el ESP-01 pero incorpora la conexión USB, la regulación del voltaje a 3.3 V y se evita el aumento de cableado al estar soldados los componentes a una placa. Además la placa MCU incluye una salida de voltaje a 5 V a través del USB lo que permite incorporar componentes que se alimenten a ese voltaje. También incluye una entrada analógica que aumenta la variedad y funcionalidad de los sensores acoplados.

Con todo esto, ya que ofrece una mayor versatilidad y el único defecto es tener un tamaño mayor pero con todos los componentes (ESP8266, mayor número de pines digitales, conexión USB y regulador de tensión) incorporados, se ha optado por utilizar la placa MCU.

### NodeMCU V3.

|  |
| --- |
|  |
| Pines de NodeMCU V3. |

Los componentes principales de la placa MCU son:

* ESP8266 (versión ESP-12E).
* Alimentación por USB de 5V.
* Chip CH340G para el manejo del USB.
* Regulador AMS1117 de tensión de 5V a 3.3V.
* Botones RST y FLASH.

#### ESP8266 (ESP-12E).

Distribución de los pines digitales (GPIO):

* Los pines GPIO 6, 7, 8 y 11 están destinados a la memoria SPI Flash externa y por ello no han sido contados como pines libres GPIO.
* Los pines GPIO 9 y 10 son pines digitales libres pero no se recomienda su uso ya que son una posible ampliación de memoria SPI Flash externa.
* Los pines GPIO 1 y 3 son pines digitales libres pero no se recomienda su uso ya que son los pines destinados al uso de la comunicación serie, usada por el USB para cargar los programas de Arduino (sketch).

|  |
| --- |
|  |
| |  |  | | --- | --- | | **Nº** | **PIN** | | 1 | RST (Reset) | | 2 | ADC | | 3 | EN (CH-PD) | | 4 | GPIO 16 | | 5 | GPIO 14 | | 6 | GPIO 12 | | 7 | GPIO 13 | | 8 | VCC (3,3 V) | | 9 | CSO, GPIO 11 | | 10 | MISO, GPIO 7 | | 11 | GPIO 9 | | 12 | GPIO 10 | | 13 | MOSI, GPIO 8 | | 14 | SCLK, GPIO 6 | | 15 | GND | | 16 | GPIO 15 | | 17 | GPIO 2 | | 18 | GPIO 0 | | 19 | GPIO 4 | | 20 | GPIO 5 | | 21 | RXD, GPIO 3 | | 22 | TXD, GPIO 1 | |
| Pines del ESP8266-12E. |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | **Característica** | **Valor** | | **Datos Generales** | | | Clock Speed | 80 MHz (máximo de 160 MHz) | | Frecuencia Cristal | 24 MHz – 52 MHz | | Memoria RAM (General) | 160 KB | | Memoria dRAM | 96 KB | | Memoria iRAM | 64 KB | | Memoria SPI Flash externa | 512 KB – 16 MB | | **Datos Eléctricos Generales** | | | Tensión Nominal | 3.3 V | | Corriente Nominal | 80 mA | | Corriente Deep Sleep | 20 µA | | Corriente TX máximo | 170 mA | | Corriente RX máximo | 56 mA | | **Datos Eléctricos GPIO** | | |  | 3.3 V | | (máxima) | 12 mA | | Input LOW | -0.3 V – 0.825 V | | Input HIGH | 2.475 V – 3.6 V | | Output LOW | 0 V – 0.33 V | | Output HIGH | 2.64 V – 3.3 V | |
| Características técnicas ESP8266. |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | **Frecuencia de Cristal (MHz)** | **Velocidad de transmisión (baudios)** | | 26 | 74880 | | 40 | 115200 | |
| Velocidad de transmisión según frecuencia de cristal. |

#### Alimentación de la placa MCU.

|  |  |
| --- | --- |
|  |  |
| Entrada µUSB de NodeMCU. | Regulación de Tensión de entrada de 5V a 3.3 V. |

#### Regulador de tensión de la placa MCU.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | **Característica** | **Valor** | | **Datos Generales** | | | Modelo | AMS1117 | | Encapsulado | SOT-223 | | **Datos Eléctricos** | | | máxima | 15 V | | Line Regulation\*1 () | 1 mV – 10 mV | | Load Regulation\*2 ( | 3 mV – 15 mV | | \*3 (con ) | 1.1 V – 1.3 V | | máxima () | 900 mA – 1500 mA (nominal 1 A) | | mínima () | 5 mA – 10 mA | | Quiescent Current () | 5 mA – 11 mA | | **Datos Térmicos** | | | Operating Junction Temperature |  | | Storage Temperature |  | | Thermal Resistance (SOT-223) |  | | Thermal Resistance (Junction to Case) |  |   \*1Line Regulation: Capacidad de mantener la tensión de salida respecto a las variaciones en la señal de entrada.  \*2Load Regulation: Capacidad de mantener la tensión de salida ante cambios en el consumo de corriente de la carga.  \*3: Mínima diferencia de tensión entre entrada-salida para que pueda realizarse la regulación de tensión correctamente. |
| Características técnicas de AMS1117. |

|  |
| --- |
| **Consumo de AMS1117**.  = rango de temperatura de la unión = -40 °C – 125 °C. Se escoge el valor máximo para el cálculo de la mayor potencia disipada por el componente (en este caso, 125 °C).  = resistencia térmica, unión-ambiente = 90 °C/W.  = temperatura ambiente = 0 °C, se escoge esta temperatura para poder ver los rangos máximos a una temperatura normalmente mínima real. |
| Consumo máximo del componente AMS1117. |

#### Botones de la placa MCU.

|  |  |
| --- | --- |
|  |  |
| Esquema eléctrico del Botón RST (Reset). | Esquema eléctrico del Botón Flash. |

Botón RST:

La placa MCU se puede reiniciar cuando en el pin RST (Reset) se aplica una señal de estado bajo. El pin RST de la placa está conectado de forma que su estado es normalmente alto, como se puede comprobar en el esquema estando conectado a la señal de 3.3 V. Cuando se pulsa el botón RST, el pin pasa a estar en estado bajo provocando el reinicio de la placa.

Si se desean aplicar los estados de la placa SLEEP o DEEP-SLEEP para ahorrar energía se debe conectar el pin GPIO 16 (D0) con el pin RST, como muestra la imagen, de esta forma cuando se señale a través del pin GPIO 16 una señal de estado bajo la placa se reiniciará al igual que si se hubiera pulsado el botón RST. Por tanto una forma de automatizarlo sería aplicando un temporizador mediante el pin GPIO 16 que “despierte” a la placa sólo cambiando el estado de salida de ese pin GPIO.

Botón FLASH:

El botón Flash permite que la placa entre en modo FLASH para que pueda almacenar los datos de los programas que se le pasará para que ejecute las operaciones deseadas. En el caso de este proyecto, permite el almacenamiento del Sketch de Arduino.

#### Modos de funcionamiento del ESP.

El ESP8266 tiene dos modos de funcionamiento:

* Modo UART, es el funcionamiento normal, ejecutando las órdenes almacenadas en el programa que le ha sido enviado.
* Modo FLASH, permite el almacenamiento de los programas en la memoria de la placa.

Para configurar modos de funcionamiento se debe cumplir las siguientes condiciones:

|  |
| --- |
|  |
| Configuración electrónica de modos de funcionamiento. |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  | | --- | --- | --- | --- | | **Modo** | **GPIO 15** | **GPIO 2** | **GPIO 0** | | UART | LOW | HIGH | HIGH | | FLASH | LOW | HIGH | LOW | |
| Configuración lógica de los modos de funcionamiento. |

* El pin GPIO 15 debe de tener un estado normalmente bajo.
* El pin GPIO 2 debe de tener un estado normalmente alto.
* El pin GPIO 0 es el que conmutará, según su estado, el modo en el que funcionará la placa .

Para poder conmutar los diferentes modos de funcionamiento de la placa se debe realizar el siguiente proceso, si se desea hacerlo de forma manual:

1. Pulsar, y mantener pulsado, el botón FLASH. Con esto dejamos el pin GPIO0 a un estado LOW, y como los pines GPIO2 y GPIO15 ya estaban fijos con sus estados correspondientes se habrá procedido a activar al modo FLASH del ESP8266.
2. Para que funcione en este modo debemos reiniciar el ESP por tanto, sin soltar aún el botón FLASH, se debe pulsar el botón RST. Por lo que reiniciamos la placa en su modo activado FLASH, comprobándolo mediante el parpadeo del LED acoplado en el ESP.
3. Finalmente, tras acabar con el modo FLASH, para volver al modo normal UART se suelta el botón FLASH.

NOTA: En la placa MCU si se usa IDE de Arduino o similares cuando se realiza la función de subir programa (sketch) este proceso se realiza de forma automática y no es necesario hacerlo.