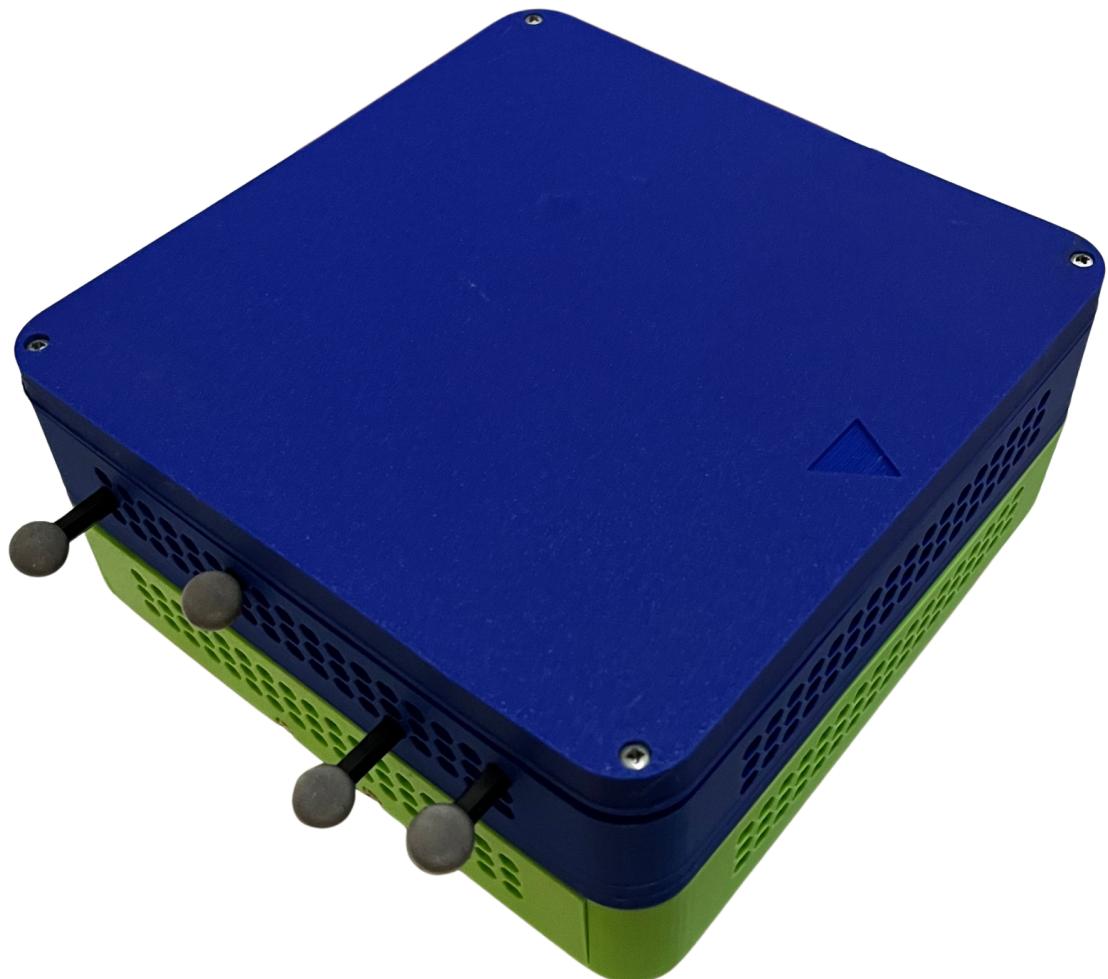


OmniBotCC: Manual de usuario

Christian Campos

8 de noviembre de 2025



OmniBotCC - Manual de Construcción
Versión 1.0 (2025)

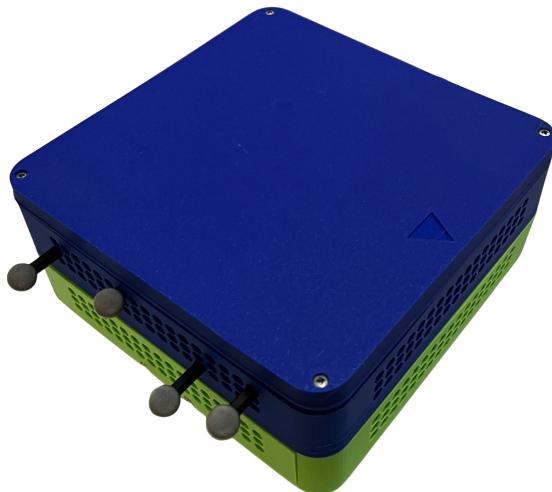
© 2025 Universidad del Valle de Guatemala. Departamento de Ingeniería Electrónica, Mecatrónica y Biomédica.
Este manual cubre el ensamblaje mecánico y las conexiones eléctricas básicas.
Fabricado en Guatemala.

Índice

1. Descripción general: ¿Qué es el OmniBotCC?	3
2. Requisitos y seguridad	4
2.1. Requisitos mínimos del equipo	4
3. Listado de Materiales (BOM)	6
4. Hardware	8
4.1. Visión general	8
4.2. Pasos a seguir	8
4.3. Impresión 3D	8
4.4. Manual de construcción	12
4.5. Esquemáticos y conexiones	18
4.6. Pruebas iniciales de <i>hardware</i>	19
5. Software/Firmware	20
5.1. Instalación del <i>software/firmware</i>	20
5.2. Protocolo de comunicación	20
5.3. API/Comandos	20
5.4. Inicio Rápido	21
6. Calibración y mantenimiento	24
6.1. Calibración AS5600	24
6.2. Calibración OmniBotCC	24
6.3. Mantenimiento OmniBotCC	25
7. Especificaciones técnicas	26
8. Soporte y licencias	27

1. Descripción general: ¿Qué es el OmniBotCC?

El OmniBotCC es un robot móvil de 190mm de largo y ancho versátil, de alto desempeño y precisión. El OmniBotCC fue diseñado para funcionar como una plataforma móvil complementaria para otros agentes robóticos. Por lo mismo, fue construido con un enfoque omnidireccional, usando ruedas *Mecanum* de 45° grados. El corazón de este robot es un microcontrolador [ESP32](#) desarrollado por Espressif. Para programarlo es necesario contar con una computadora con el entorno de desarrollo oficial de Espressif, ESP-IDF, y un cable micro usb.



El OmniBotCC cuenta con 4 ruedas [*Mecanum*](#) de 45° grados con motores paso a paso [Nema 17](#) de 23mm de grosor. Estos son controlados en lazo cerrado a través de un sistema de controladores [DRV8825](#) o [TMC2209](#) y un sensor magnético de posición [AS5600](#). Así mismo, tanto el microcontrolador como los motores están alimentados a través de 6 [baterías 18650](#), con una conexión de 3 celdas en serie y 2 celdas en paralelo (12.6V). La carga y descarga está controlada por un [BMS](#) y el voltaje de entrada para el microcontrolador y los motores son reguladas por 2 diferentes reguladores de voltaje.

2. Requisitos y seguridad

2.1. Requisitos mínimos del equipo

Para operar la plataforma OmniBotCC se necesita:

- Un computador con MATLAB o Python (versión recomendada: Python 3.8+ / MATLAB R2021a o superior).
- Red Wi-Fi (AP) con capacidad para comunicación UDP entre el computador y el ESP32.
- Fuente de alimentación y cargador para el BMS con el pack de baterías. Esta fuente de alimentación debe ser capaz de otorgar 12.6V@2.6A durante 3 horas para carga normal y 12.6V@5.2A para carga rápida durante 2.5 horas.
- Herramientas básicas: destornilladores, llave hexagonal, multímetro y pinzas.
- Impresora 3D (si se van a fabricar las piezas) con volumen de impresión mínimo de 190mmx190mm listo para las piezas del chasis.

Precauciones de seguridad

Para garantizar la seguridad del usuario y de la plataforma móvil se deben seguir estas indicaciones:

- **Baterías:** Las 18650 son celdas de ion-litio. Por lo que es indispensable evitar cortocircuitos, no dejarlas caer, cargarlas siempre a través de un cargador balanceador (BMS) y no sobrecargarlas ni las descargarlas por debajo de 2.8V por celda.
- **Conexión y prueba de motores:** Las primeras pruebas deben realizarse con el chasis asegurado y con las ruedas libres de contacto. Además, de emplear un modo de baja velocidad para estas pruebas.
- **Manipulación de PCBs:** Debe evitarse conectar las placas PCB si se nota: estática, soldaduras frías visibles o componentes sueltos.
- **Entorno de pruebas:** Se debe trabajar en un espacio despejado. Manteniendo personas y objetos frágiles fuera del área de prueba.
- **Firmware:** Antes de subir firmware nuevo es importante verificar que los límites de velocidad estén probados para evitar comportamientos inesperados. Velocidades muy altas podrían provocar: sobrecalentamiento, resonancia, vibraciones muy altas y campos magnéticos no deseados.
- **Controladores:** Antes de comenzar con cualquier tipo de pruebas, es requerimiento calibrar correctamente el amperaje máximo de los controladores de los motores.
- **Ensamblado:** Antes de cualquier prueba, se debe verificar que todo el ensamblado se haya realizado correctamente.

Señales de advertencia y qué hacer

- **Olor a quemado o humo:** Se debe apagar inmediatamente la alimentación general. Se deben inspeccionar las PCBs, buscando la fuente del humo o el olor. Si fueran las baterías, inmediatamente se deben desconectar y verificar que no haya habido corto. Recuerde que las baterías de Li-Ion son muy inestables y pueden llegar a incendiarse o incluso explotar.

- **Vibraciones o ruidos inusuales:** Debe pararse la plataforma y revisar el montaje mecánico y los acoplamientos de los motores. También debe verificarse que el entorno de trabajo sea el adecuado. La plataforma móvil está diseñada exclusivamente para moverse en superficies planas y sin ningún tipo de variaciones en el terreno.
- **Temperaturas altas (>70°C) en controladores o motores:** Debe detenerse la prueba y dejar enfriar. A pesar que los controladores tienen disipadores de calor, el uso muy extenso de la plataforma móvil puede llevar a sobrecalentamiento.

3. Listado de Materiales (BOM)

Los materiales necesarios para la construcción del OmniBotCC están descritos en el siguiente cuadro. Debe tomarse en cuenta que los distribuidores fueron consultados el 27 de agosto del 2025. Así mismo, los distribuidores son todos del país de Guatemala, si usted se encuentra afuera del país puede usar los enlaces como referencia.

Componente	Cantidad	Distribuidor
Nema 17 23mm	4	LaElectrónica
Ruedas Mecanum (par L-R)	2	Oxdea
Material PLA (Por Kg)	0.50368	Creativo3D
Regulador de Voltaje (300W)	1	LaElectrónica
Regulador de Voltaje 5V	1	LaElectrónica
AS5600	4	Oxdea
Cables JST XH Hembra (6 a 4 pines)	4	LaElectrónica
Cable Calibre 14 (Por metro)	2	LaElectrónica
Tira de Bornera (6Pines)	1	LaElectrónica
Jumpers Hembra-Macho 20cm (x10)	1	LaElectrónica
Jack Banana Hembra	2	Tetsa
Placa Alimentación (15x20 cm, referencia)	1	MakerLab/LaElectrónica
BMS (3s 20a)	1	Oxdea
Baterías 18650	6	LaElectrónica
Sujetador bat. PCB 3s	2	LaElectrónica
Placa Control (7x10 cm, referencia)	1	MakerLab/LaElectrónica
ESP32	1	LaElectrónica
DRV8825 (Driver Paso a Paso)	4	LaElectrónica
Header Hembra (40 pines)	1	Tetsa
Header Hembra (8 pines)	9	Tetsa
Conecotor Macho 4 pines	4	LaElectrónica
Switch	1	Tetsa
Bornera 15A	2	LaElectrónica
Tornillos M3x8 Allen	4	La Casa del Tornillo
Tornillos M3x8 Planos Phillips	12	La Casa del Tornillo
Tornillos M3x8 Redonda Phillips	40	La Casa del Tornillo
Tornillos M3x12 Planos Phillips	12	La Casa del Tornillo
Tornillos M3x12 Redonda Phillips	12	La Casa del Tornillo
Tornillos M3x16 Redonda Phillips	4	La Casa del Tornillo
Tornillos M3x25 Planos Phillips	8	La Casa del Tornillo
Tuerca M3	44	La Casa del Tornillo

Material	Precio Unitario	Cantidad	Precio Total
Nema 17 23mm	Q139.00	4	Q556.00
Ruedas Mecanum (par L-R)	Q99.00	2	Q198.00
Material PLA (Por Kg)	Q175.00	0.50368	Q88.14
Regulador de Voltaje (300W)	Q129.00	1	Q129.00
Regulador de Voltaje 5V	Q27.00	1	Q27.00
AS5600	Q70.00	4	Q280.00
Cables JST XH Hembra (6 a 4 pines)	Q24.00	4	Q96.00
Cable Calibre 14 (Por metro)	Q4.50	2	Q9.00
Tira de Bornera (6Pines)	Q11.00	1	Q11.00
Jumpers Hembra-Macho 20cm (x10)	Q8.00	1	Q8.00
Jack Banana Hembra	Q4.50	2	Q9.00
Placa Alimentación (referencia)	Q35.00	1	Q35.00
BMS (3s 20a)	Q45.00	1	Q45.00
Baterías 18650	Q59.00	6	Q354.00
Sujetador bat. PCB 3s	Q19.00	2	Q38.00
Placa Control (referencia)	Q10.00	1	Q10.00
ESP32	Q125.00	1	Q125.00
DRV8825 (Driver Paso a Paso)	Q38.00	4	Q152.00
Header Hembra (40 pines)	Q4.00	1	Q4.00
Header Hembra (8 pines)	Q3.00	9	Q27.00
ConeCTOR Macho 4 pines	Q1.00	4	Q4.00
Switch	Q6.00	1	Q6.00
Bornera 15A	Q2.00	2	Q4.00
Tornillos M3x8 Allen	Q3.53	4	Q14.12
Tornillos M3x8 Planos Phillips	Q1.12	12	Q13.44
Tornillos M3x8 Redonda Phillips	Q1.33	40	Q53.20
Tornillos M3x12 Planos Phillips	Q1.64	12	Q19.68
Tornillos M3x12 Redonda Phillips	Q1.64	12	Q19.68
Tornillos M3x16 Redonda Phillips	Q1.95	4	Q7.80
Tornillos M3x25 Planos Phillips	Q2.65	8	Q21.20
Tuercas M3	Q1.48	44	Q65.12
		Total	Q2,429.38

4. Hardware

4.1. Visión general

El hardware de la plataforma se compone de tres subsistemas principales:

1. **Mecánica:** chasis modular impreso en 3D, soportes de motores, ejes y ruedas *mecanum*.
2. **Electrónica de potencia:** placa PCB con un pack de baterías 3s2p y un BMS. Así como 2 reguladores de voltaje (5V para el ESP32 y 10V para los motores).
3. **Electrónica de control:** PCB con el ESP32, controladores de los motores y sensores AS5600.

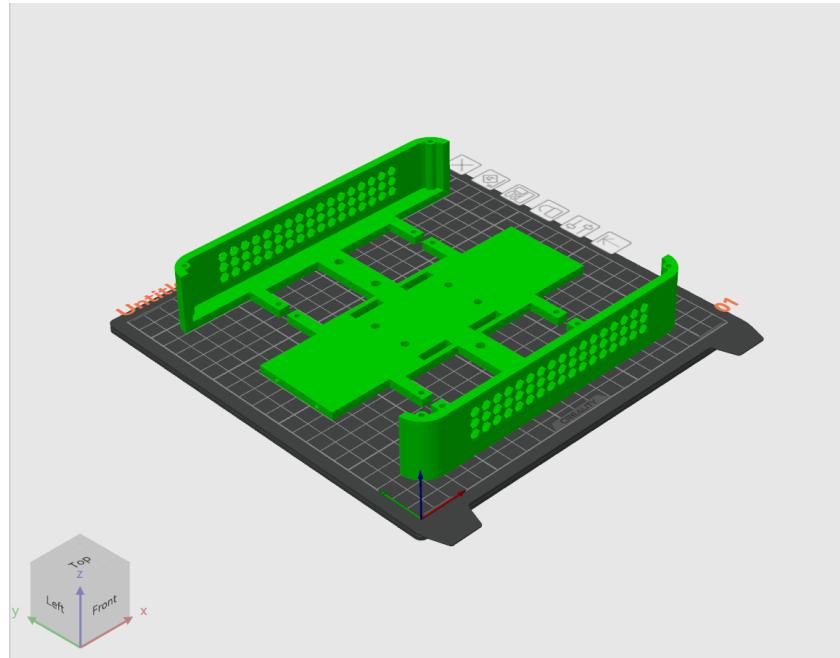
4.2. Pasos a seguir

Para el montaje se deben seguir los siguientes pasos:

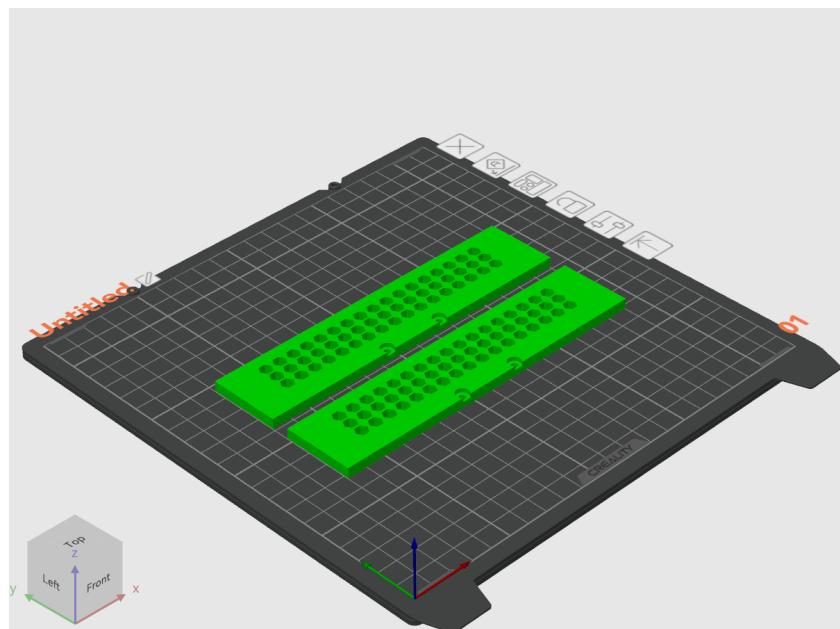
1. Imprimir las piezas en PLA, siguiendo las recomendaciones que se encuentran en la Sección 4.3.
2. Ensamblar la plataforma móvil siguiendo las instrucciones en la Sección 4.4.
3. Se instalan las PCBs en su nivel correspondiente y se hacen las conexiones siguiendo los esquemáticos mostrados en la Sección 4.5.
4. Se realizan las pruebas iniciales como se indica en la Sección 4.6

4.3. Impresión 3D

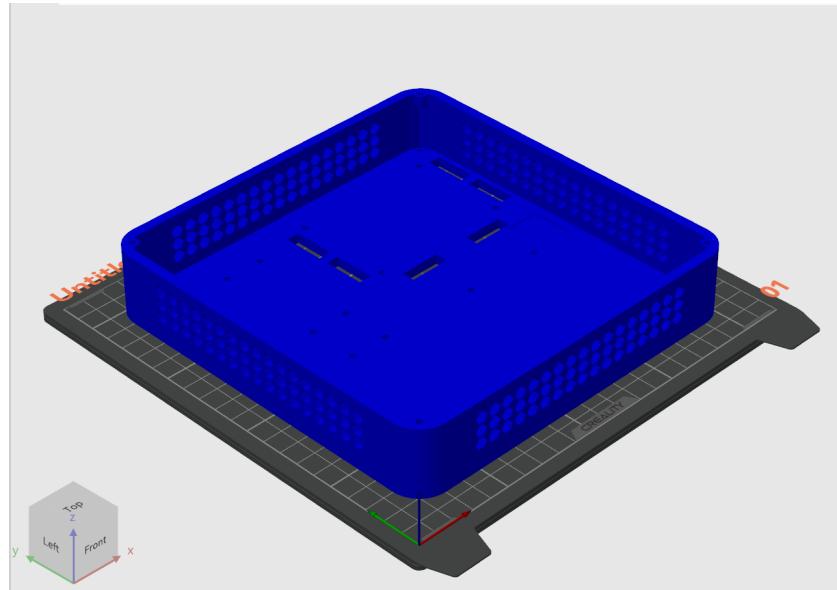
Para las piezas del chasis, se requiere de una impresora 3D que tenga un área mínima de impresión de 190mmx190mmx35mm. Los archivos STL se encuentran en el siguiente enlace: [GitHub impresión 3D](#).



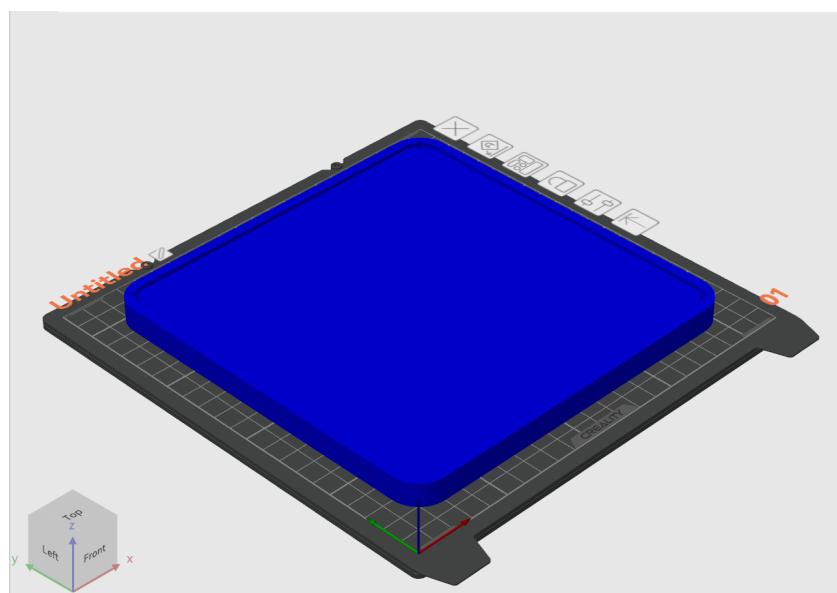
Nivel 1: En esta pieza se recomienda usar las configuraciones estándar de la impresora. No es necesario agregar soportes, aunque su uso podría mejorar la calidad de la pieza.



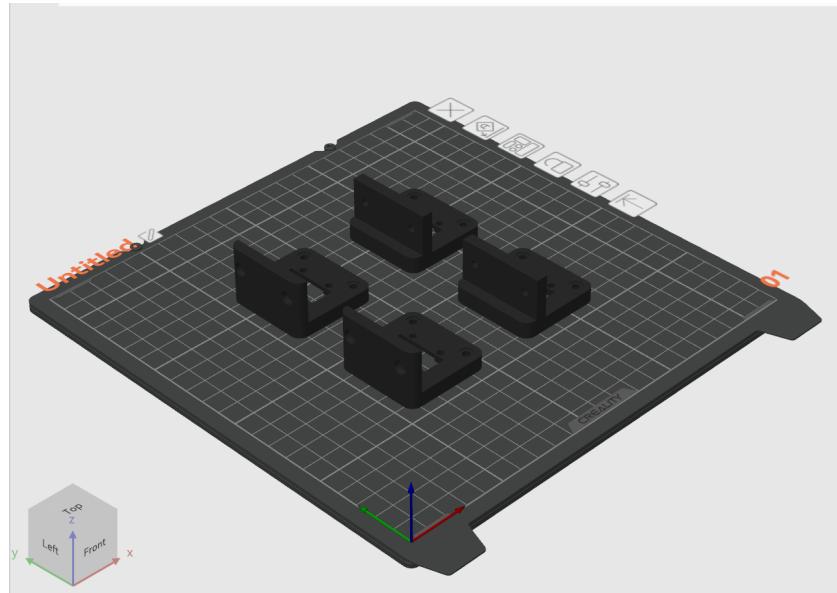
Nivel 1 (laterales): En estas piezas se recomienda usar las configuraciones estándar de la impresora. No es necesario agregar soportes.



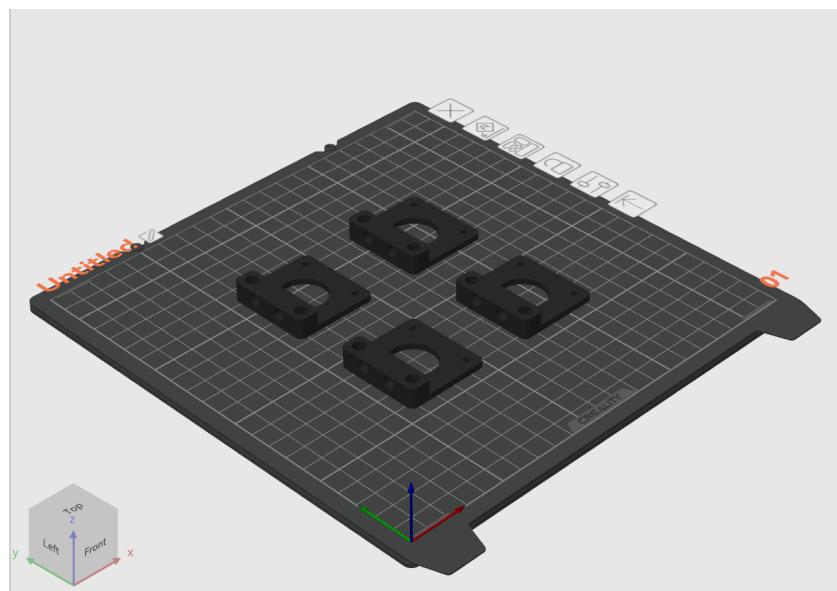
Nivel 2: En esta pieza se recomienda usar las configuraciones estándar de la impresora. Los soportes recomendados son manuales en los agujeros altos de la pieza.



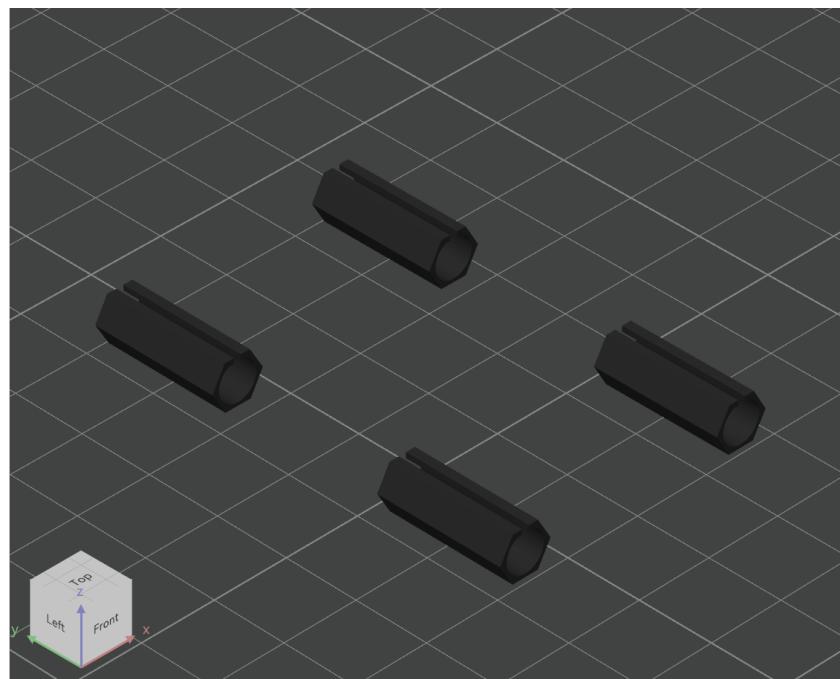
Nivel 3: En esta pieza se recomienda usar las configuraciones estándar de la impresora. Los soportes deben configurarse de tal manera que sean fáciles de extraer pues el triángulo/flecha mide solo 1mm de profundidad.



Soportes AS5600 (izquierda y abajo): En estas piezas se recomienda usar las configuraciones estándar de la impresora. Los soportes son requeridos.



Bases para los motores: En estas piezas se recomienda usar las configuraciones estándar de la impresora. Los soportes son requeridos.



Ejes para los motores: En estas piezas se recomienda usar las configuraciones de definición más alta de la impresora. Esto significa tener la menor altura de capa posible. Estas piezas deben imprimirse sin soportes.

4.4. Manual de construcción

Para los pasos del ensamblaje, se encuentra adjunto el manual de construcción. Es importante que las conexiones se realicen entre el paso 11 y 12 del ensamblado.



Manual de Construcción

Instrucciones de Ensamblado

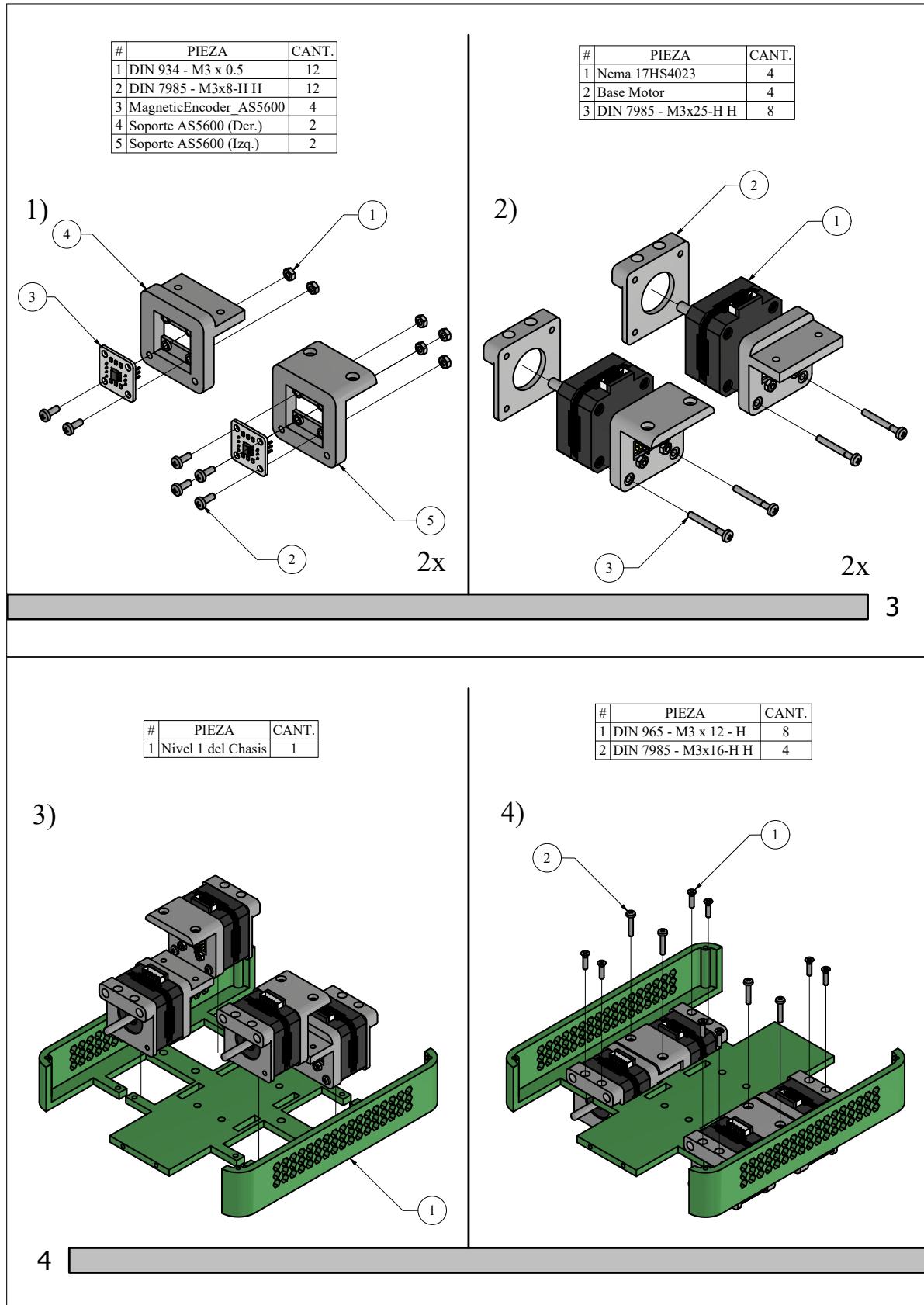
Este manual incluye las instrucciones paso a paso del ensamblado de la plataforma móvil omnidireccional *OmniBotCC*. Esto incluye todas las piezas necesarias para la construcción del chasis, así como las placas PCB que permiten el funcionamiento y el sistema de locomoción. Sin embargo, no se incluye una guía de las conexiones, estas quedan a libertad de los materiales a los que el usuario tenga acceso. Para las conexiones por favor consultar los esquemáticos de conexión de la plataforma móvil adjuntos al final de este documento. Así mismo, el código se encuentra disponible en GitHub en el siguiente repertorio: *OmniBotCC Code*.

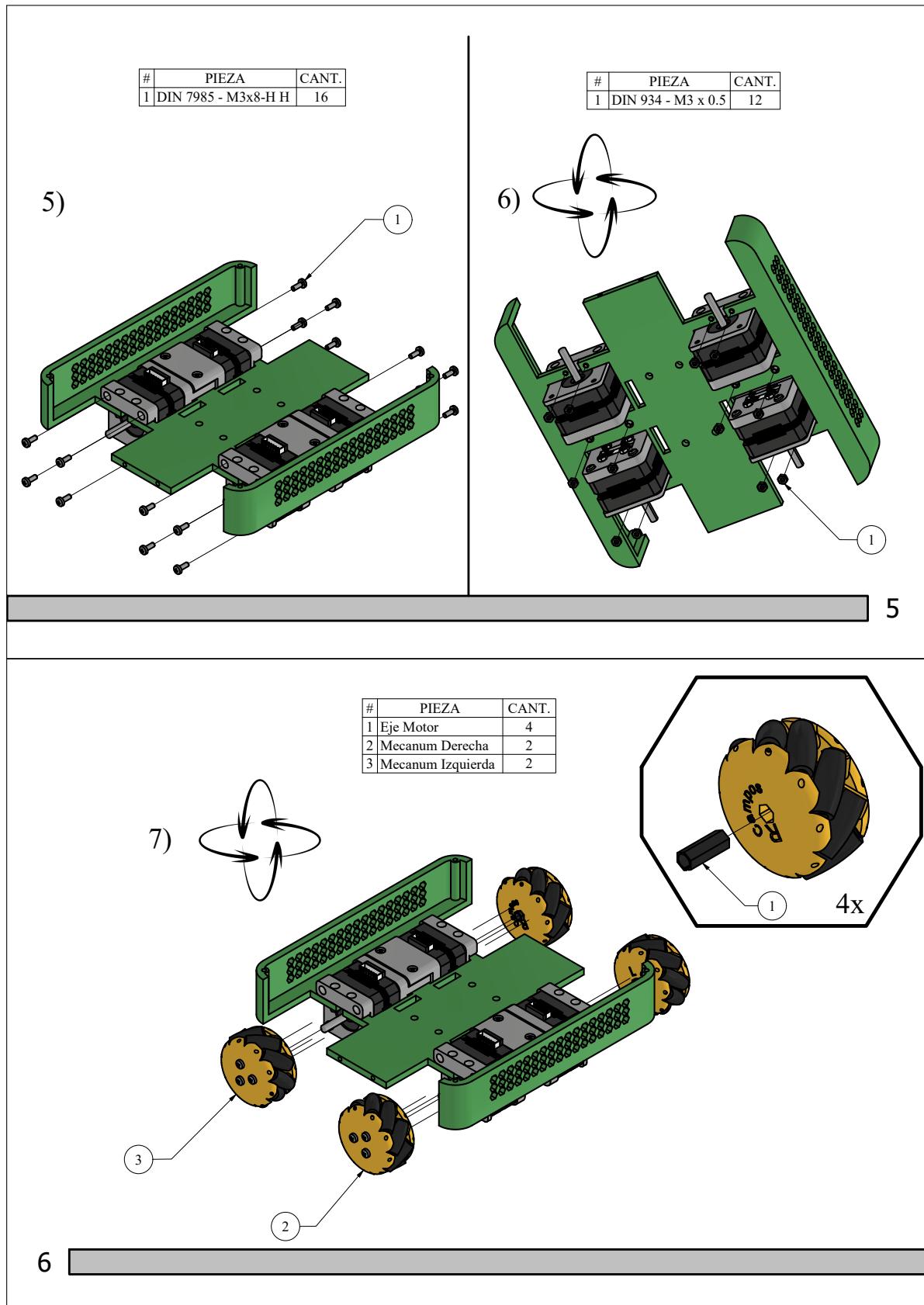


Vídeo Ensamblaje

En el código QR adjunto podrá encontrar el vídeo del ensamble para mayor claridad.

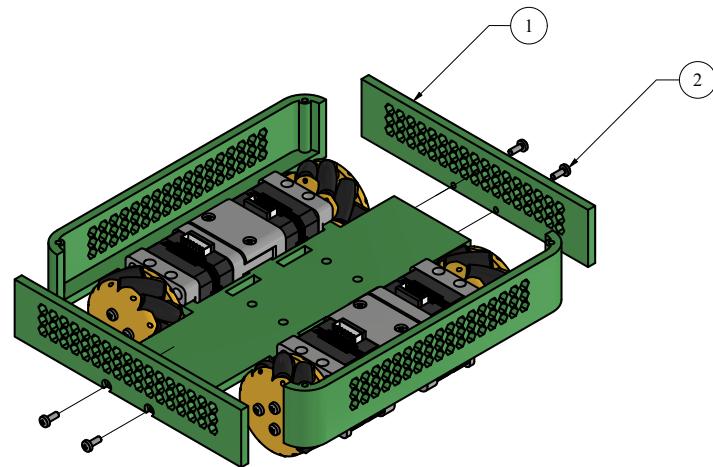






#	PIEZA	CANT.
1	Lateral Nivel 1 del Chasis	2
2	DIN 7985 - M3x8-H H	4

8)

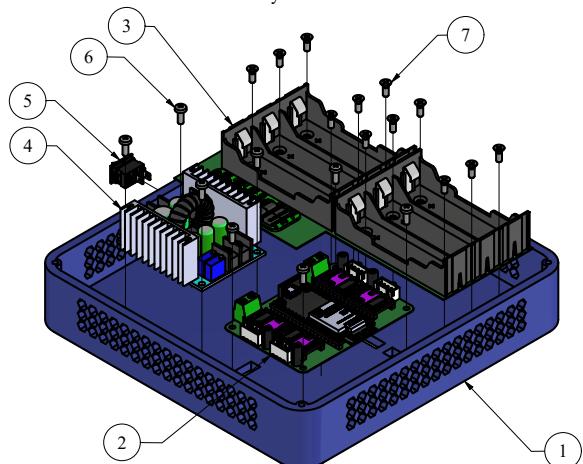


7

9)

#	PIEZA	CANT.
1	Nivel 2 del Chasis	1
2	PCB Control	1
3	PCB Alimentación	1
4	Buck Converter 200W	1
5	MINI SWITCH	1
6	DIN 7985 - M3x8-H H	8
7	DIN 965 - M3 x 8 - H	12
8	Buck Converter 5V*	1

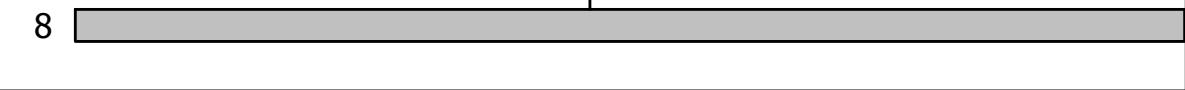
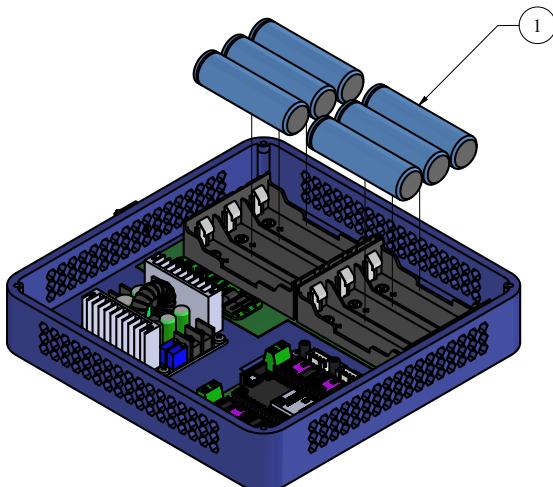
*No se incluye en el modelo.

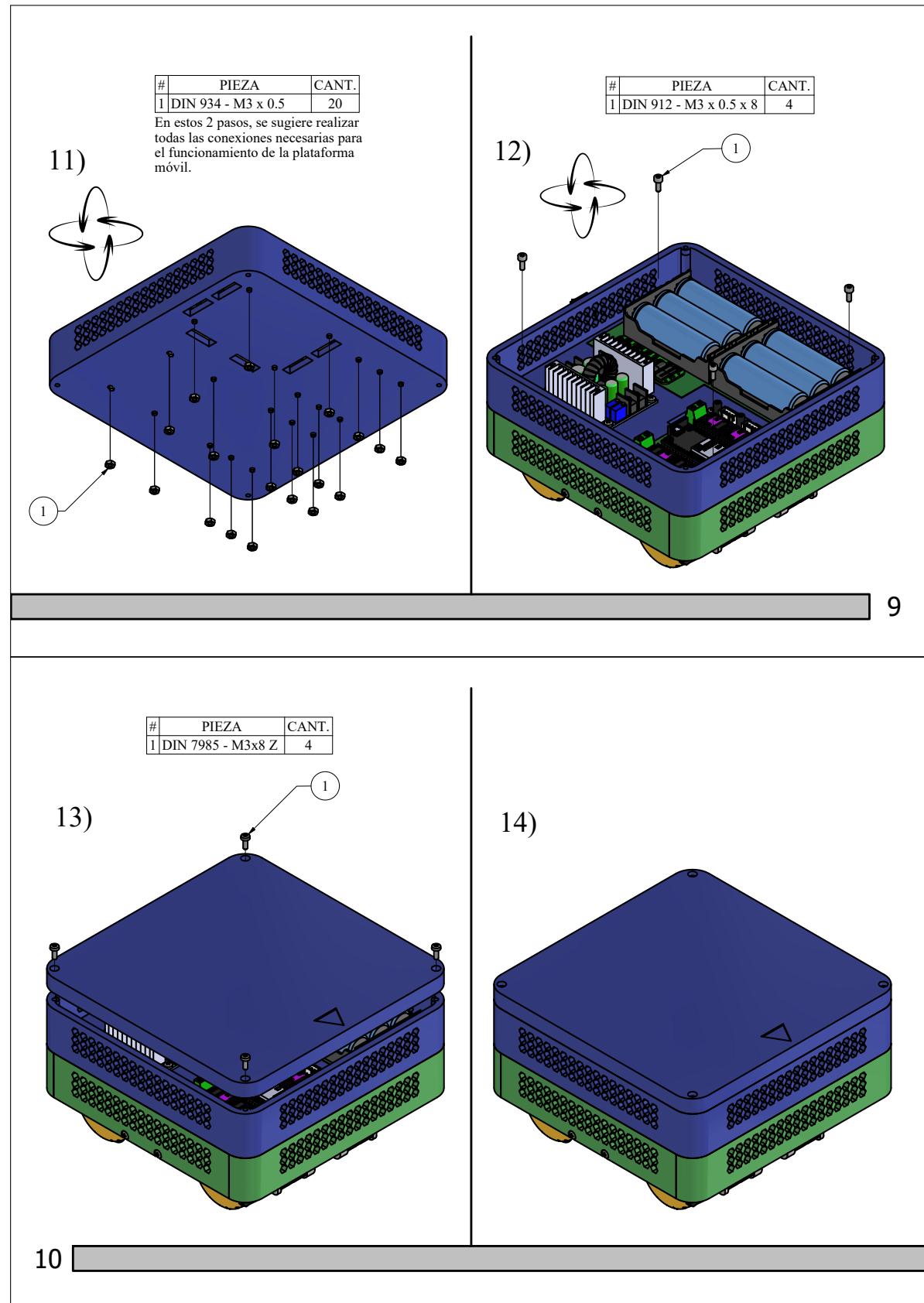


8

10)

#	PIEZA	CANT.
1	Baterías 18650	6

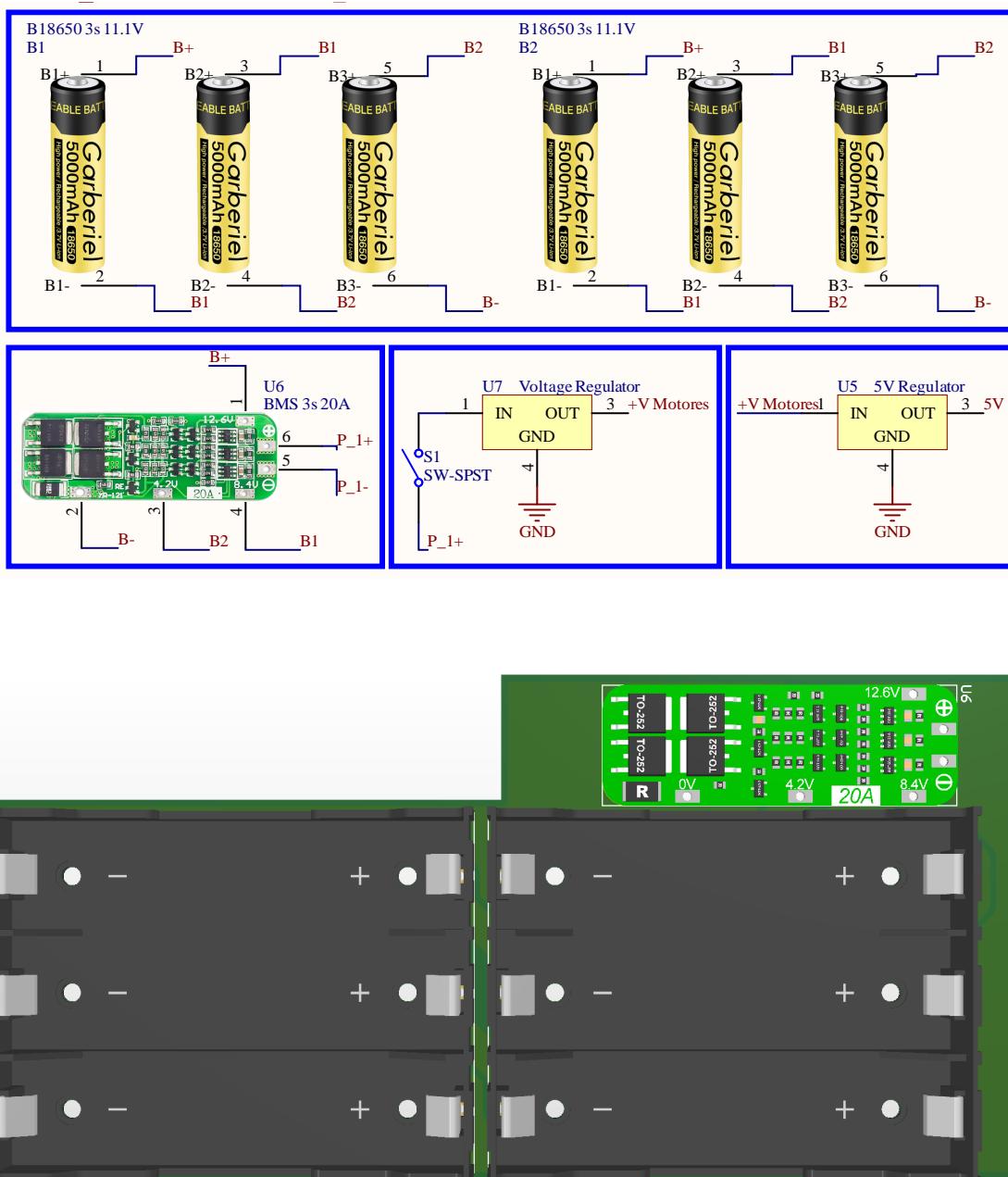




4.5. Esquemáticos y conexiones

4.5.1. Sistema de alimentación

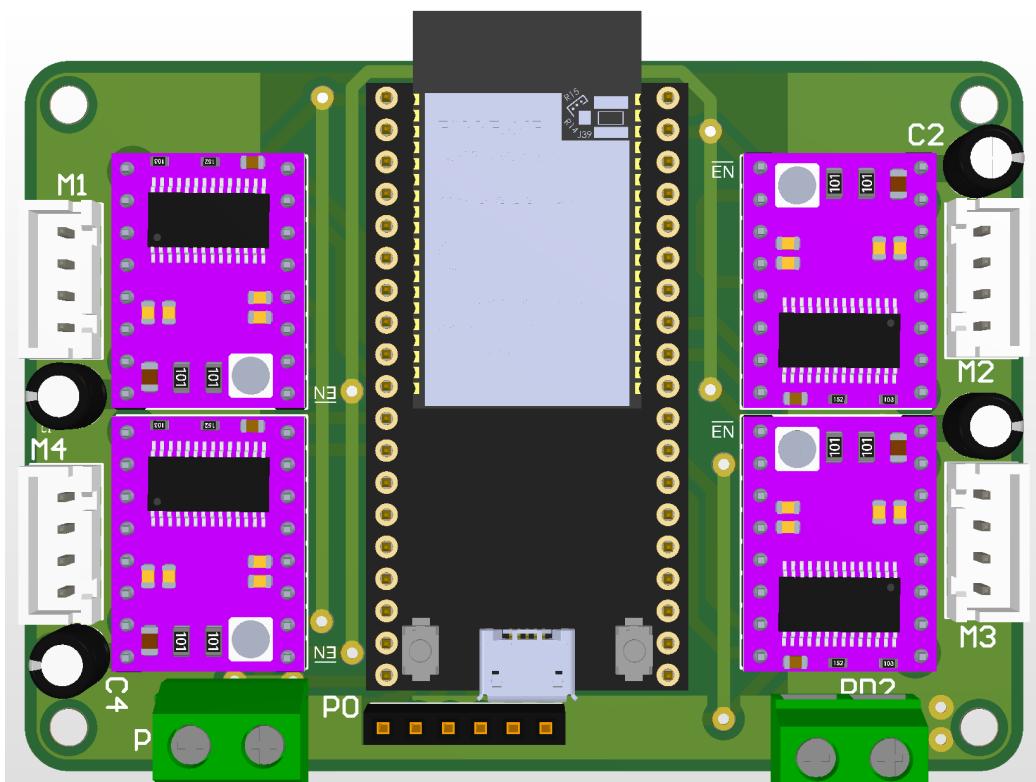
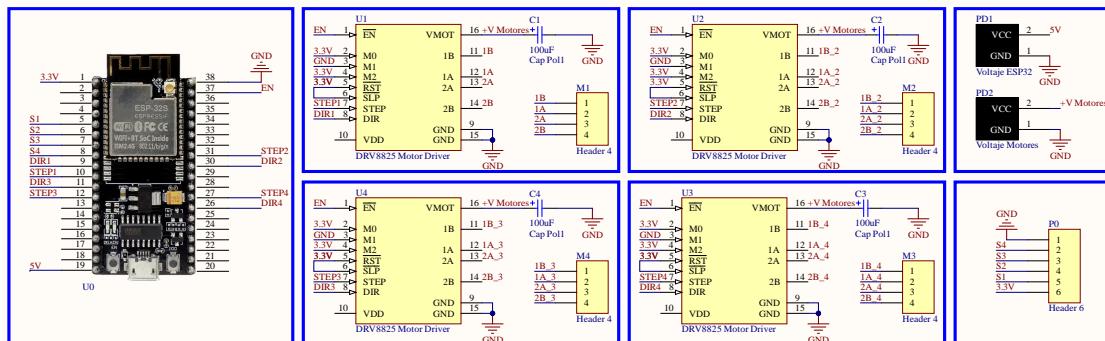
El esquemático del circuito y la placa de alimentación del OmniBotCC son los siguientes:



Es importante que al cargar las baterías, se haga siempre mediante el BMS, a 12.6V@2.6A durante 3 horas para una carga normal. Si se deseará una carga rápida la corriente puede subir hasta 5.2A durante 2 horas y media, sin embargo no es recomendable pues podría desgastar rápidamente las baterías.

4.5.2. Sistema de control

El esquemático del circuito y la placa de control del OmniBotCC son los siguientes:



4.6. Pruebas iniciales de hardware

Antes de integrar *firmware* deben realizarse:

1. Verificación de tensiones en la placa de distribución (sin estar conectada a la placa de control).
2. Comprobación de continuidad en el cableado de motores.
3. Test de drivers con límite de corriente bajo y medición térmica.

5. Software/Firmware

La plataforma móvil está diseñada para poder programarse siempre fuera del ensamblaje. Por lo mismo, el código puede ser cualquiera, sin embargo a continuación se presenta un código que cumple con el funcionamiento básico de la plataforma móvil. Esto a través de comunicación Wi-Fi con un protocolo UDP y del control en lazo cerrado de los motores paso a paso.

5.1. Instalación del software/firmware

Para la instalación del *firmware* designado para el OmniBotCC debe compilarse y depurarse el código disponible en la siguiente [carpeta de GitHub](#) para el ESP32 a través del entorno ESP-IDF. Sin embargo, antes de esto debe confirmarse que el nombre y contraseña de la red sea el correcto. Por la naturaleza de la plataforma móvil, cuenta con el nombre y contraseña del Wi-Fi del laboratorio del Robotat de la UVG. Si tiene alguna duda con respecto al código, se le recomienda leer los comentarios que el mismo contiene. Así mismo, se sugiere la lectura de [la guía de programación de ESP-IDF](#) para comprender el uso de las librerías usadas en la programación del OmniBotCC.

5.2. Protocolo de comunicación

El protocolo de comunicación que usa el OmniBotCC es UDP (*User Datagram Protocol*). Es un protocolo de comunicación inalámbrica enfocado en la velocidad, que sacrifica la garantía de envío y recibimiento de datos por la velocidad. Por la importancia de poder realizar el control de los motores en lazo cerrado, al mismo tiempo que se reciben los comandos de control, este protocolo fue el escogido para el OmniBotCC.

5.3. API/Comandos

Los comandos que debe recibir el ESP32 para funcionar son los siguientes:

Comando	Descripción
start/stop	Arrancan o detienen los cuatro motores simultáneamente.
mX_rpm= RPMS	Fija la velocidad del motor X en RPM. Sólo establece la magnitud (la dirección se define por separado).
mX_dir= CW/CCW	Define la dirección de giro del motor X (CW = <i>Clockwise</i> , CCW = <i>Counter-Clockwise</i>).
mX_start / mX_stop	Arrancan o detienen de forma individual el motor X.

Para una mayor facilidad del control se recomienda usar las funciones de Matlab/Python disponibles en la siguiente [carpeta de GitHub](#). Las funciones que se encuentran disponibles en esta carpeta son las siguientes:

Función	Descripción / Notas
robot_connect()	Configura el puerto/endpoint UDP para comunicarse con el ESP32. (Wi-Fi del Robotat)
robot_start() / robot_stop()	Inician o detienen el robot usando los últimos valores recibidos.
robot_set_velocities()	Envía un vector con las RPM de las cuatro ruedas (determinantes de ω_{bx} , v_{bx} , v_{by}).

5.4. Inicio Rápido

Para un inicio rápido y comprobación del correcto funcionamiento del OmniBotCC, puede correrse el programa llamado [Inicio_Rapido.m](#). En este podrá seguir un paso a paso para visualizar el funcionamiento de la plataforma móvil. Este control se realiza en lazo abierto, por lo que los movimientos no serán precisos.

```
1 % -----
2 % Inicio Rapido del OmniBotCC
3 %
4 % Proposito
5 %   % Mostrar al usuario los comandos basicos del OmniRobotCC. Incluye:
6 %   % - Demostracion de conexion y desconexion al Robot.
7 %   % - Demostracion de inicio, parada y seteo de velocidades.
8 %
9 % Uso rapido
10 %   1) Ejecucion del script seccion por seccion (antes de la ultima seccion)
11 %      para visualizar
12 %      el correcto funcionamiento de los comandos.
13 %   2) Ejecucion de la ultima seccion para visualizar un movimiento simple
14 %      sin retroalimentacion.
15 %
16 % Variables importantes (entrada / salida)
17 %   Entradas configurables:
18 %     - Vel (rpms)
19 %
20 % Comportamiento y supuestos
21 %   - El Robot deberia tener suficiente espacio para moverse durante 2
22 %     segundos hacia adelante e intentar regresar a su punto de inicio. Al no
23 %     tener retroalimentacion no se asegura que esto sea cierto.
24 %   - El codigo no es de control, solo de demostracion.
25 %
26 % Tips rapidos de uso / debugging
27 %   - Verifique que el Robot tenga espacio a sus alrededores.
28 %   - Verifique que el Robot este conectado al Wi-Fi del Robotat, esto lo
29 %     indica la LED Azul del ESP32.
30 %   - Si el ESP32 no conecta, reinicie el ESP32 y compruebe el estado del
31 %     Wi-Fi del Robotat.
32 %
33 % Autores / version
34 %   - Autor: Christian Campos
35 %   - Version: 1.0
36 %   - Fecha: 2025-10-13
37 %
38 %% Comando para conexion con el Robot
39 robot_connect();
40
41 %% Comando para enviar velocidades al Robot (aun no esta en movimiento)
42 rpm_vector = -10*[0 0 0 1];
43 robot_set_velocities(rpm_vector);
44
45 %% Comando para comenzar el movimiento del Robot con las velocidades guardadas
46 robot_start();
47
48 %% Comando para enviar velocidades al Robot (modifica las velocidades actuales)
49 rpm_vector = -10*[-1 1 1 -1];
50 robot_set_velocities(rpm_vector);
51
52 %% Comando para detener el Robot (los motores siguen encendidos)
53 rpm_vector = [0 0 0 0];
54 robot_set_velocities(rpm_vector);
55
56 %% Comando para detener el Robot y apagar los motores.
57 robot_stop();
```

```
59 %% Comando para desconexion con el Robot
60 robot_disconnect();
61
62 %% Demostracion movimiento sencillo sin retroalimentacion.
63
64 robot_connect();
65
66 % Velocidad base en rpm (se puede ajustar hasta a 400rpms)
67 vel = 40;
68
69 % Definicion de los vectores de velocidad de cada rueda
70 % Orden: [m1 m2 m3 m4]
71 movimientos = {
72     'Adelante',    vel * [ 1   1   1   1 ];      % +Y (avanza hacia adelante)
73     'Atras',       vel * [-1  -1  -1  -1];      % -Y (retrocede)
74     'Derecha',    vel * [ 1  -1  -1   1 ];      % +X (desplaza a la derecha)
75     'Izquierda',  vel * [-1   1   1  -1];      % -X (desplaza a la izquierda)
76     'Giro CW',    vel * [ 1  -1   1  -1];      % Rotacion horario
77     'Giro CCW',   vel * [-1   1  -1   1];      % Rotacion antihorario
78 };
79 pause(1);
80
81 % Velocidades 0 antes de iniciar
82 robot_set_velocities([0 0 0 0]);
83
84 % Iniciar robot con velocidades 0
85 robot_start();
86
87 % Vamos a realizar todos los movimientos
88 for k = 1:size(movimientos,1)
89     nombre = movimientos{k,1};
90     rpm_vector = movimientos{k,2};
91
92     disp(['Movimiento: ', nombre]);
93
94     % Asignar velocidades
95     robot_set_velocities(rpm_vector);
96     tic;
97     while toc < 2
98         % loop de espera de 2 segundos
99     end
100    % Detener Robot (sin apagar motores)
101    robot_set_velocities([0 0 0 0]);
102    pause(1); % pequena pausa entre movimientos
103
104    % Detener Robot
105    robot_stop();
106
107    disp('Rutina completa.');
108
109    % Desconectar Robot
110    robot_disconnect();
111
112    %% Demostracion con velocidades X y Y
113    % Parametros fisicos del robot
114    w = ((46.8+82.3)/2)/1000;    % m
115    l = ((74.65+32.35)/2)/1000; % m
116    r = 48/1000;                 % radio rueda [m]
117    % Limites motores
118    rpm_max = 80;               % rpm
119
120    H0 = [ -1 - w,  1,  -1;
121          1 + w,  1,   1;
122          1 + w,  1,  -1;
123          -1 - w, 1,   1 ];
```

```
125 H0_pinv = pinv(H0);
126
127 % Magnitudes deseadas
128 v_linear = 0.12;      % m/s -> velocidad lineal de referencia para traslacion
129 omega_turn = 0.8;     % rad/s -> velocidad angular de referencia para giro
130 rpm_max=400;
131 % Nota sobre convenciones:
132 % Vb = [ omega; v_bx; v_by ] donde
133 % - v_bx = avance (forward) en m/s
134 % - v_by = lateral (derecha positiva) en m/s
135 % - omega: rotacion positiva.
136
137 % Mapear movimientos a Vb
138 movimientos = {
139     'Adelante', [0; v_linear; 0];      % avanzar +X del cuerpo (forward)
140     'Atras', [0; -v_linear; 0];         % retroceder -X del cuerpo
141     'Derecha', [0; 0; v_linear];        % desplazamiento lateral +Y del cuerpo (derecha)
142     'Izquierda', [0; 0; -v_linear];
143     'Giro CW', [-omega_turn; 0; 0];    % giro horario --> omega negativo (si conv. positive=CCW)
144     'Giro CCW', [omega_turn; 0; 0];
145 };
146 robot_connect();
147 % Inicio seguro
148 robot_set_velocities([0 0 0 0]);
149 robot_start();
150 pause(0.2);
151
152 for k = 1:size(movimientos,1)
153     robot_start();
154     nombre = movimientos{k,1};
155     Vb = movimientos{k,2}; % [omega; v_bx; v_by]
156
157 % Calcular velocidades angulares de ruedas (rad/s)
158 % u_rad_s = (1/r) * H0 * Vb
159 u_rad_s = (1/r) * (H0 * Vb);
160
161 % Convertir a RPM
162 u_rpm = u_rad_s * 60 / (2*pi);
163 if any(abs(u_rpm) > rpm_max)
164     scale = rpm_max / max(abs(u_rpm));
165     u_rpm = u_rpm * scale;
166 end
167
168 fprintf('Movimiento: %s -> RPMs: [%f %f %f %f]\n', ...
169 nombre, u_rpm(1), u_rpm(2), u_rpm(3), u_rpm(4));
170 robot_set_velocities(round(u_rpm(:)));
171
172 % Mantener durante 2 s
173 tic;
174 while toc < 2
175     % espera activa
176     pause(0.01);
177 end
178
179 % Parar entre movimientos
180 robot_set_velocities([0 0 0 0]);
181 pause(0.6);
182 end
183
184 robot_stop();
185 robot_disconnect();
186 disp('Rutina completa.');
```

6. Calibración y mantenimiento

6.1. Calibración AS5600

Nombre	Dirección	Tipo (R/W)	Bits	Descripción
ZMCO	0x00	R	1:0	Número de veces que se ha realizado BURN_ANGLE (máx. 3)
ZPOS	0x01-0x02	R/W	11:0	Posición cero configurada (<i>offset</i> del ángulo)
MANG	0x05-0x06	R/W	11:0	Rango máximo del ángulo de medición
CONF	0x07-0x08	R/W/P	WD (bit 7)	Activación del Watch-dog
			FTH (6:4)	Límite del filtro rápido
			SF (3:2)	Filtro lento
			PWMF (1:0)	Frecuencia PWM
			OUTS (3:2)	Configuración de la salida
			HYST (1:0)	Histéresis
			PM (1:0)	Modo de alimentación
STATUS	0x0B	R	MD (bit 5)	Imán detectado
			ML (bit 4)	Imán muy lejos
			MH (bit 3)	Imán muy cerca
RAW_ANGLE	0x0C-0x0D	R	11:0	Ángulo sin procesar (valor bruto)
ANGLE	0x0E-0x0F	R	11:0	Ángulo corregido según posición cero
BURN	0xFF	W	7:0	Registro especial para grabar configuración en memoria OTP

El cuadro muestra los registros y direcciones para comunicación I2C del AS5600. Para grabar la configuración de posición (ZPOS y MPOS) en la memoria, debe enviarse el valor 0x80 al registro BURN (comando BURN_Angle). Esto solo puede realizarse hasta tres veces, lo cual se refleja en el valor del registro ZMCO. Para grabar los ajustes de configuración (CONF y otros), se debe enviar 0x40 al mismo registro (BURN_Setting); esta operación solo puede hacerse una vez y únicamente si no se ha realizado ningún BURN_ANGLE previamente. Durante todo este proceso es necesario que el pin GPO este conectado a GND (tierra). Así mismo, este proceso puede realizarse usando el código: [AS5600_Configuration.c](#).

Sin embargo, hay versiones del AS5600 que para permitir la lectura analógica solo es necesario colocar el mismo voltaje de alimentación en el pin GPO.

6.2. Calibración OmniBotCC

Si se usa el sistema *OptiTrack* que se encuentra en el ecosistema Robotat, debe especificarse un cuerpo rígido dentro del sistema de *Motion*. En ese caso, para obtener correctamente la posición y orientación de la plataforma dentro del Robotat, se deben seguir los siguientes pasos de calibración:

1. Colocar la plataforma en el centro del área de captura del Robotat, con la dirección positiva del eje X tanto de la plataforma como del Robotat alineadas.
2. Visualizar la posición actual de los marcadores en la interfaz del sistema.
3. Leer los valores de posición y orientación desde MATLAB o Python; estos valores serán las compensaciones para determinar con precisión el centro de la plataforma.

Cabe aclarar, que el OmniBotCC solo puede variar su ángulo ψ (yaw) y su posición en el plano XY. Por lo mismo, solo estos datos son requeridos para usarse posteriormente como *offsets* dentro del control. Para este proceso puede usarse el scrypt [Calibracion_Robot.m](#)

6.3. Mantenimiento OmniBotCC

El OmniBot CC es una plataforma móvil que hasta la fecha es muy reciente. Por lo que los intervalos necesarios mantenimiento se desconocen con certeza. Sin embargo, en base a los componentes con los que cuenta, algunos intervalos recomendados, considerando un uso continuo, son los siguientes:

- **Cada 2 semanas:** Revisar apriete de tornillos, estado de ejes y acoplamientos. Como se pudo visualizar en la Sección 4.3, los acoplamientos entre los ejes y las ruedas son bastante delgados. Esto los vuelve muy vulnerables al desgaste, en especial si se usan velocidades muy altas.
- **Cada mes:** Revisar desgaste de ruedas; reemplazar si hay deterioro. Esto debido a que las ruedas *mecanum* requieren que sus rodillos tengan suficiente fricción con el suelo para lograr un movimiento omnidireccional.
- **Cada 3 meses:** Comprobar salud del pack de baterías y balanceo; reemplazar celdas defectuosas. Esto debe hacerse con extrema precaución, pues como se indicó anteriormente, las celdas 18650 son peligrosas. Asegúrese de no realizar corto-circuito en ninguna de las celdas, de no tener ningún circuito conectado al momento de manipularlas y sobre todo al realizar un reemplazo, de colocar celdas con las mismas características que las originales o en su defecto, cambiar todas las celdas.
- **Cada actualización de firmware:** Verificar parámetros de seguridad y restaurar límites si fue necesario. Esto incluye valores del control en lazo cerrado de los motores, como valores de velocidades máximas y mínimas de la plataforma móvil.

Se recomienda, como en todo sistema eléctrico o mecánico, llevar un registro de mantenimiento. Para el mismo, se encuentra el siguiente cuadro de ejemplo:

Fecha	Actividad	Observaciones

7. Especificaciones técnicas

- **Dimensiones:** 190 mm (largo) x 190 mm (ancho) x 90 mm (alto).
- **Masa:** apróximadamente 3.2 kg (con baterías).
- **Tensión nominal:** 12.6 V (pack 3s2p).
- **Corriente máxima continua:** 20 A (limitado y controlado por el BMS).
- **Motores:** NEMA17 de 23mm de grosor, torque nominal 0.45 Nm.
- **Velocidad máxima nominal de translación:** 2.0 m/s (400 rpms)
- **Velocidad máxima nominal de rotación:** 17.0 rad/s (400 rpms)
- **Sensores:** 4 x AS5600 (resolución angular de 12 bits).
- **Comunicación:** Wi-Fi (UDP), puerto por defecto 12345 e IP estática 192.168.50.222.
- **Controlador embebido:** ESP32 (dual-core, 240 MHz) con FreeRTOS.

8. Soporte y licencias

Si se necesita soporte:

- Diseñador: Christian Alessandro Campos López — cam21760@uvg.edu.gt
- Tutor académico: MSc. Miguel E. Zea A. — mezea@uvg.edu.gt