

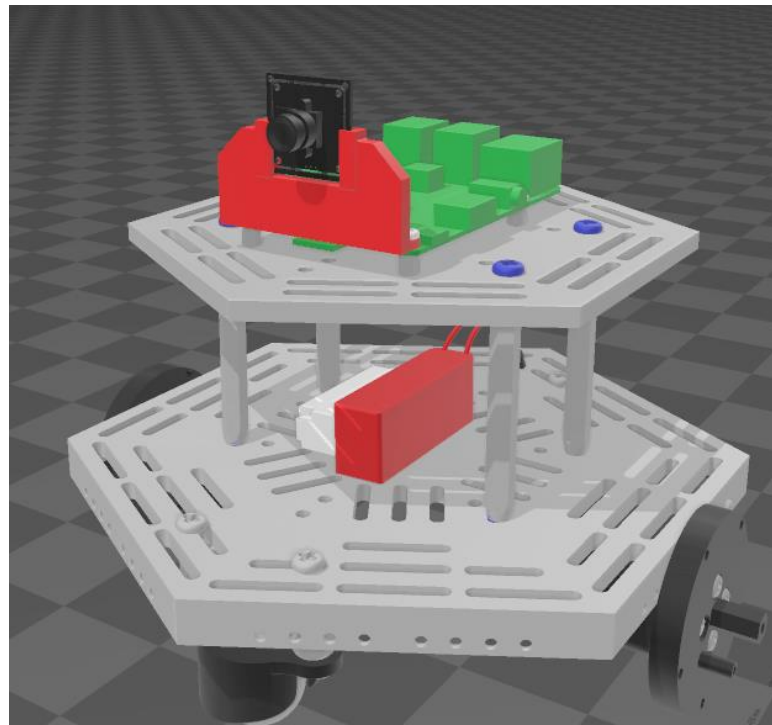


INSTITUTO POLITÉCNICO NACIONAL

CENTRO DE INVESTGACIÓN EN COMPUTACIÓN

Manual de ensamble y conexión del robot:

“EIN”



ÍNDICE

Componentes del sistema	3
Ensamblado.....	4
Chasis	4
Conexión electrónica.....	14
Flujo de la programación demostrativa del sistema.....	16

Componentes del sistema

- Dos motores AX12 Dynamixel con sus respectivos tornillos.
- Dos ruedas compatibles con los motores AX12 dynamixel con sus respectivos tornillos.
- Dos ruedas locas.
- Batería tipo LiPo de 11.1 v.
- Un BEC con salida de 5 volts a 5 amperes.
- Una Raspberry Pi 4 de 4gb de RAM.
- Raspberry Pi camera v1.3.
- Chasis impreso PLA.
- Cinturones de plástico.
- Un Arduino Nano para comunicar a la Raspberry con los motores (cuestiones técnicas).
- Varios jumpers tipo hembra-hembra y tipo macho-hembra.
- Ocho piezas de base para placa de bronce hembra-hembra con sus respectivos tornillos (2 tornillos por cada pieza) de media pulgada las primeras 4, las otras restantes dependerá de la altura entre el impreso PLA y las ruedas de motores AX12. Tenga en cuenta también el tamaño de la rueda loca.
- Cuatro piezas de base para placa de bronce hembra-hembra de 5 pulgadas con sus respectivos tornillos.
- Tornillos de media pulgada aproximadamente.
- Cable USB macho a micro-USB macho.

Ensamblado

Chasis

El ensamblado del chasis consiste en primero en instalar las 4 piezas de latón de media pulgada en la tapa superior de los impresos en PLA como se ve en la figura 1. Se deben ajustar utilizando 4 tornillos.

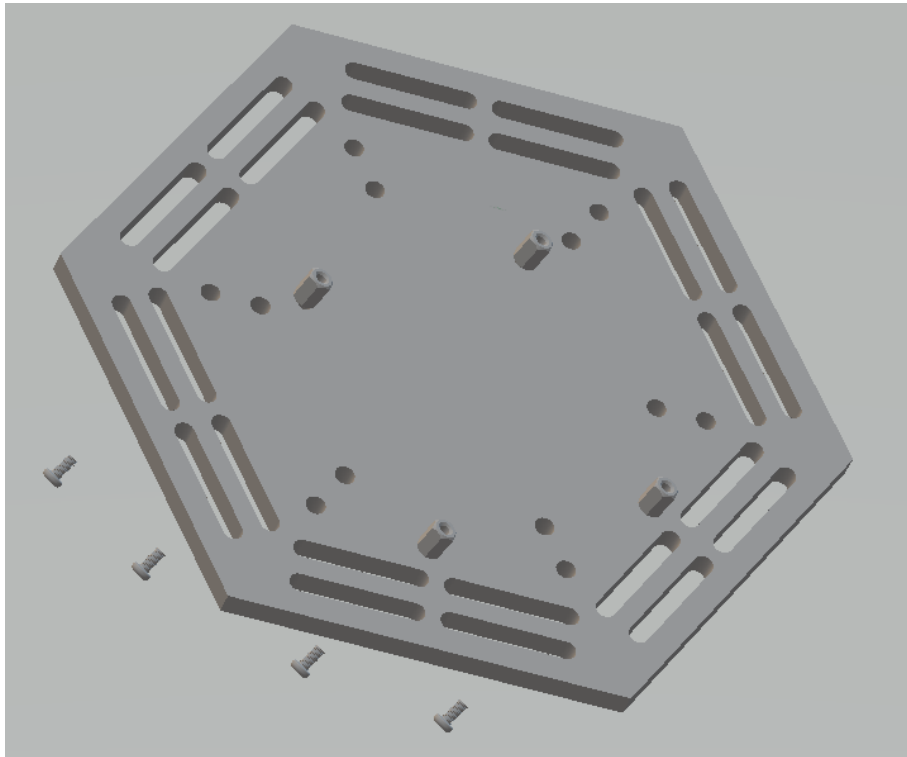


Figura 1. Colocación de piezas de latón.

Posteriormente, debemos ensamblar los motores AX12 Dynamixel los cuales son colocado en el impreso inferior PLA. Deben ir colocados a los lados del impreso de forma simétrica, como lo muestra la figura 2 y figura 3, usando los orificios marcados en círculos color rojo. Estos motores pueden ir ensamblados con tornillos como se ve en la figura 3 o con cinturones de plástico siempre y cuando vayan bien ajustados.

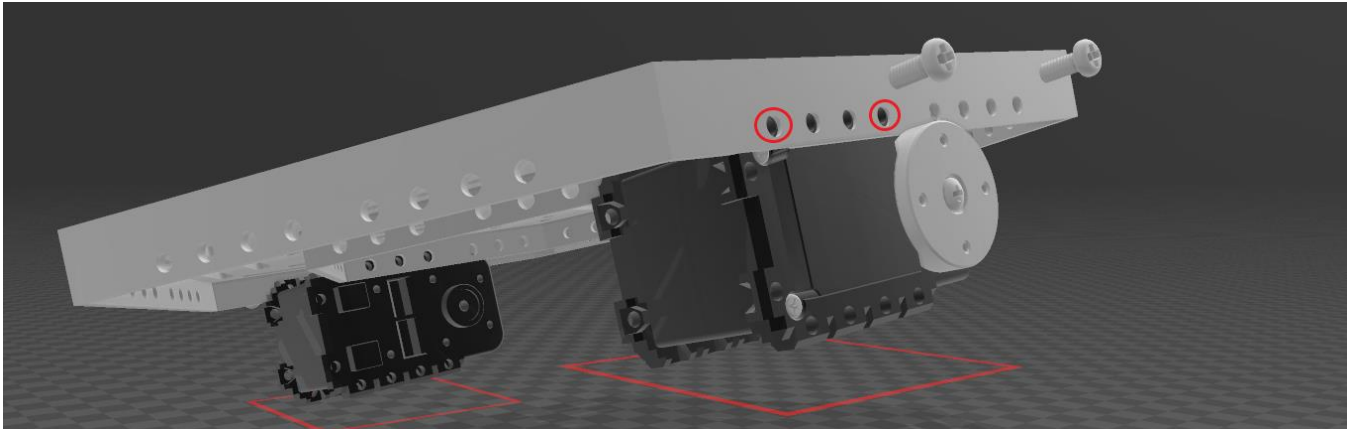


Figura 2. Ensamblado y atornillado de motores AX12.

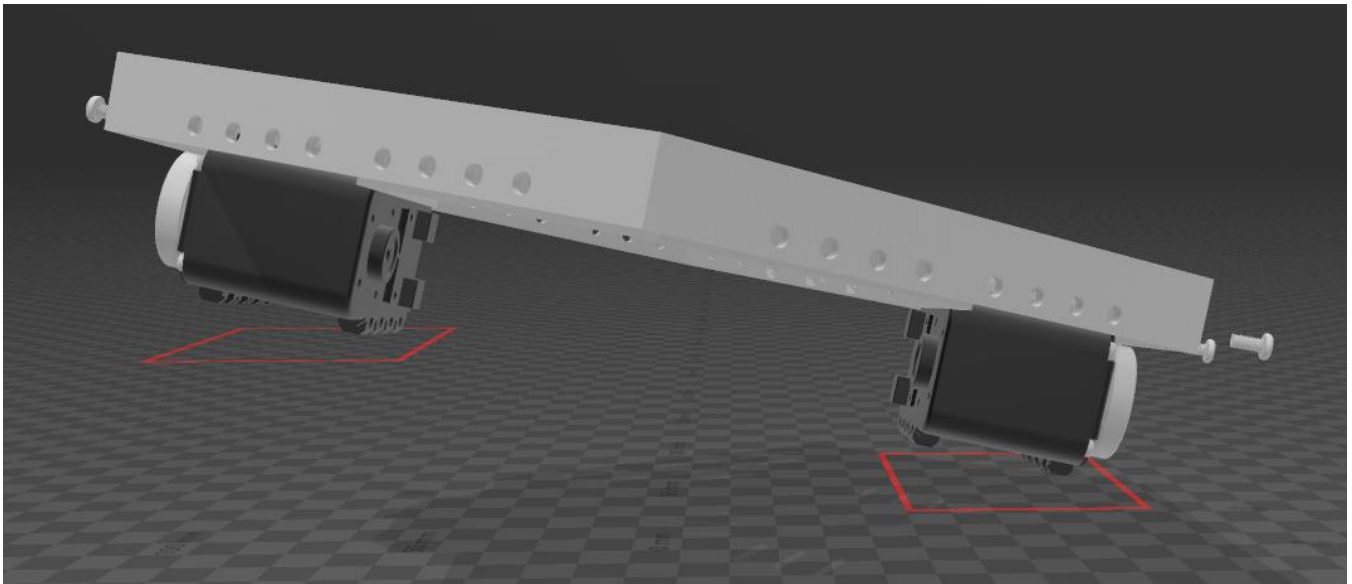


Figura 3. Viste frontal del impreso PLA inferior.

Ahora debemos colocar la batería tipo LiPo de 11.1v y el BEC de 5v en la parte central del impreso PLA, en donde se colocó previamente los motores AX12 dynamixel, como podemos ver en la figura 4. Debemos ajustar con cinturones de plástico de tal manera de que ni la batería ni el BEC se puedan mover. Se recomienda ajustar los componentes usando 3 cinturones de plástico, pasándolos por los agujeros que indican los círculos en rojo.

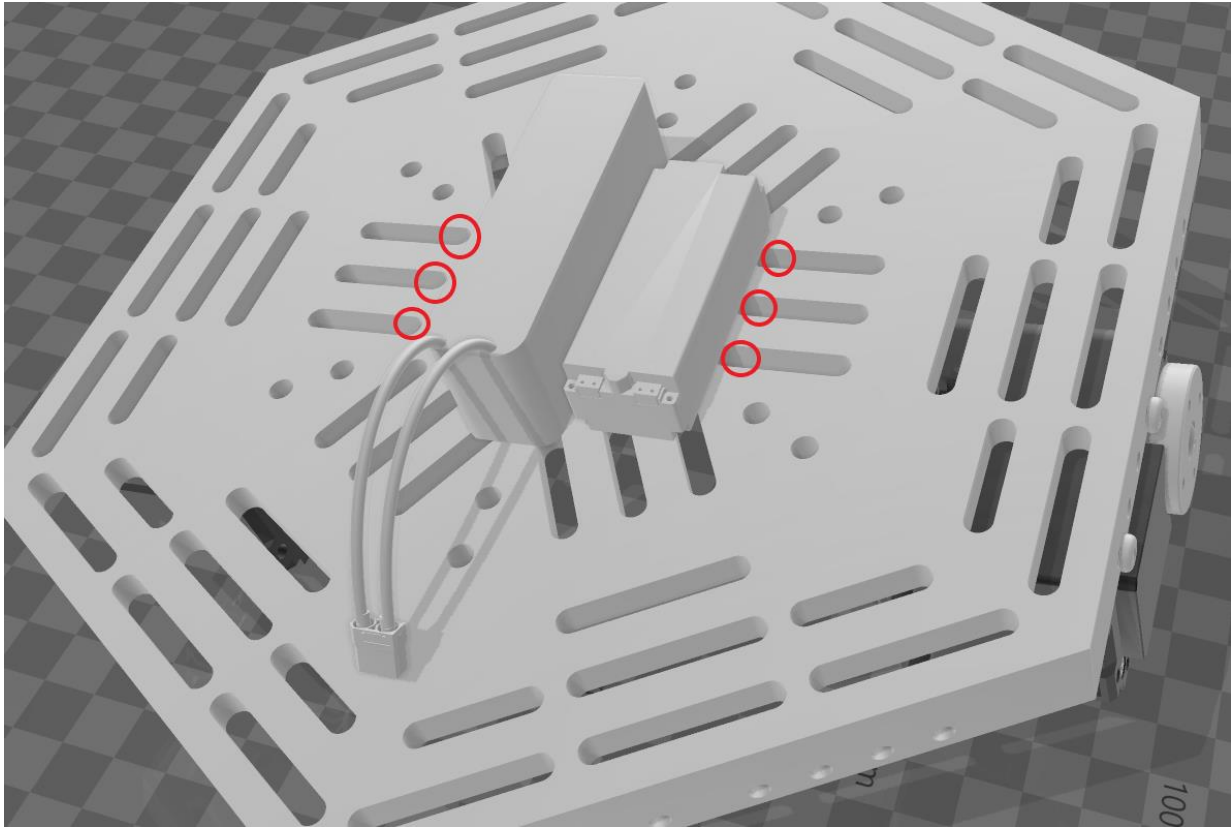


Figura 4. Colocación de la batería y el BEC.

Como siguiente paso, debemos ensamblar las ruedas de los dos motores AX12 dynamixel usando sus respectivos tornillos o pijas, según lo que se incluya en la compra. Cada rueda lleva un total de 4 tornillos como se ve en la primera imagen de la figura 5. Revise que los cuatro orificios de la rueda coincidan con los 4 orificios del motor AX12 del respectivo lado en donde ensamble primero, así como lo muestra en la segunda imagen de la figura 5. Asegúrese que las ruedas estén fijas y no se muevan de los motores, esto para evitar desestabilidad en el robot.

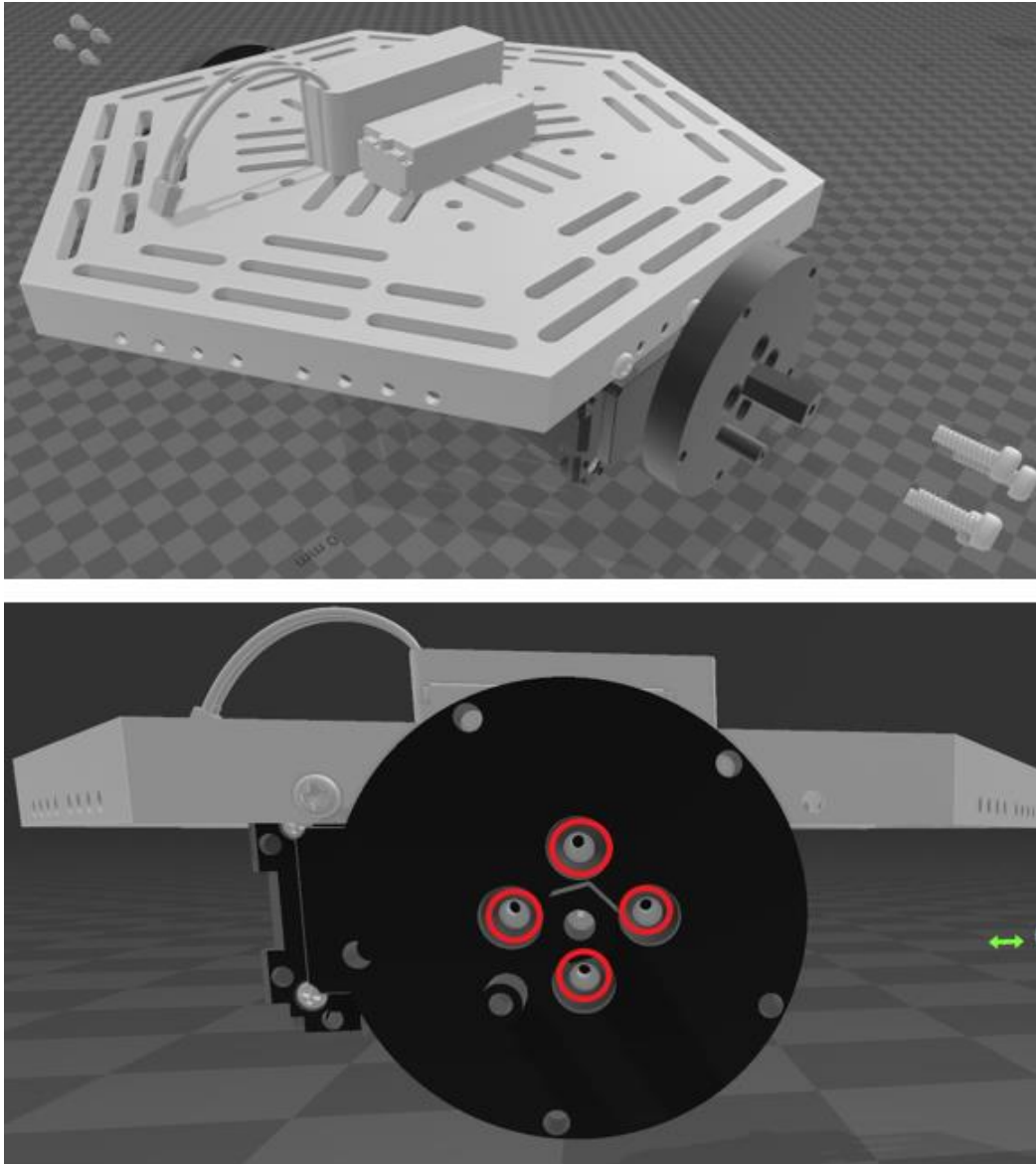


Figura 5. Ensamble de las ruedas a los motores AX12.

El siguiente paso es ensamblar las ruedas locas al chasis inferior, las cuales deben ser colocadas con dos piezas de base para placa de bronce usando 4 tornillos por cada rueda loca como lo muestra la figura 6. La posición de ensamble recomendada se puede ver en la figura 6 sin embargo, las ruedas locas pueden variar en tamaño, si es su caso, use los demás segmentos huecos. Asegúrese que ambas ruedas sean simétricas entre sí y que toquen el piso. Si no es así, considere cambiar las piezas de latón por otra de diferente tamaño.

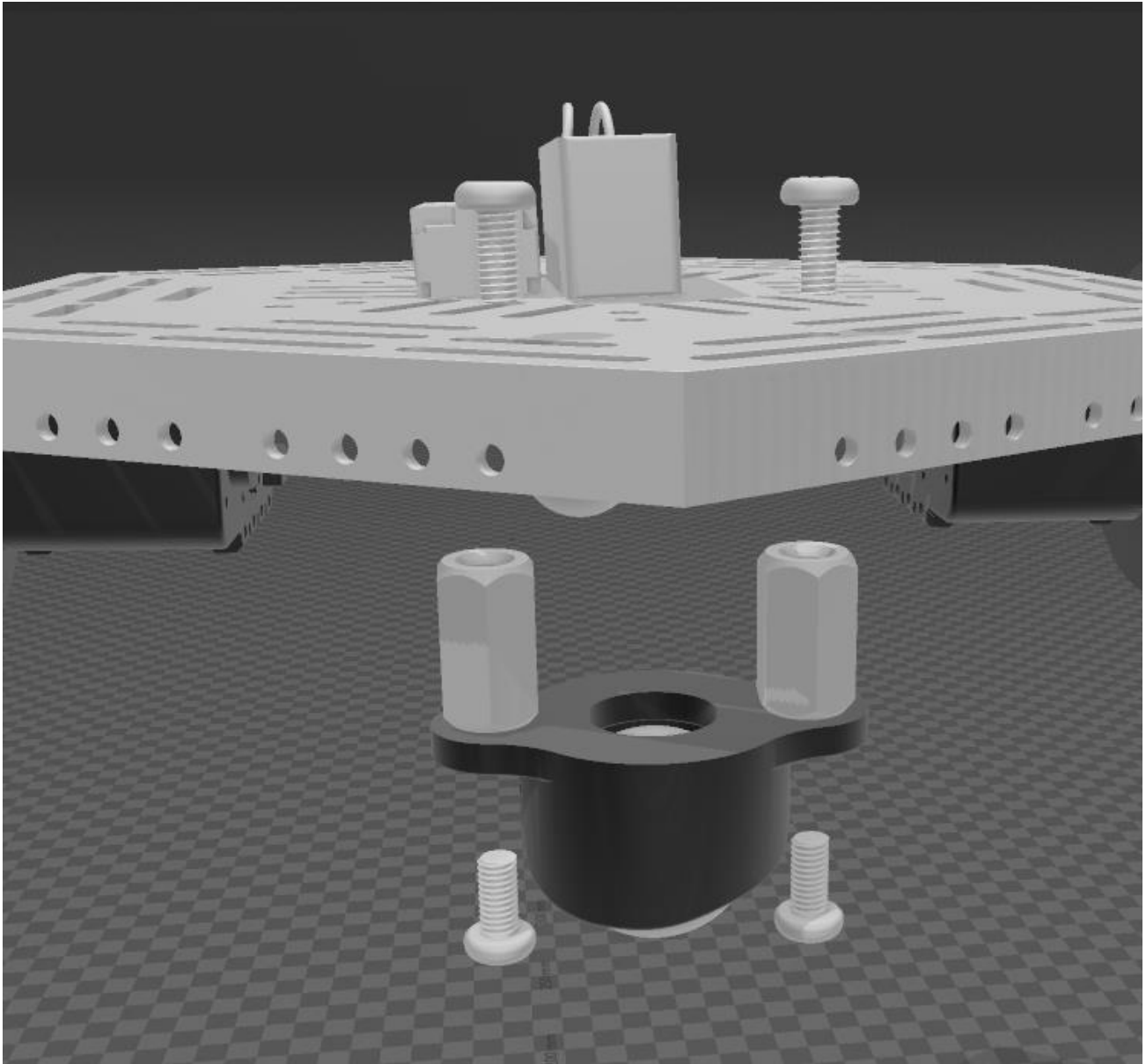


Figura 6. Ensamble de una rueda loca.

El ensamble que se ha llevado a cabo en el impreso inferior PLA debe ser similar al que se muestra en la figura 7.

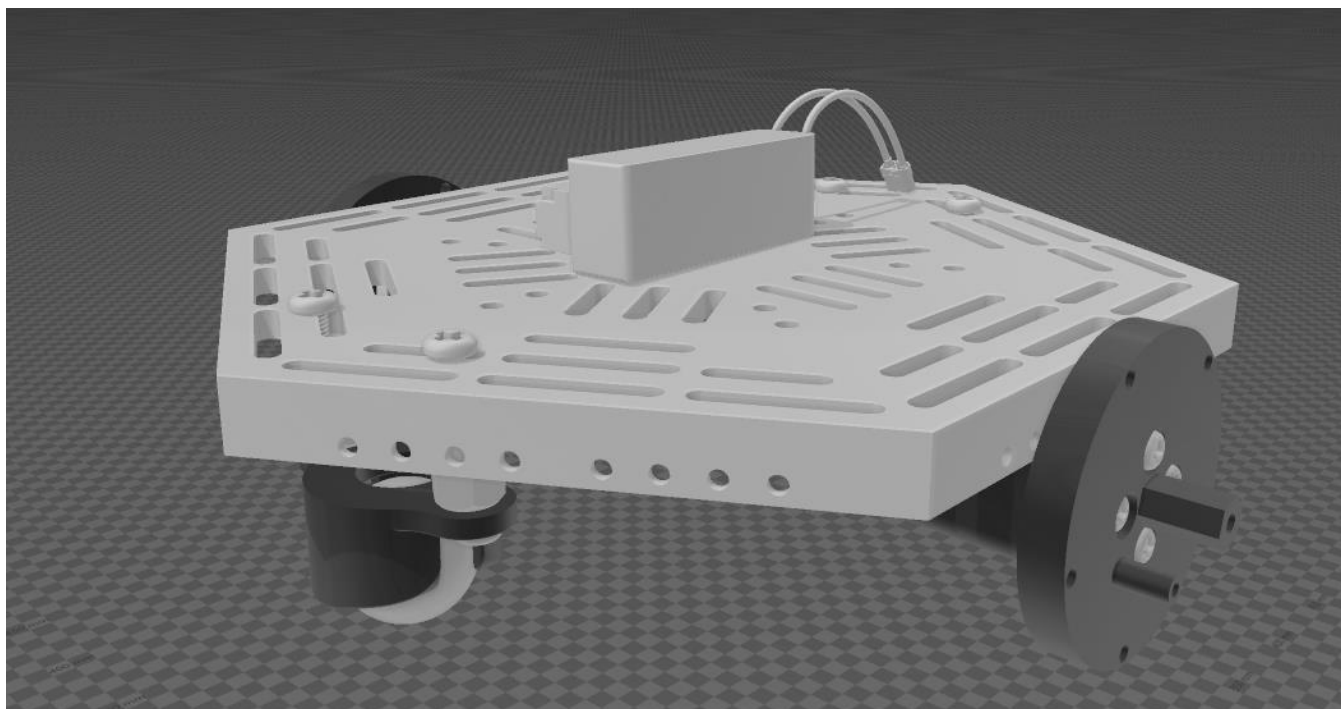


Figura 7. Vista del impreso inferior PLA armado.

A continuación, volvemos con el impreso PLA superior para ensamblar la base de la cámara y la Raspberry Pi. Se coloca como lo muestra la figura 8. Vemos que primero se pone la Raspberry Pi sobre los soportes de placa de bronce, después se sobrepone la base de la cámara del lado de las Raspberry Pi que no tiene las conexiones a periféricos y, por último, se ajustan con 4 tornillos (asegúrese que tengan un tamaño adecuado). Revise que la base de la cámara este colocada del lado en donde esté más cerca de la orilla del impreso como se puede ver en la misma figura 8.

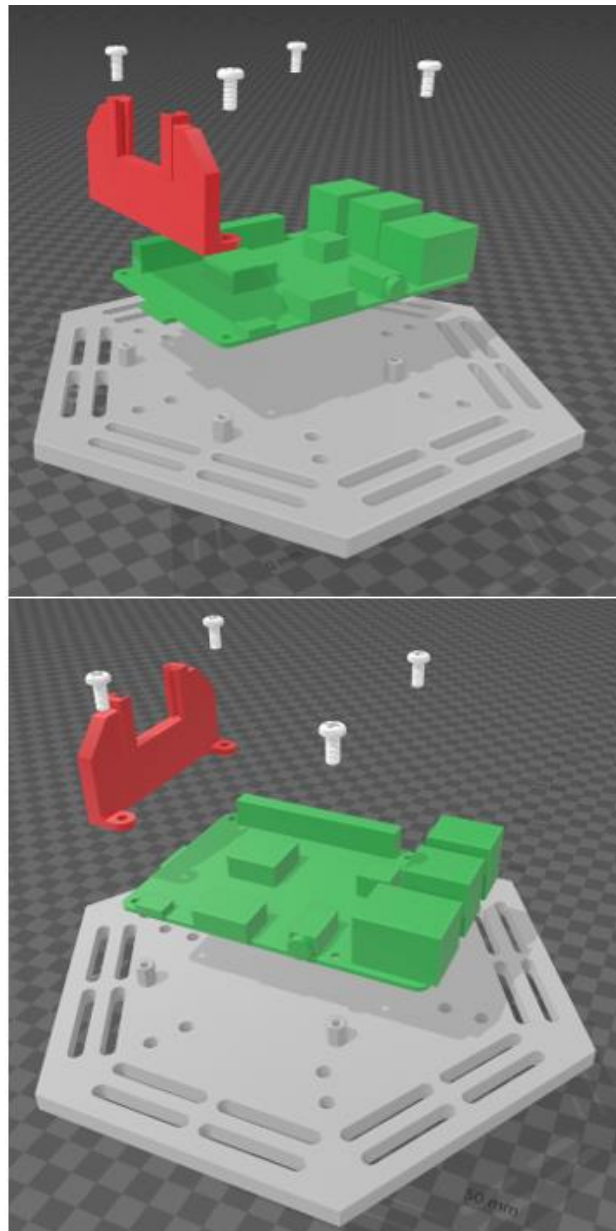


Figura 8. Ensamble de la base de la cámara y Raspberry Pi.

Siguiendo con el armado, uniremos los impresos PLA superior e inferior a través de las bases para placa de 5 pulgadas. Para esto, uniremos el PLA superior con las cuatro bases para placa de bronce usando sus respectivos tornillos como se ve en la figura 9. Revise que los tornillos entren en los agujeros correctos como lo muestra la segunda imagen de la figura. Asegúrese de apretar bien los tornillos.

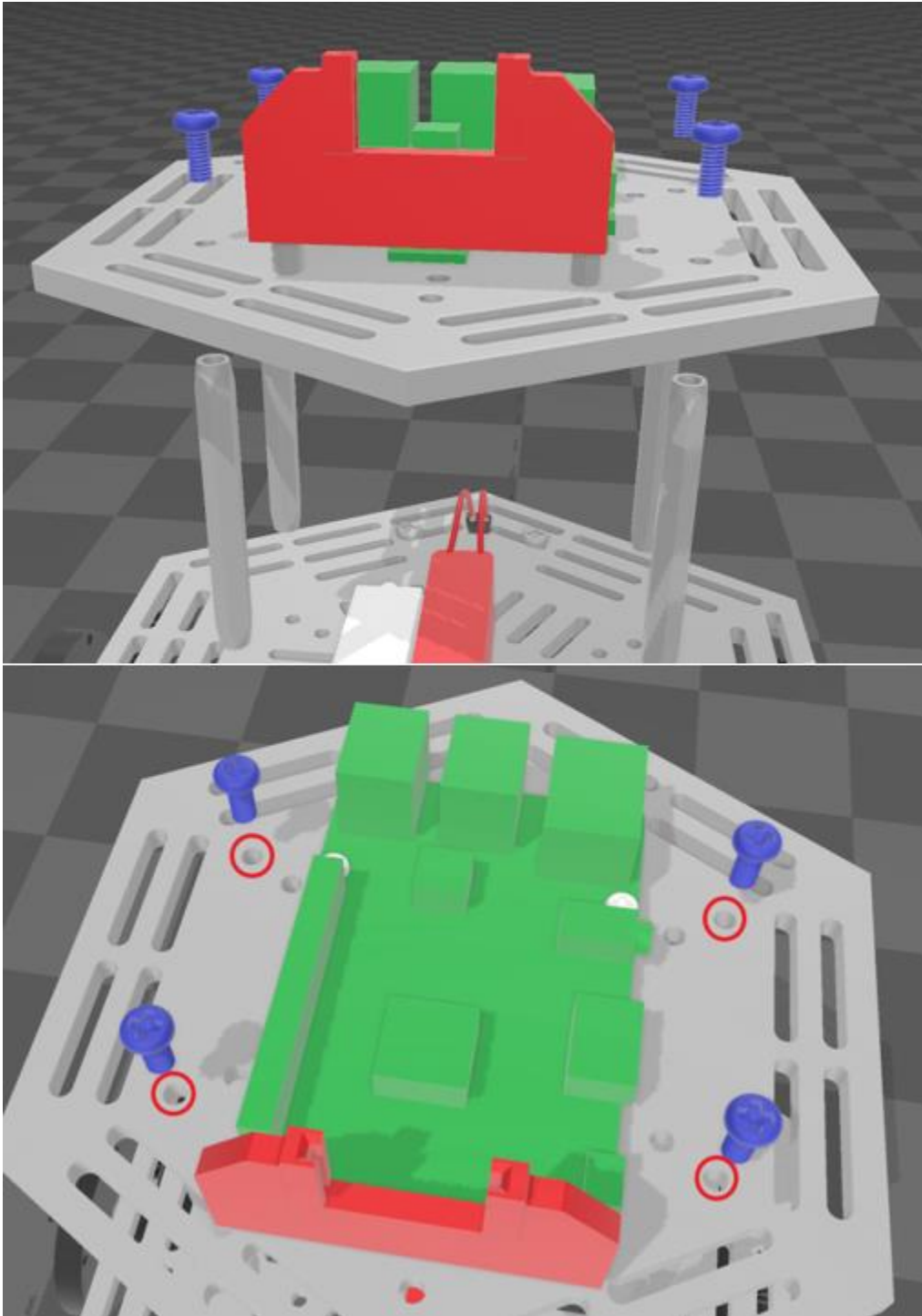


Figura 9. Unión de las bases para placa con el impreso PLA superior.

Como penúltimo paso, uniremos los PLA. Voltee de cabeza el PLA superior e inferior, ajuste el PLA inferior y sus agujeros con los agujeros de los extremos de las bases para placa de bronce de tal manera que, al insertar el tornillo, este una el impreso inferior con la pieza de bronce. Los agujeros del PLA inferior están marcados en la figura 10. Use esta figura como guía.

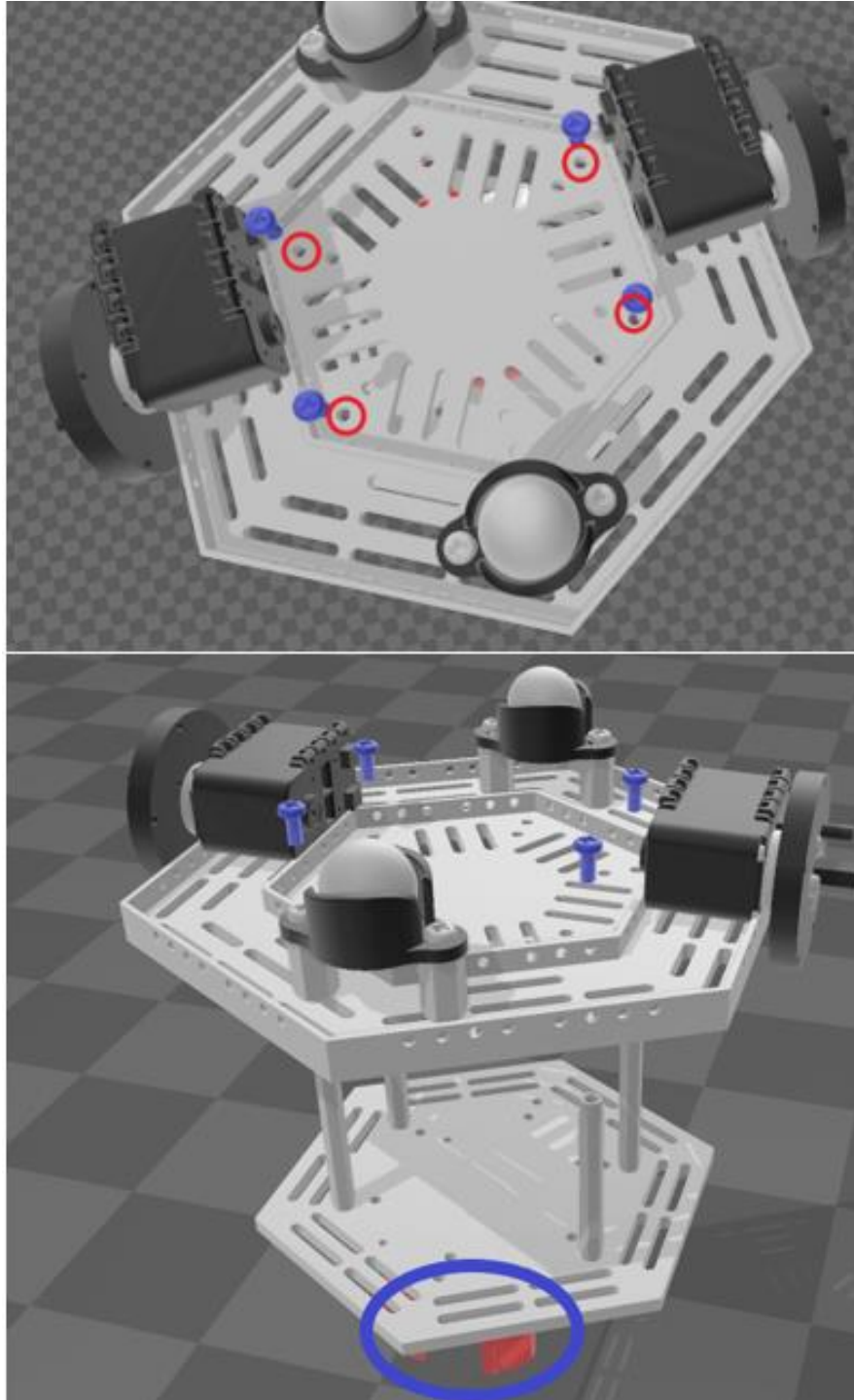


Figura 10. Unión final de impreso superior con el inferior.

Por ultimo, ensamblamos la cámara de Raspberry Pi en su respectiva base, como lo muestra la figura 11. Tenga en cuenta la posición de la cámara para evitar que quede de cabeza y por lo tanto la imagen capturada por ella.

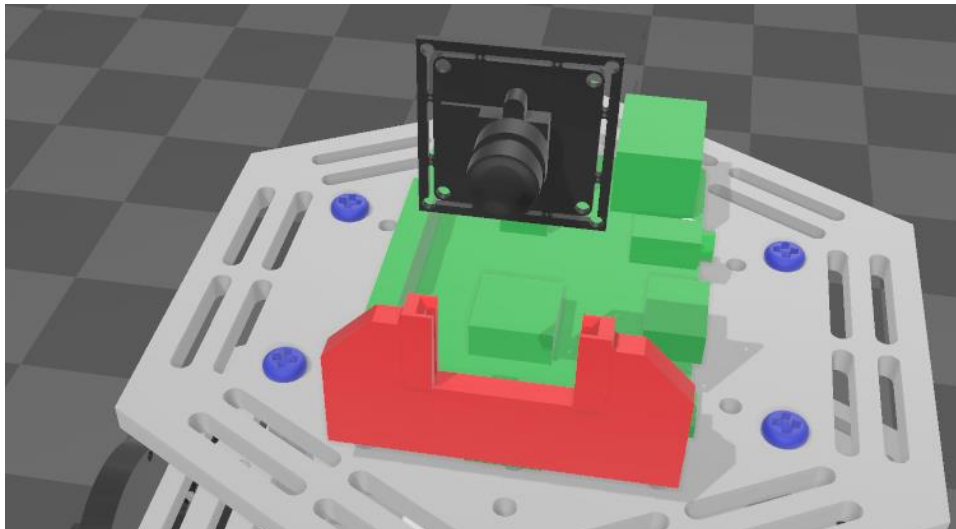


Figura 11. Ensamble de cámara de Raspberry Pi.

Al final del ensamble, el robot debe lucir algo similar a lo que se muestra en la figura 12.

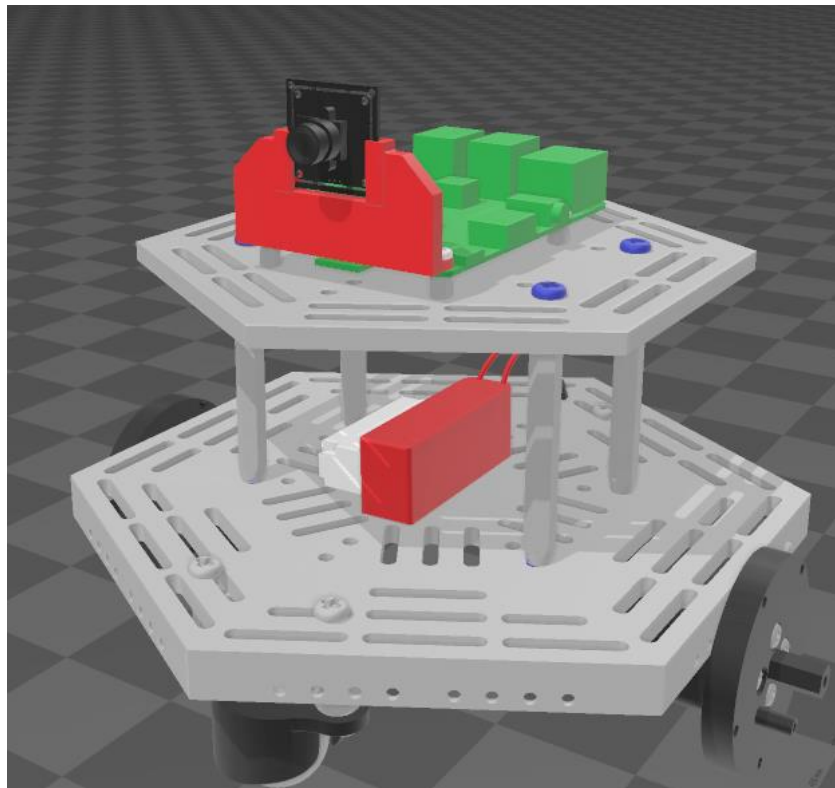


Figura 12. Ensamble total de Ein.

Conexión electrónica

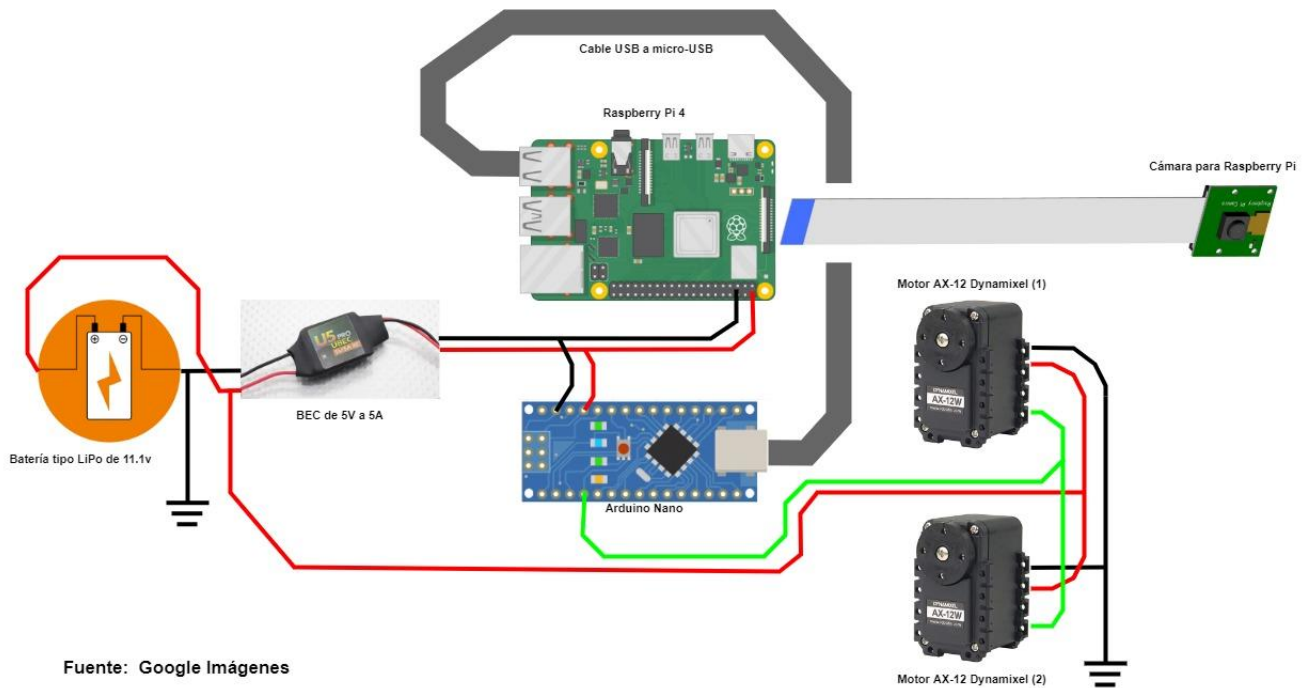


Figura 13. Conexión electrónica de Ein.

Como podemos ver en el diagrama de la figura 13, se realiza una conexión entre el positivo de la batería de 11.1 volts con la entrada positiva del BEC de 5 volts. El lado negativo de la batería, así como la entrada negativa del BEC se conectan a GND.

Las salidas del BEC van conectadas tanto a la entrada de 5 volts de las Raspberry Pi como a la entrada de 5 volts del Arduino Nano, así como lo muestra el diagrama de conexión.

Entre la Raspberry Pi y el Arduino Nano se llevará a cabo una comunicación serial, por lo que se necesita conectar un cable USB macho a micro-USB macho. El lado USB macho del cable se conectará al puerto USB de la Raspberry Pi y el lado micro-USB macho del cable al puerto micro-USB del Arduino Nano.

Las entradas positivas de ambos motores AX12 Dynamixel se conectan también al positivo de la batería, y ambas entradas negativas se conectan a la salida negativa de la batería (GND). Su alimentación es de 11.1 voltios.

Con un jumper, se debe conectar la salida D9 del Arduino Nano a ambas entradas "Data" de los motores AX12 Dynamixel.

Puede optar por agregar una o más protoboards, según lo que considere para realizar las conexiones, por ejemplo, en la figura 14 se puede ver que se empleó un protoboard pequeño para ajustar el Arduino Nano con sus respectivas conexiones y fue ajustado usando un cinturón de plástico, usando los huecos que el mismo inferior contiene.

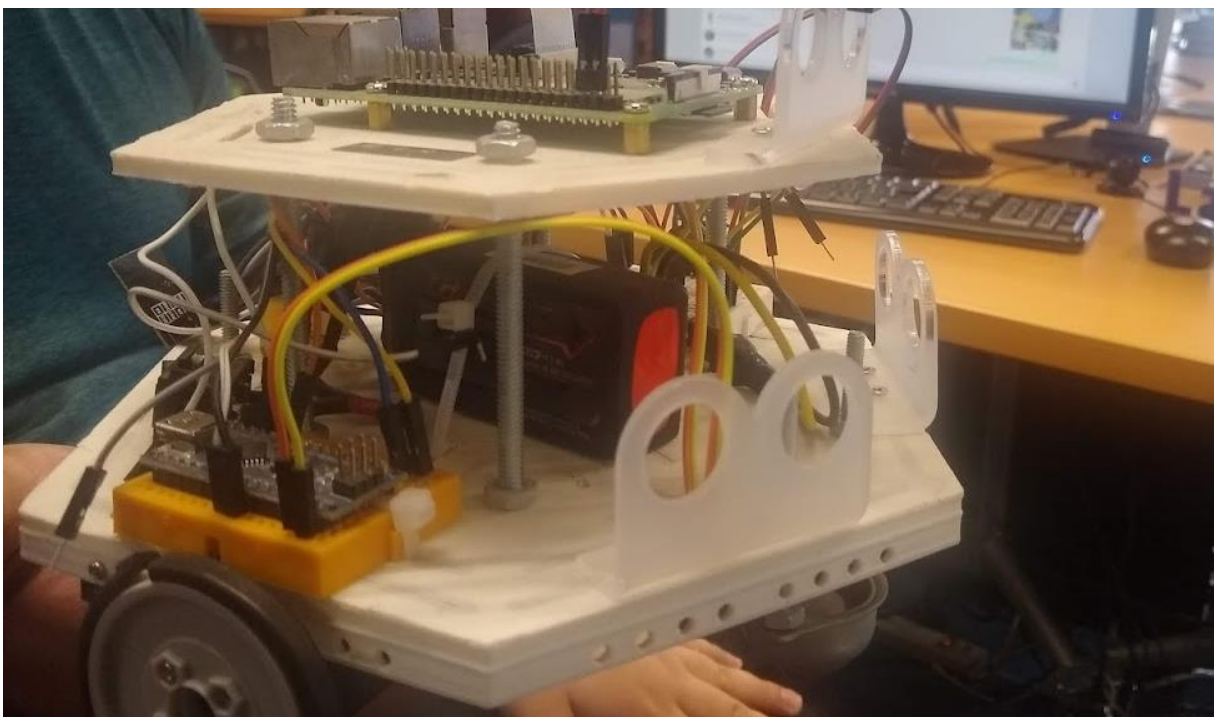


Figura 14. Uso de protoboard para ensamble del Arduino Nano.

Flujo de la programación demostrativa del sistema

El programa demostrativo esta realizado en el framework ROS (Robot Operating System) usando Python 2 y una distribución Linux Raspbian, los cuales están instalados en la Raspberry Pi.

El sistema tiene como entrada de datos en un ciclo “infinito” con movimientos básicos como adelante, atrás, derecha o izquierda, los cuales se ejecutan cada 5 segundos consecutivamente y se encuentran predefinidas el archivo **pasarela.py**. Este mismo archivo contiene la configuración del nodo llamado **pasarela** y del tópico llamado **navcom** el cual nos ayudará a publicar los datos arrojados por el nodo.

El nodo **pasarela** publica sus datos a través de datos de tipo “string” al nodo **easynav** el cual está configurado e inicializado en el archivo **easynav.py**. El nodo **easynav** tiene como función convertir el tipo de movimiento recibido (forward, backward, right y left) en valores para la velocidad de los motores del robot. Estos datos son publicados a través del tópico **velocity** de tipo “string”.

El nodo **motors**, cuya configuración se encuentra en archivo **motors.py**, es el encargado de recibir la información del nodo **easynav** y enviarla a los motores a través de una conexión serial entre la Raspberry Pi y el Arduino Nano.

Adicionalmente, se pueden escribir los movimientos en un archivo llamado **Demo.txt**, donde se especifica el tipo de movimiento a ejecutar y el tiempo en segundos que “dormirá” el programa.

El nodo que se encarga de leer estos datos se llama **reader**, cuya configuración se encuentra en el archivo **reader.py** y publica su contenido al topico **velocity**.