

Test de Hardware y Software

1. Un robot móvil:
 - a. Siempre tiene ruedas
 - b. Siempre tiene gps
 - c. Siempre tiene encoders
 - d. Ninguna de las anteriores

2. El driver de un motor:
 - a. Realiza el control de motores de corriente continua.
 - b. Incluyen un controlador interno.
 - c. Incluyen la etapa de potencia
 - d. Todas las anteriores

3. En los motores de corriente continua, el ángulo girado se mide con:
 - a. Sensores de efecto Hall.
 - b. Encoders.
 - c. Telémetro láser.
 - d. Ninguna de las anteriores.

4. En los motores brushless, el ángulo girado se mide con:
 - a. Sensores de efecto Hall.
 - b. Encoders.
 - c. Telémetro láser.
 - d. Ninguna de las anteriores.

5. Para medir distancias a los objetos el sensor más indicado es:
 - a. Telémetro laser.
 - b. Sensores de ultrasonidos.
 - c. Varios controladores conectados en BUS.
 - d. Todas las anteriores

6. Para integrar múltiples sensores digitales se emplea:
 - a. Un telémetro.
 - b. Un driver.
 - c. Una tarjeta de entradas salidas (DAQ)
 - d. Ninguna de las anteriores

7. Para el correcto uso de las baterías:
 - a. Deben dejarse las baterías totalmente descargadas.
 - b. Debe utilizarse un cargador adecuado.
 - c. Deben dejarse las baterías totalmente cargadas.
 - d. Ninguna de las anteriores

8. Las setas y sistemas de emergencia de un robot:
 - a. Deben actuar directamente sobre los actuadores del robot.
 - b. Deben desconectar los sistemas de control.
 - c. Se colocan en paralelo entre si.
 - d. Ninguna de las anteriores.

9. El control del robot se puede realizar:
 - a. Mediante computador tipo PC.
 - b. Mediante micro controladores.
 - c. Mediante combinación de computadores y microcontroladores.
 - d. Todas las anteriores.

10. Los micro robots:
 - a. Son plataformas de bajo coste.
 - b. Existe mucha información para su desarrollo en internet.
 - c. Permite la iniciación en la robótica.
 - d. Todas las anteriores

11. El framework ROS:
 - a. Permite la integración de nodos escritos en C++.
 - b. Permite el trabajo con Matlab desde una máquina virtual.
 - c. Permite la integración de nodos escritos en Java.
 - d. Todas las anteriores

12. Gazebo:
 - a. Es un simulador de robots para ROS.
 - b. Es una librería de procesamiento de imágenes.
 - c. Es un lenguaje de programación de robots.
 - d. Ninguna de las anteriores.

13. Matlab:
 - a. Contiene toolboxes de robótica.
 - b. Permite la conexión con un robot real.
 - c. Permite probar y programar software para robots sin necesidad de compilación.
 - d. Todas las anteriores.

14. La variable de entorno ROS_IP
 - a. Guarda la IP de ese ordenador para que ROS sepa interpretar a quién le tiene que enviar un topic. .
 - b. Guarda la IP del ordenador que está proporcionando los servicios del master al resto de nodos.
 - c. Guarda la IP de ese ordenador para que ROS sepa interpretar a quién le tiene que enviar un topic. Pero si también se ha utilizado la variable ROS_HOSTNAME, esta última es la que tiene preferencia.
 - d. Ninguna de las anteriores.

15. Matlab, ROS y Gazebo:
- Son incompatibles entre si.
 - Son las únicas herramientas para programar robots.
 - Se integran entre sí permitiendo desarrollar y programar algoritmos de control de robots.
 - Ninguna de las anteriores.
16. El Algoritmo A*
- Encuentra el camino de menor coste entre un nodo origen y uno objetivo, siempre y cuando se cumplan unas determinadas condiciones.
 - Se guía en exclusiva por una función heurística.
 - No realiza ninguna búsqueda en anchura.
 - Todas las anteriores.
17. El Algoritmo “Pure Pursuit”
- Es un algoritmo para encontrar el camino óptimo en un grafo.
 - Es un algoritmo para seguir trayectorias.
 - Es un algoritmo de relocalización.
 - Es un algoritmo de exploración.
18. Si estoy con un Turtlebot en el entorno virtual de las prácticas y quiero obtener la orientación del robot en forma de un ángulo con el eje X
- Me suscribo a /odom, capturo el mensaje en un objeto del tipo correspondiente, accedo al campo “orientation” y obtengo la orientación en forma de cuaternio, que tendré que transformar a ángulos de Euler.
 - Me suscribo a /odom, capturo el mensaje en un objeto del tipo correspondiente, accedo al campo “orientation” y obtengo un vector cuyo componente “theta” es el ángulo que busco.
 - Me suscribo a /mobile_base/commands/velocity, capturo el mensaje en un objeto del tipo correspondiente, accedo al campo “orientation” y obtengo la orientación en forma de cuaternio, que tendré que transformar a ángulos de Euler.
 - Me suscribo a /mobile_base/commands/velocity, capturo el mensaje en un objeto del tipo correspondiente, accedo al campo “orientation” y obtengo un vector cuyo componente “theta” es el ángulo que busco.
19. Si desde Matlab he creado un mensaje en la variable “msg” para publicar en el topic “topic” y quiero conocer la estructura del mensaje, escribo
- rosmessage(msg).
 - rosmessage(topic)
 - showdetails(msg)
 - showdetails(topic)
20. ¿Qué sucede si en un nodo de ROS (cuyo main no contiene ningún bucle) que publica en el topic T1 y está suscrito al topic T2 se olvida poner ros::spin()?
- No se publicarán mensajes en T1 pero la función de callback se ejecuta con normalidad cuando se recibe un mensaje en T2.
 - No funciona nada: ni se publica en T1 ni se ejecuta la función de callback cuando se recibe un mensaje en T2.
 - No se ejecutará la función de callback que debería ejecutarse cuando se recibe un mensaje en T2, pero los mensajes se publican con normalidad en T1.
 - No pasa nada.