

0.- Presentación

[1] Buenos días miembros del tribunal. Mi nombre es Juan Ruiz Tovar y les presento mi proyecto fin de grado titulado: Desarrollo de un rover explorador para terrenos abruptos. Rover Ulysses para el proyecto UGRASP.

1.- Introducción

[2] Este proyecto se sitúa en el marco de la iniciativa UGR Aerospace Program o UGRASP. Esta iniciativa del Departamento de Arquitectura y Tecnología de Computadores de la UGR se propone ayudar al desarrollo de las habilidades en ingeniería de los estudiantes colaboradores mediante la realización de misiones espaciales simuladas.

El rover Ulysses sobre el que versa este proyecto es un vehículo autónomo para la exploración de terrenos abruptos.

[3] Ulysses tiene asignada la misión de explorar una formación rocosa rodeada de agua conocida como Púlpito de Canales situada en el embalse de Canales, Granada, España.

****Explicar los requisitos de la misión****

[4] Para el desarrollo de la misión, Ulysses cuenta con los siguientes elementos y capacidades.**

2.- Configuración previa

[5] La configuración de la Raspberry Pi consta de la instalación del Sistema Operativo en la tarjeta SD y de la configuración del mismo. El SO elegido es Raspbian, una distribución Linux basada en Debian muy popular en la comunidad de Raspberry Pi y que está ampliamente documentada y mantenida. Para la configuración mínima del SO hemos hecho uso de la herramienta preinstalada raspi-config con la cual hemos activado la comunicación por el bus I2C y el acceso por SSH.

[6] El módulo GSM está conectado a la Raspberry Pi mediante el conector GPIO y para encenderlo es necesario activar en alta la salida GPIO 13 durante 2 segundos para lo cual se ha creado el script python GSMStart.py. Posteriormente lo configuramos instalando los paquetes wvdial y ppp que sirven para realizar la "llamada" (dialing) de configuración hacia el proveedor de telefonía móvil y para configurar la interfaz de red a usar por el módulo GSM respectivamente.

[7] Debido a la limitación por parte del proveedor de telefonía móvil que consiste en el bloqueo de determinados puertos, entre ellos el 22 usado por ssh, surge la necesidad de crear una conexión mediante la técnica del tunel inverso para poder redireccionar el tráfico del puerto 22. Para ello usaremos como servidor intermedio el servidor atcproyectos.ugr.es el cual tiene abierto el puerto 2022. Esta conexión se automatiza con el paquete autossh y un script al inicio.**

3.- Diseño Hardware

[8] Respondiendo a la necesidad de crear un sistema con bajo nivel de acople al igual que en las misiones espaciales reales, la funcionalidad se encuentra dividida en subsistemas conectados mediante el bus I2C y un lenguaje de comunicación propio llamado Código UL.

[9] El subsistema de computación es el encargado de procesar datos y proporcionar funcionalidad de alto nivel como: comunicación inalámbrica GSM, toma de datos GPS, captura y procesamiento de imágenes.

[10] El subsistema de control se encarga de operar el rover a bajo nivel en tareas de movimiento y adquisición de datos.

[11] El diseño mecánico del rover Ulysses está basado en el sistema de locomoción para vehículos autónomos conocido como Shrimp rover publicado en el estudio de Roland Siegwart "Innovative design for wheeled locomotion in rough terrain". Sus características principales son:

- El sistema de estabilización es pasivo, es decir, no necesita actuadores para mantener la estabilidad en terreno abrupto. Esto reduce la complejidad, tamaño, peso y consumo energético.
- Poder superar obstáculos del tamaño de 1.5 veces el diámetro de la rueda. Ulysses usa ruedas de 114 mm, por lo tanto de forma teórica puede superar obstáculos de 171 mm. En la práctica puede superar obstáculos mayores.

[12] [13] Los procesos CAD** / CAM** / CNC** permiten el diseño y fabricación automatizada de componentes.

La herramienta CAD usada para diseñar Ulysses es SketchUp y la herramienta CAM VCarve Pro, ambas elegidas por la facilidad de uso y baja curva de aprendizaje. La herramienta CNC está incluida en la terminal integrada en la propia máquina CNC.

Análogamente al desarrollo software, el diseño CAD ha seguido el modelo de desarrollo conocido como Modelo en Espiral que se detalla a continuación.

4.- Desarrollo Software

[14] Diseño en espiral o ciclo de vida en espiral, o basado en refinamientos de prototipos

[15] La comunicación entre subsistemas se realiza mediante el bus I2C. Típicamente, la configuración del bus I2C se basa en el modelo de comunicación Maestro-Esclavo** con la posible existencia de varios Maestros. Los subsistemas de Ulysses están conectados en un modelo Mestro-Maestro en el que Raspberry Pi tiene el control de Arduino a alto nivel. Podemos definir esta configuración como Maestro-Falso Maestro.

[16] Debido a la necesidad de modularizar el diseño tanto en software como en hardware, Ulysses utiliza un formato de comunicación propio entre subsistemas. Esto permite desacoplar el diseño de forma que los subsistemas puedan remplazarse si se desea. La comunicación se lleva a cabo en el bus I2C en la dirección 0x04** mediante ráfagas de Bytes con el siguiente formato.

****Explicar transparencia****

5.- Presupuesto, fabricación y montaje

[17] A continuación se muestra el presupuesto de fabricación del rover excluyendo la mano de obra y aproximando los costes derivados a la compra de los productos.**

[18] [19] Las siguientes diapositivas recogen imágenes sobre el proceso de fabricación y montaje.**

6.- Conclusiones y trabajo futuro

[20] Tras el desarrollo del proyecto se ha conseguido diseñar y fabricar el rover Ulysses así como dotarle de un sistema software fiable y ampliable para el manejo del mismo.

[21] Este proyecto ha conseguido sentar las bases del desarrollo iterativo del rover Ulysses y en el futuro se espera:** También se propone la actualización del hardware de control con alternativas como Raspberry Pi Zero o Udoo Neo.**