Dron de vuelo autónomo con reconocimiento basado en inteligencia artificial.

Autonomous flight drone with recognition based on artificial intelligence.



Trabajo de fin de grado del grado en ingeniería informática.

Curso académico 2019-2020.

Realizado por Juan Carlos De Alfonso Juliá.

Dirigido por Juan Jiménez Castellanos.

Índice

[Índice 2](#_Toc31278921)

[Resumen. 4](#_Toc31278922)

[Palabras clave. 4](#_Toc31278923)

[Introducción. 4](#_Toc31278924)

[Etapa 1: Diseño y construcción. 4](#_Toc31278925)

[Diseño hardware 4](#_Toc31278926)

[Primer acercamiento: elementos que podemos necesitar. 4](#_Toc31278927)

[Segundo acercamiento: Selección concreta de componentes. 4](#_Toc31278928)

[Construcción final del prototipo. 5](#_Toc31278929)

[Impresión 3d (cambiar nombre) 5](#_Toc31278930)

[Integración de los componentes. 5](#_Toc31278931)

[Etapa 2: software del prototipo 6](#_Toc31278932)

[Sistema operativo 6](#_Toc31278933)

[Mpu9250 IMU. 6](#_Toc31278934)

[MS5611. 6](#_Toc31278935)

[RGB LED. 6](#_Toc31278936)

[Controlador PWM. 6](#_Toc31278937)

[Etapa 4: Diseño de una aplicación de control. 6](#_Toc31278938)

[Introducción. 6](#_Toc31278939)

[Protocolo de comunicación. 6](#_Toc31278940)

[Modelo cliente servidor. 6](#_Toc31278941)

[Protocolo TCP. 6](#_Toc31278942)

[Funcionamiento interno. 6](#_Toc31278943)

[Estructura de la aplicación. 7](#_Toc31278944)

[Logo de la aplicación. 7](#_Toc31278945)

[Ventanas de la aplicación. 7](#_Toc31278946)

[9](#_Toc31278947)

[Estructura interna. 9](#_Toc31278948)

[Etapa 5: Diseño del reconocedor basado en IA. 10](#_Toc31278949)

[TensorFlow API. 10](#_Toc31278950)

[Etapa 6: Pruebas finales. 10](#_Toc31278951)

[Conclusiones. 10](#_Toc31278952)

[Bibliografía. 10](#_Toc31278953)

Resumen.

Palabras clave.

Dron, PID

Introducción.

Etapa 1: Diseño y construcción.

# Diseño hardware

## Primer acercamiento: elementos que podemos necesitar.

De forma abstracta resulta sencillo el saber que elementos podemos necesitar para construir el aparato:

* Chasis donde poder montar el resto de componentes.
* Ordenador que actúe como “cerebro” y sea capaz de al recibir información de los sensores tomar decisiones durante el vuelo.
* Una cámara para la etapa de reconocimiento.
* Una batería.
* Cuatro motores con sus respectivos variadores y hélices.
* Sensores para poder conocer la inclinación del aparato, sus coordenadas, la altura del o el nivel de batería.
* Un módulo de comunicación inalámbrica (wifi, bluetooth…).

## Segundo acercamiento: Selección concreta de componentes.

Para el chasis se ha elegido el modelo F450 de la marca DJI dado que al tener una superficie amplía la colocación del resto de componentes será más sencilla además de contar con un tablero con una pcb integrada lo que hace mucho más fácil el soldar y colocar los cables.

Para el ordenador de a bordo hay dos opciones claras(Arduino o Raspberry) en concreto dado que vamos a necesitar poder ejecutar un algoritmo de machine learning y alguna forma de conectarnos al ordenador lo más sencillo es usar una Raspberry, en concreto el modelo 3B+ que cuenta con bluetooth y wifi de serie.

Los sensores es quizá la parte más importante pues deben estar colocados de manera correcta en el aparato para poder funcionar correctamente, por la cantidad de sensores necesarios se decidió comprar un HAT para la raspberry llamado NAVIO2 compatible con nuestro modelo de raspberry.

Imagen que contiene electrónica, circuito

Descripción generada automáticamente

En concreto este HAT cuenta con un barómetro de alta resolución (), dos sensores IMU (), la compatibilidad de mediante un adaptador poder ser alimentado por una batería LIPO, un módulo gps y 14 canales pwm para poder controlar los motores.

biblio

…

# Construcción final del prototipo.

## Impresión 3d (cambiar nombre)

Con el propósito de que la estrucutra fuera más sólida algunos componentes se han unido al chasis usando piezas impresas en 3d mediante una impresora Prusa mk3s.

## Integración de los componentes.

Etapa 2: software del prototipo

# Sistema operativo

# Mpu9250 IMU.

# MS5611.

# RGB LED.

# Controlador PWM.

Etapa 4: Diseño de una aplicación de control.

# Introducción.

La aplicación de nombre Pathfinder en honor a la sonda enviada a explorar Marte la he diseñado desde cero en los lenguajes java y XML usando Android Studio como entorno de desarrollo. El propósito de dicha aplicación es poder gestionar el recorrido del dron de forma integra permitiendo la selección de un plan de vuelo incluyendo el recorrido, la altitud o el lugar de aterrizaje.

La conexión entre el dispositivo móvil y el prototipo se ha realizado mediante un modelo cliente-servidor a través de un protocolo TCP explicado más adelante.

Nota: la aplicación ha sido testeada en un móvil Xiaomi MI A2 con una versión 10 de Android por lo que en versiones anteriores no se asegura el funcionamiento de esta, además es necesario conceder los permisos de ubicación a esta aplicación para poder usarla.

# Protocolo de comunicación.

## Modelo cliente servidor.

El modelo cliente servidor es una estructura de comunicación entre dos dispositivos (cliente y servidor) a través de internet.

## Protocolo TCP.

## Funcionamiento interno.

En el caso concreto del dron y el teléfono el primero actuará como servidor ejecutando un bucle de espera hasta que un cliente (el teléfono) se conecte y transmita la información.

Únicamente se transmitirán dos mensajes, el primero por parte del teléfono que contendrá un línea del estilo siguiente:

6.3|40.34523,-3.43463|42.343243,-3.3532…

En donde el primer valor corresponde a la altura que debe llevar el dron durante todo su recorrido, y el resto de valores corresponden a pares de coordenadas por donde debe pasar el aparato.

El segundo mensaje corresponde a una confirmación de recibido (ACK) que pondrá fin a la comunicación.

El código del cliente se encuentra ….

# Estructura de la aplicación.

## Logo de la aplicación.

## Ventanas de la aplicación.

La aplicación la forman cinco ventanas.

La primera de estas corresponde a la que vemos al ejecutar la aplicación y corresponde a un menú en donde se pueden seleccionar las opciones de iniciar vuelo, ajustes e información junto con el logo ampliado de la aplicación.

Si seleccionamos el botón de información se abrirá una nueva ventana con información acerca de la app (instrucciones, autoría…).

En el caso de seleccionar ajustes, la interfaz mostrará una serie de parámetros configurables que se aplicarán durante el vuelo del dron o la conexión con este, en concreto encontramos dos submenús, el primero llamado Protocolo TCP permite configurar dos parámetros relativos a la comunicación:

* Dirección IP: dirección ip asignada a la raspberry dentro de la red.
* Puerto: identificador único del puerto del servidor de la raspberry, por defecto usaremos el puerto 12345.

El segundo submenú referente a la navegación permite configurar lo siguiente:

* Altura: la altura en metros a la que debe volar el aparato.
* Vuelta al origen: representa un switch que en el caso de estar seleccionado indicará al dron que al finalizar el recorrido debe volver al punto del que partió y aterrizar ahí, en caso de no seleccionarse aterrizará en el último punto establecido por el usuario.

La tercera de las ventanas a la que se llega si se selecciona el botón “iniciar vuelo” abrirá una interfaz más compleja que las anteriores formada por un mapa en donde aparecerá la ubicación de nuestro aparato así como dos botones “iniciar” y “reset”.

En esta ventana debemos seleccionar el recorrido que queremos que haga el aparato pulsando en el mapa, esto hará que se creen diferentes puntos (con un máximo de diez) que se recorrerán en orden de selección, el botón reset permite borrar todos los puntos y volver a crear el recorrido, y el botón iniciar nos lleva a la última ventana explicada a continuación.

La última ventana es meramente informativa pues cuando se inicia, gráficamente no realiza ninguna acción, es código de esta en cambio es muy importante pues realizará la comunicación inalámbrica enviando todos los datos necesarios y usando la dirección ip y el puerto de los ajustes para dicha acción.

Imagen que contiene captura de pantalla

Descripción generada automáticamenteImagen que contiene captura de pantalla

Descripción generada automáticamente

## Imagen que contiene captura de pantalla Descripción generada automáticamenteImagen que contiene captura de pantalla Descripción generada automáticamente

Fig …

## Estructura interna.

Los códigos referentes a la aplicación se encuentran … y se subdividen en los siguientes archivos:

Archivos .java

Corresponden a clases usadas..

Java//AjustesActivity.java

Java//InfoActivity.java

Java//MainActivity.java

Es la clase principal llamada al inicio de la aplicación, está formada por varios métodos que actúan como liseners de los botones de la interfaz.

Es la encargada de pasar los parámetros de configuración de vuelo y tcp a la siguiente clase teniendo en cuenta que los parámetros por defecto han podido ser modificados, esta acción se realiza recogiendo los datos enviados por los ajustes con la siguiente porción de código:

Java//VueloActivity.java

Java//VueloActivity.java

Archivos.xml

Manifest/AndroidManifest.xml

res/layout/activity\_ajustes.xml

res/layout/activity\_info.xml

res/layout/activity\_main.xml

res/layout/activity\_vuelo\_msg.xml

res/layout/activity\_vuelo\_map.xml

Etapa 5: Diseño del reconocedor basado en IA.

# TensorFlow API.

Etapa 6: Pruebas finales.

Conclusiones.

Bibliografía.