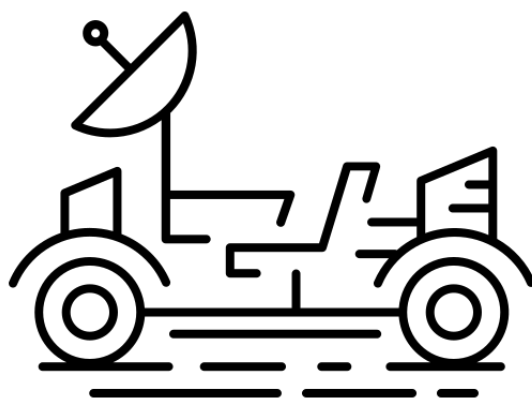




Software Asistente Vuelo Virtual

Manual de instalación y uso



Índice de contenidos

Introducción	3
Repositorio de código	4
Instalación y uso del componente: App Android	4
Instalación Software	4
Modificación Software	5
Uso de la APP	6
Instalación del componente: Arduino	8
Implementación Hardware	8
Instalación Software	8

Introducción

El presente documento tiene como objetivo describir los pasos de instalación del software que forma parte de proyecto del “Carrito asistente de vuelo virtual autónomo”, también conocido por sus siglas en inglés: “AAA: Automated Aircraft Assistant”.

Este proyecto tecnológico ha sido desarrollado por alumnos del Máster de Formación del Profesorado de la **UPM** para su uso en el aula por parte de los alumnos, profesores y profesionales del **C.P.E.E. PRÍNCIPE DE ASTURIAS (Aranjuez)**.

El software está diseñado para su utilización en los dispositivos y elementos físicos específicos implementados por la UPM para dicho proyecto.

La solución está formada por los siguientes módulos funcionales independientes:



- **Módulo 1:** App Móvil: Aplicación móvil para dispositivos Android. Es la interfaz del usuario para controlar el movimiento del carrito a distancia a través de conexión Bluetooth.
- **Módulo 2:** Carrito acoplable: Carrito de asistente de vuelo para el transporte de bebidas, alimentos en el avión. Fabricación con madera.
- **Módulo 3:** Estructura de sujeción y electrónica: Cajetín de madera que sirve como pieza de integración entre el carrito y el patinete eléctrico, además de ser el contenedor de la electrónica necesaria para el control del patinete. En su interior se encuentra instalado una placa controladora Arduino UNO con un sensor Bluetooth HC5. La placa Arduino se encuentra cableada hacia la fuente de alimentación del patinete y las conexiones de intensidad del motor para controlar la velocidad deseada.

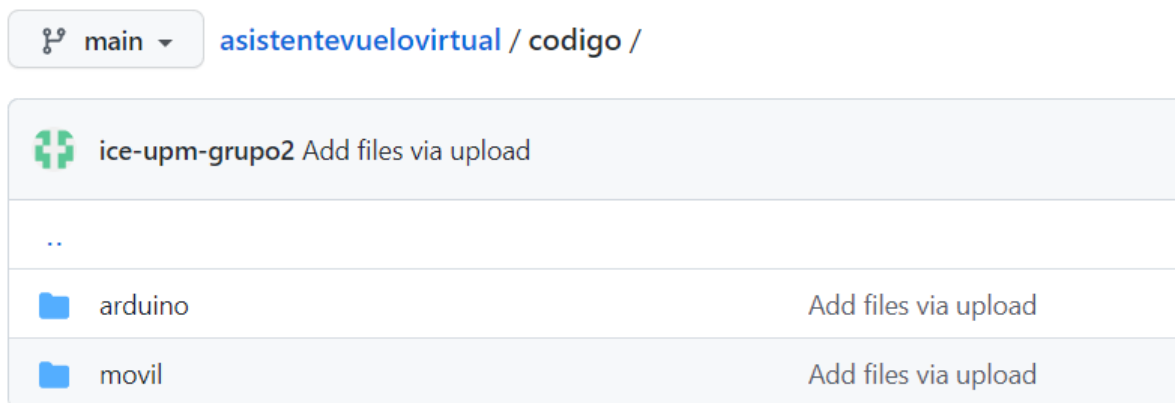
- **Módulo 4:** Patinete eléctrico: Patinete eléctrico estándar que realiza las funciones requeridas para la movilidad de la solución.

Repositorio de código

Todo el código y documentación asociado al Software de este proyecto puede encontrarse y descargarse del siguiente repositorio público de GitHub:

<https://github.com/ice-upm-grupo2/asistentevuelovirtual>

Además de encontrar la documentación en la raíz del repositorio, en la rama “main” bajo la carpeta de “código”, se encuentran 2 carpetas con lo necesario para instalar y/o modificar el Software si fuera necesario:



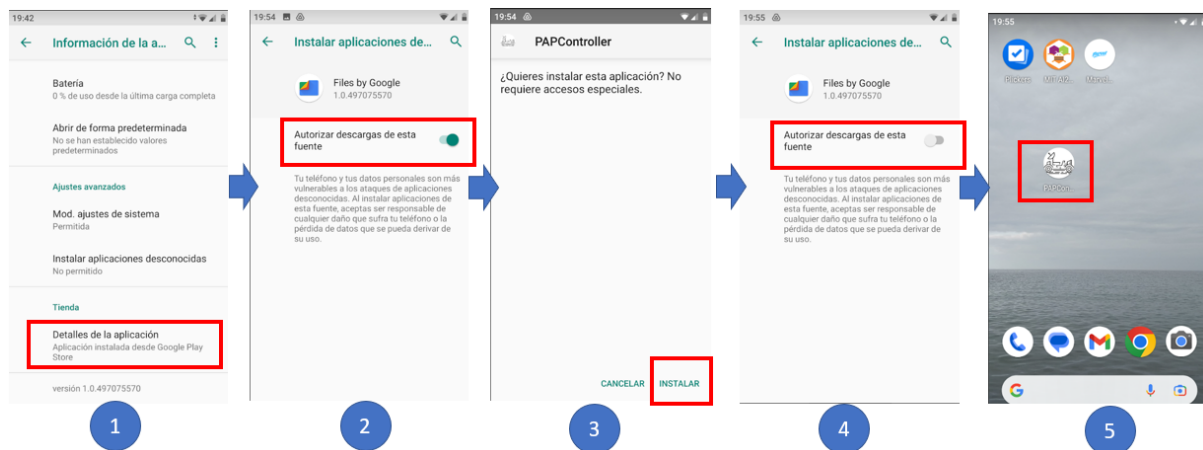
Instalación y uso del componente: App Android

Instalación Software

La aplicación móvil ha sido diseñada para ser instalada en dispositivos Android. Se ha comprobado su funcionamiento en móvil Xiaomi A1, y Tablet.

Para su instalación se proporciona un fichero APK ya construido en el repositorio: <https://github.com/ice-upm-grupo2/asistentevuelovirtual/tree/main/codigo/movil>

Una vez descargado el archivo .APK al dispositivo móvil donde se desea realizar la instalación según se muestra en las imágenes que se describen a continuación:



1. En primer lugar, hay que dar permiso a la aplicación de explorador de archivos de Android para: “Autorizar descargas de esta fuente”. Desde el menú de configuración de aplicaciones, se selecciona la aplicación de explorador a utilizar para ejecutar la APK. En este caso “Files by Google”, y se hace clic en “**Detalles de la aplicación**”.
2. Hacer clic en “**Autorizar descargas de esta fuente**”.
3. Hacer clic en “**Instalar**”.
4. **IMPORTANTE:** Por seguridad, una vez instalada la aplicación, **volver a dejar desactivada** la opción de “**Autorizar descargas de esta fuente**”.
5. La aplicación aparecerá instalada en el dispositivo móvil con el siguiente icono:



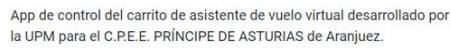
Modificación Software

Si se desea realizar cualquier modificación a la APP entregada, esto podrá hacerse mediante el proyecto publicado en la plataforma “App Inventor” a través del siguiente enlace:

<https://gallery.appinventor.mit.edu/?galleryid=b7eac395-92c9-431d-9273-7a9c840fd543>

Con este enlace se navegará a la galería donde se encuentra la APP. Haciendo clic en el enlace “**Load App into MIT App Inventor**”, el usuario podrá cargar el proyecto en su propio entorno de desarrollo de este servicio proporcionado por el MIT. Una vez dentro del entorno, podrá cambiar el diseño y/o el código de la APP y generar una nueva APK con los nuevos cambios si se desea:

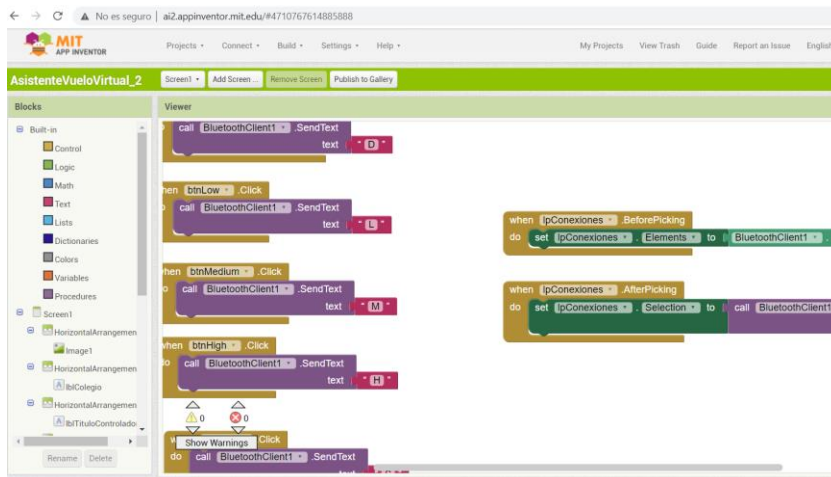
Jan. 28, 2023, 1:56 p.m. Likes: 0 ❤️



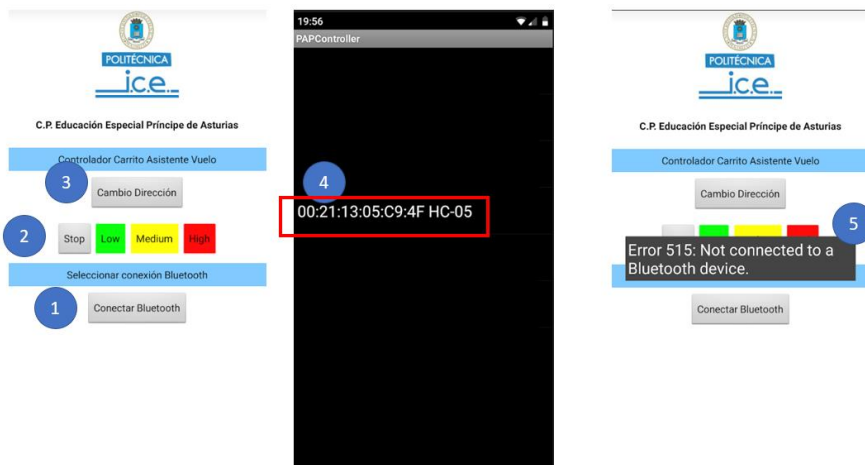
Other projects by same author

Report Project

Permanent link: <https://gallery.appinventor.mit.edu/?galleryid=b7eac395-92c9-431d-9273-7a9c840fd543>



Una vez instalada la aplicación en un dispositivo móvil, se puede acceder a ella. En las siguientes imágenes se describe la funcionalidad:



1. La primera imagen por la izquierda muestra el panel principal de control. En primer lugar, lo que hay que hacer es conectarse al dispositivo Bluetooth del patinete. Para ello, pulsar en el botón “Conectar Bluetooth” situado en la parte inferior de la pantalla. Aparecerá un listado del que hay que seleccionar el dispositivo Bluetooth del Arduino. (En el nombre aparecerá **HC-05**).

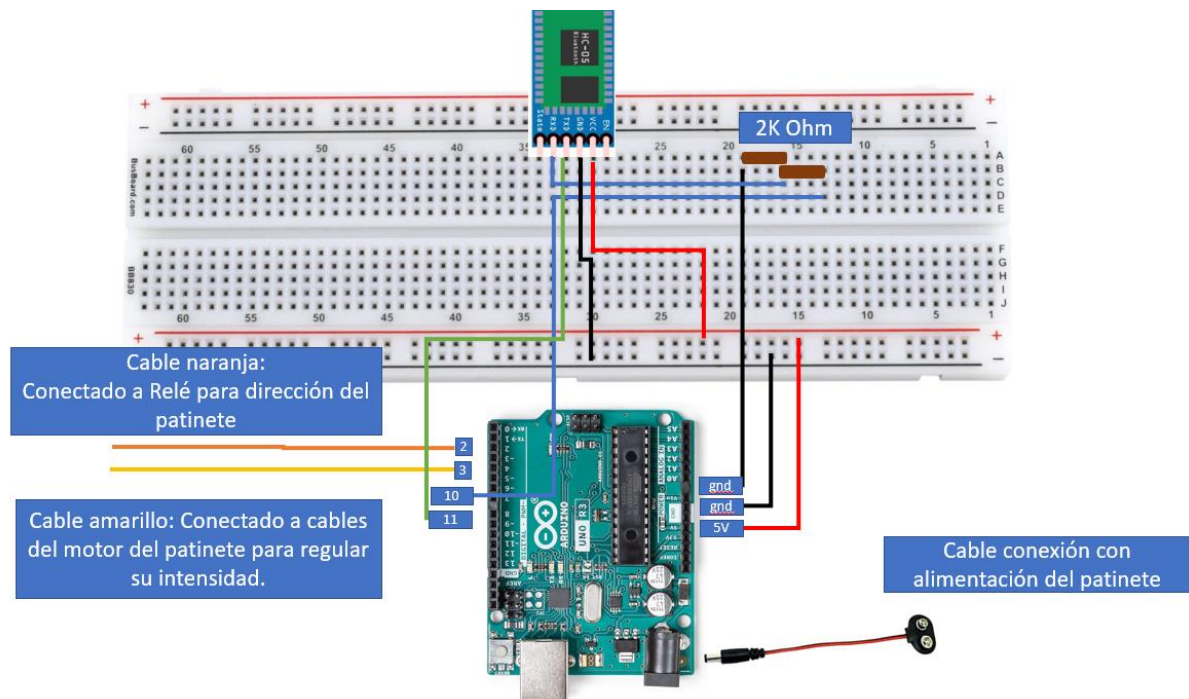
IMPORTANTE:

- Para que el sensor bluetooth del patinete aparezca en la lista a seleccionar de la APP, es necesario haber activado previamente el bluetooth del dispositivo móvil, y haberse conectado al menos una vez con anterioridad. Al establecer esta conexión, se solicitará un pin que en este caso es “**1234**”.
 - En caso de obtener un error de permisos para el uso del Bluetooth al conectar, asegúrese que la aplicación tiene los permisos necesarios para el uso del Bluetooth dentro de la configuración de aplicaciones Android.
2. La botonera principal se encuentra en la mitad de la pantalla y cuenta con los siguientes botones que pueden ser utilizados una vez se ha establecido la conexión bluetooth con el patinete:
 - Low: Baja velocidad. Proporciona al motor del patinete una intensidad baja.
 - Medium: Velocidad media. Proporciona al motor del patinete una intensidad media.
 - High: Alta velocidad. Proporciona al motor del patinete una intensidad alta.
 - Stop: Para el patinete. Deja de proporcionar intensidad al motor del patinete.
 3. Botón de cambio de dirección: activa un relé instalado en la electrónica del patinete que permite que el motor gire en sentido inverso, pudiendo dar marcha atrás al patinete.
 4. Listado de selección de bluetooth que aparece tras presionar el botón explicado en el punto 1.
 5. Mensaje de error que aparece en caso de que se intente interactuar con el patinete a través de la botonera principal sin haber conectado con el bluetooth previamente.

Instalación del componente: Arduino

Implementación Hardware

Como se ha comentado anteriormente, la electrónica que hace posible la interacción con el patinete y su movimiento está implementada en una placa microcontroladora de Arduino UNO R3. A continuación se describe el esquema de conexiones instalado en el módulo.



1. Pin 2: Digital. Conectado al relé para controlar la dirección del patinete
2. Pin 3: PWM. Conectado a cable de intensidad de motor del patinete para regular su velocidad.
3. Pin 10: PWM / Digital. Conectado a sensor bluetooth RXD (recepción)
4. Pin 11: PWM / Digital. Conectado a sensor bluetooth TXD (transmisión)
5. Gnd: Conexiones de tierra
6. 5V: Conexión que proporciona alimentación al sensor bluetooth.
7. Cable de alimentación a Arduino desde fuente de alimentación del patinete.

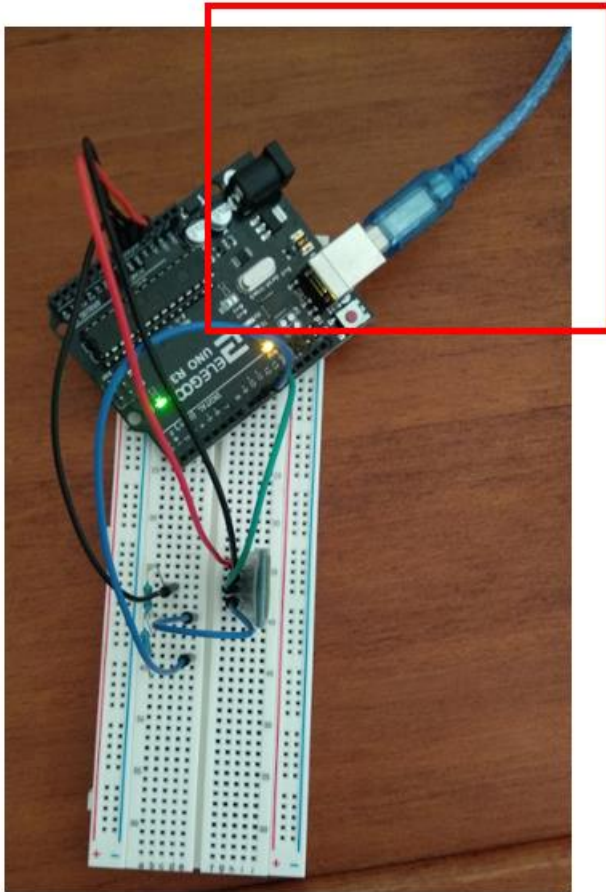
Instalación Software

El software para cargar en el controlador Arduino UNO se puede encontrar en la siguiente ruta del repositorio proporcionado:

<https://github.com/ice-upm-grupo2/asistentevuelovirtual/tree/main/codigo/arduino>

Este software ya viene preinstalado en la tarjeta de Arduino, pero si en algún momento se requiere volver a instalar por cualquier motivo, esto debe de hacerse utilizando el IDE de Arduino descargable desde: <https://www.arduino.cc/en/software>

Para ello se conectará el cable USB del PC al Arduino de la siguiente manera, y una vez hechos los cambios se subirán a la controladora mediante el botón de “Upload” del IDE.



Este código puede modificarse si se desea a partir del IDE. Algunos parámetros a tener en cuenta son los que se encuentran al inicio del código donde se definen los distintos pines que se utilizan y los valores utilizados para las distintas intensidades del motor:

```
SoftwareSerial miBT(11,10);

//Variable para leer el valor enviado desde la APP
char DATO = 0;
//PIN para Relé de cambio de dirección (Simulación LED Rojo)
int LEDROJO = 2;
//PIN para intensidad del motor del patinete (Simulación LED Verde)
int LEDVERDE = 3;
//Variables para niveles de intensidad
int INTENSIDAD_LOW = 75;
int INTENSIDAD_MEDIUM = 100;
int INTENSIDAD_HIGH = 250;
int INTENSIDAD_PARADA = 0;
```