

# Establecimiento de protocolo de comunicación entre X-Plane y microcontroladores externos

Por Germán Quijada

Profesor guía:  
Bernardo Hernández

12 de julio de 2023

## Concepto

- Actores en la comunicación

- Condiciones de diseño

## Establecimiento de protocolo

- Soluciones existentes

- Diseño preliminar

## Implementación

- Características

## Conclusión

En el simulador de vuelo del laboratorio de técnicas aeroespaciales, establecer un *protocolo de comunicación* entre el software de simulación *X-Plane* y *microcontroladores externos*.

# **Actores en la comunicación**

- ▶ Software de simulación de vuelo

- ▶ Software de simulación de vuelo
- ▶ Utilizado en entornos de entrenamiento certificados<sup>1</sup>

---

<sup>1</sup><https://x-plane.helpscoutdocs.com/article/31-faa-certification>

- ▶ Software de simulación de vuelo
- ▶ Utilizado en entornos de entrenamiento certificados<sup>1</sup>
- ▶ Herramienta ingenieril<sup>2</sup>

---

<sup>1</sup><https://x-plane.helpscoutdocs.com/article/31-faa-certification>

<sup>2</sup>[https://www.x-plane.com/desktop/meet\\_x-plane](https://www.x-plane.com/desktop/meet_x-plane)

- ▶ Software de simulación de vuelo
- ▶ Utilizado en entornos de entrenamiento certificados<sup>1</sup>
- ▶ Herramienta ingenieril<sup>2</sup>
- ▶ Funcionalidad agregada con plug-ins

---

<sup>1</sup><https://x-plane.helpscoutdocs.com/article/31-faa-certification>

<sup>2</sup>[https://www.x-plane.com/desktop/meet\\_x-plane](https://www.x-plane.com/desktop/meet_x-plane)



- ▶ Software de simulación de vuelo
- ▶ Utilizado en entornos de entrenamiento certificados<sup>1</sup>
- ▶ Herramienta ingenieril<sup>2</sup>
- ▶ Funcionalidad agregada con plug-ins
- ▶ Interfaz de comunicación UDP

---

<sup>1</sup><https://x-plane.helpscoutdocs.com/article/31-faa-certification>

<sup>2</sup>[https://www.x-plane.com/desktop/meet\\_x-plane](https://www.x-plane.com/desktop/meet_x-plane)



STEAM

[TIENDA](#)

[COMUNIDAD](#)

[ACERCA DE](#)

[SOPORTE](#)

[Instalar Steam](#)

[Iniciar sesión](#)

[Idioma](#)



[Tu tienda](#)

[Nuevo y destacable](#)

[Categorías](#)

[Tienda de puntos](#)

[Noticias](#)

[Laboratorios](#)



[Todos los juegos](#) > [Simuladores](#) > [X-Plane 12](#)

## X-Plane 12

[Punto de encuentro](#)



Ultra-Realistic Flight Simulation

RESEÑAS RECIENTES: **Variadas** (95)

RESEÑAS GENERALES: **Mayormente positivas** (1,401)

FECHA DE LANZAMIENTO: 17 DIC 2022

DESARROLLADOR: **Laminar Research**

EDITOR: **Laminar Research**

Etiquetas populares para este producto:

[Simulación](#)

[Vuelo](#)

[Simulador inmersivo](#)

[3D](#)



1

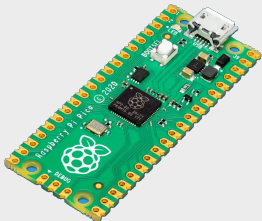
<sup>1</sup>[https://store.steampowered.com/app/2014780/XPlane\\_12/](https://store.steampowered.com/app/2014780/XPlane_12/)

- ▶ Pequeños dispositivos electrónicos integrados

- ▶ Pequeños dispositivos electrónicos integrados
- ▶ Especializados

- ▶ Pequeños dispositivos electrónicos integrados
- ▶ Especializados
- ▶ Interfaces estandarizadas ( $I^2C$ , Serial, UART)

- ▶ Pequeños dispositivos electrónicos integrados
- ▶ Especializados
- ▶ Interfaces estandarizadas ( $I^2C$ , Serial, UART)
- ▶ Bajo costo



Raspberry Pi Pico<sup>1</sup>



Arduino UNO<sup>2</sup>

<sup>1</sup><https://www.raspberrypi.com/products/raspberry-pi-pico/>

<sup>2</sup><https://store.arduino.cc/products/arduino-uno-rev3>

## Posibles aplicaciones



## Posibles aplicaciones

- ▶ Desarrollar sistemas de control

## Posibles aplicaciones

- ▶ Desarrollar sistemas de control
- ▶ Evaluar rendimiento de maniobras o en misiones

## Posibles aplicaciones

- ▶ Desarrollar sistemas de control
- ▶ Evaluar rendimiento de maniobras o en misiones
- ▶ Escribir código portable

El protocolo de comunicación establecido debe

El protocolo de comunicación establecido debe

- ▶ Permitir acceso general a las variables internas de simulación.

El protocolo de comunicación establecido debe

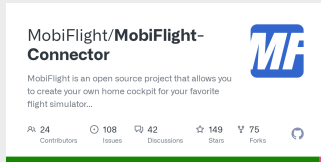
- ▶ Permitir acceso general a las variables internas de simulación.
- ▶ Funcionar en los microcontroladores más populares.

El protocolo de comunicación establecido debe

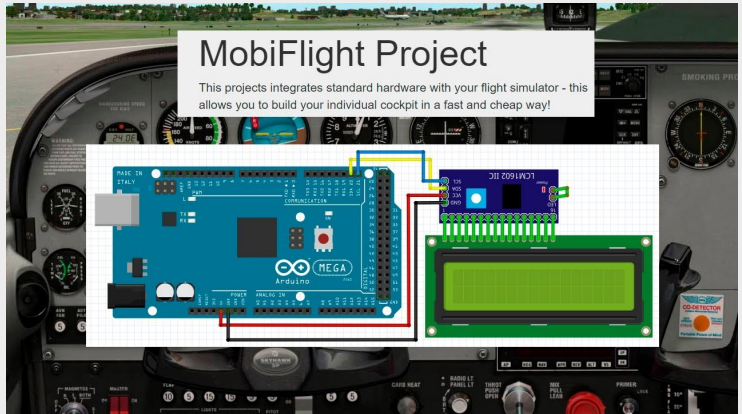
- ▶ Permitir acceso general a las variables internas de simulación.
- ▶ Funcionar en los microcontroladores más populares.
- ▶ Ser accesible para alumnos del departamento.

## **Soluciones existentes**





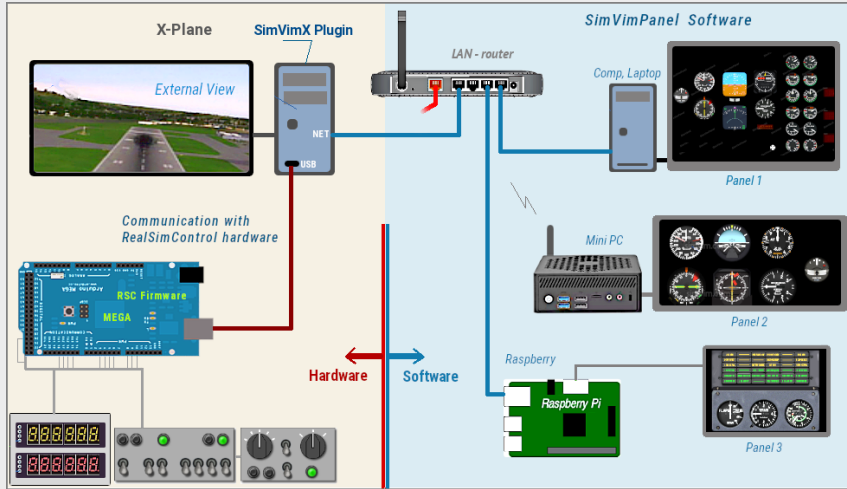
1



2

<sup>1</sup><https://github.com/MobiFlight/MobiFlight-Connector>

<sup>2</sup>Youtube - Mobiflight: Como usar un LCD Display / Using an LCD Display.



1

<sup>1</sup><https://simvim.com/>

The screenshot shows the SIMMARKET website. The top navigation bar is dark blue with the SIMMARKET logo on the left and links for Sales, Free, Vouchers, Reviews, Vendors (with a dropdown arrow), and Install App. A search bar is on the right. Below the navigation bar is a light blue bar with category links: MSFS (with a dropdown arrow), P3D (with a dropdown arrow), X-PLANE (with a dropdown arrow), FSX & LEGACY (with a dropdown arrow), TRAINS (with a dropdown arrow), MORE (with a dropdown arrow), and TRAINING (with a dropdown arrow). The main content area has a white header with the text 'FSUIPC7 FOR MSFS'. Below this is a section titled 'Description' in bold. The description text reads: 'FSUIPC7: Flight Simulator Universal Inter-Process Communication (for MSFS only)' followed by two paragraphs explaining the utility's function and its importance as an add-on for MSFS.

**FSUIPC7 FOR MSFS**

**Description**

**FSUIPC7: Flight Simulator Universal Inter-Process Communication (for MSFS only)**

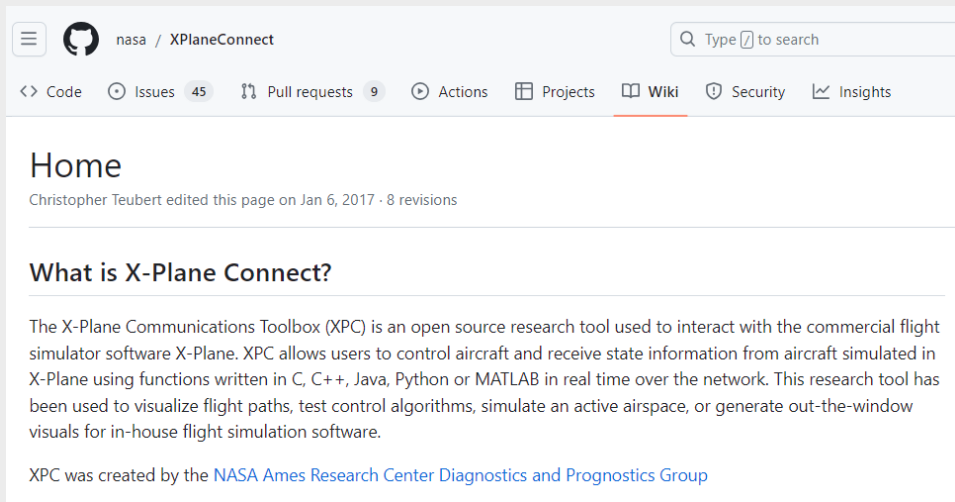
FSUIPC7 is a utility program for MSFS which provides an interface for other programs to read and write all sorts of pertinent data relating to the simulation, and in many ways even to control the actual process itself.

It is an essential ingredient in many add-on FS programs, and, in fact, may be installed on your system by any one or more of them already. For most such programs you do not need to actually purchase FSUIPC7.

1

---

<sup>1</sup><https://secure.simmarket.com/john-dowson-fsuipc7-for-msfs.phtml>



The screenshot shows the GitHub repository page for 'nasa / XPlaneConnect'. The repository has 45 issues and 9 pull requests. The 'Wiki' tab is selected, showing the 'Home' page. The page content includes a heading 'Home' with a note that Christopher Teubert edited the page on Jan 6, 2017, with 8 revisions. Below this is a section titled 'What is X-Plane Connect?' which describes the X-Plane Communications Toolbox (XPC) as an open source research tool used to interact with the commercial flight simulator software X-Plane. It mentions that XPC allows users to control aircraft and receive state information from aircraft simulated in X-Plane using functions written in C, C++, Java, Python or MATLAB in real time over the network. The text also states that XPC has been used to visualize flight paths, test control algorithms, simulate an active airspace, or generate out-the-window visuals for in-house flight simulation software. At the bottom, it says 'XPC was created by the NASA Ames Research Center Diagnostics and Prognostics Group'.

Home

Christopher Teubert edited this page on Jan 6, 2017 · 8 revisions

## What is X-Plane Connect?

The X-Plane Communications Toolbox (XPC) is an open source research tool used to interact with the commercial flight simulator software X-Plane. XPC allows users to control aircraft and receive state information from aircraft simulated in X-Plane using functions written in C, C++, Java, Python or MATLAB in real time over the network. This research tool has been used to visualize flight paths, test control algorithms, simulate an active airspace, or generate out-the-window visuals for in-house flight simulation software.

XPC was created by the [NASA Ames Research Center Diagnostics and Prognostics Group](#)

1

<sup>1</sup><https://github.com/nasa/XPlaneConnect/wiki>

Observaciones

## Observaciones

- ▶ MobiFlight y SimVim están diseñados para crear paneles de instrumentos con microcontroladores.

## Observaciones

- ▶ MobiFlight y SimVim están diseñados para crear paneles de instrumentos con microcontroladores.
- ▶ FSUIPC y X-Plane Connect son de uso general, pero su documentación no contempla microcontroladores.

## Observaciones

- ▶ MobiFlight y SimVim están diseñados para crear paneles de instrumentos con microcontroladores.
- ▶ FSUIPC y X-Plane Connect son de uso general, pero su documentación no contempla microcontroladores.

## Conclusión

- ▶ Crear una nueva solución que cumpla las condiciones de diseño.



# Diseño preliminar

# Características de nueva solución

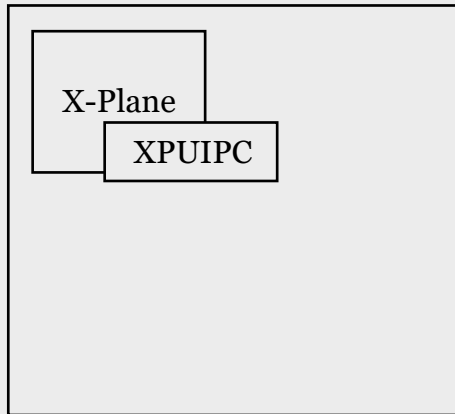


- ▶ Ser un programa intermediario entre el simulador y el microcontrolador.

- ▶ Ser un programa intermediario entre el simulador y el microcontrolador.
- ▶ Interactuar con simulador por medio de FSUIPC.

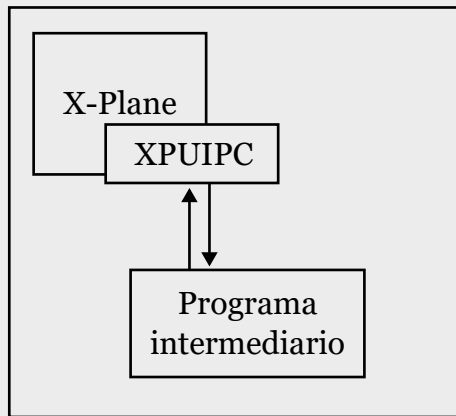
- ▶ Ser un programa intermediario entre el simulador y el microcontrolador.
- ▶ Interactuar con simulador por medio de FSUIPC.
- ▶ Extender FSUIPC con canales accesibles por microcontroladores.

# Diagrama



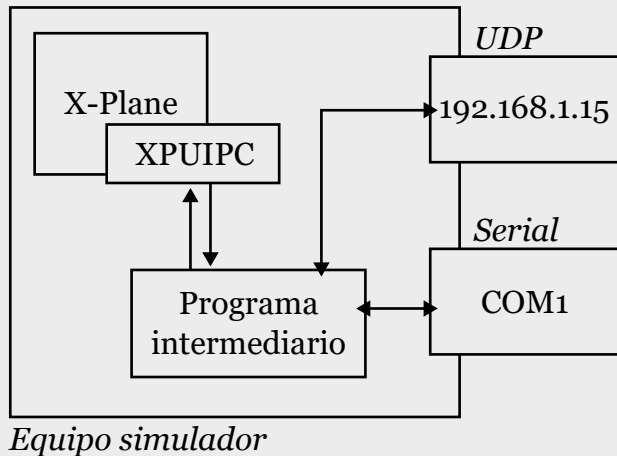
*Equipo simulador*

# Diagrama



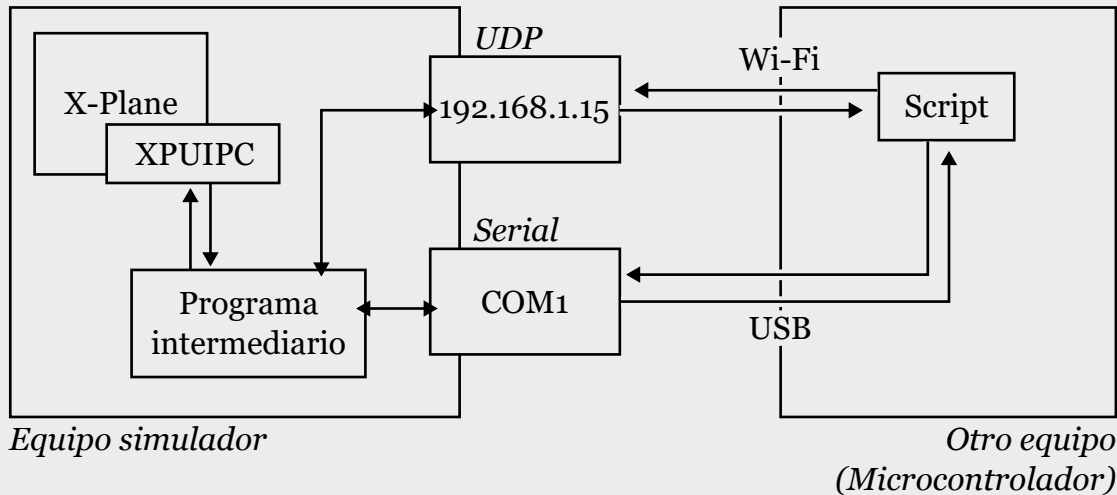
*Equipo simulador*

# Diagrama





# Diagrama



# Implementación



## Características

## Características

- ▶ Fácil instalación.

## Características

- ▶ Fácil instalación.
- ▶ Abre canales Serial y UDP por red local.

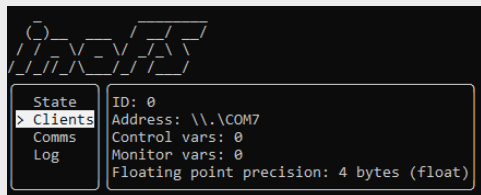
## Características

- ▶ Fácil instalación.
- ▶ Abre canales Serial y UDP por red local.
- ▶ Permite el acceso a variables internas del simulador por medio de FSUIPC.

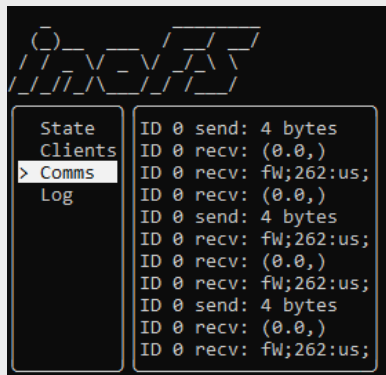
## Características

- ▶ Fácil instalación.
- ▶ Abre canales Serial y UDP por red local.
- ▶ Permite el acceso a variables internas del simulador por medio de FSUIPC.
- ▶ Interfaz gráfica sencilla para ayudar a identificar problemas.





Pestaña Clientes



Pestaña Comunicaciones



- ▶ Mensajes estructurados
  - ▶ Lectura
  - ▶ Escritura
  - ▶ Monitorear (Lectura continua)
  - ▶ Controlar (Escritura continua)

- ▶ Mensajes estructurados
  - ▶ Lectura
  - ▶ Escritura
  - ▶ Monitorear (Lectura continua)
  - ▶ Controlar (Escritura continua)
- ▶ Comandos enviados como cadenas de caracteres
- ▶ Datos enviados como bytes crudos

# Protocolo - Comando Escribir

```
1  # Header de mensaje
2  header = bytes("inofs", "ascii")
3  # Comando con variables a sobrecribir
4  cmd = bytes("fW;310A:c;089A:s;", "ascii")
5  # Variables
6  input = 8
7  throttle = (2/10) * 16384
8  data = bytes(struct.pack("<ff", input, throttle), "ascii")
9  # Evaluar largo del mensaje
10 size = struct.pack("<i", len(cmd + data))
11 # Empacar y enviar
12 msg = header + size + cmd + data
13 sys.stdout.buffer.write(msg)
14
```

Mensaje enviado por Serial

“header” + “bytes del mensaje” + “comando” + “datos”

- ▶ Lista de variables de FSUIPC  
<https://www.projectmagenta.com/all-fsuihc-offsets/>
- ▶ Documentación inoFS en repositorio  
<https://github.com/qgerman2/inoFS>

# Conclusión

# Conclusión





- ▶ Se establece el protocolo de comunicación de acuerdo a las condiciones de diseño.

- ▶ Se establece el protocolo de comunicación de acuerdo a las condiciones de diseño.
- ▶ El programa es funcional, pero necesita más desarrollo.

- ▶ Se establece el protocolo de comunicación de acuerdo a las condiciones de diseño.
- ▶ El programa es funcional, pero necesita más desarrollo.
- ▶ Con debida documentación, el programa podría ser utilizado por la comunidad.

**Gracias por su atención**