

CONEXIÓN Y COMUNICACIÓN DEL AUTOPILOTO PIXHAWK CON UNA RASPBERRY PI MODELO 4B USANDO MAVLINK

Alexis Fernando Cruz Baños `acruzba.2100@alumno.ipn.mx`

Dr. Juan Carlos Herrera Lozada `jlozada@ipn.mx`

M. en C. Miguel Hernández Bolaños `mbolanos@ipn.mx`

Centro de Innovación y Desarrollo Tecnológico en Cómputo
Instituto Politécnico Nacional

Resumen

En el presente artículo se explica de manera concreta cómo se realiza la conexión entre un autopiloto PixHawk versión 2.4.8 con una computadora de bajo costo Raspberry Pi versión 4B, con el propósito de lograr una arquitectura sencilla que pueda implementarse en cualquier vehículo de navegación (dron, rover, carro omnidireccional, hovercrafts, entre otros). Se utiliza comunicación serial a través de los puertos de telemetría del autopiloto o el puerto USB y el protocolo de comunicación MAVLink. Es necesario conocer la arquitectura de ambos dispositivos y tener conocimientos de configuración y puesta en marcha de una Raspberry Pi, así como la previa configuración del autopiloto usando un software de telemetría como Mission Planner, lo cual se muestra de manera breve en este documento. Esta arquitectura en conjunto con un Vehículo Aéreo no Tripulado (VANT) puede ser usada en aplicaciones que requieran comunicación de baja latencia y alto procesamiento como: procesamiento de imágenes, visión computacional, seguimiento de trayectorias, modelado 3D de terrenos, entre otros.

Palabras Clave: MAVLink, PixHawk, Raspberry Pi, VANT, comunicación serial.

Abstract

This article explains in a straightforward manner how to connect a PixHawk autopilot version 2.4.8 to a low-cost Raspberry Pi version 4B computer, with the purpose of achieving a simple architecture that can be implemented in any navigation vehicle (drone, rover, omnidirectional car, hovercrafts, among others). Serial communication is used through the telemetry ports of the autopilot or the USB port, using the MAVLink communication protocol. This architecture, together with an Unmanned Aerial Vehicle (UAV), can be used in applications that require low-latency communication and high processing power, such as image processing, computer vision, path tracking, 3D terrain modeling, among others.

Keywords: MAVLink, PixHawk, Raspberry Pi, UAV, serial communication.

I. Introducción

Desde el siglo pasado, la necesidad de utilizar vehículos de navegación no tripulados ha ido incrementando con el paso del tiempo; de ahí surge la necesidad del uso de controladores de navegación que cumplen con funciones como proveer algoritmos de guía o navegación y, en algunos casos dependiendo la aplicación requerida, establecer una comunicación con estaciones fijas de control. Este módulo que integra las funciones descritas es conocido como *autopiloto*. Los autopilotos son ampliamente usados en diferentes vehículos no tripulados; sin embargo, tienen limitaciones de memoria y procesamiento. Para aplicaciones que requieren un alto procesamiento pero al mismo tiempo una respuesta de comunicación rápida, un autopiloto en conjunto con un sistema embebido o una computadora de compañía permite crear un sistema económico, eficiente y con respuesta de baja latencia para aplicaciones más complejas.

El autopiloto PixHawk es un dispositivo desarrollado por PX4 de bajo costo y con características competitivas que lo hacen un gran dispositivo en términos de costo-beneficio frente a otros autopilotos. Aunque mediante el uso de un software de telemetría permite realizar aplicaciones autónomas en los VANT, no

tiene la capacidad física de realizar aplicaciones que requieran un alto procesamiento o el uso de sensores externos. Es por eso que, utilizando una computadora portátil como la Raspberry Pi, se puede utilizar el autopiloto como central de comunicación, dejando como unidad central de procesamiento a esta tarjeta *single on chip* (RPI) [?].

II. Configuración de la Raspberry Pi 4B

Antes de comenzar con la conexión al autopiloto es necesario configurar la RPi 4B para que la comunicación serial sea llevada a cabo con éxito. Los pasos son los siguientes:

1. **Instalar el sistema operativo.** El más recomendable es Raspi OS; en la página de Raspberry se encuentran las instrucciones detalladas de instalación [?]. Como recomendación, para la configuración inicial es necesario tener un monitor extra; posteriormente se puede acceder a la RPi vía protocolo SSH.
2. **Configurar SSH.** Una vez finalizada la configuración inicial, es necesario configurar la RPi para comunicación remota mediante SSH, que permite acceder de manera remota a la RPi mediante la terminal de comandos de Raspi OS. En [?] se explica de manera detallada la configuración de la RPi para usarla mediante SSH, así como la instalación del software PuTTY en Windows. En la Figura 1 se muestra el acceso remoto a la RPi desde una computadora con Windows.

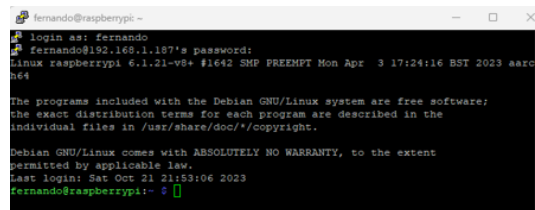


Figura 1 Conexión con RPi vía SSH.

3. **Configurar el puerto serial (UART).** Dentro de la terminal abierta por SSH se introduce el comando `sudo raspi-config` para abrir el RasPi Configuration Utility que se muestra en la Figura 2.

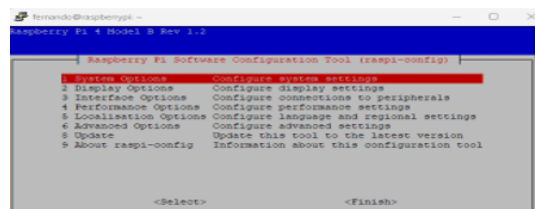


Figura 2 RasPi Configuration Utility.

Se sigue la ruta Interface Options / Serial y posteriormente aparecerán 2 ventanas emergentes con las siguientes opciones:

- *Would you like a login Shell to be accessible over serial?* Seleccionar **No**.
- *Would you like the serial port hardware to be enabled?* Seleccionar **Yes**.
- Finalmente se reinicia la RPi y ya se tendrá activo el puerto serial.

4. **Instalación de MAVProxy.** MAVProxy es una herramienta de código abierto que actúa como intermediario entre MAVLink y una computadora o estación terrestre. Para instalar MAVProxy en la RPi basta con ejecutar los siguientes comandos desde la terminal:

```
sudo apt-get install python3-dev python3-opencv python3-wxgtk4.0 \
python3-pip python3-matplotlib python3-lxml python3-pygame
pip3 install PyYAML mavproxy --user
```

```
echo "export PATH=$PATH:$HOME/.local/bin" >> ~/.bashrc
```

Una vez ejecutados los comandos anteriores, se puede verificar la instalación con el comando `mavproxy.py --version`; si la instalación ha sido exitosa, en la terminal aparecerá la versión instalada de MAVProxy como se observa en la Figura 3.

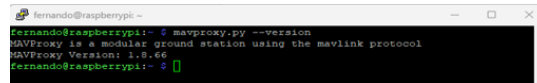


Figura 3 Instalación de MAVProxy en la RPI.

- 5. Modificar el firmware boot.** Se hace una modificación en el firmware boot para evitar mala conexión entre los puertos seriales de la RPi. Para esto se utiliza el comando `sudo nano /boot/config.txt`. En la última línea del archivo se agregan las siguientes 2 líneas:

```
enable_uart=1
dtoverlay=disable-bt
```

Finalmente, con las combinaciones de teclas `Ctrl+X`, `Ctrl+Y` y `Enter`, se guarda el archivo ya modificado.

III. Configuración del autopiloto PixHawk

Para iniciar la configuración del PixHawk es necesario contar con una PC e instalar algún software de telemetría; en este caso se utilizó Mission Planner debido a su alta compatibilidad con la arquitectura PX4.

- Una vez instalado Mission Planner se procede a conectar el autopiloto mediante su puerto USB a la computadora y se instala el firmware requerido, en este caso el firmware de Quad Copter. En [?] se encuentra información detallada de la instalación del firmware y la puesta en marcha del PixHawk por primera vez.
- Cuando se haya verificado que el autopiloto puede establecer comunicación con Mission Planner de manera correcta, se selecciona la pestaña `CONFIG` en la parte superior del software y en la ventana izquierda se selecciona `Full Parameter List`, ambos marcados para referencia en la Figura 4.

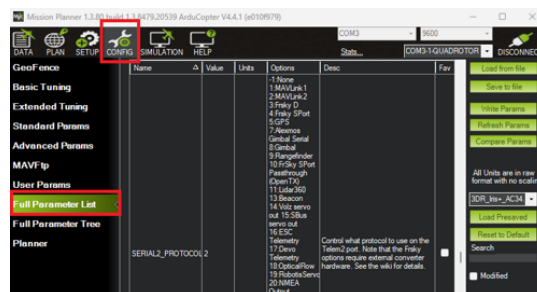


Figura 4 Configuración de Mission Planner.

En la ventana central se encuentra la lista de parámetros dentro de los cuales se modificarán los siguientes con los valores indicados:

- `SERIAL2_PROTOCOL = 2` para habilitar MAVLink en el puerto Telem2 del PixHawk.
- `SERIAL2_BAUD = 57` de esta manera el autopiloto se comunicará con la RPi a 57,600 baudios.

Se procede a reiniciar el autopiloto; de esta manera quedará configurado y listo para la comunicación con la RPi.

IV. Conexión PixHawk -- Raspberry

La conexión física entre la RPi y el PixHawk se puede llevar a cabo de 2 maneras. La primera es mediante el uso del puerto de telemetría del PixHawk (TELEM2) a los pines de Rx y Tx de la RPi, como se observa en la Figura 5. Para que esta conexión funcione de manera correcta es necesario contar con un conector Dupont pre armado, pues si se utiliza alguna adaptación o un cable no completamente compatible, puede provocar que la transmisión de datos no sea correcta.

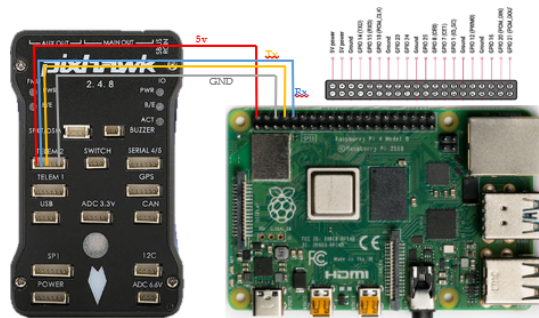


Figura 5 Conexión RPi - PixHawk mediante TELEM2.

Otra forma de establecer la conexión es mediante el puerto micro USB del PixHawk con un puerto USB de la RPi; esta es la manera más sencilla y segura, puesto que no hay interferencia en la transmisión y los GND de ambos dispositivos se encuentran debidamente conectados.

Cuando la conexión física está correcta, se hace conexión SSH con la RPi y se verifica que la RPi esté reconociendo la conexión con el PixHawk, ingresando el siguiente comando en la terminal:

```
ls /dev/ttyACMO
```

Nota: El comando anterior es para la conexión mediante el puerto USB. Si la conexión se hace mediante el puerto TELEM2, es necesario sustituir ACMO por AMAO.

Si la RPi reconoce que la PixHawk está conectada, en la terminal se verá el puerto en color amarillo como se muestra en la Figura 6.

```
fernando@raspberrypi:~$ ls /dev/ttyACMO
/dev/ttyACMO
```

Figura 6 RPi reconociendo una conexión serial en el puerto ACMO.

Ya con la conexión verificada, solo queda inicializar MAVProxy en la RPi para comenzar la comunicación entre ambos dispositivos. Para inicializar la comunicación es necesario poner el siguiente comando en la terminal:

```
mavproxy.py --master=/dev/ttyACMO --baudrate 57600
```

Nota: Si se está utilizando la conexión por el puerto TELEM2, es necesario sustituir ttyACMO por ttyAMA0 en el comando anterior.

Si la conexión ha resultado exitosa, en la terminal se indica el reconocimiento de un vehículo y los parámetros preconfigurados en Mission Planner, como se observa en la Figura 7.

```
fernando@raspberrypi:~$ sudo mavproxy.py --master=/dev/ttyACM0 --baudrate 57600
Connect /dev/ttyACM0 source_system=255
Log Directory:
Telemetry log: mav.tlog
Waiting for heartbeat from /dev/ttyACM0
MAV: Detected vehicle 1:1 on link 0
online system 1
STABILIZE Mode STABILIZE
fence present
AP: ArduCopter V4.4.1 (e010f979)
AP: ChibiOS: 17a50e3a
AP: Pixhawk1-LM 0030003E 34395102 30323935
AP: RCOut: PWM1-14
AP: RMO: fast sampling enabled 8.0kHz/1.0kHz
AP: Frame: QUAD/X
Received 870 parameters (ftp)
Saved 870 parameters to mav.parm
Servo volt 4.3
AP: 2M flash - use Pixhawk1 firmware
```

Figura 7 Conexión RPi - PixHawk mediante MAVProxy.

Una vez lograda la conexión de manera exitosa, se pueden comenzar a implementar diferentes métodos de control o arquitecturas. Entre las que más destacan se encuentra DroneKit, que es un proyecto de código abierto que, en conjunto con el protocolo MAVLink, permite crear múltiples aplicaciones para los VANT utilizando lenguaje Python. También se puede utilizar sistemas de visión artificial como OpenCV e incluso arquitecturas más sofisticadas como TensorFlow o PyTorch para aplicaciones que requieren el uso de *Machine Learning* o *Deep Learning*.

V. Conclusiones

En este trabajo se ha explorado la conexión entre una Raspberry Pi 4 y un autopiloto PixHawk. Esta integración representa la creación de una arquitectura de bajo costo y versátil que puede ser implementada en una amplia gama de vehículos no tripulados. La capacidad resultante de esta arquitectura permite la realización de aplicaciones que demandan comunicación de baja latencia y un alto poder de procesamiento, como la visión computacional, el seguimiento de trayectorias y el modelado 3D de terrenos, entre otras. El uso de la comunicación serial a través de los diversos puertos de estos dispositivos ha demostrado ser una solución efectiva para lograr esta integración, abriendo así la puerta a una variedad de aplicaciones en el campo de los vehículos no tripulados. Finalmente, esta arquitectura puede ser un punto de partida para que, en conjunto con tecnologías emergentes, se puedan realizar aplicaciones cada vez más complejas.

Referencias

- Cruz, A., Vazquez, R., Herrera, J., Hernández, M., & Herrera, S. (1 de julio de 2023). Un sistema de navegación de bajo costo para vehículos no tripulados: el autopiloto PixHawk. *Boletín UPIITA*, 18(97). <https://www.boletin.upiita.ipn.mx/index.php/ciencia/1045-cyt-numero-97/2176-un-sistema-de-navegacion-de-bajo-costo-para-ve>
- Raspberry Pi. (s.f.). Instalación de sistema operativo con Pi Imager. <https://raspberrypi.cl/2021/05/13/instalacion-de-sistema-operativo-con-pi-imager/>
- IONOS. (24 de septiembre de 2020). Acceso remoto a una Raspberry Pi con SSH: ¿Cómo habilitarlo? <https://www.ionos.mx/digitalguide/servidores/configuracion/activar-ssh-en-raspberry-pi/>
- Ardupilot DevTeam. (2023). Mission Planner Overview. *Ardupilot.org*. <https://ardupilot.org/planner/>

Cómo citar este artículo en APA: Cruz Baños, A. F., Herrera Lozada, J. C., & Hernández Bolaños, M. (1 de enero de 2024). Conexión y comunicación del autopiloto PixHawk con una Raspberry Pi Modelo 4B usando MAVLink. *Boletín UPIITA*, 18(100).