

G6 CoPilot NanoCore

Configuration Guide

Multirotor

October 2025



1 Features of the G6 Flight Controller

The G6 Flight Controller represents a state-of-the-art platform designed to empower multi-copter drones with exceptional stability, versatility, and computational capabilities. Below is an overview of the key features that make the G6 a powerful tool for drone enthusiasts and professionals alike:

Advanced Stabilization and Control. The G6 is engineered to stabilize multi-copter drones with four motors, offering robust attitude and altitude control. This capability is driven by an on-board inertial measurement unit (IMU) and a precision barometer, ensuring precise and responsive flight dynamics in a wide range of environments.

Embedded Computing Platform. The G6 integrates a high-performance computing platform that combines a dedicated Flight Controller core with four additional application cores. This architecture supports advanced flight management while providing a flexible framework for running custom high-level applications.

Seamless Telemetry and Mission Control. The G6 enables dynamic interaction between its Flight Controller core and application cores. Telemetry and control messages are efficiently shared, allowing high-level applications to retrieve real-time flight status and actively influence mission execution.

Comprehensive Sensor and Actuator Integration. The Flight Controller core handles communication with critical on-board sensors such as the IMU, pressure sensor, electronic speed controllers (ESCs), compass, and optional external magnetic and GNSS sensors. These interfaces ensure smooth operation and precise control under various flight conditions.

Cutting-Edge Vision and AI Capabilities. Equipped with a MIPI CSI-2 camera port and on-board VPU, the G6 supports hardware-accelerated H.264 video compression up to 1080p. Combined with its hardware-accelerated machine learning capabilities, the G6 is ideal for advanced vision-based applications such as object recognition and tracking.

Improved Configurability with Integrated WiFi. The G6 Flight Controller features a built-in WiFi module, enabling convenient short-range telemetry and effortless configuration.

Extensive Connectivity and Expandability. The G6 provides a variety of interfaces for enhanced connectivity and expandability:

- USB Host and USB OTG interfaces for external devices.
- Dedicated GPIO pins for integrating user-provided sensors or actuators.
- Four ESC outputs supporting the DSHOT protocol for high-speed motor control.
- Optional interfaces, including:
 - I2C for external magnetic sensors.
 - UART for external GNSS sensors.
 - CAN for additional communication needs.

With its powerful combination of stabilization, computational capability, and extensibility, the G6 Flight Controller is a versatile solution for multi-copter drone enthusiasts and developers aiming to push the boundaries of aerial technology.

2 Connectivity

Figure 1 provides an overview of the various connectors available on the G6 Flight Controller. These connectors are **JST-SUR type** with a 0.8 mm pitch. As **insulation displacement connectors (IDC)**, they eliminate the need for crimping or soldering. To attach a wire, simply insert a thin **AWG-32 wire** into the bottom of the plug and press it firmly into the recess.

The pinout configurations for the G6 Flight Controller's external ports are shown in Figures 3 through 10. Refer to Fig 2 for the mounting points and intended flight direction.

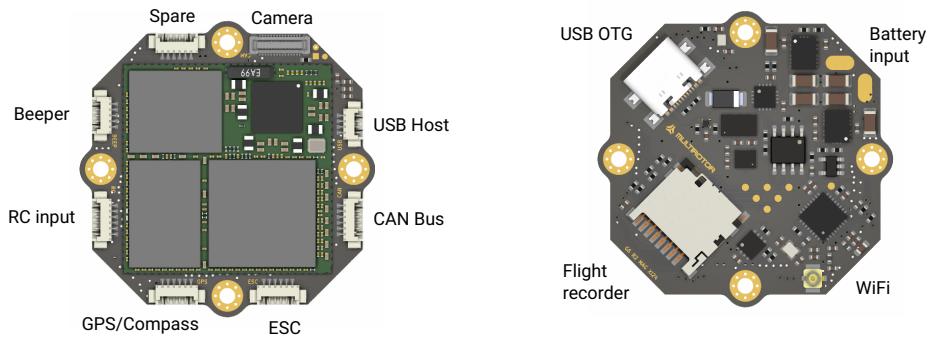


Figure 1: Top (left) and bottom (right) view of the Flight Controller connections

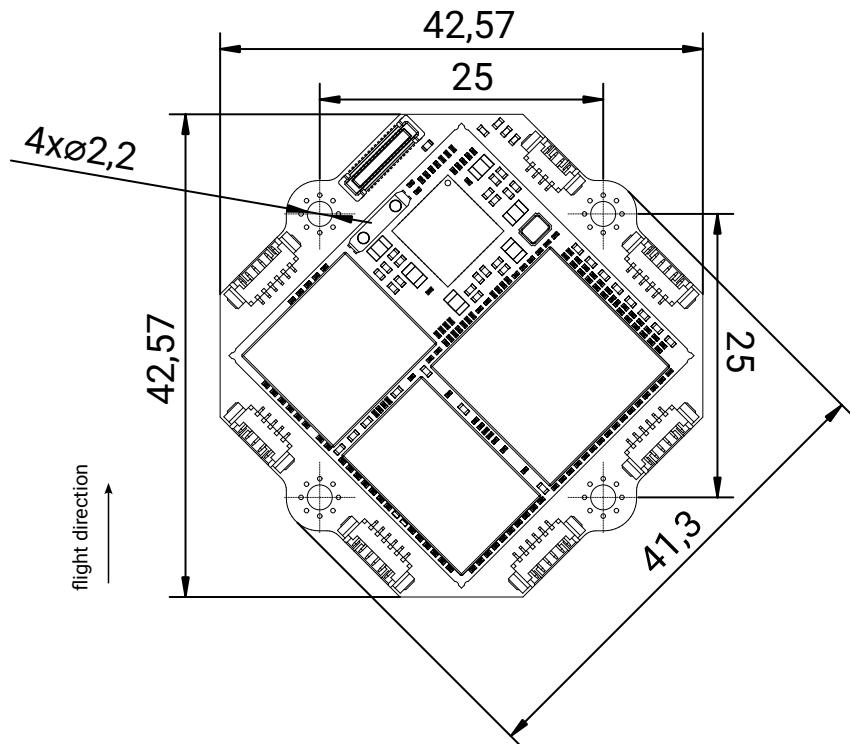


Figure 2: Mounting options, dimensions and intended flight direction.

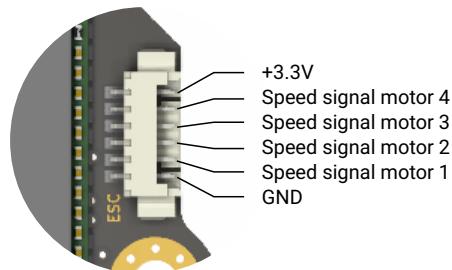


Figure 3: Pinout of the ESC port on the G6 Flight Controller.

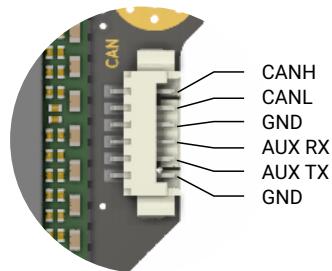


Figure 4: Pinout of the CAN port on the G6 Flight Controller.

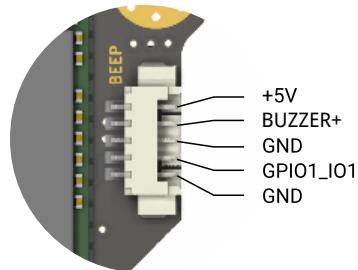


Figure 5: Pinout of the BUZZ port on the G6 Flight Controller.

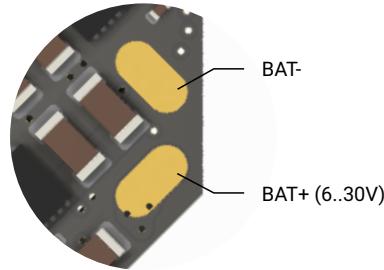


Figure 6: Pinout of the Battery connection to the G6 Flight Controller. Provide battery voltage between 6 and 30 V

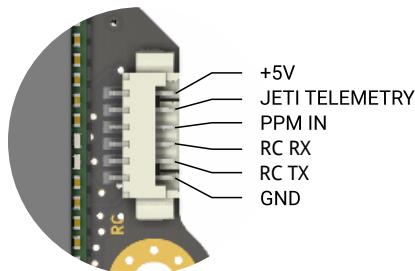


Figure 7: Pinout of the RC port on the G6 Flight Controller.

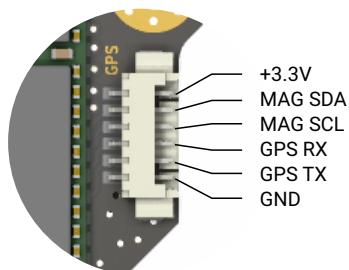


Figure 8: Pinout of the GPS port on the G6 Flight Controller.

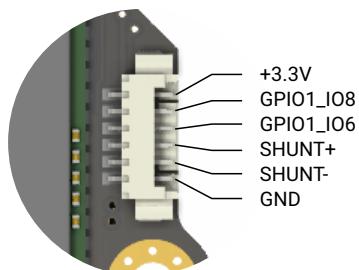


Figure 9: Pinout of the SPARE port on the G6 Flight Controller.

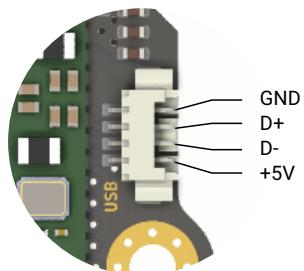


Figure 10: Pinout of the USB Host interface on the G6 Flight Controller.

3 Configuration

The G6 flight controller operates as a Wi-Fi access point. Because of its compact antenna, the range is limited but sufficient for configuration and short-range telemetry. Clients can connect to the network with SSID MAVLink-Bridge and password `mavlink123`.

Once connected, a MAVLink stream is available at `UDP://10.0.0.1:14550`. Ensure that only one client accesses the stream at a time to avoid communication conflicts. When connected via USB during startup, the G6 also enumerates as a virtual serial port, which can be used for a MAVLink connection.

The G6 flight controller is compatible with QGroundControl (QGC v5.0.6 or later), enabling most configuration tasks to be completed through the dedicated QGC user interfaces. Parameters can also be adjusted manually using QGC's parameter table if required. For low-level interaction, a PX4 shell is available via *Analyze Tools* → *MAVLink Console*.

- Wi-Fi access point: SSID MAVLink-Bridge, password `mavlink123`.
- MAVLink stream: `UDP://10.0.0.1:14550` or virtual serial port.
- QGroundControl: version 5.0.6 or later recommended.

3.1 Blackbox

An SD card is required for proper operation, as it is used to store parameters, waypoint data, and flight logs. The flight controller expects a VFAT-formatted microSD card. When the G6 is powered on for the first time or when parameters are saved, the necessary system files are automatically created on the card.

- Use a VFAT-formatted microSD card for non-volatile storage.

3.2 Radio Setup

The G6 currently supports SBUS, Spektrum DSM, and CSRF radio protocols. More protocols will be supported in future firmware releases. The radio protocol is detected automatically by the system and does not require manual configuration. Channel assignment is performed within QGroundControl under *Vehicle Setup* → *Radio*, where users can verify and calibrate their transmitter inputs.

- Supported protocols: SBUS, Spektrum DSM, CRSF with automatic protocol detection.
- Channel assignment and calibration: QGC *Vehicle Configuration* → *Radio*.

3.3 Servo and ESC Setup

Servo and ESC output functions are assigned through the parameters `MULTIG6IO_FUNC1` to `MULTIG6IO_FUNC4`, available in QGC's parameter table. Each parameter corresponds to a specific output function (e.g., 101 for Motor1 through 104 for Motor4). The PWM frequency is set using the parameter `MULTIG6IO_PWM` in the range of 20–400 Hz. Setting this parameter to `-1` enables DSHOT mode. When DSHOT is used, the bitrate is configured via `MULTIG6IO_DSHOT` in the range of 150000–600000 bit/s. The PWM values for disarmed, minimum, and maximum output states are defined by the parameters `MULTIG6IO_DIS`, `MULTIG6IO_MIN`, and `MULTIG6IO_MAX`, respectively.

- Output function mapping: `MULTIG6IO_FUNC1` to `MULTIG6IO_FUNC4`.
- PWM frequency: set via `MULTIG6IO_PWM` (20–400 Hz); use `-1` for DSHOT.
- DSHOT bitrate: `MULTIG6IO_DSHOT` (150000–600000 bit/s).
- PWM limits:
 - `MULTIG6IO_DIS` — PWM when disarmed.
 - `MULTIG6IO_MIN` — minimum PWM while armed.
 - `MULTIG6IO_MAX` — maximum PWM while armed.

3.4 Current and Voltage Sensing

The G6 includes built-in voltage and current sensing circuitry. Voltage is measured at the `BAT+` supply with respect to system ground (`BAT-`). Current can be measured either as the voltage drop across an external shunt resistor (connected to `SHUNT+` and `SHUNT-`) or as a ratiometric voltage referenced to system ground (connected to `SHUNT+`). The specific configuration is factory-defined and cannot be changed without hardware modification. Refer to Fig. 11 to identify which configuration is implemented on your board.

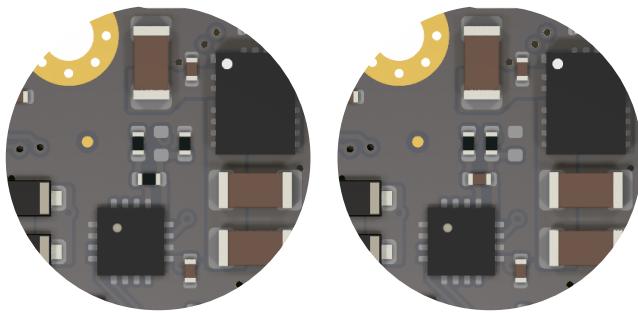


Figure 11: Current sensing options of the G6 Flight Controller (rear view). Left: ratiometric current sensing; right: external shunt sensing configuration.

In ratiometric mode, the input voltage must not exceed 3.3 V. When using this mode (for example, when interfacing with a 4-in-1 ESC), the parameter `BAT1_A_PER_V` should be set to $43.5 \times G$, where G represents the current-sense gain of the ESC in A/V. In external shunt mode, this parameter should instead be set to $1/R_s$, where R_s is the resistance of the shunt resistor in ohms.

The G6 does not monitor voltage rails. Hence, set the parameter `CBRK_SUPPLY_CHK` to 894281 in order to disable the pre-arm system power check.

- Current sensing modes (fixed at production, ensure using the correct one):
 - Ratiometric input (SHUNT+ to GND, max 3.3 V).
 - External shunt resistor (SHUNT+ / SHUNT-).
- Parameter settings:
 - Ratiometric: $BAT1_A_PER_V = 43.5 \times G$.
 - External shunt: $BAT1_A_PER_V = 1/R_s$.
- Set `CBRK_SUPPLY_CHK` to 894281 to disable system power pre-arm check.

3.5 External Magnetic and GPS Sensors

An external GPS receiver using the u-blox protocol can be connected to the GPS port. Refer to Fig. 8 for detailed pin assignments. In the current firmware revision, only the internal magnetometer is supported. Future firmware releases will add support for external magnetic sensors connected via the MAG SCL and MAG SDA lines.

4 Onboard Computer

A debug console can be accessed through the AUX UART port (see Fig. 4), providing direct command-line access to the onboard Linux system. Configure the port to 115200 Baud, 8 data bits, no parity, 1 stop bit. The OS employs a read-only file system. If changes have to be made permanent, please remount the root filesystem using the command `mount -o remount,rw /`.

4.1 Transferring Files

Files can be transferred between a host computer and the flight controller via the USB OTG interface. Prior to starting the Linux operating system, the USB interface must be set to mass-storage mode using a terminal emulator such as `minicom` or `picocom` connected to the AUX UART port (see Fig. 4). The UART connection should be configured for 115200 Baud, 8 data bits, no parity, and 1 stop bit.

Connect the USB OTG port to the host computer. The flight controller is powered from USB but can also be powered from battery, if the host computer is not capable of supplying enough current. When the bootloader starts, press any key

to interrupt the boot process before Linux loads. Enter the command `ums 0 mmc 0:2` to enable USB mass-storage mode. The G6's root partition will then appear on the host computer as a removable storage device, allowing direct file transfer. Ensure the partition is properly unmounted before rebooting.

- Connection: AUX UART port (115200 Baud, 8N1).
- Enable USB mass-storage mode in u-boot: `ums 0 mmc 0:2`.

4.2 Setting Up a Development Environment

A development environment for the G6 can be prepared using a clean Ubuntu 22.04 Docker image. Begin by installing the required packages:

```
apt-get update && apt-get install -y sudo curl vim git \
    python-is-python3 build-essential chrpath cpio \
    debianutils diffstat file gawk gcc git iputils-ping \
    libacl1 liblz4-tool locales python3 python3-git \
    python3-jinja2 python3-pexpect python3-pip python3-subunit \
    socat texinfo unzip wget xz-utils zstd
```

Next, install the Google repo tool:

```
curl https://storage.googleapis.com/git-repo-downloads/repo > \
    /usr/bin/repo && chmod a+x /usr/bin/repo
```

Follow the repository setup instructions (branch `scarthgap`) provided at:

```
https://github.com/nxp-imx/imx-manifest/blob/imx-linux-scarthgap/README.md
```

To build the image, run the following commands:

```
MACHINE=imx8mpevk DISTRO=fsl-imx-xwayland source ./imx-setup-release.sh \
    -b <build-dir>
    bitbake -c populate_sdk core-image-minimal
```

Execute the generated installer from `<build-dir>/tmp/deploy/sdk/`, then source the environment setup script:

```
source <path-to-sdk>/environment-setup-armv8a-poky-linux
```

After these steps, the development environment is ready to build binaries compatible with the G6 Linux console.

- Base system: Ubuntu 22.04 (Docker recommended).
- Repository: NXP's i.MX Yocto branch `scarthgap`.
- Build target: `MACHINE=imx8mpevk, DISTRO=fsl-imx-xwayland`.
- SDK build command: `bitbake -c populate_sdk core-image-minimal`.
- Environment setup: `source <path-to-sdk>/environment-setup-armv8a-poky-linux`.

4.3 CAN Interface

The CAN interface can be accessed by any Linux user-space program running on the G6. To configure the CAN bus bitrate, use the following command on the Linux console:

```
ip link set can0 type can bitrate <rate_in_bit/s>
```

Once configured, the interface can be enabled with:

```
ip link set can0 up
```

To verify functionality, test data can be transmitted using:

```
cansend can0 123#DEADBEEF
```

- Bitrate setup: `ip link set can0 type can bitrate <bit/s>`.
- Enable interface: `ip link set can0 up`.
- Test command: `cansend can0 123#DEADBEEF`.