DroneCAN M9N/M8N

Overview

Overview

> □□ _[Image — to be added]_ > □□ _[Image — to be added]_

Descriptions□

The Holybro DroneCAN GPS has an UBLOX M8N or M9N module, BMM150 compass, tri-colored LED indicator. It has adopted the DroneCAN protocol for communication. It is better in dealing with electromagnetic interference compare to serial connection, making it more reliable. It does not occupy any serial port of the flight controller, and different CAN devices can be connected to the same CAN bus via a CAN splitter board.

Features and Specifications:

	DroneCAN M8N & M9N
GNSS Receiver	Ublox NEO M8N
Number of Concurrent GNSS	M8N - Up to 3 GNSS M9N - Up to 4 GNSS
Processor	STM32G4 (170MHz, 512K FLASH)
Compass	BMM150 or IST8310
Frequency Band	GPS: L1C/A GLONASS: L10F Beidou: B1I Galileo: E1B/C
GNSS Augmentation System	SBAS: WAAS, EGNOS, MSAS, QZSS
Navigation Update	5Hz Default(10Hz MAX)
Navigation sensitivity	–167 dBm
Cold starts	~ 26s
Accuracy	2.5m
Speed Accuracy	0.05 m/s
Max # of Satellites	22+
Default CAN BUS data rate	1MHz
Communication Protocol	DroneCAN @ 1 Mbit/s

	DroneCAN M8N & M9N
Firmware Support	PX4, Ardupilot
Port Type	GHR-04V-S
Antenna	25 x 25 x 4 mm ceramic patch antenna
Voltage	4.7-5.2V
Power consumption	Less than 200mA @ 5V
Temperature	-40~80C
Size	Diameter: 54mm Thickness: 14.5mm
Weight	36g
Cable Length	26cm
Other	• LNA MAX2659ELT+ RF Amplifier • Rechargeable Farah capacitance • Low noise 3.3V regulator • 26cm cable included

Pinout

Pinout

> □□ *[Image — to be added]*

> **i Info**
>
> Pinout is the same for all Holybro DroneCAN GPS

Dimension

Dimension

> **i Info**
>
> DroneCAN M9N & M8N has the same dimensions

> *[Image — to be added]*

DroneCAN GPS Setup Guide

DroneCAN GPS Setup Guide

Using Ardupilot Firmware

Using one DroneCAN GPS

Connect the 4-pin CAN cable connector to the CAN1 or CAN2 port on the flight controller.

Power the flight controller and connect it to Mission Planner. Go to "Config/Tuning > Full Parameter List" and modify the following parameters:

CAN_D1_PROTOCOL: 1 set virtual driver of CAN1 to DRONECAN

CAN_D2_PROTOCOL: 1 set virtual driver of CAN 2 to DRONECAN

CAN_P1_DRIVER: 1 set this parameter to enable the CAN1 bus

CAN_P2_DRIVER: 1 set this parameter to enable the CAN2 bus

GPS_TYPE: 9 set the communication protocol type of GPS 1 to DRONECAN

NTF_LED_TYPES: 231 Set to DRONECAN for LED type

There is no external safety switch. Set BRD_SAFETYENABLE as 0 to disable the safety switch, or connect an physical external safety switch

Click "Write Params" when done. CAN functions will be available after rebooting the flight controller.

Using two DroneCAN GPS

As the document is written, the firmware used for flight control is ArduCopter 4.1.5, which automatically allocates 2 node IDs for the GPSs; you can perform the following operation directly.

Connect two CAN cables to the CAN1 and CAN2 ports of the flight controller

Power up the flight controller and connect to Mission Planner. Go to "Config/Tuning > Full Parameter List" and modify the following parameters:

CAN_D1_PROTOCOL: 1 set virtual driver of CAN1 to DRONECAN

CAN_D2_PROTOCOL: 1 set virtual driver of CAN 2 to DRONECAN

CAN_P1_DRIVER: 1 set this parameter to enable the CAN 1 bus

CAN_P2_DRIVER: 1 set this parameter to enable the CAN 2 bus

GPS_TYPE: 9 set the communication protocol type of GPS 1 to DRONECAN

GPS_TYPE2 9 sets the communication protocol type of GPS 2 to DRONECAN

NTF_LED_TYPES: 231 Set to DRONECAN for LED type

There is no external safety switch on the DroneCAN GPS. You can set BRD_SAFETYENABLE to 0 to disable the safety switch, or connect a physical external safety switch

Click "Write Params" when done. CAN functions will be available after rebooting the flight controller.

Using PX4 firmware

Load PX4 firmware into the controller. Connect the 4-pin CAN connector from the DroneCAN GPS to the CAN1 or CAN2 port on the flight control.

Connect to the flight control and set the parameter "UAVCAN_ENABLE" to "Sensor Automatic Config".

About DroneCAN

About DroneCAN

DroneCAN

DroneCAN is the primary CAN protocol used by the ArduPilot and PX4 projects for communication with CAN peripherals. It is an open protocol with open communication, specification, and multiple open implementations.

Relationship with UAVCAN

DroneCAN was created to continue the development of the widely used UAVCAN v0 protocol. This protocol has proven itself as robust and feature rich and has been widely deployed in the commercial drone industry, and enjoys broad support among industry partners. The proposed introduction of the UAVCAN v1 protocol involved changes to UAVCAN that increased complexity and did not offer a smooth migration path for existing deployments. After extended discussions within the UAVCAN consortium, it was decided that the best solution was to continue development of DroneCAN v0 under the name DroneCAN.

DroneCAN Key Features

The first version of DroneCAN, known as DroneCAN v1, is identical to the existing UAVCAN v0 protocol. This means that the large number of existing UAVCAN v0 devices used throughout the drone industry are already DroneCAN v1 compliant. Features inherited from UAVCAN v0 include:

- Detailed protocol specification
- DSDL message description language for message description
- DNA (dynamic node allocation) for assignment of CAN node IDs
- Multiple open DSDL compilers that produce C and C++ bindings
- Rich python implementation
- Feature rich graphical user interface for diagnostics and device configuration
- Mature implementations in ArduPilot and PX4
- AP_Periph and PX4 can node multiple toolkits for easy creation of feature rich peripherals

Continued Evolution

DroneCAN is a continually evolving protocol. Starting with DroneCAN v1, the protocol will evolve to add new features to assist in the widespread adoption of CAN throughout the UAV industry. The DroneCAN project is committed to ensuring this evolution is done in a manner that retains compatibility with existing DroneCAN devices. Key features planned for DroneCAN in the near future:

- Support for FDCAN, allowing for higher data rates and larger frame sizes
- A node capability message to allow the DNA server to determine both the hardware and software capabilities of connected nodes, to facilitate smooth transitions to updated protocol versions
- Support for extending message definitions while retaining compatibility with existing implementations
- A comprehensive rework of existing DSDL message structures to improve efficiency and flexibility

Development

The DroneCAN project has an active development community.

- Discussions on Discord at https://dronecan.org/discord
- Development on GitHub at https://github.com/DroneCAN

More DroneCAN Information: https://dronecan.github.io/

Downloads

Downloads

3D CAD File

> *[Downloadable file — to be added]*