Heartbeat/Connection Protocol

The heartbeat protocol is used to advertise the existence of a system on the MAVLink network, along with its system and component id, vehicle type, flight stack, component type, and flight mode.

The heartbeat allows other components to:

- discover systems that are connected to the network and infer when they have disconnected. A component is considered to be *connected to the network* if its HEARTBEAT message is regularly received, and disconnected if a number of expected messages are not received.
- handle other messages from the component appropriately, based on component type and other properties (e.g. layout a GCS interface based on vehicle type).
- route messages to systems on different interfaces.

Message/Enum Summary

Message	Description
HEARTBEAT	Broadcast that a MAVLink component is present and responding, along with its type (MAV_TYPE) and other properties.

Enum	Description
MAV_TYPE	Type of the component. Flight controllers must report the type of the vehicle on which they are mounted (e.g. MAV_TYPE_OCTOROTOR). All other components must report a value appropriate for their type (e.g. a camera must use `MAV_TYPE_CAMERA`).
MAV_AUTOPILOT	Autopilot type / class. Set to `MAV_AUTOPILOT_INVALID` for components that are not flight controllers (e.g. ground stations, gimbals, etc.).
MAV_MODE_FLAG	System mode bitmap.

Enum	Description
MAV_STATE	System status flag.

HEARTBEAT Broadcast Frequency

Components must regularly broadcast their `HEARTBEAT` and monitor for heartbeats from other components/systems.

The rate at which the `HEARTBEAT` message must be broadcast, and how many messages may be "missed" before a system is considered to have timed out/disconnected from the network, depends on the channel (it is not defined by MAVLink). On RF telemetry links, components typically publish their heartbeat at 1 Hz and consider another system to have disconnected if four or five messages are not received.

A component may choose not to send or broadcast information on a channel (other than the `HEARTBEAT`) if it does not detect another system, and it will continue to send messages to a system while it is receiving heartbeats. Therefore it is important that systems:

- broadcast a heartbeat even when not commanding the remote system.
- do not broadcast a heartbeat when they are in a faulted state (i.e. do not publish a heartbeat from a separate thread that is unaware of the state of the rest of the component).

Connecting to a GCS or MAVLink API

The `HEARTBEAT` may also used by GCS (or Developer API) to determine if it **can** connect to a vehicle in order to collect telemetry and send missions/commands.

For example, *QGroundControl* will only connect to a vehicle system (i.e. not another GCS, gimbal, or onboard controller), and also checks that it has a non-zero system ID before displaying the vehicle connected message. QGC also uses the specific type of vehicle and other heartbeat information to control layout of the GUI.

INFO

The specific code for connecting to *QGroundControl* can be found in MultiVehicleManager.cc ↗ (see `void MultiVehicleManager::_vehicleHeartbeatInfo`).

Component Identity

The *type* of a component is obtained from its `HEARTBEAT.type` (`MAV_TYPE`) and `HEARTBEAT.autopilot` (`MAV_AUTOPILOT`) fields:

- A flight controller component must use a `MAV_TYPE` corresponding to a particular vehicle (e.g. `MAV_TYPE_FIXED_WING`, `MAV_TYPE_QUADROTOR` etc.), and set `HEARTBEAT.autopilot` to a valid flight stack.
- All other components must use a `MAV_TYPE` corresponding to the actual type (e.g.: `MAV_TYPE_GIMBAL`, `MAV_TYPE_BATTERY`, etc.), and should set `HEARTBEAT.autopilot` to `MAV_AUTOPILOT_INVALID`.

TIP

The recommended way to recognise an autopilot component is to check that `HEARTBEAT.autopilot` is not `MAV_AUTOPILOT_INVALID`.

Every component must have a system-unique component id, which is used for routing and for identifying multiple instances of a particular component type.

WARNING

Historically the component id was also used to determine the component type. New code must not make any assumption about the type from the id used (type is determined from `HEARTBEAT.type`).

MAVLink recommends that *by default* components use a type-appropriate component id from MAV_COMPONENT, and provide an interface to change the component id if needed. For example, a camera component might use any of the MAV_COMP_ID_CAMERA `n` ids, and should not use `MAV_COMP_ID_GPS2`.

TIP

Using type-specific component ids:

- makes id clashes less likely "out of the box" (unless two components of the same type are present on the same system).
- reduces the impact on legacy code that determines component type from the id.

Component Capabilities

The basic properties and capabilities of an autopilot can be determined by requesting the AUTOPILOT_VERSION message using MAV_CMD_REQUEST_MESSAGE, and for other components by requesting COMPONENT_INFORMATION_BASIC. This should normally be done on discovery of a new component, and the results cached.

The information includes hardware and software versioning information, and also the `capabilities`, a bitmap of the MAVLink services/protocols (MAV_PROTOCOL_CAPABILITY) supported by the component.

Edit on GitHub

Previous page
Microservices

Next page
Mission Protocol