Protocole MAVLINK pour Caméra

Auteurs :
Pedro CARVALHO MENDES
Cyprien QUIVET

Février 2020

Table des matières

1 Échanges MAVLINK natifs implémentés

1.1 Heartbeat

Nous utilisons le message HEARTBEAT pour annoncer l'existence d'une caméra sur le réseau MAVLink, ainsi que son ID système.

Le HEARTBEAT permet au système maître de découvrir les équipements connectés au réseau et d'en déduire lorsqu'ils se sont déconnectés. Un système est considéré comme étant connecté au réseau si son message HEARTBEAT est régulièrement reçu, et déconnecté si un certain nombre de messages attendus ne sont pas reçus.

1.2 Camera Informations

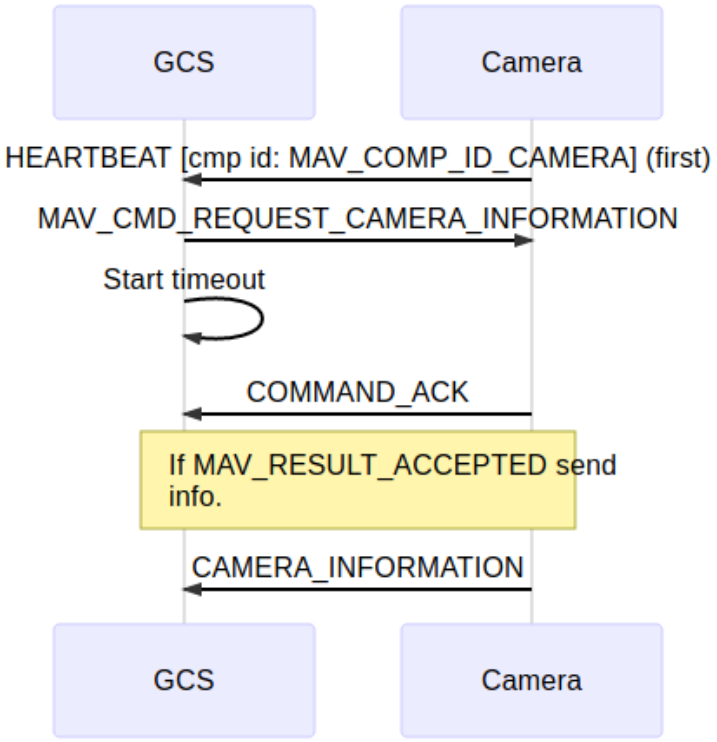

FIGURE 1 – Diagramme de demande d'informations

La station de contrôle doit suivre les étapes comme sur la figure ci-dessus pour obtenir les informations générales de la caméra.

CAMERA_INFORMATION est un message MAVLINK 2 contenant une structure avec les informations suivantes :

CAMERA_INFORMATION (#259)

[Message] **(MAVLink 2)** Information about a camera

Field Name	Type	Units	Values	Description
time_boot_ms	uint32_t	ms		Timestamp (time since system boot).
vendor_name	uint8_t[32]			Name of the camera vendor
model_name	uint8_t[32]			Name of the camera model
firmware_version	uint32_t			Version of the camera firmware (v << 24 & 0xff = Dev, v << 16 & 0xff = Patch, v << 8 & 0xff = Minor, v & 0xff = Major)
focal_length	float	mm		Focal length
sensor_size_h	float	mm		Image sensor size horizontal
sensor_size_v	float	mm		Image sensor size vertical
resolution_h	uint16_t	pix		Horizontal image resolution
resolution_v	uint16_t	pix		Vertical image resolution
lens_id	uint8_t			Reserved for a lens ID
flags	uint32_t		CAMERA_CAP_FLAGS	Bitmap of camera capability flags.
cam_definition_version	uint16_t			Camera definition version (iteration)
cam_definition_uri	char[140]			Camera definition URI (if any, otherwise only basic functions will be available). HTTP- (http://) and MAVLink FTP- (mavlinkftp://) formatted URIs are allowed (and both must be supported by any GCS that implements the Camera Protocol).

FIGURE 2 – Champs des informations de la caméra

1.3 Camera Settings

1.3.1 Get Camera Settings

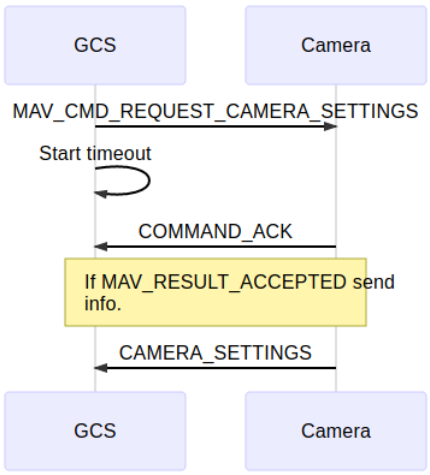

FIGURE 3 – Diagramme de demande de paramètres

Pour obtenir les paramètres de la caméra et savoir dans quel mode elle se trouve (CAMERA_MODE_IMAGE ou CAMERA_MODE_VIDEO), le système maître doit envoyer la requête correspondante (MAC_CMD_REQUEST_INFORMATION). La caméra transmet alors la structure suivante :

CAMERA_SETTINGS (#260)

[Message] **(MAVLink 2)** Settings of a camera, can be requested using MAV_CMD_REQUEST_CAMERA_SETTINGS.

Field Name	Type	Units	Values	Description
time_boot_ms	uint32_t	ms		Timestamp (time since system boot).
mode_id	uint8_t		CAMERA_MODE	Camera mode
zoomLevel **	float			Current zoom level (0.0 to 100.0, NaN if not known)
focusLevel **	float			Current focus level (0.0 to 100.0, NaN if not known)

FIGURE 4 – Champs des paramètres de la caméra

1.3.2 Set Camera Settings

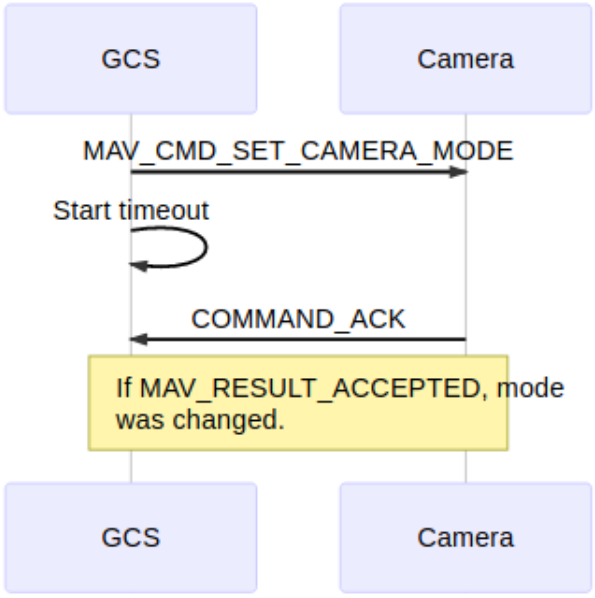

FIGURE 5 – Diagramme d'écriture du mode

Pour changer le mode d'une caméra(CAMERA_MODE_IMAGE ou CAMERA_MODE_VIDEO), le focus et le zoom , la station de contrôle doit transmettre la commande suivante (MAC_CMD_SET_CAMERA_MODE). La caméra répond alors par un acquittement.

1.4 Camera Streaming

La diffusion vidéo de la caméra peut être activée ou désactivée avec deux commandes MAVLINK.

1.4.1 Camera Start Streaming

MAV_CMD_VIDEO_START_STREAMING est la commande du protocole MAVLINK qui permet d'activer la caméra. Le flux de sortie vidéo ne passe pas par MAVLINK,

il est géré indépendamment. Pour lancer un enregistrement vidéo, le dialogue suivant est établit :

1. La station de contrôle envoie MAV_CMD_VIDEO_START_STREAMING à la caméra. Cette commande contient en paramètre "StreamID". "StreamID" est un identifiant de streaming vidéos pouvant aller de 0 à n.
2. La caméra répond à la commande par un acquittement qui peut contenir MAV RESULT_ACCEPTED ou MAV_RESULT_DENIED.

1.4.2 Camera Stop Streaming

MAV_CMD_VIDEO_STOP_STREAMING est la commande du protocole MAV-LINK qui permet de stopper la diffusion vidéo.

Pour ce faire la commande MAV_CMD_VIDEO_STOP_STREAMING est transmise à la caméra et la diffusion vidéo est stoppée si l'acquittement émit par la caméra correspond à MAV_RESULT_ACCEPTED.

2 Personnalisation du protocole MAVLINK

Il est possible de créer des messages dont les paramètres sont personnalisés par l'utilisateur. L'intérêt est de pouvoir échanger des données qui ne sont pas incluses dans les messages existants en utilisant tout de même le protocole MAVLINK.

2.1 Solution envisageable

Les messages MAVLINK sont définis dans des fichiers .XML nommés dialectes. Le dialecte "Common.xml" contient la définition de tous les principaux messages du protocole MAVLINK. Cependant pour des applications spécifiques, d'autres dialectes ont été crées. C'est par exemple le cas du dialecte "ArdupilotMega.xml" crée par la suite logiciel de pilotage automatique de véhicule sans pilote Ardupilot.

Une fois les messages personnalisés ajoutés dans les dialectes, il faut pouvoir les générer dans un langage de programmation pour les implémenter. Pour ce faire il existe l'outil MAVGEN qui permet de générer en langage C, C++, Java et Python les dialectes crées.

Lorsque les dialectes sont générés, il faut inclure les fichiers sources au projet. Il est alors possible d'échanger les nouveau messages via le protocole MAVLINK.

2.2 Exemple

2.2.1 Création d'un dialecte

Nous voulons échanger un nouveau message MAVLINK contenant les 4 paramètres suivants :

$var1$, $var2$, $var3$, $var4$, qui sont des entiers non signés sur 32 bits.

La première étape est donc de créer un nouveau dialecte que l'on nommera "tachyssema.xml" et qui contiendra la définition du nouveau message personnalisé. (voir figure ci-dessous).

```
<?xml version="1.0"?>
<mavlink>
  <include>common.xml</include>
  <!-- <version>9</version> -->
  <enums>
  </enums>
  <messages>
    <message id="402" name="Custom_Struct">
      <wip/>
      <description>MESSAGE PERSONNALISE TEST</description>
        <field type="uint32_t" name="var1" enum="variable">Comment variable 1</field>
        <field type="uint32_t" name="var2" enum="variable">Comment variable 2</field>
        <field type="uint32_t" name="var3" enum="variable">Comment variable 3</field>
        <field type="uint32_t" name="var4" enum="variable">Comment variable 4</field>
    </message>
  </messages>
</mavlink>
```

Listing 1 – Nouveau dialecte

2.2.2 Génération de la librairie en langage C

A présent, il faut généré le dialecte crée en un fichier exploitable en langage C. Pour ce faire, on ouvre le générateur de librairie MAVGEN comme sur la figure ci-dessous : Le GUI

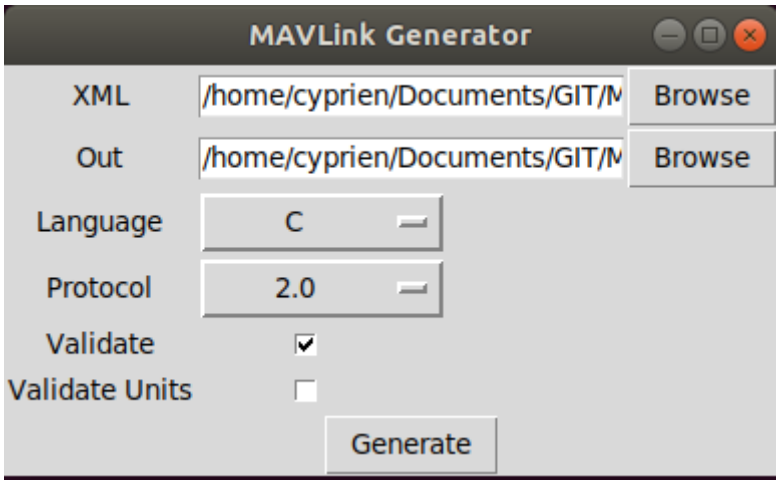

FIGURE 6 – Générateur MAVGEN

permet de choisir le fichier source, le repertoire cible ainsi que le langage et la version de MAVLINK désirée. Une fois la génération terminée en langage C, on retrouve nos fichiers au sein de la librairie MAVLINK :

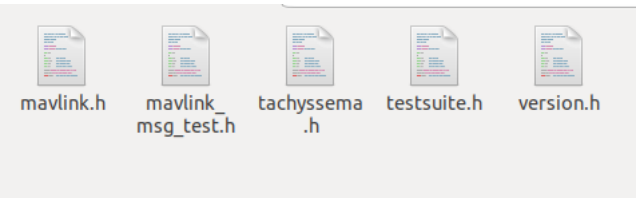

FIGURE 7 – Fichiers .C générés

2.2.3 Utilisation du message dans un projet

A présent, il est possible de transmettre le nouveau message via le protocole MAVLINK. On retrouve bien dans les fichiers générés la structure de notre message :

```
MAVPACKED(
  typedef struct __mavlink_custom_struct_t {
    uint32_t var1; /*<  Comment variable 1*/
    uint32_t var2; /*<  Comment variable 2*/
    uint32_t var3; /*<  Comment variable 3*/
    uint32_t var4; /*<  Comment variable 4*/
  }) mavlink_custom_struct_t;
```

Listing 2 – Structure personnalisée

3 Mise en œuvre du protocole MAVLINK pour le système OptSys

3.1 Utilisation des fonctions incluses dans le protocole MAVLINK

3.1.1 Heartbeat

Il faudra utiliser l'échange Heartbeat précédemment décrit pour s'assurer que la caméra est bien connectée au réseau. Il reste cependant à définir à quelle période ce type de message doit être émit par la caméra pour la considérer connectée ou non au réseau.

3.1.2 Informations et paramètres de la caméra

Pour obtenir les informations de la caméra, l'échange « Camera informations » sera établit et permettra à la caméra de transmettre la structure de données « CAMERA _INFORMATIONS » renvoyant l'ensemble des paramètres présents sur la figure 2.

Les messages MAV_CMD_REQUEST_CAMERA_SETTINGS et MAV_CMD_SET _CAMERA_MODE permettront d'obtenir et/ou modifier les paramètres suivants :
- Le zoom de la caméra
- Le focus de la caméra
- Le mode dans lequel se trouve la caméra (prise d'image ou de vidéo)

3.1.3 Lancement et arrêt d'une diffusion vidéo

Pour activer la diffusion vidéo de la caméra, nous utiliserons la commande MAV _CMD_VIDEO_START_STREAMING.

Pour arrêter la diffusion vidéo de la caméra, nous utiliserons la commande MAV _CMD_VIDEO_STOP_STREAMING.

3.2 Ajout de fonctions publiques au protocole MAVLINK

La création d'un dialecte publique sera mis en œuvre pour ajouter des définitions de messages publiques non existants via le protocole MAVLINK qui seront utiles à la caméra. Par exemple l'ajout de messages pour gérer la saturation et la teinte de la vidéo.

3.3 Ajout de fonctions privées au protocole MAVLINK

Certains attributs de la caméra ne seront pas visibles par le client. Comme par exemple :
- L'offset
- La matrice
- Le seuil
- La colorimétrie
- Le numéro de série etc ..

Pour se faire, un dialecte privé contentant la définition de ces messages sera établit.