



8 décembre 2025

PINSARD Achile - MENICHE tinhinen - SOO Lianne - MARION Astrid -
LACHGUER Nihal - BRUYERE Thomas

Référent : Lucia BERGANTIN

Client : CEREMA

Rapport intermédiaire client

Groupe 10



IMT Atlantique

Bretagne-Pays de la Loire
École Mines-Télécom



Table des matières

1 Schémas explicatifs	3
2 Résolution du capteur	4
3 Etude comparative des caméras	5
3.1 Analyse et comparaison des données	6
4 Fréquence de prise de vue et temps de vol	6
4.1 Fréquence d'acquisition	6
4.2 Vitesse d'écriture sur la carte SD	7
4.3 Temps d'enregistrement d'une photo	7
5 Analyse et solutions	8

1 Schémas explicatifs

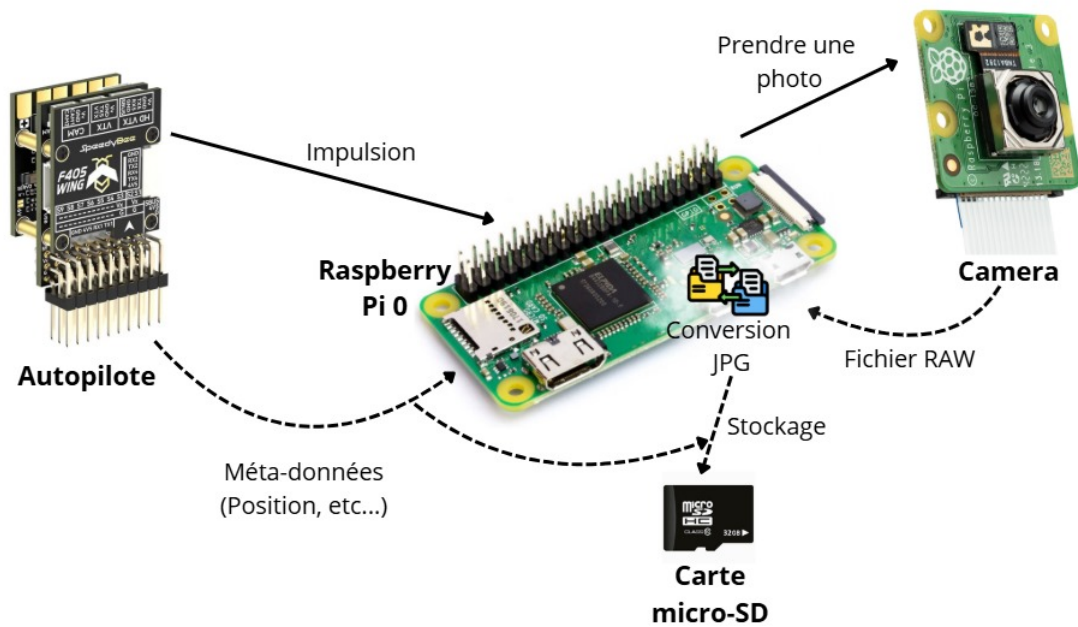


FIGURE 1 – Schéma de câblage global

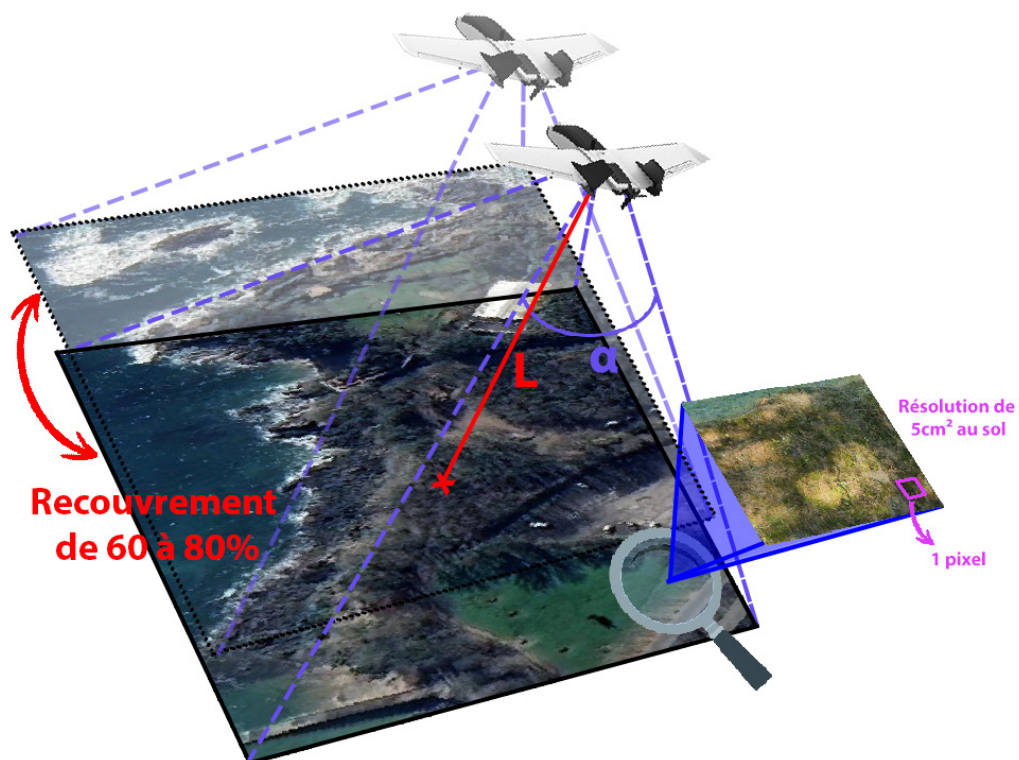


FIGURE 2 – Schéma de vulgarisation du recouvrement longitudinal et de la résolution au sol

2 Résolution du capteur

Pour calculer la résolution du capteur nous utilisons la formule suivante :

$$\text{Résolution}_{\text{capteur}} = \frac{4L^2}{\text{Résolution}_{\text{sol}}} \tan^2\left(\frac{\alpha}{2}\right)$$

Avec :

- $\text{Résolution}_{\text{capteur}}$ en px
- $\text{Résolution}_{\text{sol}}$ en m²/px
- L la hauteur du drone en m
- α l'angle de prise de vue en rad

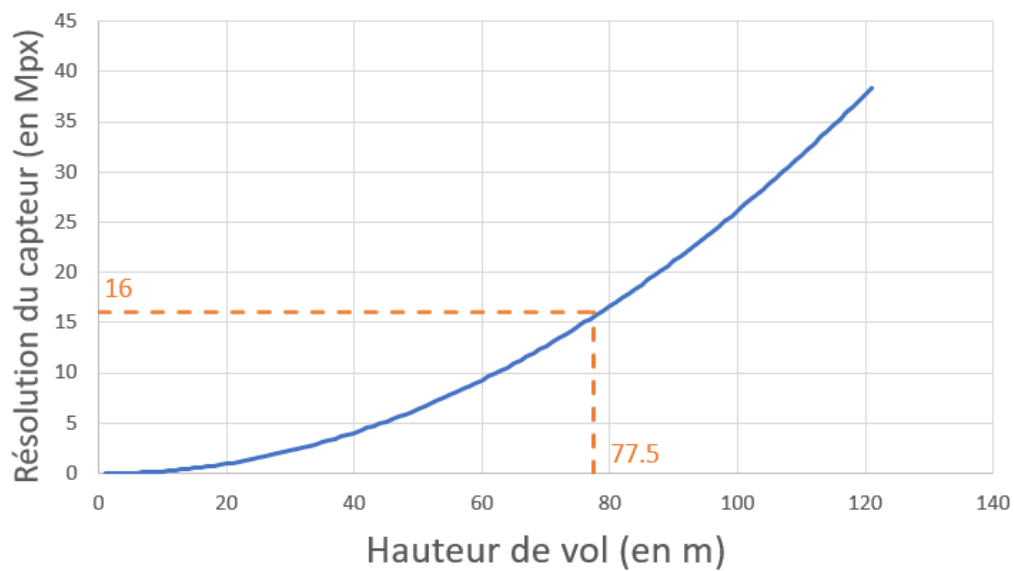


FIGURE 3 – Résolution du capteur par rapport à la hauteur de vol pour un angle α de 60° et une résolution au sol de 5 cm²/px

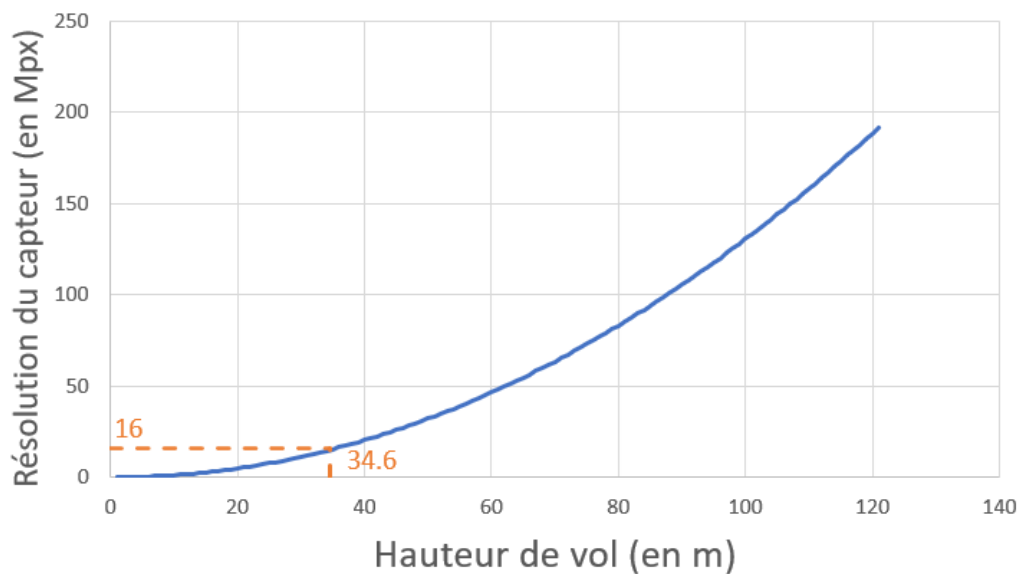


FIGURE 4 – Résolution du capteur par rapport à la hauteur de vol pour un angle α de 60° et une résolution au sol de $1\text{ cm}^2/\text{px}$

Pour une résolution au sol de $5\text{ cm}^2/\text{px}$, une résolution de capteur de 16 Mpx (taille du capteur de la solution trouvée) est correcte car elle permet d'aller jusqu'à 77.5 m d'altitude. Pour une résolution au sol de $1\text{ cm}^2/\text{px}$, l'altitude est très limitée, puisqu'avec un capteur de 16 Mpx, on ne pourrait pas aller plus haut que 34.6 m d'altitude.

Or, le marché ne propose pas de caméra possédant un capteur de plus de 20 Mpx et correspondant aux autres attentes (poids ...), et pour la suite (stockage notamment) un trop gros capteur risque de poser des problèmes.

Donc nous choisissons d'utiliser une résolution au sol de $5\text{ cm}^2/\text{px}$, qui permet de conserver une résolution correcte tout en permettant d'atteindre une altitude élevée.

3 Etude comparative des caméras

Pour l'étude comparative de caméras, nous nous sommes basés sur les contraintes suivantes :

- 16 à 18 MP minimum
- poids le plus faible possible (se rapprocher de 15g maximum)
- Port compatible Pi 0
- Possibilité de convertir en JPEG

L'angle d'ouverture étant plutôt constant entre les modules, nous n'avons pas pris ce critère en compte. Le CEREMA nous a indiqué que le prix était un critère secondaire.

Cela nous a donc conduits au tableau récapitulatif suivant :

Malheureusement malgré nos recherches, nous n'avons pas pu trouver assez d'informations concernant leurs consommations énergétiques pour ajouter un critère de comparaison.

4. Fréquence de prise de vue et temps de vol

Caméra	Résolution	Capteur	Shutter	AF	Poids (g)	Compatibilité Pi Zero	Format	Prix (€)
Raspberry Pi Global Shutter caméra (IMX296)	1.58 MP (1456×1088)	Sony IMX296	Global	Non	34	Oui (adapt. 22→15)	RAW10	60
Arducam OV9281 1MP Mono GS	1 MP (1280×800)	OmniVision OV9281	Global	Non	17	Oui	RAW8/RAW10	43
OV7251 Monochrome GS	0.3 MP (640×480)	OmniVision OV7251	Global	Non	10	Oui	RAW8	30
Arducam AR0234 2.3MP Couleur GS	2.3 MP (1920×1200)	OnSemi AR0234	Global	Non	15	Partielle (Jetson/Orin)	RAW10	100
Arducam 16 MP Autofocus (IMX519)	16 MP (4656×3496)	Sony IMX519	Rolling	Oui	10	Oui (adapt. 22→15)	RAW10/JPEG	30
Arducam 16 MP Noir (IMX519)	16 MP (4656×3496)	Sony IMX519	Rolling	Non	9	Oui (adapt. 22→15)	RAW10/COMP8	30
Arducam Mini 16 MP (IMX519)	16 MP (4656×3496)	Sony IMX519	Rolling	Non	6	Oui (22 broches direct)	RAW10/COMP8	30
Arducam 18 MP (AR1820HS PiVariety)	18 MP (4912×3684)	OnSemi AR1820HS	Rolling	Non	11	Oui	RAW10/JPEG (ISP)	120
Leopard IMX 20MP (AR2020)	20 MP (5120×3840)	OnSemi AR2020	Rolling	Non	25	Non (Jetson MIPI)	RAW10	300
SinoSense 20 MP Autofocus (IMX230)	20 MP (5344×4016)	Sony IMX230	Rolling	Oui	12	Partielle (MIPI smartphone)	RAW10	80
Phytec VM-116 PhyCam-P Mini	13–20 MP	Sony IMX214/IMX2xx	Rolling	Oui	10	Partielle (adapt. requis)	RAW10	200

TABLE 1 – Comparaison des caméras avec liens cliquables

3.1 Analyse et comparaison des données

Dans un premier temps, nous constatons que les capteurs **global shutter** ont une définition moindre. C'est pourquoi, après discussion, ce n'est plus un critère de contrainte. Dans un second temps pour certains modules, il nous a été compliqué de trouver des informations suffisantes pour les considérer comme viables, c'est le cas du **Leopard IMX** ainsi que du **phytech VM 116** (nous laissons une trace de ces modules pour permettre toutes recherches supplémentaires possibles).

Enfin, nous avons fait le choix de sélectionner les meilleurs modules ainsi que de trouver des caméras utiles dans certaines situations particulières. Les plus intéressantes sont donc :

- l'**Arducam 16 MP Autofocus (IMX519)** car elle possède un **autofocus** et est compatible avec l'ISP de la Raspberry tout en gardant une masse acceptable et une résolution qui répond à nos attentes.
- la **SinoSense 20 MP Autofocus (IMX230)**, qui réunit une bonne résolution et un **autofocus**. Cependant, elle ne possède pas d'ISP de compression JPEG. Il faut toutefois confirmer sa compatibilité avec la Raspberry Pi Zéro.

D'autres restent utiles dans des cas particuliers :

- Si le drone peut voler à une altitude constante et que le temps de stockage des images doit être réduit, l'**Arducam 18 MP (AR1820HS Pivariety)** possède un ISP qui permet de faire la conversion en JPEG par la caméra et une bonne qualité (mais n'a pas d'autofocus).
- En cas de problème de poids et que 16 MP suffisent la meilleure solution est l'**Arducam Mini 16 MP (IMX519)**.
- Pour des basses luminosités l'**Arducam 16 MP Noir (IMX519)** est très intéressante.

4 Fréquence de prise de vue et temps de vol

4.1 Fréquence d'acquisition

Voici la formule utilisée pour déterminer la fréquence d'acquisition :

$$f = \frac{V}{D \cdot (1 - R)} \quad \text{où} \quad D = 2L \cdot \tan\left(\frac{\alpha}{2}\right)$$

4. Fréquence de prise de vue et temps de vol

Avec L la hauteur du drone, D la largeur du champ au sol, V la vitesse, α l'angle de la caméra et R le recouvrement de 0 à 1 (voir Figure 2).

TABLE 2 – Fréquence d'acquisition (images/sec)
avec un recouvrement de 60 %

Vitesse (km/h)	30 m	60 m	90 m
45	1.81	0.91	0.61
60	2.41	1.21	0.81
90	3.62	1.81	1.21

TABLE 3 – Fréquence d'acquisition (images/sec)
avec un recouvrement de 80%

Vitesse (km/h)	30 m	60 m	90 m
45	4.52	2.26	1.51
60	6.03	3.02	2.01
90	9.05	4.52	3.02

4.2 Vitesse d'écriture sur la carte SD

Le temps qui s'écoule entre la prise de vue de la caméra et la fin du stockage de la photo sur la carte SD est **majoritairement limité par le temps de stockage sur la carte SD**.

Nous avons donc mesuré celui-ci afin de choisir une fréquence de prise de vue réaliste en fonction du temps d'enregistrement d'une photo.

Pour tester efficacement la rapidité d'écriture sur la carte nous avons utilisé la ligne de commande suivante directement depuis le terminal :

```
dd if=/dev/zero of=/dev/mmcblk2 bs=4M count=10 status=progress
```

Où :

- **dd** est un outil bas niveau qui copie des blocs tels quels, sans se préoccuper du système de fichiers.
- **if=/dev/zero** permet de définir la source des données à écrire (ici **/dev/zero** génère un flux de zéros (0x00) à l'infini).
- **of=/dev/mmcblk2** permet de définir la destination (ici le nom de la carte SD).
- **bs=4M** est la taille des blocs à écrire.
- **count=10** est la taille des blocs à écrire.
- **status=progress** affiche la progression en temps réel.

RÉSULTAT :

On obtient une vitesse d'écriture :

$$v_{writeSD} = 1.5MB/s$$

4.3 Temps d'enregistrement d'une photo

En considérant le modèle de caméra **Arducam 16 MP Autofocus (IMX519)**, nous pouvons calculer le temps d'enregistrement d'une photo grâce à la valeur de $v_{writeSD}$ trouvée précédemment.

La photo est en couleur, donc chaque pixel contient 3 composantes (R,V,B) chacune codée sur 8 bits (= 1 octet). Donc avant compression, une image de 16 Mpx pèse environ :

$$16 \times 3 = \mathbf{48 \text{ MB}}.$$

Il est donc essentiel de compresser les photos en JPEG au vu de la valeur de $v_{writeSD}$. Grâce à libcaméra, il est possible de choisir le taux de compression :

TABLE 4 – Taille d'une photo en fonction du taux de compression

Qualité indicative (en %)	Taux	Taille d'une photo (en MB)	Temps d'écriture de la photo (en s)	Fréquence d'acquisition max (en s^{-1})
95 (Très haute)	x5	9.6	6.4	0.1
90 (Haute)	x10	4.8	3.2	0.3
85 (Bonne)	x15	3.2	2.1	0.5
80 (Moyenne)	x20	2.4	1.6	0.6

5 Analyse et solutions

En confrontant les tables 2 et 3 avec la table 4, on observe que la qualité de l'image permettant d'atteindre une fréquence de prise de vue exploitable est de 80%, ce qui n'est pas très élevé et pas idéal pour de la photogrammétrie. Cette qualité permet d'obtenir une fréquence maximale d'acquisition de 0.6 s^{-1} , et les tables 2 et 3 indiquent que cette fréquence n'est accessible qu'avec un **recouvrement de 60 % à 90 m d'altitude et une vitesse de 45 km/h**. Cependant, d'après la section 2, pour conserver une résolution au sol de maximum $5 \text{ cm}^2/\text{px}$, on ne peut pas dépasser **77.5 m d'altitude**, ce qui est inférieur au 90 m souhaités.

Il nous faut donc trouver une alternative pour diminuer le temps de stockage d'une photo.

Une première alternative serait de **remplacer la carte SD par une clef USB**. Cela permettrait de transférer les images avec un temps de transfert plus court et donc sans recourir à une compression élevée. Cela ajouterait par contre d'autres contraintes :

La clef est **plus imposante et plus lourde**.

La consommation énergétique est plus importante, et pourrait impacter directement le temps de vol.

Une seconde alternative reviendrait simplement à faire le stockage durant le vol **sur la carte SD de la carte Raspberry Pi Zéro**, puis, après le vol transférer les images sur la seconde carte SD pour les récupérer. En effet cette solution permettra de passer à **0.5 seconde par images** (avec une qualité indicative de 90%). Cette solution pourrait poser problème avec l'OS sur la carte SD, endommager des fichiers ou faire planter le module.