

OPÉRATION : AUTONOMIE

MISSION DE PROGRAMMATION ET GESTION DE L'ÉNERGIE

[ACT MODE: STANDBY]

[SYS STATUS: INACTIVE]

ALT 0012M

R7 SR

VS 0.0 M/S

ALT 0012M

0.00W

VS 0.0 N/S

[SYS STATUS: INACTIVE]

[SYS STATUS: INACTIVE]

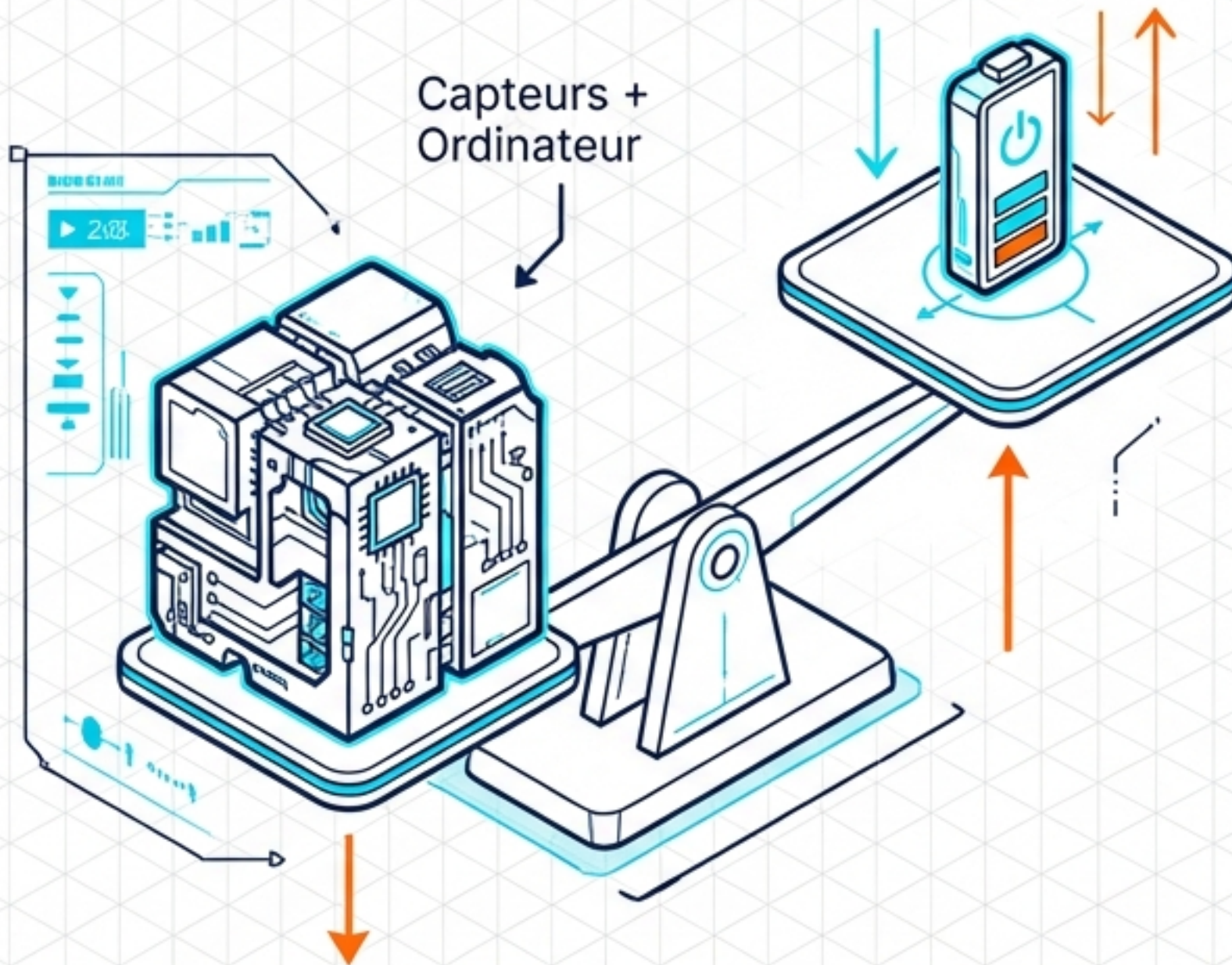


Bienvenue, Pilote. Votre drone est un bijou de technologie, mais il a une faiblesse critique. Votre mission : programmer son cerveau pour éviter le crash.

[ENERGY LEVEL: CRITICAL 05%]

[FIRMWARE UPDATE: REQUIRED]

LA GUERRE CONTRE LA GRAVITÉ



Un drone est bourré de capteurs et transporte un ordinateur de bord. Tout cela pèse lourd.

Pour économiser du poids et pouvoir décoller, le drone ne dispose que d'une petite batterie.

Conséquence : L'énergie délivrée ne permet d'assurer que quelques minutes de vol. Chaque seconde compte.

LE SCÉNARIO CATASTROPHE

Le drone doit retourner régulièrement à la station de recharge. Si l'énergie électrique vient à manquer en plein ciel, la sanction est immédiate : il tombe.

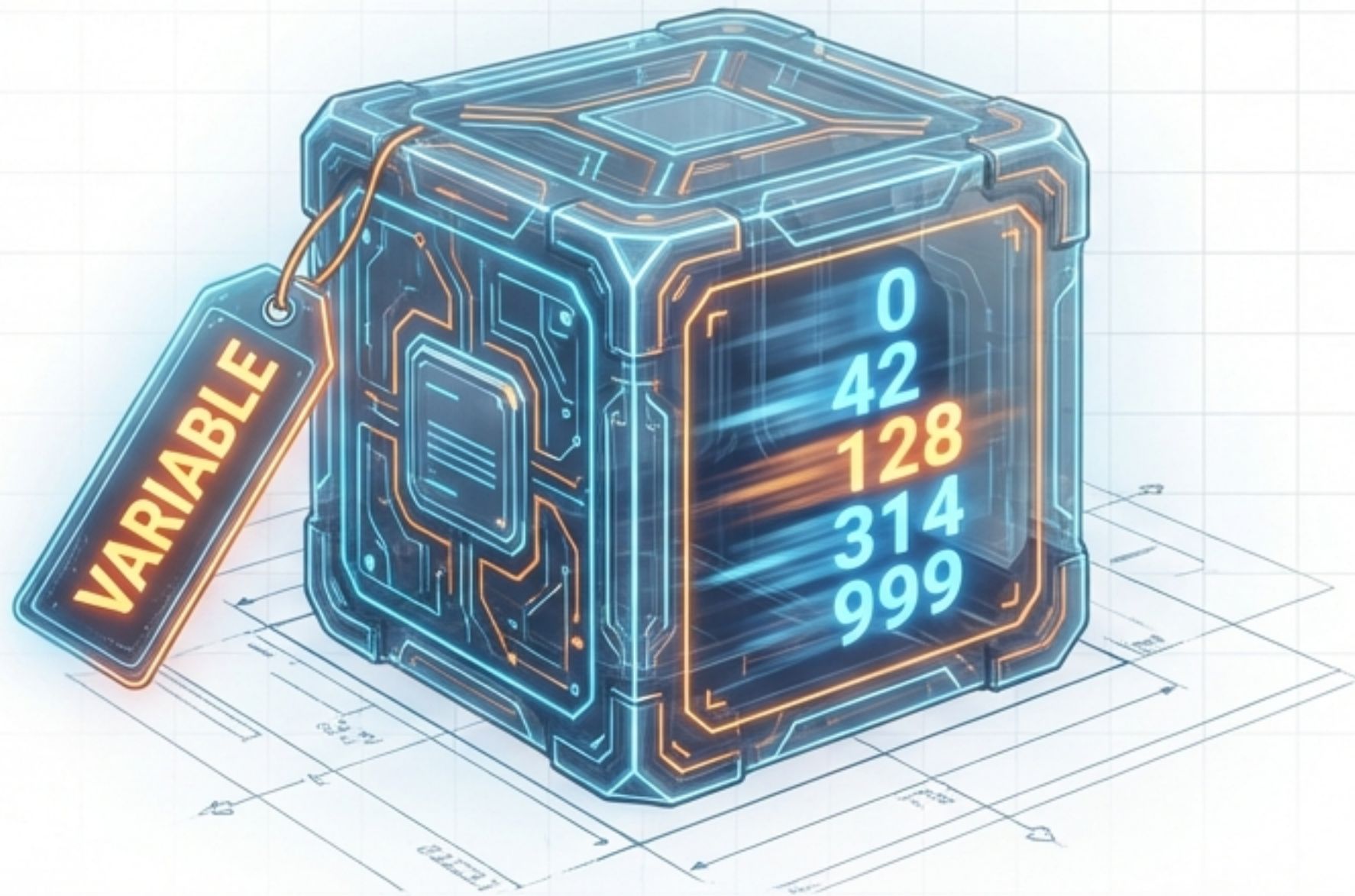


MISSION NOTE :

Le pilote ne peut pas deviner l'énergie restante. C'est au programme de pilotage de vérifier impérativement le niveau de charge.

L'ARME SECRÈTE DU CODEUR : LA VARIABLE

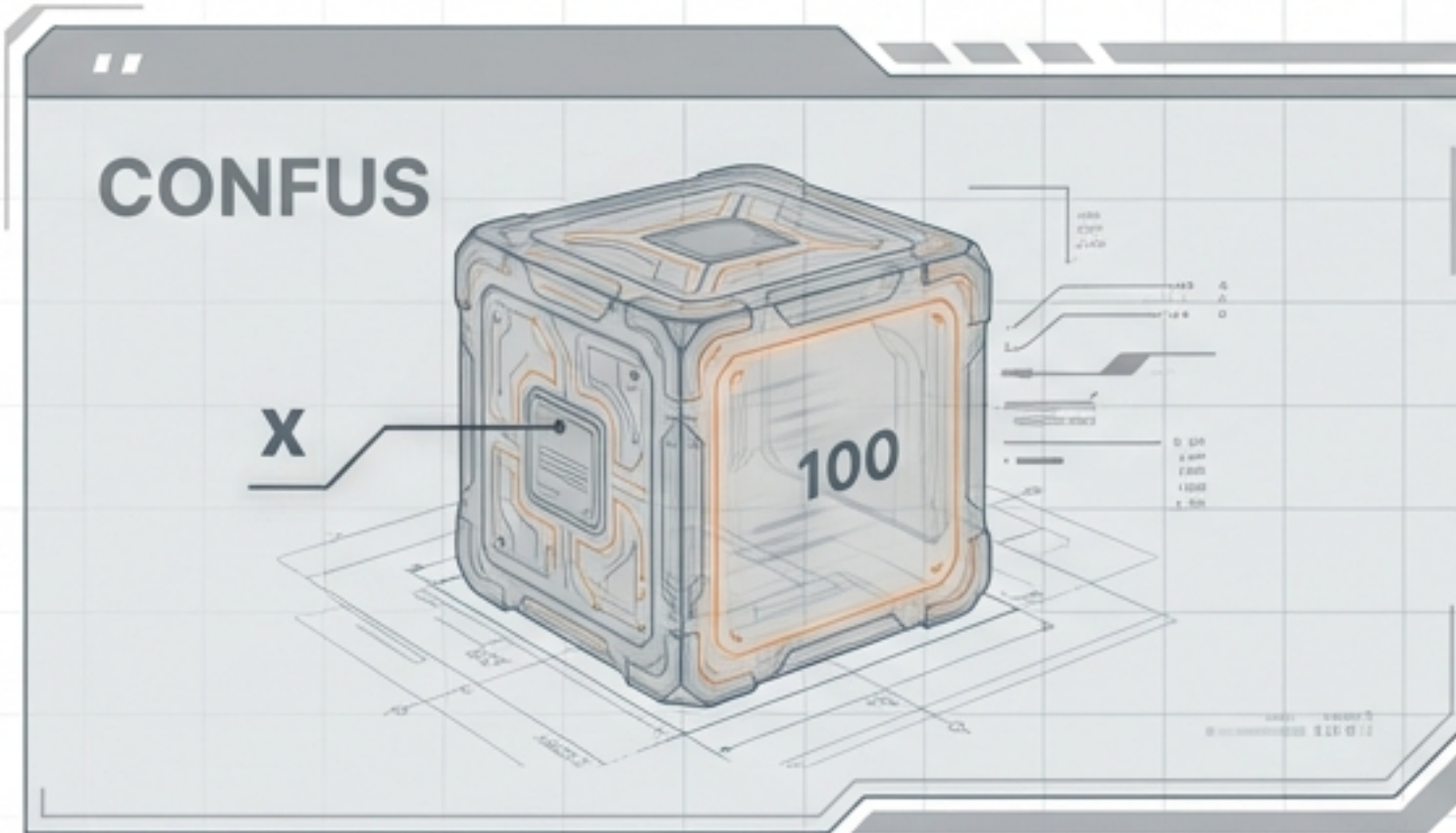
Lorsqu'un programme tourne, il doit stocker des informations. On utilise pour cela des variables.



Définition : Comme son nom l'indique, la valeur d'une variable change durant l'exécution du programme. C'est une boîte où l'information est modifiable à tout moment.

NOM DE CODE : EXPLICITE

Comme les sous-programmes, les variables ont besoin d'un nom clair. Le nom doit définir le type d'information concernée au premier coup d'œil.



Coding Terminal

> **Action** : Pour cette mission, nous allons créer une nouvelle variable nommée : « batterie ».

PHASE 1 : INITIALISATION

Au démarrage d'un programme, les variables doivent toujours être réinitialisées à une valeur prédéfinie.



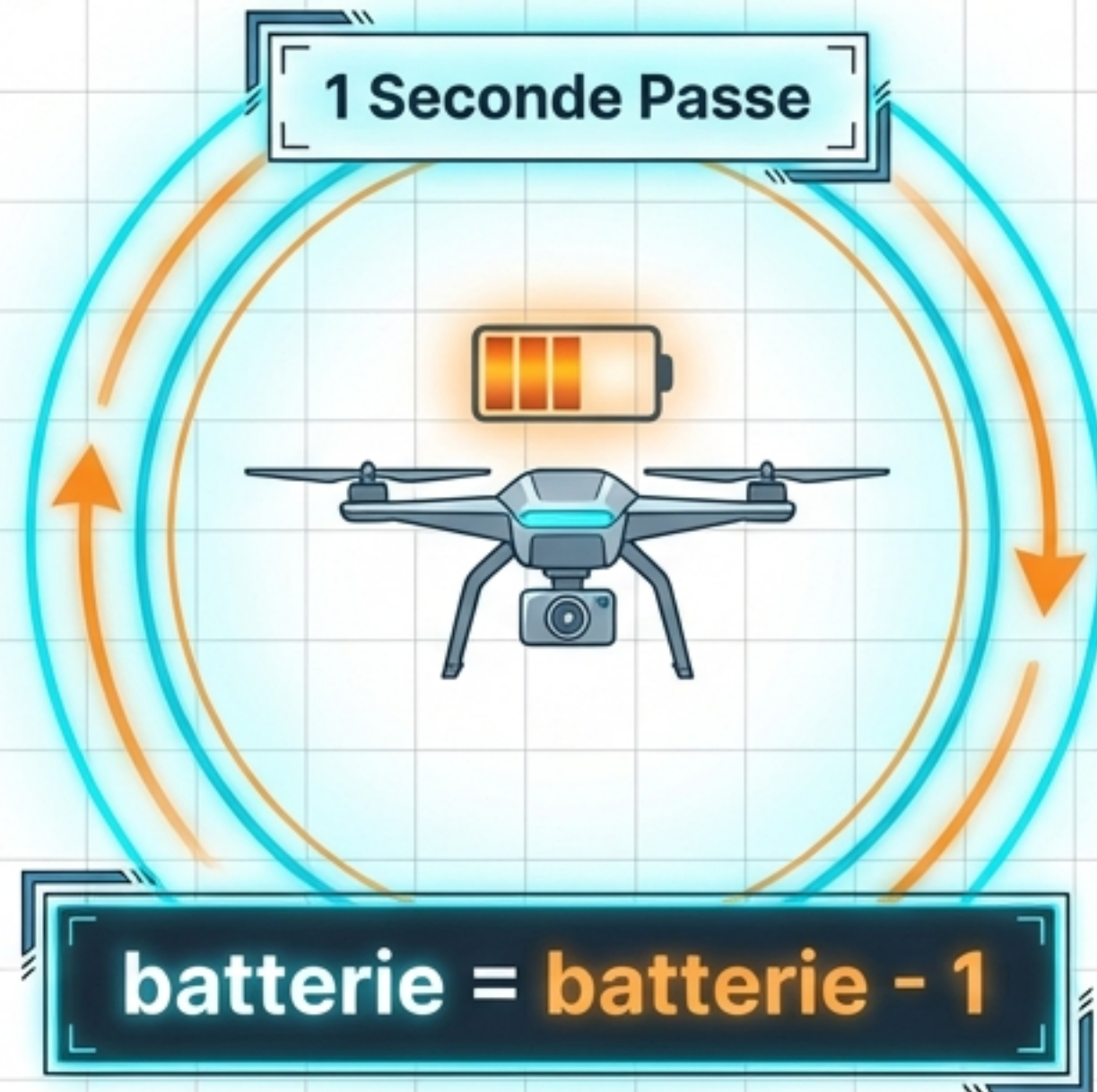
1. Condition : Le drone est sur la zone de départ.

2. Action : Il est considéré comme chargé.

3. Code : La variable « batterie » doit afficher le niveau 100.

PHASE 2 : LA CONSOMMATION

Une fois en l'air, l'énergie diminue. Dans notre modèle de simulation, la règle est stricte :
La batterie du drone perd 1% de sa charge par seconde.



Tu dois ajouter un sous-programme qui effectue ce calcul en boucle.

PHASE 3 : VISUALISATION TÊTE HAUTE

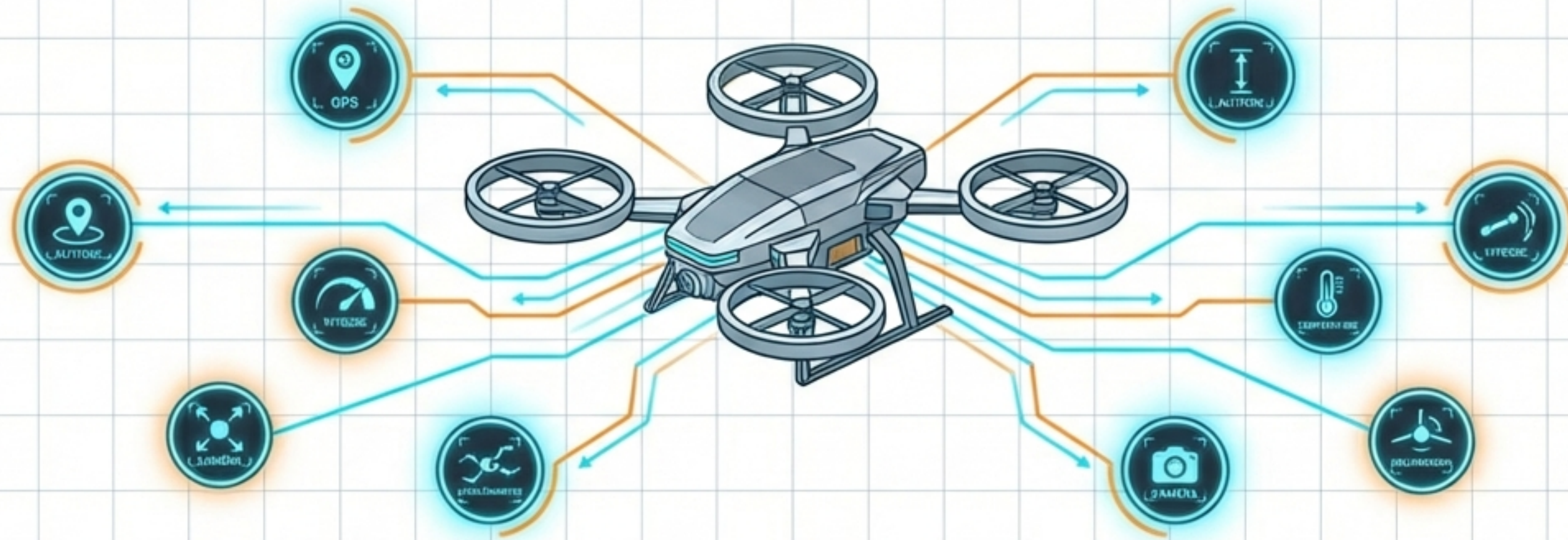
Une donnée invisible ne sert à rien. Il faut faire en sorte que la variable soit affichée pendant la simulation.



Grâce à ton code, l'ordinateur surveille la variable « batterie » et informe le pilote en temps réel.

LE COCKPIT COMPLET

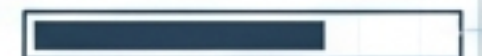
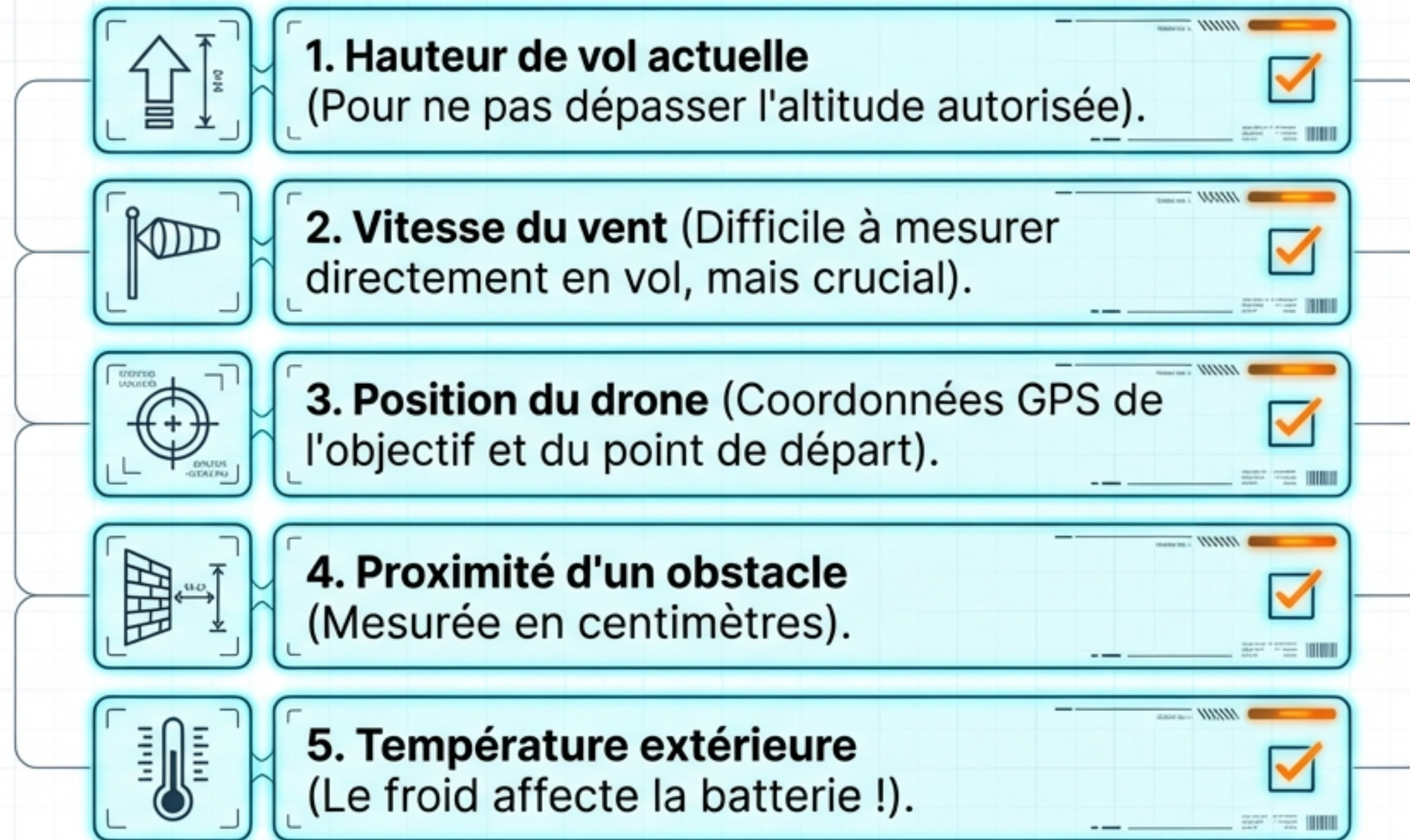
La batterie n'est pas la seule information vitale. Un vrai drone possède de nombreux capteurs qui fournissent constamment des données à l'ordinateur embarqué.



Quels autres types de valeurs mesurées le drone pourrait-il stocker sous forme de variables ?

PARAMÈTRES DE VOL CRITIQUES

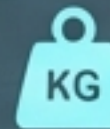
5 exemples de données essentielles à stocker dans des variables :



DIAGNOSTICS AVANCÉS

Pour les missions complexes, le drone analyse aussi :

- > DIAGNOSTIC LOG START >
- > SYSTEM CHECK: ROTOR STATUS
 - État des rotors (Un moteur est-il défectueux ? Les pales sont-elles arrêtées ?)
- > PAYLOAD ASSESSMENT
 - Masse de la charge utile (Ce qu'on transporte)
- > ENVIRONMENTAL SENSORS
 - Capteur de pluie (Météo)
- > HISTORICAL FLIGHT DATA
 - Itinéraire du vol précédent (Donnée complexe)
- > END LOG <

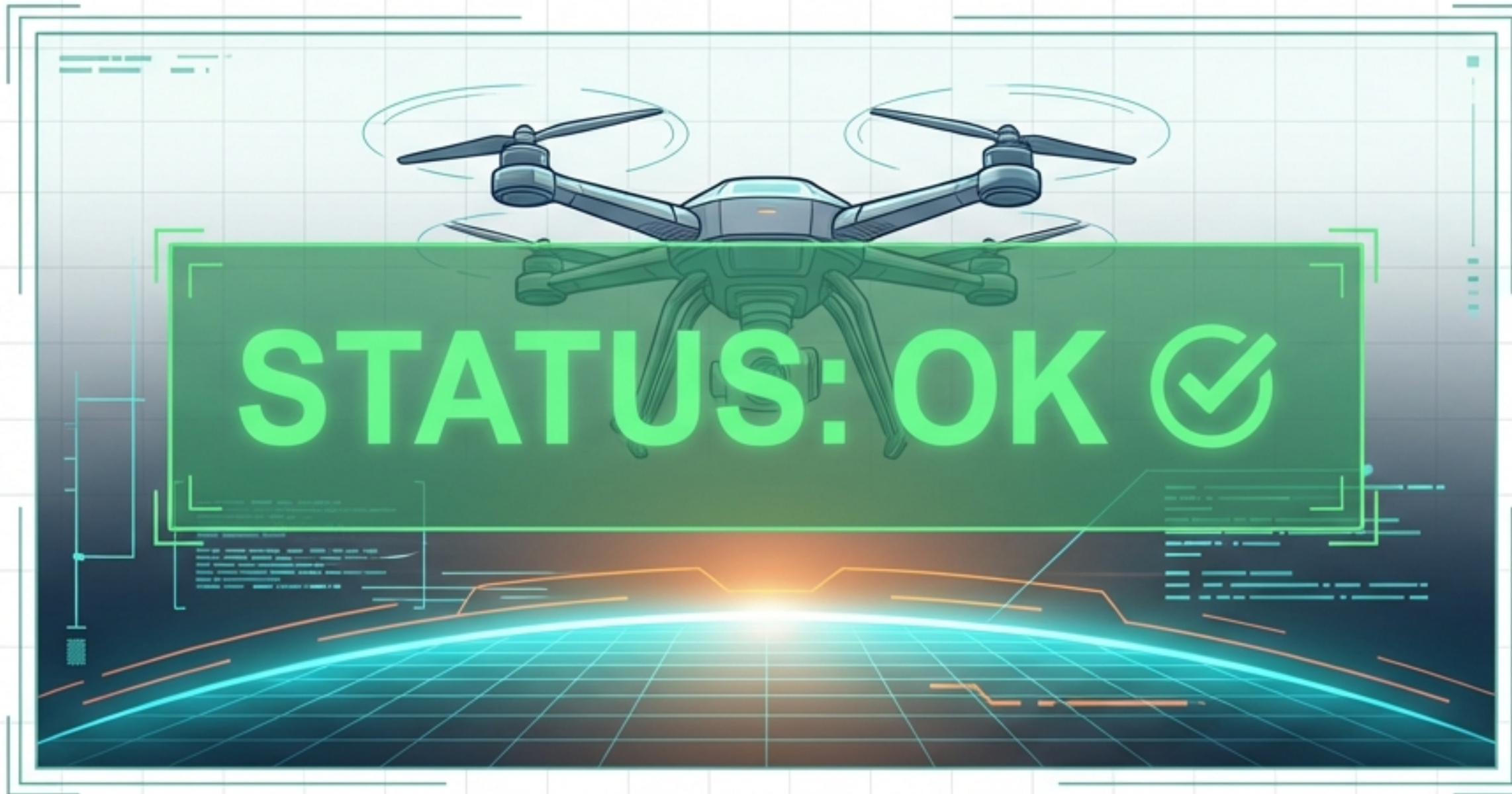


SYSTEM STATUS: ADVANCED DIAGNOSTICS LOG



MISSION ACCOMPLIE

Tu as transformé un risque (la panne) en une information maîtrisée (la variable).



- Initialiser à 100 sur la zone de départ.
- Soustraire 1% par seconde en vol.
- Afficher la variable pour le pilote.

Le programme est prêt. Bon vol !