

# KYRONEX

KINETIC YIELDING RESPONSIVE ONBOARD NEURAL EXPERT

## MODE D'EMPLOI

Intelligence Artificielle Vocale Multilingue – Embarquée

NVIDIA Jetson Orin Nano Super 8 Go

■ LLM GPU

■ TTS CUDA

■ STT CUDA

■ VISION

■ VIGILANCE

■ MULTILINGUE

Version 4.0 – 20 février 2026

CRÉÉ PAR MANIX – EMMANUEL GELINNE

on3egs@icloud.com | KITT Franco-Belge

# TABLE DES MATIÈRES

<b>1.</b>	<b>Présentation</b>	
	Qu'est-ce que KYRONEX ?	
<b>2.</b>	<b>Démarrage</b>	
	Lancer le système	
<b>3.</b>	<b>Interface Web</b>	
	Tableau de bord	
<b>4.</b>	<b>Identification</b>	
	Prénom et langue persistants	
<b>5.</b>	<b>Chat vocal</b>	
	Parler à KYRONEX	
<b>6.</b>	<b>Vision par caméra</b>	
	Voir avec YOLOX-S	
<b>7.</b>	<b>Mode Vigilance</b>	
	Surveillance caméra automatique	
<b>8.</b>	<b>Sons d'ambiance</b>	
	Ambiance MOTHER / AMB	
<b>9.</b>	<b>Messages proactifs</b>	
	Alertes et salutations auto	
<b>10.</b>	<b>Statistiques connexions</b>	
	Panneau IPs et sessions	
<b>11.</b>	<b>Architecture technique</b>	
	Sous le capot	
<b>12.</b>	<b>Dépannage</b>	
	Diagnostic et solutions	
<b>13.</b>	<b>Crédits et licence</b>	
	Informations légales	

# 1. PRÉSENTATION

KYRONEX (Kinetic Yielding Responsive Onboard Neural EXpert) est une intelligence artificielle vocale embarquée au style rétro-futuriste, conçue par Manix (Emmanuel Gelinne). Elle tourne entièrement en local sur un NVIDIA Jetson Orin Nano Super – sans connexion cloud obligatoire.

KYRONEX répond dans cinq langues (français, anglais, allemand, italien, portugais), reconnaît les utilisateurs par leur prénom, mémorise leur langue préférée, surveille son environnement via une caméra et envoie des alertes proactives. L'interface web adopte une esthétique MU-TH-UR 6000 / Knight Rider – fond noir, rouge, vert phosphore.

## > Capacités principales

- LLM GPU : Qwen 2.5 3B (Q5\_K\_M) via llama.cpp – ~1 400 ms
- STT GPU : faster-whisper base CUDA (CTranslate2) – ~221 ms
- TTS GPU : Piper fr\_FR-tom CUDA – ~265 ms (6x plus rapide que CPU)
- Vision : YOLOX-S ONNX (Apache 2.0, Megvii) – ~580 ms, 80 classes
- Mode Vigilance : surveillance caméra automatique + alertes temps réel
- Multilingue : fr/en/de/it/pt – langue mémorisée par utilisateur
- Identification : prénom demandé à la 1re connexion, mémorisé (MAC)
- Statistiques : connexions actives, 24 h, 7 jours avec liste IPs
- Sons MOTHER / Alien : ambiance oscillateur + double bip fin de message
- Orbe 3D réactif : 130 points Fibonacci réactifs à la voix
- Messages proactifs : salutations horaires, alertes température/RAM
- Mémoire persistante : faits utilisateur entre sessions
- Recherche web : DuckDuckGo sur mots-clés actualité/météo/prix
- Latence totale : ~2,3 s (parole → réponse audio complète)

## FONCTIONNEMENT 100 % HORS LIGNE

Aucune donnée n'est envoyée vers un serveur distant.

LLM, STT et TTS tournent sur le GPU du Jetson.

La recherche web (DuckDuckGo) est optionnelle et ne se déclenche que sur les mots-clés d'actualité.

## 2. DÉMARRAGE

### > Lancement normal

```
cd /home/kitt/kitt-ai  
bash start_kyronex.sh
```

Le script démarre automatiquement :

1. llama-server sur le port 8080 (modèle LLM sur GPU)
2. kyronex\_server.py sur le port 3000 (interface web HTTPS)
3. Boucle de surveillance : redémarre tout si l'un des deux plante

### > Service systemd (démarrage automatique au boot)

```
# Activer le démarrage automatique :  
sudo systemctl enable kitt-kyronex.service  
  
# Démarrer / arrêter manuellement :  
sudo systemctl start kitt-kyronex.service  
sudo systemctl stop kitt-kyronex.service  
  
# Voir les logs en direct :  
journalctl -u kitt-kyronex -f
```

### > Mode tunnel (accès distant)

```
TUNNEL=1 bash start_kyronex.sh
```

Active un tunnel Cloudflare. L'URL publique est affichée dans /tmp/cloudflared.log. Un mot de passe est requis (défaut : 1982, variable KYRONEX\_PASSWORD).

### > Accès à l'interface

#### ADRESSES D'ACCÈS

Depuis le Jetson : <https://localhost:3000>  
Depuis le réseau : <https://192.168.1.4:3000>

Le certificat HTTPS est auto-signé – acceptez l'exception de sécurité dans votre navigateur.  
Chrome : cliquer 'Paramètres avancés > Continuer'.

### > Indicateurs de démarrage réussi

```
[OK] Whisper prêt (CUDA float16)  
[OK] TTS fr GPU ready (265ms)  
[OK] Serveur HTTPS actif : https://localhost:3000
```

La barre rouge en bas de l'interface affichera << EN LIGNE >>.

### 3. INTERFACE WEB

L'interface adopte une esthétique rétro-futuriste : fond noir avec grille verte horizontale et rouge verticale, scanner KR descendant, orbe 3D réactif, terminal MU-TH-UR 6000 en bas à droite.

#### > Header – boutons de contrôle

Bouton	Rôle
VIG (rouge)	Mode Vigilance – surveillance caméra auto
AMB (vert)	Sons d'ambiance MOTHER – drone 60 Hz + harmoniques

#### > Zone d'entrée – boutons de chat

Bouton	Rôle
MIC (rouge)	Push-to-talk – clic pour parler, clic pour arrêter
AUTO (vert)	Écoute continue VAD – détecte la parole automatiquement
WAKE (violet)	Wake word -- réagit uniquement à KYRONEX
CAM (bleu)	Capture caméra + analyse YOLOX-S
ENVOYER	Envoi du message texte saisi

#### > Panneau de connexions (gauche)

Panneau translucide affiché à gauche (sur écrans > 700 px). Mis à jour toutes les 20 secondes :

- EN LIGNE : nombre de sessions actives (heartbeat 30 s)
- 24 H : nombre de connexions uniques sur les dernières 24 heures
- 7 J : nombre de connexions uniques sur les 7 derniers jours
- IPs : liste des adresses IP des sessions actives

#### > Terminal MU-TH-UR 6000 (bas droite)

Panneau de diagnostic style Alien – affiché sur grands écrans. Mis à jour toutes les secondes avec :

- CORE STATUS : NOMINAL / WARNING selon RAM/température
- TEMP : température du SoC (°C)
- RAM : mémoire disponible (Mo)
- SESSIONS : nombre de connexions actives

#### > Voicebox et orbe

La voicebox (barres verticales rouges animées) réagit à l'audio TTS via l'AnalyserNode Web Audio. L'orbe 3D (sphère Fibonacci 130 points) est visible en arrière-plan et réagit également à l'amplitude audio.

## 4. IDENTIFICATION – PRÉNOM ET LANGUE PERSISTANTS

À la première connexion depuis un appareil, KYRONEX affiche une fenêtre d'identification style MU-TH-UR 6000. Elle demande votre prénom et votre langue préférée. Ces informations sont mémorisées de façon permanente – elles survivent aux redémarrages.

### > Fenêtre d'identification (1re connexion)

- Fond noir, titre vert phosphore MU-TH-UR 6000
- Saisie du prénom : champ texte (Entrée ou bouton CONFIRMER)
- Sélection de la langue : boutons FR / EN / DE / IT
- Mémorisation par adresse MAC – identifie l'appareil

### > Comportement aux connexions suivantes

- KYRONEX vous salue par votre prénom : Bonsoir Manix.
- La langue choisie est verrouillée – KYRONEX ne change jamais de langue
- Même après un redémarrage complet du Jetson
- Même si vous lui parlez dans une autre langue

### > Changer la langue

Depuis l'interface web, les boutons FR / EN / DE / IT sont visibles dans la fenêtre d'identification (avant confirmation) ou via le panneau de configuration. La langue est envoyée au serveur via POST /api/set-lang.

#### RÈGLE ABSOLUE DE LANGUE

Une fois la langue choisie, KYRONEX répond UNIQUEMENT dans cette langue, quelle que soit la langue de l'interlocuteur.  
Si vous choisissez l'anglais, il répond en anglais même si vous lui parlez en français.

## 5. CHAT VOCAL

### > Mode Push-to-Talk (MIC)

Cliquez sur le bouton microphone (rond rouge). Il pulse pendant l'enregistrement. Parlez, puis cliquez à nouveau pour envoyer. La transcription est automatique (Whisper GPU).

### > Mode Auto-écoute (AUTO / VAD)

Cliquez sur AUTO. KYRONEX écoute en permanence et détecte automatiquement la parole. Paramètres :

- Seuil VAD : 0.008 (sensible, capte les voix douces)
- Durée silence : 800 ms (fin de phrase rapide)
- Parole minimum : 500 ms (évite les bruits courts)
- Anti-écho : sourdine pendant TTS + 1,5 s après

### > Mode Wake Word (WAKE)

Cliquez sur WAKE (violet). KYRONEX écoute passivement en continu et ne réagit que si vous prononcez KYRONEX (ou approximation). Après détection :

- Fenêtre de 6 secondes pour poser votre question
- Ou dites KYRONEX seul -> attente de la commande
- Utile pour utilisation mains-libres

### > Sons d'indication (style MOTHER/Alien)

- Pendant la réflexion LLM : sons sawtooth/square graves irréguliers
- Fin de message KYRONEX : double bip 523 Hz + 659 Hz (square, Q=6)
- Alerte vigilance : 4 bips 880/660/880/440 Hz (square, rapide)

### > Exemples de phrases

- Bonjour, comment vas-tu ? (exemple)
- KYRONEX, quelle est la météo à Paris ?
- What do you see? (en anglais si langue EN choisie)
- Regarde devant toi. -> déclenche automatiquement la caméra
- Qu'est-ce que je porte ? -> analyse vision + couleurs vêtements

#### RECHERCHE WEB AUTOMATIQUE

Sur les mots-clés : actualité, météo, prix, définition, aujourd'hui, bitcoin, etc. – KYRONEX consulte DuckDuckGo et injecte les résultats dans sa réponse.  
Entités privées (Manix, etc.) sont exclues du web search.

## 6. VISION PAR CAMÉRA

KYRONEX peut voir grâce à la webcam connectée. Il utilise YOLOX-S (Apache 2.0, Megvii) via cv2.dnn pour détecter les objets et analyser les couleurs. Le daemon vision tourne en mémoire – modèle chargé une seule fois.

Paramètre	Valeur
Modèle	YOLOX-S ONNX (35 Mo)
Temps d'inférence	~580 ms
Classes	80 (COCO) – personnes, animaux, objets
Seuil détection	0.35 (réduit les faux positifs)
Résolution	640 × 480 (letterbox 640 × 640)

### > Utilisation manuelle

- Cliquer sur le bouton CAM (icône appareil photo)
- Le bouton devient bleu pendant la capture (~600 ms)
- KYRONEX décrit ce qu'il voit en langage naturel
- Tapez une question avant de cliquer pour contextualiser

### > Déclenchement automatique par mots-clés

Ces mots dans votre message activent automatiquement la caméra :

regarde-moi, devant toi, caméra, qu'est-ce que tu vois,  
comment je suis habillé, de quelle couleur, tu me vois,  
décris-moi, analyse-moi, scanne

Cooldown : 30 s minimum entre deux captures auto.

### > Capacités d'analyse

- Détection personnes, animaux, véhicules, objets du quotidien
- Couleurs dominantes de chaque objet (rouge, bleu, vert...)
- Pour les personnes : couleur haut du corps + bas du corps
- Description transmise au LLM pour une réponse naturelle

## 7. MODE VIGILANCE

Le Mode Vigilance transforme KYRONEX en gardien silencieux. Lorsqu'il est activé, KYRONEX surveille la caméra toutes les 20 secondes et envoie une alerte si une présence est détectée dans des conditions suspectes.

### > Activer / désactiver

- Cliquer sur le bouton VIG (rouge, en haut à droite du header)
- VIG rouge clignotant = surveillance active
- VIG sombre = surveillance désactivée
- L'état est synchronisé avec le serveur via POST /api/vigilance

### > Conditions d'alerte

Situation	Message envoyé
0 → 1 personne détectée, terminale in	Présence détectée sur terminal inactif. Identité non confirmée.
1 → 2+ personnes détectées	Présence non identifiée dans la zone.

### > Affichage de l'alerte

- Message rouge dans le chat avec étiquette [ VIGILANCE ]
- Animation flash rouge (3 cycles)
- Son d'alarme : 4 bips square 880/660/880/440 Hz
- Synthèse vocale du message (TTS GPU)

#### CONSEIL D'UTILISATION

Activez VIG avant de quitter votre poste.  
KYRONEX vous alertera si quelqu'un s'approche  
du terminal pendant votre absence (> 5 min inactif).  
Nécessite une webcam connectée à /dev/video0.

### > Exigences techniques

- Webcam disponible sur /dev/video0
- Aucun autre service ne doit utiliser la caméra (kitt-recognition OFF)
- KYRONEX doit avoir au moins 1 client WebSocket connecté
- VISION\_SCRIPT doit exister (/home/kitt/kitt-ai/vision.py)

## 8. SONS D'AMBIANCE – BOUTON AMB

Le bouton AMB (vert, header) active un son d'ambiance continu généré par la Web Audio API – sans téléchargement de fichier audio. Inspiré des sons de MÈRE (MU-TH-UR 6000) dans Alien (1979).

### > Composition du son d'ambiance

- Drone principal : oscillateur 60 Hz (sawtooth), gain 0.018
- Harmonique : oscillateur 120 Hz (triangle), gain 0.010
- Bruit filtré : buffer noise + filtre passe-bas 200 Hz (BiquadFilter)
- LFO : modulation lente 0.08 Hz sur le gain principal
- Fondu d'entrée 2,5 s / sortie 1,5 s

### > Sons de réflexion LLM (style MOTHER/Alien)

Pendant que KYRONEX réfléchit (appel LLM), des sons courts irréguliers sont joués toutes les 600-1500 ms :

- 19 sons prédéfinis : sawtooth + square, graves 55-1319 Hz
- Filtre résonant (BiquadFilter, Q = 2-12) pour le timbre MOTHER
- Attaque rapide 0.08 s / extinction rapide
- S'arrêtent dès que la réponse commence à s'afficher

### > Son de fin de message

Après chaque réponse complète de KYRONEX (audio terminé) :

Double bip : 523 Hz + 659 Hz (square, Q=6) – 0.12 s chacun

## 9. MESSAGES PROACTIFS

KYRONEX peut envoyer des messages spontanés sans que vous le demandiez, via le WebSocket proactif (/api/proactive/ws). Ces messages apparaissent en orange italique dans le chat.

### > Salutations horaires

Heure	Message
6 h 00	Mes systèmes sont en ligne. Une nouvelle journée commence.
7 h 00	Tous mes capteurs sont opérationnels. Prêt pour la mission.
12 h 00	Peut-être une pause s'impose ? Mes circuits ne connaissent pas la faim.
18 h 00	Bonsoir. La journée a été productive, j'espère.
22 h 00	Je reste vigilant, mais vous devriez envisager du repos.
0 h 00	Mon scanner veille. Bonne nuit, Manix.

### > Alertes automatiques

Déclencheur	Alerte envoyée
Température > 85 °C	ALERTE CRITIQUE – surchauffe
Température > 75 °C	Température élevée – surveillance
Température > 70 °C	Information température (max 1/2 min)
RAM disponible < 500 Mo	RAM critique – ralentissement possible

Les alertes Vigilance (type vigilance\_alert) sont affichées en rouge avec le son d'alarme, distinctes des messages proactifs normaux.

## 10. STATISTIQUES CONNEXIONS

KYRONEX enregistre chaque connexion dans conn\_stats.json (roulement 2 000 entrées). Le panneau gauche de l'interface affiche les statistiques en temps réel.

### > Panneau gauche (connexions)

- EN LIGNE : sessions actives (heartbeat /api/ping toutes les 30 s)
- 24 H : connexions uniques sur les dernières 24 heures
- 7 J : connexions uniques sur les 7 derniers jours
- IPs : liste des IP actives (max 8 affichées)
- Rafraîchissement automatique toutes les 20 s

### > API disponibles

Endpoint	Description
GET /api/stats	Statistiques JSON (current/24h/7d/sessions)
POST /api/ping	Heartbeat session (renouvelle la présence)
POST /api/set-lang	Changer la langue persistante
GET /api/whoami	Infos session courante (nom, langue, MAC)

### > Fichiers de données

- users.json : { MAC → {name, lang} } – prénom + langue par appareil
- conn\_stats.json : historique des connexions (IP, MAC, nom, heure)

## 11. ARCHITECTURE TECHNIQUE

### > Vue d'ensemble du flux

```
Navigateur (port 3000 HTTPS)
|
+-- Texte/Voix --> /api/chat/stream --> LLM (~1400ms)
+-- Voix --> /api/stt --> Whisper GPU (~221ms)
+-- Vision --> /api/vision --> YOLOX-S (~580ms)
+-- Vigilance --> /api/vigilance --> toggle mode
|
+-- Audio <-- MultilingualTTS CUDA + SoX (~265ms fr)
+-- Proactif <-- WebSocket /api/proactive/ws
+-- Monitor <-- WebSocket /api/monitor/ws
|
+-- llama-server port 8080 (GPU CUDA, API OpenAI-compatible)
```

### > Composants logiciels

Composant	Détail
Serveur	Python aiohttp, port 3000 HTTPS self-signed
LLM	Qwen 2.5 3B (Q5_K_M), llama.cpp, SM 87
STT	faster-whisper base, CTranslate2 CUDA float16
TTS fr	Piper fr_FR-tom-medium, ORT CUDA (permanent)
TTS autres	Piper en/de/it/pt, CPU, LRU cache x1
Vision	YOLOX-S ONNX via cv2.dnn, daemon persistant
Détection langue	langdetect (~3 ms, seed=0)
GPU	NVIDIA Orin Nano Super, CUDA 12.6, SM 87
ORT GPU	onnxruntime-gpu 1.23.0 (compilé pour Tegra)
CTranslate2	4.5.0 avec CUDA (compilé depuis sources)

### > Paramètres llama-server

Paramètre	Valeur
--n-gpu-layers	99 (tout sur GPU)
--ctx-size	1024 (anti-OOM)
--batch-size	512
--flash-attn	on (Flash Attention)
--threads	4

### > Arborescence des fichiers clés

```
kitt-ai/
    kyronex_server.py      # Serveur principal
    piper_gpu.py           # TTS GPU + MultilingualTTS
    vision.py              # Vision YOLOX-S (daemon)
    start_kyronex.sh       # Script démarrage + surveillance
    generate_manual.py     # Ce générateur de PDF
    static/index.html      # Interface web (PWA)
    models/                # LLM .gguf + TTS .onnx + YOLOX .onnx
    audio_cache/           # Cache audio temp. (nettoyé 5 min)
    users.json              # Prénoms + langues par MAC
    conn_stats.json         # Historique connexions (2000 max)
    memory.json             # Mémoire persistante utilisateur
    certs/                 # Certificats HTTPS auto-signés
    logs/                  # Journaux JSONL conversations
    site/                  # Site web GitHub Pages
```

## 12. DÉPANNAGE

### > KYRONEX ne démarre pas

- Vérifier les paquets : ffmpeg, sox, libportaudio2
- Vérifier LD\_LIBRARY\_PATH dans start\_kyronex.sh
- Consulter : journalctl -u kitt-kyronex -f
- Tester llama-server séparément : curl http://localhost:8080/health

### > LLM HORS LIGNE dans l'interface

- llama-server charge le modèle (~14 s au boot – attendre)
- Vérifier : curl http://localhost:8080/health
- Le fichier .gguf est-il présent dans models/ ?
- Mémoire insuffisante ? Vérifier : free -m

### > Le microphone ne fonctionne pas

- Autoriser l'accès micro dans le navigateur
- HTTPS requis pour le micro (sauf localhost)
- Vérifier les appareils audio : arecord -l

### > La caméra ne fonctionne pas / Vigilance inactive

- Vérifier : ls /dev/video\* (doit afficher video0)
- Tester : /usr/bin/python3 vision.py --test
- Un seul programme peut utiliser la caméra à la fois
- Arrêter kitt-recognition si actif : sudo systemctl stop kitt-recognition
- kitt-driver utilise aussi /dev/video0 – le stopper si nécessaire

### > La voix TTS est lente ou absente

- Vérifier SoX : sox --version
- Vérifier le modèle TTS : ls models/\*.onnx
- ORT GPU disponible ? Relancer le serveur et vérifier les logs
- Vérifier la sortie audio : aplay /usr/share/sounds/alsa/Front\_Center.wav

### > Surveiller la mémoire (anti-OOM)

```
# RAM / VRAM en temps réel  
tegrastats | grep 'RAM'  
  
# Log VRAM KYRONEX  
tail -f /tmp/kitt_vram.log  
  
# Statut service  
systemctl status kitt-kyronex.service
```

### **PROTECTION OOM**

ctx-size 1024 + flash-attn activés par défaut.  
Si crash OOM répété : baisser ctx-size à 512.  
Le service redémarre automatiquement en cas de crash  
(Restart=always, jusqu'à 5 fois par 5 minutes).

## 13. CRÉDITS ET LICENCE

### CRÉATEUR

**Manix**

(Emmanuel Gelinne)

#### > Contact

E-mail	on3egs@icloud.com
Site web	<a href="https://on3egs.github.io/manix-kitt/">https://on3egs.github.io/manix-kitt/</a>
GitHub	<a href="https://github.com/on3egs">https://github.com/on3egs</a>
Groupe	KITT Franco-Belge (Facebook)

#### > Licence

##### ELASTIC LICENSE 2.0 (ELv2)

Copyright (c) 2026 Manix (Emmanuel Gelinne)

Utilisation libre pour usage personnel et éducatif.

INTERDIT sans accord écrit de l'auteur :

- Offrir ce logiciel comme service commercial hébergé
- Retirer ou modifier les notices de licence
- Créer des produits dérivés commerciaux

L'auteur conserve tous les droits commerciaux.

Fichier LICENSE inclus dans le projet.

#### > Technologies libres utilisées

- NVIDIA Jetson Orin Nano Super – CUDA 12.6, SM 87 (EULA commercial OK)
- llama.cpp – MIT – inférence LLM GPU optimisée
- Qwen 2.5 3B Instruct – Apache 2.0 – modèle de langage
- Piper TTS – MIT – synthèse vocale neuronale multilingue
- faster-whisper / CTranslate2 – MIT – STT GPU
- onnxruntime-gpu 1.23.0 – MIT – inférence ONNX CUDA
- YOLOX-S (Megvii) – Apache 2.0 – détection d'objets
- OpenCV – Apache 2.0 – vision par ordinateur
- SoX – GPL-2.0 – effets audio (outil externe, pas de contamination)
- FFmpeg – LGPL/GPL – conversion audio (outil externe)
- Python aiohttp – Apache 2.0 – serveur web asynchrone
- langdetect – Apache 2.0 – détection de langue



