MONTRE CONNECTÉE nRF5340

Projet de Systèmes Embarqués

Groupe : TimeX

Rafelimanana Deraniaina (Project owner) **Khanh-Phuong NGUYEN**

HAJJAJ Houda **EL KILI Rim**

Enseignants référents :

RICHARD Noël
MBOEGLEN Hervé

Année scolaire 2025/2026

Ce document a été rédigé dans le cadre du module de Systèmes Embarqués (SEC). Il présente la conception et l'analyse d'une montre connectée simplifiée basée sur la plateforme nRF5340 et l'OS temps réel Zephyr. Ce projet s'inspire du design ouvert ZSWatch et exploite le shield X-NUCLEO-IKS01A3, un écran graphique tactile, la connectivité Bluetooth Low Energy et un module RTC pour la gestion de l'heure et du calendrier.

Table des matières

Table des figures

Liste des tableaux

Chapitre 1

Analyse du projet

1.1 Contexte et référence ZSWatch

Le projet ZSWatch propose une montre entièrement open source, basée sur un module u-blox NORA-B10 intégrant un SoC nRF5340, un écran circulaire 240×240 tactile, une IMU, une mémoire externe et une interface haptique. Sur le plan logiciel, il utilise Zephyr RTOS, un ensemble d'applications (watchfaces, notifications, musique, boussole, etc.) et une communication BLE avec l'application Android GadgetBridge.

Notre projet reprend les grandes briques fonctionnelles (OS temps réel, UI graphique, capteurs, Bluetooth, RTC) en les adaptant au matériel disponible (nRF5340 DK, shield IKS01A3, écran TFT 320×240, RTC RV-8263-C8).

1.2 Éléments principaux et schéma bloc

Les fonctions principales retenues sont :

- Système de base : SoC nRF5340 (double cœur Cortex-M33), mémoire interne, alimentation batterie.
- Interface utilisateur : écran TFT 320×240 tactile, piloté via SPI, framework LVGL pour l'interface graphique.
- Capteurs de mouvement et d'environnement : LSM6DSO (accéléromètre + gyroscope) et LIS2MDL (magnétomètre), ainsi que les capteurs de température, humidité et pression du shield X-NUCLEO-IKS01A3.
- Communication : Bluetooth Low Energy pour la liaison avec un smartphone.
- Gestion du temps : module RTC externe RV-8263-C8 pour l'heure et le calendrier.
- Gestion de l'énergie : mesure du niveau de batterie et modes basse consommation (sleep, deep sleep).

Le schéma bloc de la figure 1.1 résume l'architecture : le nRF5340 au centre, les périphériques connectés (écran, capteurs, RTC, batterie), la liaison BLE avec le smartphone et les entrées utilisateur.

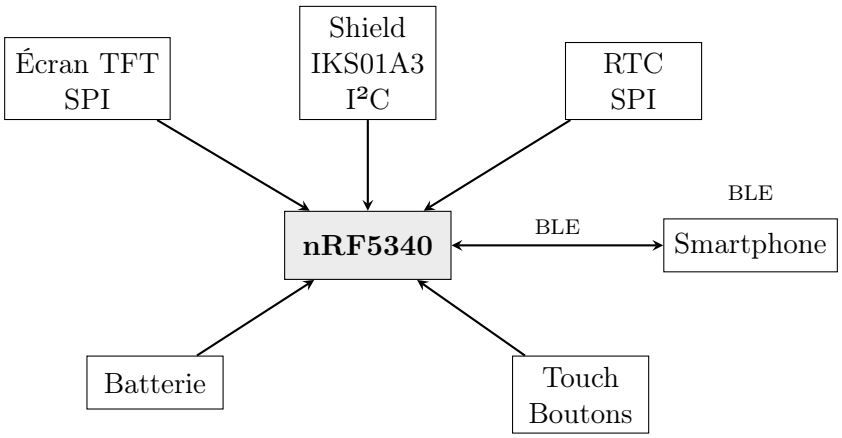

FIGURE 1.1 – Schéma bloc du système (montre connectée nRF5340)

1.3 Contraintes et choix logiciels

Zephyr impose une structuration par devicetree et overlays pour décrire le hardware, par Kconfig pour activer les modules (BLE, capteurs, display, RTC) et par une organisation en modules et threads pour structurer l'application. Le framework LVGL est utilisé pour l'interface graphique, avec génération d'écrans via SquareLine Studio et intégration dans Zephyr.

La machine d'états fournie dans le cours SEC servira de base pour organiser les modes de la montre (veille, heure, menu, activité, synchronisation, configuration), avec une séparation claire entre logique d'application et drivers matériels.

1.4 Tables des entrées et sorties

Les entrées du système incluent : capteurs du shield (LSM6DSO, LIS2MDL, HTS221, LPS22HH), écran tactile, boutons, module RTC. Les sorties incluent : affichage LCD, LED, communication BLE. Les tableaux ci-dessous résument les entrées et sorties.

TABLE 1.1 – Entrées du système

Entrée	**Description / Interface**
LSM6DSO	Accéléromètre + gyroscope (shield IKS01A3, I²C)
LIS2MDL	Magnétomètre (shield IKS01A3, I²C)
HTS221	Température et humidité (shield IKS01A3, I²C)
LPS22HH	Pression atmosphérique (shield IKS01A3, I²C)
Écran tactile	Détection des appuis (SPI / GPIO)
Boutons	Commande utilisateur (GPIO)
Module RTC (RV-8263-C8)	Heure et calendrier (SPI)

TABLE 1.2 – Sorties du système

Sortie	**Description / Interface**
Écran LCD TFT	Affichage graphique (SPI, LVGL)
LED	Indication d'état (GPIO)
Communication BLE	Liaison avec smartphone

Accès au code source :
Lien du dépôt Git : `https://github.com/votre-repo/montre-nrf5340.git`

Chapitre 2

Spécifications externes

2.1 Objectifs du projet

L'objectif de ce projet est de concevoir et réaliser une montre connectée complète basée sur un système embarqué nRF5340 et Zephyr RTOS. La montre doit :

- Exploiter un écran graphique tactile et le framework LVGL
- Intégrer les capteurs de mouvement et d'environnement (shield X-NUCLEO-IKS01A3)
- Assurer une connectivité Bluetooth Low Energy avec un smartphone
- Gérer l'heure et le calendrier via un module RTC externe

2.2 Fonctionnalités possibles

Le tableau ci-dessous résume les fonctionnalités envisagées pour la montre, par catégorie.

TABLE 2.1 – Tableau des fonctionnalités de la montre connectée

Catégorie	**Fonctionnalité**
Base	Affichage de l'heure et de la date (format 12/24h), basé sur la RTC RV-8263-C8.
Base	Affichage du niveau de batterie et indication de charge.
Base	Alarmes, minuteur et chronomètre.
Base	Plusieurs cadrans (watchfaces) sélectionnables (thèmes minimalistes, analogiques, numériques).
Capteurs	Podomètre et estimation de distance (accéléromètre LSM6DSO).
Capteurs	Détection de gestes (réveil écran, tap, double-tap).
Capteurs	Boussole numérique (magnétomètre LIS2MDL).
Capteurs	Température, humidité, pression (HTS221, LPS22HH).
Capteurs	Détection d'activité (marche, repos) et éventuellement chute (IMU).
BLE	Synchronisation heure/date avec le smartphone.
BLE	Réception de notifications (SMS, appels, applications).
BLE	Synchronisation des données de capteurs vers une application mobile.
BLE	Contrôle multimédia (play/pause, volume).
IHM	Menus graphiques (horloge, capteurs, réglages, activité).
IHM	Écrans capteurs : accélération, boussole, météo.
IHM	Réglages : cadran, heure, date, unités.
IHM	Tactile (SquareLine Studio, LVGL).
Avancées	Modes économie d'énergie.
Avancées	Journalisation locale, synchronisation différée.
Avancées	Balise BLE (direction finding).

Chapitre 3

Modélisation du système

3.1 Bords du modèle

Cette section décrit les frontières du système : entrées (capteurs, utilisateur, communication) et sorties (écran, LED, stockage, communication). Le schéma ci-dessous illustre le nRF5340 au centre, connecté à l'écran TFT, au shield IKS01A3, au module RTC et aux interfaces BLE.

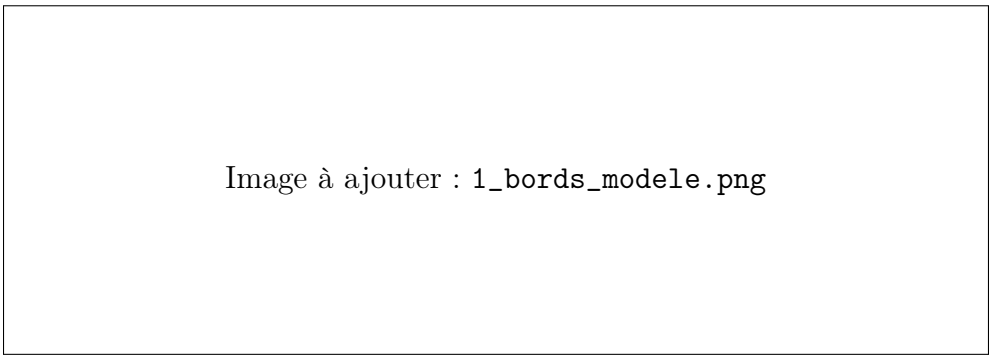

FIGURE 3.1 – Bords du modèle

3.2 Flot de données

Description du flux des données depuis les capteurs et les entrées utilisateur vers le traitement (Zephyr, machine d'états) et les sorties (affichage LVGL, BLE, stockage).

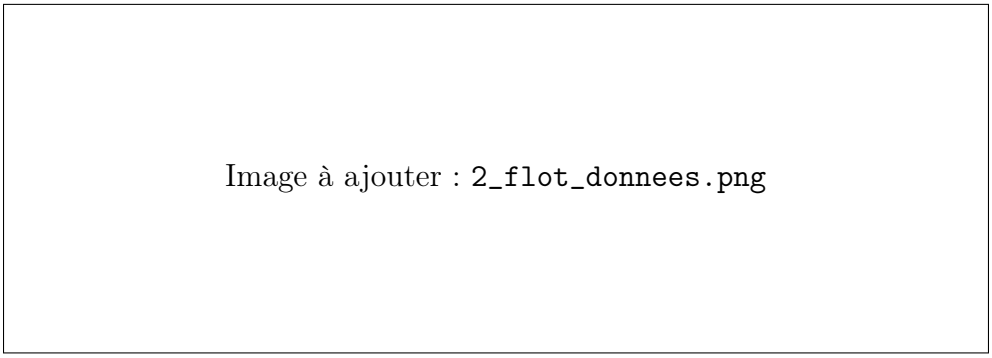

FIGURE 3.2 – Flot de données

3.3 Flot d'événements

Les événements (tactile, boutons, alarmes RTC, données BLE) déclenchent des traitements et des changements d'état. Le schéma ci-dessous décrit le flot d'événements.

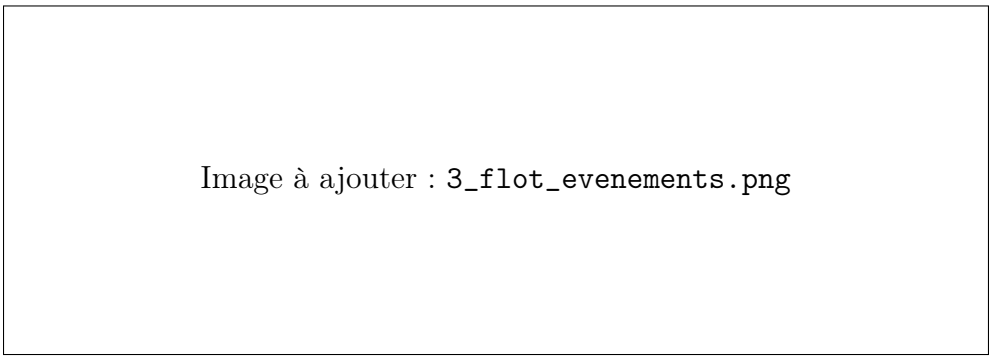

FIGURE 3.3 – Flot d'événements

3.4 Machine d'états

La machine d'états organise les modes de la montre (veille, heure, menu, activité, synchronisation, configuration), avec une séparation claire entre logique d'application et drivers matériels.

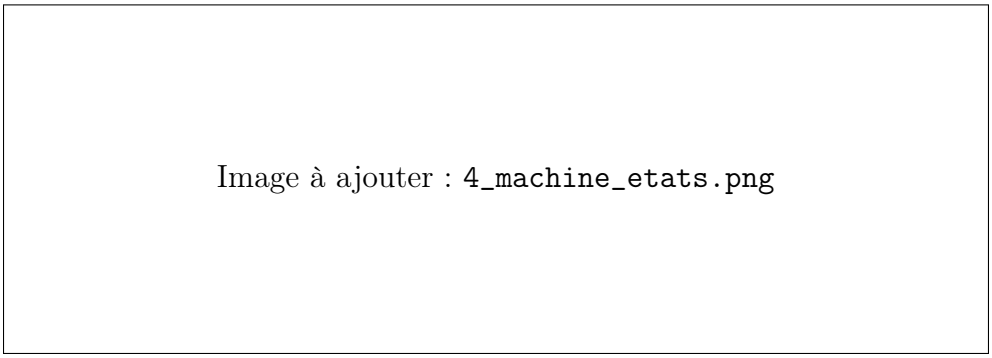

FIGURE 3.4 – Machine d'états de gestion du système

Chapitre 4

Développement et validation

4.1 Stratégie de test

La validation de la montre repose sur plusieurs niveaux :
— Tests unitaires des drivers Zephyr (capteurs, écran, RTC, BLE) pour vérifier les communications matérielles (I^2C, SPI, GPIO).
— Tests d'intégration pour valider la cohérence entre machine d'états, interface graphique LVGL et services BLE.
— Tests système sur cible réelle (nRF5340 DK + shield IKS01A3 + écran) pour évaluer les performances, la réactivité et la consommation.

4.2 Cas de test principaux

Exemples de cas de test :
— Vérification de l'affichage correct de l'heure et de la date après configuration de la RTC et après synchronisation BLE.
— Test de la détection de mouvements (réveil de l'écran, comptage de pas) et comparaison grossière avec un podomètre de référence.
— Test de la boussole : vérification qualitative de la direction affichée par rapport à une boussole physique.
— Test des notifications : réception d'un SMS ou d'une notification d'application et affichage sur la montre.
— Test des modes basse consommation : mesure approximative de la consommation dans différents états (veille, affichage actif, BLE actif).

4.3 Critères de validation

Le projet sera considéré comme validé si :
— Les fonctionnalités minimales imposées (Zephyr, LVGL, capteurs, BLE, RTC) sont opérationnelles.
— L'interface utilisateur permet une navigation fluide entre les principales applications.

— La machine d'états fonctionne sans blocage (pas de deadlock, comportement cohérent).
— La consommation reste compatible avec une utilisation sur batterie (ordre de grandeur raisonnable au vu du matériel de développement).

Chapitre 5

Interface Homme-Machine (IHM)

5.1 Choix technologiques

L'interface graphique repose sur le framework **LVGL** intégré sous Zephyr, avec un écran TFT 320×240 tactile piloté via SPI. La conception des écrans peut être réalisée avec SquareLine Studio puis intégrée dans le projet.

5.2 Structure de l'interface

L'IHM permet de naviguer entre les applications (horloge, capteurs, réglages, activité) via des menus graphiques. Les écrans dédiés aux capteurs (graphe d'accélération, boussole, météo locale) et les réglages (choix du cadran, heure, format de date, unités) sont gérés par des widgets LVGL et des événements tactiles (boutons virtuels, sliders).

5.3 Affichage des données

Les données des capteurs (heure, température, humidité, pression, pas, direction) et les notifications BLE sont affichées sur l'écran. L'architecture sépare la logique d'acquisition, le traitement des données et la couche d'affichage pour une maintenance facilitée.

5.4 Machine d'états de l'IHM

La machine d'états de l'interface décrit les écrans et les transitions (tactile, boutons) entre horloge, menus, capteurs et réglages. Le schéma ci-dessous illustre la ME de l'IHM.

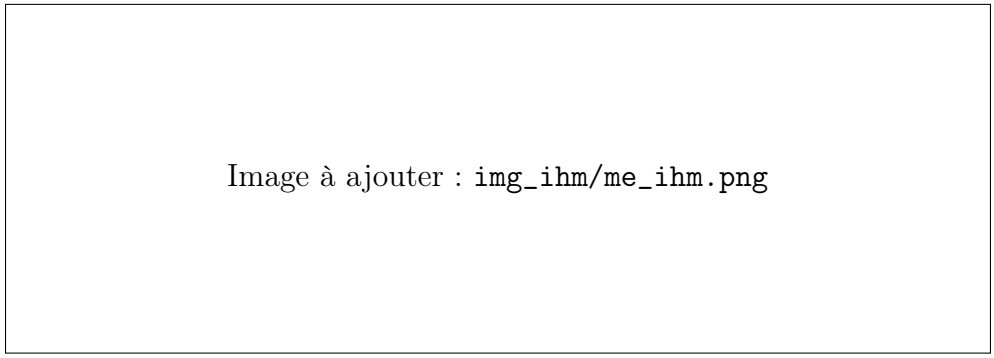

FIGURE 5.1 – Machine d'états de l'IHM

Gestion de projet

5.5 Composition de l'équipe

Groupe TimeX : Rafelimanana Deraniaina (Project owner), Khanh-Phuong NGUYEN, HAJJAJ Houda, EL KILI Rim.

5.6 Répartition des tâches

La répartition des tâches proposée pour le groupe est la suivante :

5.6.1 Membre A : Plateforme et OS

- Configuration Zephyr (devicetree, overlays, Kconfig) pour le nRF5340 DK, l'écran et le module RTC.
- Intégration des drivers de base (display, input, RTC, GPIO).
- Mise en place de la machine d'états générale et des threads Zephyr associés.

5.6.2 Membre B : Capteurs et BLE

- Intégration du module `sensor` de Zephyr avec le shield X-NUCLEO-IKS01A3 (LSM6DSO, LIS2MDL, capteurs environnementaux).
- Traitement des données (podomètre, boussole, température, etc.).
- Mise en œuvre de la communication BLE (services, caractéristiques, protocole d'échange avec le smartphone).

5.6.3 Membre C : Interface graphique et UX

- Conception des écrans avec SquareLine Studio (watchfaces, menus, écrans capteurs, réglages).
- Intégration LVGL dans Zephyr et gestion des événements tactiles.
- Affichage des notifications et des données de capteurs dans l'interface.

Une phase commune de tests et d'intégration globale est prévue, impliquant tous les membres.

5.7 Suivi des tâches

Le tableau ci-dessous permet de suivre l'avancement par membre et par tâche (à compléter en cours de projet).

TABLE 5.1 – Suivi de l'avancement des tâches par membre

Membre	**Tâche**	**Fait**	**Implanté**	**Validé**
Membre A	Config. Zephyr (devicetree, overlays, Kconfig)	–	–	–
Membre A	Drivers de base (display, input, RTC, GPIO)	–	–	–
Membre A	Machine d'états et threads Zephyr	–	–	–
Membre B	Module sensor + shield IKS01A3	–	–	–
Membre B	Traitement données (podomètre, boussole, etc.)	–	–	–
Membre B	Communication BLE	–	–	–
Membre C	Écrans SquareLine Studio (watchfaces, menus)	–	–	–
Membre C	Intégration LVGL et tactile	–	–	–
Membre C	Affichage notifications et données capteurs	–	–	–
Tous	Tests et intégration globale	–	–	–
Tous	Rédaction du rapport	–	–	–

Chapitre 6

Conclusion

Ce projet de montre connectée basée sur le nRF5340 et Zephyr RTOS nous a permis d'explorer une architecture complète d'objet connecté, depuis les drivers bas niveau jusqu'à l'interface utilisateur graphique et à la connectivité Bluetooth. En nous inspirant de ZSWatch et en exploitant le shield de capteurs X-NUCLEO-IKS01A3, nous avons pu définir un ensemble de fonctionnalités réalistes, tout en respectant les contraintes de temps et de complexité du module.

La démarche imposée (machine d'états, structuration en modules, utilisation rigoureuse de l'écosystème Zephyr) prépare à la conception de systèmes embarqués plus complexes et constitue une base solide pour de futurs développements dans le domaine des wearables et de l'IoT.