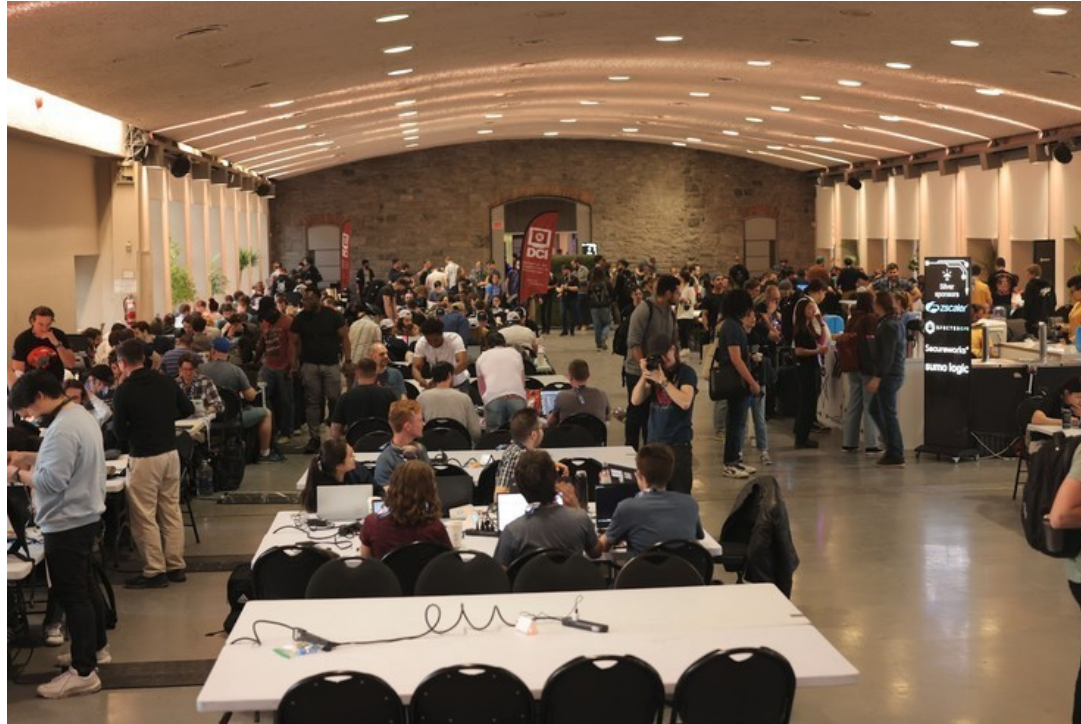


Transformer un appareil de compétition en jouet

Commencer

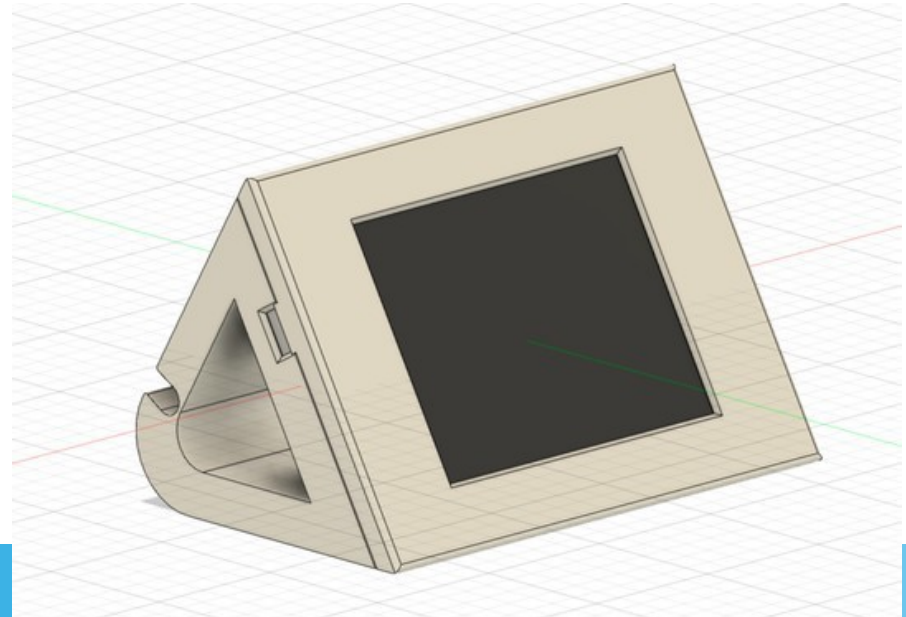


- Dispositif remis lors de la compétition North Sec CTF en 2023 nsec.io



Compétition ? NSEC ?

- Qu'est-ce que Capture The Flag ?
 - Jeu en équipe pendant une fin de semaine
 - Objectif : Gagner des points en piratant des logiciels ou du matériel pour récupérer un drapeau textuel. *FLAG-C'EST-UNE-DEMO*
- Utilisation de ce dispositif
 - Permet d'afficher des informations à l'aide d'un réseau maillé
 - Nombreux drapeaux disponibles
 - DEL de couleur adapté au badge de la compétition



La carte

- L'appareil est le `ESP32-2432S028R`, connu sous le nom de ESP Cheap Yellow Display.
- Processeur `ESP32-D0WDQ6` avec wifi et bluetooth.
- Logiciel de développement ESP-IDF `Espressif IoT Development Framework`
- Contrainte : L'appareil a été flashé à l'aide d'un démarrage sécurisé par l'équipe NSEC.
 - Ils ont publié le code source avec les clés de démarrage sécurisées et des outils de construction.
 - Il n'est pas réversible, car lorsqu'il est activé, il brise un fusible pour éviter le retour en arrière.
 - Langage C++ et interface utilisateur avec LVGL v7
 - Système d'exploitation en temps réel avec FreeRTOS

Enquête sur l'article de réemploi

- **ÉCHEC** : Évaluation l'utilisation pour d'autres choses et annulé car nécessite de nombreuses recherches pour utiliser le démarrage sécurisé comme pour :
 - ESPHome
 - Développement de l'ESP
 - Le démarrage sécurisé sur ESP ne peut pas être désactivé
- **Nouvel objectif** : Créer un jouet
 - Idée : Score de jeu pour les figurines
 - Date d'achèvement prévue : 1 mois
 - Temps réel : 4 mois quelques heures le week-end

Comment cela s'est passé

Prendre le code source de l'équipe NSEC et le modifier

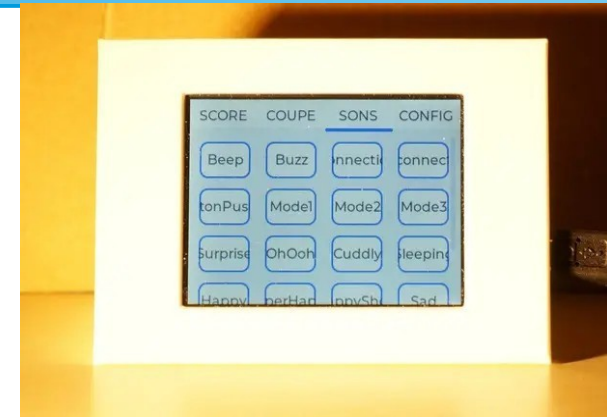
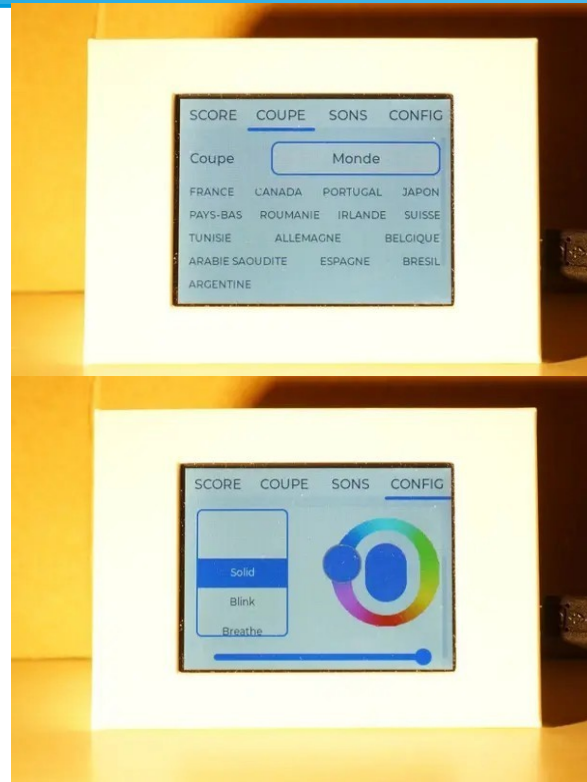
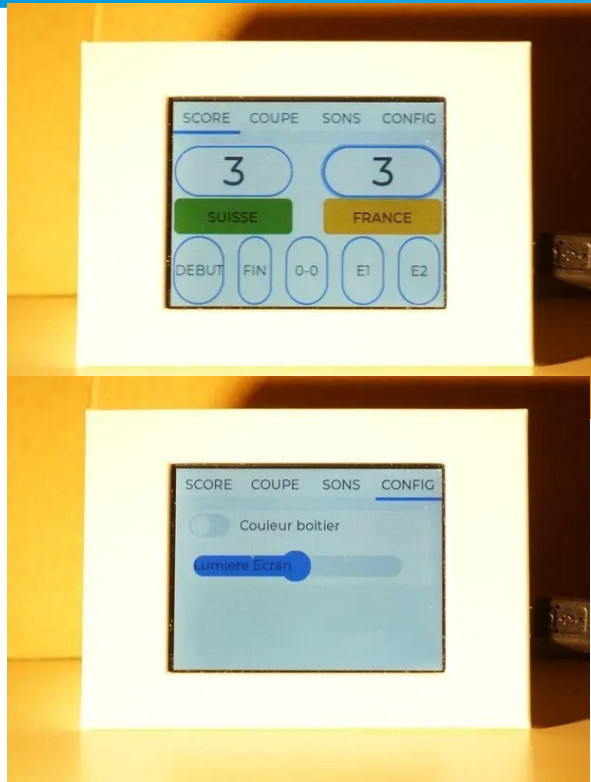
- Langue C++ avec ESP IDF
- Mise à jour des scripts ne fonctionnait pas par défaut
- Reconstruction et flashage pour s'assurer que tout fonctionne
- Retirer les composants non utilisés (Wifi, bluetooth, carte SD, ...)
- Construire une nouvelle interface utilisateur
 - Modifier l'existant en essayant et en échouant
 - Utiliser un simulateur en langage C
 - Adaptation à du style

Ce qu'il contient / caractéristiques*

- Score
- Couleur composants
- Sélection des équipes
- Sons
- Coupes avec différentes équipes
- Utilisation d'une LED embarquée avec sélection de couleur (déjà disponible)
- Rétro-éclairage de l'écran pour préserver la batterie
- Quelques paramètres

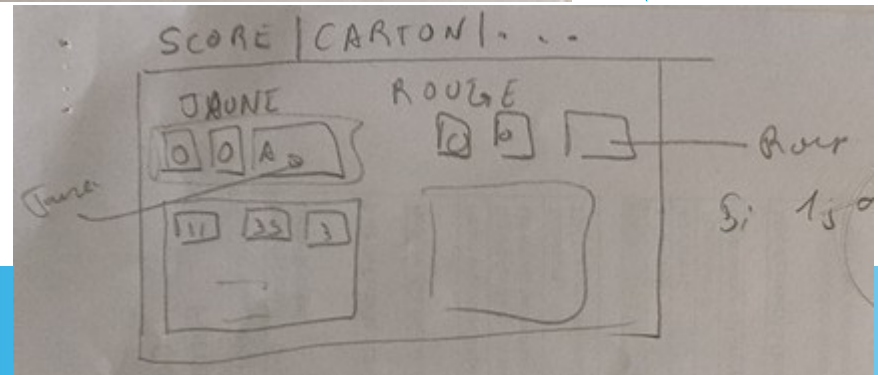
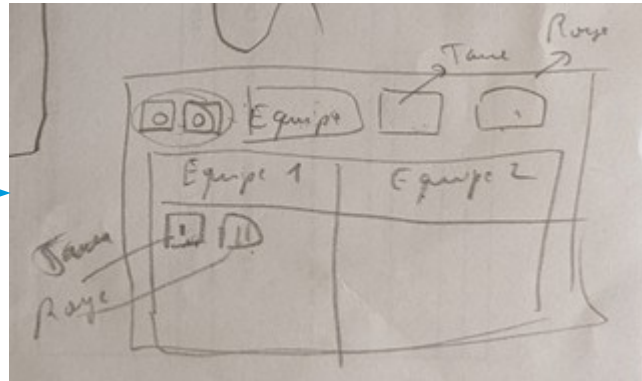
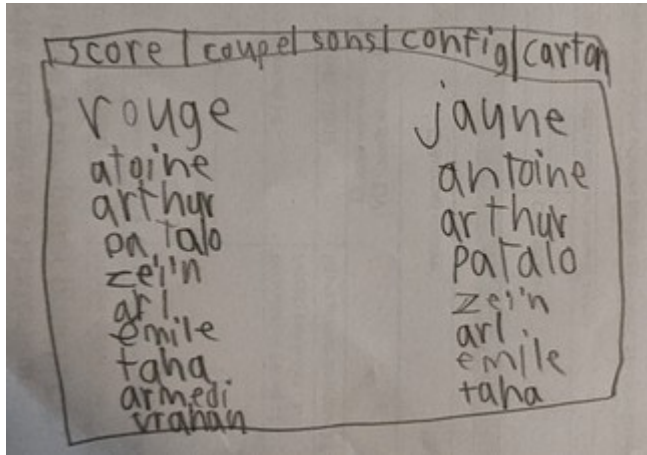
DEMO

DEMO



Dernière caractéristique

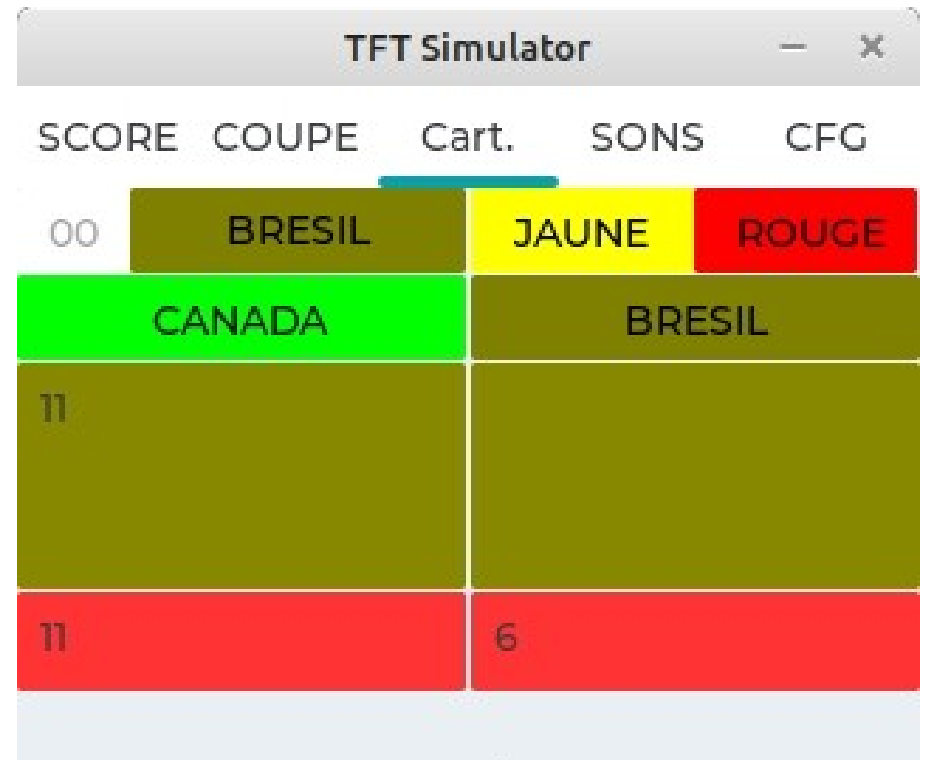
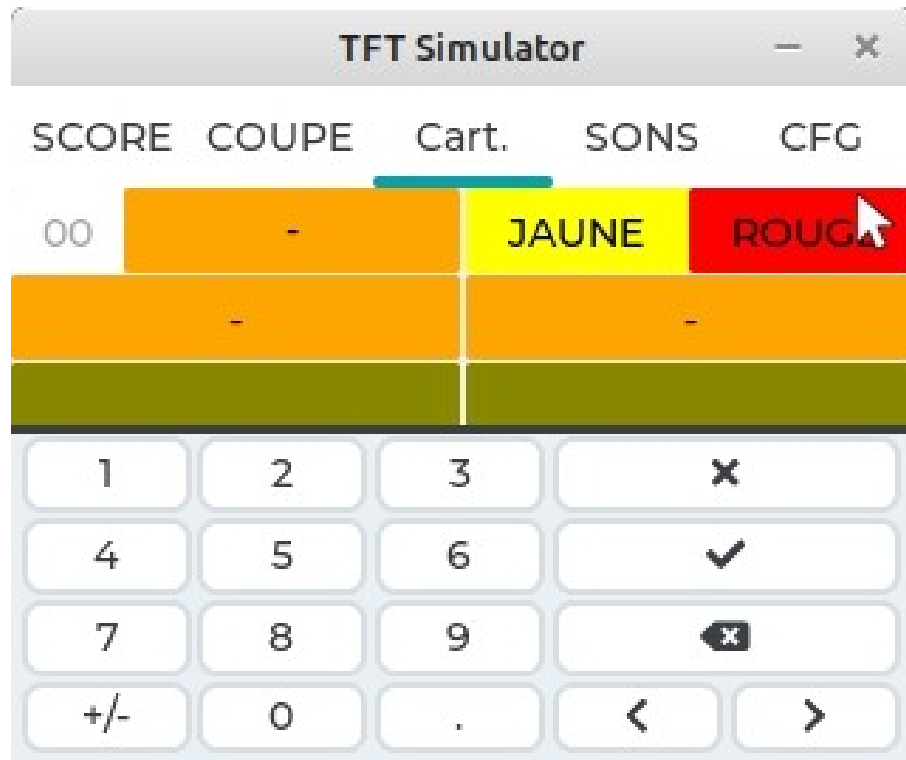
En septembre, pendant un mois, ajout de cartes jaunes et rouges Travailler avec mon enfant sur le design



Défi

- Utiliser LVGL pour afficher exactement à une position de l'écran
- Pas de bordures pour optimiser l'espace de l'écran
- Se souvenir de la structure du code
- Stocker les réglages en mémoire non volatile

Échantillons



Ce que j'ai appris

- La bibliothèque LVGL pour les appareils embarqués est simple et facile à utiliser
- Le C++ n'est pas aussi simple que je le pensais
 - Seed pour la génération de nombres aléatoires
 - Bibliothèques ESP modifiées par Espressif
 - Mémoires en C et C++ avec #IFDEF pour traiter le même code
 - ESP IDF met en place beaucoup d'actions de compilation
- Le code final n'est pas optimal ni même maintenable pour un produit réel
 - **C'EST UN JOUET !**
- Chose non réalisé : Supprimer certaines marges et bordures

La fin

GitHub : <https://github.com/jingl3s/nsec-badge-display-other>