intégrer l'IA périphérique avec l'ESP32-P4

Anant Gupta, Sun Xiangyu, et Xie Wei (Espressif)

L'ESP32-P4 est le dernier-né de la famille des SoC haute performance d'Espressif, conçu pour révolutionner le secteur des systèmes embarqués. Cette puissante puce est dotée de fonctionnalités avancées, notamment de solides capacités d'intelligence artificielle, ce qui en fait un choix idéal pour les développeurs désireux de concevoir des appareils connectés et intelligents. Dans cet article, nous explorerons en détail les capacités d'IA de l'ESP32-P4, les bibliothèques d'IA qu'elle prend en charge et présenterons un exemple concret d'application qui illustre son immense potentiel.

L'ESP32-P4 d'Espressif est un microcontrôleur à double cœur basé sur un processeur RISC-V capable d'atteindre une vitesse d'horloge allant jusqu'à 400 MHz. Comme le montre le schéma fonctionnel de la **figure 1**, l'ESP32-P4 dispose d'un sous-système de mémoire très flexible et adaptable, doté de 768 Ko de SRAM intégrée, 8 Ko de RAM TCM à latence nulle, et une PSRAM extensible. La puce intègre également une large gamme de périphériques, notamment SPI, I2S, I2C, LED PWM, MCPWM, RMT, ADC, UART et TWAI. Il offre également des fonctionnalités avancées pour les interfaces homme-machine telles que MIPI-CSI avec ISP intégré, MIPI-DSI, 14 entrées tactiles

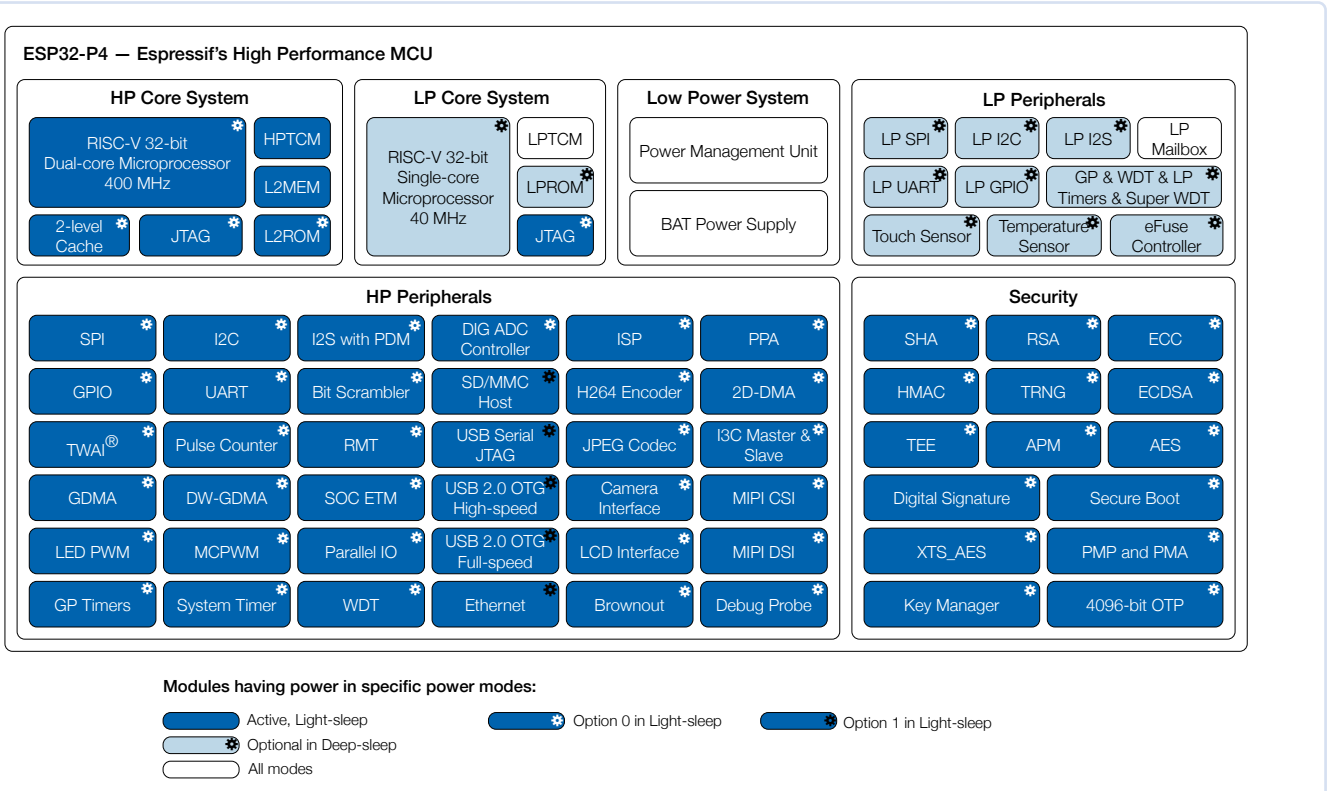

Figure 1. Schéma fonctionnel de l'ESP32-P4.

capacitives, etc. Il est équipé d'accélérateurs matériels notamment l'accélérateur de traitement de pixels (PPA), l'encodage H.264 et le 2D-DMA. Grâce à ses hautes performances et sa connectivité E/S complète, l'ESP32-P4 promet de transformer radicalement le domaine des systèmes embarqués.

Capacités IA de l'ESP32-P4

L'ESP32-P4 est conçue pour prendre en charge une vaste gamme d'applications d'IA, des modèles simples d'apprentissage automatique aux algorithmes complexes d'apprentissage profond. Les capacités d'IA de la puce reposent sur son processeur RISC-V à double cœur, qui constitue une base solide pour le traitement de l'IA. L'ESP32-P4 est équipée d'un ensemble d'accélérateurs matériels et d'instructions optimisées pour l'IA, qui facilitent les calculs complexes, ce qui en fait un choix particulièrement adapté aux applications exigeant un traitement d'IA de haute performance.
L'une des principales caractéristiques des capacités d'IA de l'ESP32-P4 est sa compatibilité avec l'extension *Xai* du jeu d'instructions RISC-V. L'extension Xai fournit une gamme d'instructions spécifiquement conçues pour les applications d'IA et d'apprentissage automatique, notamment les :

- *Instructions vectorielles* : elle permettent à la puce de traiter de grands vecteurs de données, une fonctionnalité essentielle pour de nombreux algorithmes d'IA et d'apprentissage automatique.
- *Instructions matricielles* : elles permettent à la puce d'effectuer des opérations sur de grandes matrices de données, ce qui est une exigence commune à de nombreux algorithmes d'apprentissage profond.

L'extension Xai offre également une gamme d'autres instructions spécifiquement conçues pour les applications d'IA et d'apprentissage automatique, notamment des instructions pour la manipulation et le déplacement de données vectorielles alignées et non alignées, l'arrondi configurable et les modes de saturation.

ESP-SR : cadre de reconnaissance vocale

ESP-SR est un environnement complet destiné aux développeurs souhaitant intégrer la reconnaissance vocale dans leurs applications grâce à l'IA. Ce cadre inclut divers modules, notamment le traitement audio frontal, la détection des mots de réveil et la reconnaissance des commandes vocales. Avec ESP-SR, les développeurs peuvent créer des applications capables de reconnaître et de répondre aux commandes vocales, ce qui en fait un choix idéal pour des applications telles que la domotique intelligente et les assistants vocaux (**figure 2**).
Le cadre ESP-SR offre plusieurs fonctionnalités clés :

- *Traitement audio frontal (alogrithmes AFE)* : ce module offre un ensemble d'API pour le traitement audio, incluant la suppression du bruit profond, l'annulation d'écho et la séparation aveugle des sources.
- *Détection des mots de réveil (WakeNet)* : ce module permet aux développeurs de détecter les mots de réveil, tels que « Alexa-ou» «OK Google", ou tout autre mot de réveil personnalisé, et de déclencher des commandes personnalisées.
- *Reconnaissance des commandes vocales (Mul)* : ce module propose un ensemble d'API pour la reconnaissance vocale, permettant aux développeurs de créer des applications capables de reconnaître et de répondre aux commandes vocales.

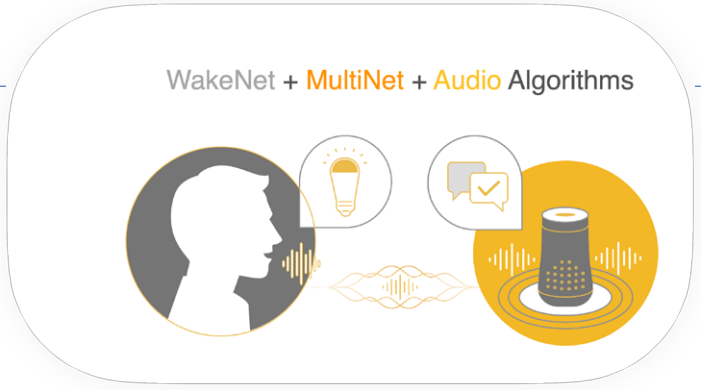

Figure 2. ESP-Speech Recognition.

ESP-DL : bibliothèque d'apprentissage profond

ESP-DL est une bibliothèque d'apprentissage profond qui propose des API pour l'inférence de réseaux neuronaux, le traitement d'images et les opérations mathématiques (**figure 3**). Avec ESP-DL, les développeurs peuvent déployer leurs modèles d'apprentissage profond sur l'ESP32-P4, ouvrant ainsi la porte à un large éventail d'applications utilisant l'IA.
La bibliothèque ESP-DL offre de nombreuses fonctionnalités :

Format de modèle standard ESP-DL

Ce format binaire utilisé pour stocker le graphique du modèle, les poids et d'autres informations. Ce format est similaire au format de modèle ONNX, tout en substituant les Protobuf par des FlatBuffers, ce qui rend nos modèles plus légers et permet une désérialisation directe. Donc, l'accès aux données sérialisées ne nécessite pas de les copier dans une partie distincte de la mémoire. L'accès aux données devient significativement plus rapide par rapport aux formats exigeant une manipulation plus complexe, tels que les Protobuf.

Implémentation efficace et précise des opérateurs

Grâce à des directives d'IA, nous avons implémenté efficacement les opérateurs IA courants, tels que Conv2d, Pool2D, Gemm, Add, Mul, etc. Parallèlement, nous avons corrigé les erreurs de précision des opérateurs 8 bits de versions antérieures. Nous avons implémenté l'opérateur PyTorch afin de garantir que les résultats obtenus par notre outil de quantification soient cohérents avec les résultats obtenus sur ESP-DL.

Planificateur de mémoire statique

Le planificateur de mémoire statique évalue la taille de mémoire maximale requise et les décalages de mémoire de chaque variable en fonction de la séquence topologique des opérateurs. Cela permet d'éviter la surcharge de temps et la fragmentation potentielle de la mémoire causée par l'allocation de la mémoire pendant l'exécution du modèle. Nous avons conçu un nouveau planificateur de mémoire statique pour la structure de mémoire RAM/PSRAM interne commune. Étant donné que la RAM interne a une vitesse d'accès plus rapide mais une capacité limitée, nous fournissons une API qui permet aux utilisateurs de personnaliser la taille de la RAM interne que le modèle peut utiliser. Le planificateur de mémoire répartit automatiquement les différentes couches dans les zones de mémoire optimales de la mémoire en fonction de la taille de la RAM interne spécifiée par l'utilisateur, maximisant ainsi l'efficacité de l'exécution tout en minimisant l'utilisation de la mémoire.

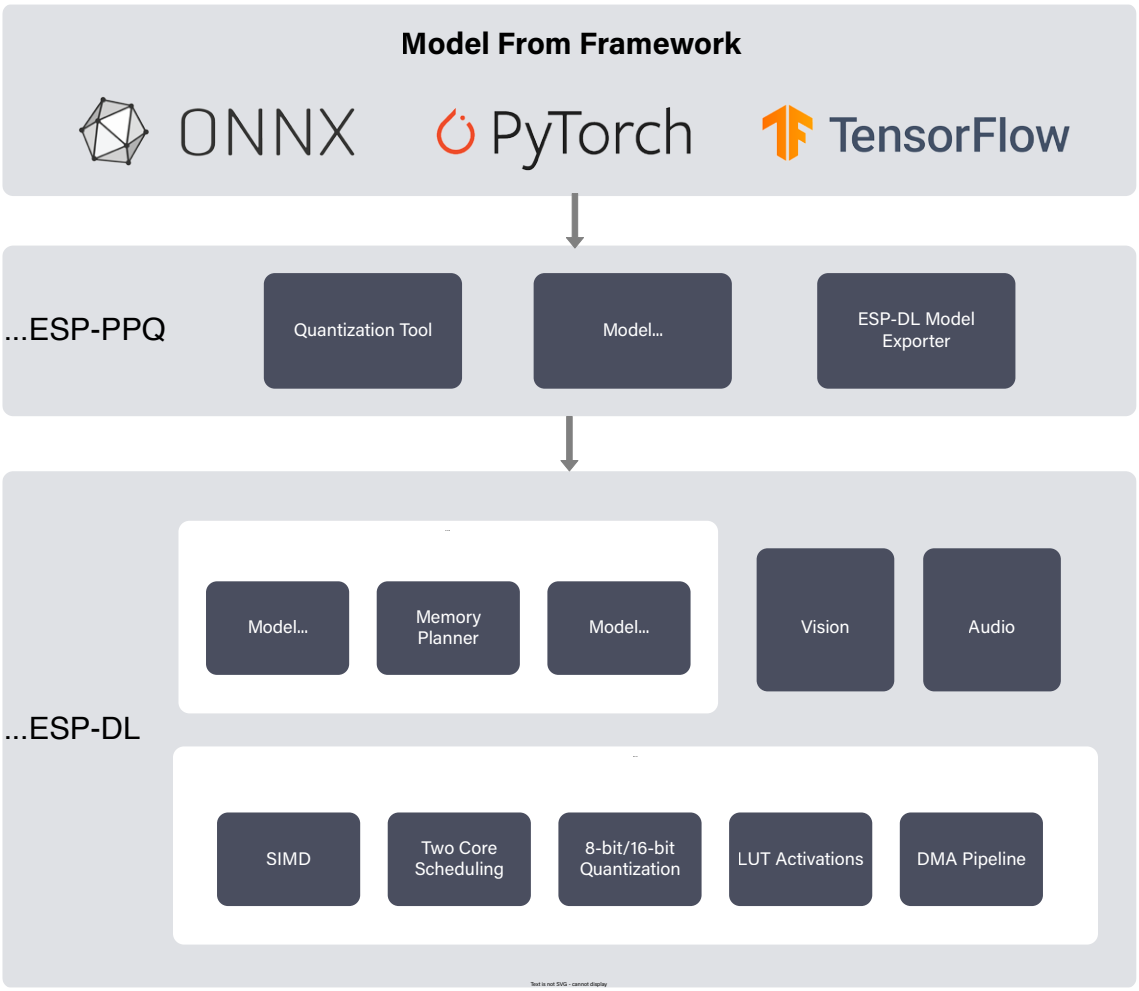

Figure 3. Framework ESP-DL.

› Planification double cœur

Nous avons ajouté une planification automatique à double cœur pour certains opérateurs à grande intensité de calcul, permettant ainsi aux modèles d'utiliser pleinement la puissance de calcul des architectures à double cœur. Actuellement, Conv2D et DepthwiseConv2D prennent en charge cette planification. Voici quelques-uns de nos résultats expérimentaux : pour les couches à forte intensité de calcul, l'utilisation d'un double cœur peut réduire le temps de moitié.

Cas d'utilisation : classification d'images

L'ESP32-P4, grâce à ses capacités avancées d'intelligence artificielle, se révèle être une plateforme idéale pour la classification d'images. Grâce à la bibliothèque ESP-DL, les développeurs peuvent implémenter des modèles d'apprentissage profond capables de détecter et de classifier différents objets à partir d'images. Cette application présente un large éventail d'utilisations potentielles, des systèmes de sécurité à la domotique intelligente. Grâce à la haute performance de l'ESP32-P4, ses capacités d'IA et à sa connectivité E/S étendue, les développeurs peuvent créer un système de classification d'images à la fois précis et efficace.

Pour implémenter la classification d'images sur l'ESP32-P4, les développeurs peuvent utiliser la bibliothèque ESP-DL pour déployer des modèles d'apprentissage profond pré-entraîné comme le MobileNet V2. Le modèle est entraîné sur un ensemble de données d'images. Il est donc capable de détecter et de classer différentes images. Grâce au moteur PPA et 2D-DMA de l'ESP32-P4, il est possible d'accélérer significativement le traitement et la gestion du flux vidéo, permettant à la puce d'effectuer des traitements complexes d'IA en temps réel.

Nous utiliserons la carte ESP32-P4-Function-EV-Board illustrée à la **figure 4**. Cette carte de développement multimédia est basée sur la puce ESP32-P4. et est compatible avec USB2.0, MIPI-CSI, MIPI-DSI et plusieurs autres périphériques. Grâce à toutes ces caractéristiques exceptionnelles, cette carte constitue un choix idéal pour le développement de produits audio et vidéo connectés, à faible coût, à hautes performances et à faible consommation d'énergie.

Portage de MobileNet V2 sur ESP32-P4 avec ESP-DL

MobileNet V2 est un modèle d'apprentissage profond léger, adapté aux performances sur les appareils mobiles et embarqués. Il est largement utilisé dans divers domaines, tels que la classification d'images, la détection d'objets, la reconnaissance faciale, entre autres. Utilisons MobileNet V2 comme exemple pour comprendre comment déployer le modèle avec ESP-DL.

1. **Préparer un modèle pré-entraîné** : nous pouvons directement obtenir un modèle pré-entraîné via l'interface PyTorch et exporter le modèle vers un fichier ONNX pour le processus de quantification.

```
torchvision.models.mobilenet.
mobilenet_v2(pretrained=True)
torch.onnx.export(
    model=model,
    args=tuple(
        [
            torch.zeros(
                size=[1] + input_shape[1:],
                device=self.device_str,
```

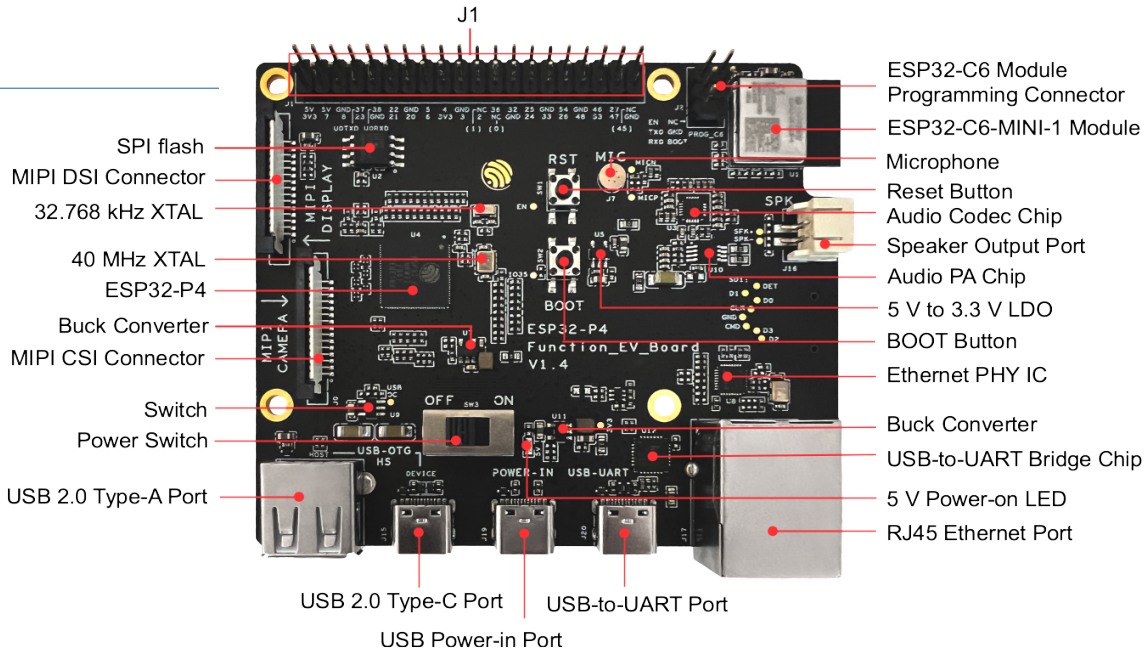

Figure 4. ESP32-P4-Function-EV-Board.

```
            dtype=self.input_dtype,
        )
        for input_shape in self.input_shape
    ]
),
f=orig_onnx_path,
opset_version=11,
do_constant_folding=True,
)
```

2. **Préparer un jeu de données de calibration et quantifier le modèle** : sur la base de l'outil de quantification ESP-PPQ, la fonction encapsulée `quantize_model_wrapper` permet de charger le fichier ONNX de l'étape 1 et d'effectuer la quantification à partir du jeu de données de calibration. Une fois la quantification terminée, exportez le modèle pour obtenir un fichier avec le suffixe *espdl*.

```
ppq_graph, executor = quantize_model_wrapper(
            onnx_import_file=orig_onnx_path,
            calib_dataloader=self.cali_iter,
            calib_steps=self.calib_steps,
            input_shape=self.input_shape,
            platform=self.platform,
            input_dtype=self.input_dtype,
            dispatching_override=None,
            dispatching_method="conservative",
            collate_fn=collate_fn,
            device=self.device_str,
            verbose=1,
        )

PFL.Exporter(platform=self.platform).export(
    file_path=export_onnx_path,
    graph=ppq_graph,
    config_path=export_config_path,
    modelVersion=self.model_version,
    valuesForTest=valuesForTest,
)
```

3. **Déploiement du modèle** : chargez le fichier de modèle *espdl* dans la mémoire flash. Lors du chargement avec ESP-DL, celui-ci alloue automatiquement la mémoire nécessaire aux processus intermédiaires. À ce stade, transmettez les données d'entrée à l'interface *Model::run*. Les résultats de l'inférence finale peuvent être récupérés grâce à la bibliothèque d'opérateurs d'accélération d'ESP-DL.

```
Model *model = new Model("model",
fbs::MODEL_LOCATION_IN_FLASH_PARTITION);
std::map graph_test_inputs = get_graph_test_inputs(model);
model->run(graph_test_inputs);
std::map outputs = model->get_outputs();
```

L'avenir de l'IA embarquée avec l'ESP32-P4

L'ESP32-P4 se distingue comme un puissant SoC prêt à révolutionner le domaine des systèmes embarqués. Doté de capacités d'IA avancées, d'une connectivité E/S complète, et supportant des bibliothèques d'IA telles que ESP-SR et ESP-DL, ce SoC est parfaitement adapté aux développeurs désireux de concevoir des dispositifs intelligents et interconnectés. Le cas d'utilisation de la classification d'images avec MobilieNet V2 illustre le potentiel de l'ESP32-P4 pour exploiter les capacités d'IA et comment un modèle pré-entraîné peut être efficacement porté via le framework ESP-DL, démontrant comment la puce peut être utilisée pour créer des systèmes précis et efficaces alimentés par l'IA. À mesure que le domaine de l'IA continue de se développer, l'ESP32-P4 est destiné à jouer un rôle essentiel dans l'évolution future des systèmes embarqués.

240568-04

Questions ou commentaires ?
Contactez Elektor (redaction@elektor.fr).

produits

- **Elektor Special: Espressif Guest-Edited Edition 2023**
  www.elektor.com/EP-0526
- **Elektor Special: Espressif Guest-Edited Edition (PDF)**
  www.elektor.com/ED-0526