

M4 + 2xA7 + GPU : équipe de rêve ou presque !

Nouveau SoC STM32MP1 : pour vos exigences les plus élevées

Viacheslav Gromov (leser@gromov.de)

Avec la multiplication d'applications telles que l'Internet des Objets, la cryptographie, et les interfaces personne-machine (IHM) de plus en plus complexes, même un ARM-Cortex-M7 tutoie ses limites. Des fabricants renommés proposent des processeurs multicoeurs hétérogènes (PMH), qui regroupent dans un même boîtier des gros cœurs applicatifs pour l'IHM et des cœurs plus petits pour les applications en temps réel. Procédons à un examen détaillé des premiers représentants de PMH de STMicroelectronics particulièrement intéressants du point de vue du prix et de la documentation. Parmi les avantages substantiels de cette famille MP1, on trouve son rendement, une GPU-3D, le support de Linux et d'Android, et une sécurité accrue, entre autres par le cloisonnement interne en zones sécurisées.

La plupart des thèmes de cet article (bien trop bref) sont inépuisables, il ne peut donc offrir qu'un premier contact avec le monde des PMH. Il s'adresse avant tout aux développeurs avancés du domaine ARM-Cortex-M et de Linux embarqué avec une première expérience approfondie du Raspberry Pi.

Tableau 1. Signes distinctifs des membres de base de la famille MP1. Il existe les variantes A et C, sans ou avec unité matérielle intégrée de cryptage.

Désignation	Cortex-A7 (650 MHz)	Cortex-M4 (209 MHz)	Vivante 533 MHz avec MIPI-DSI	FD-CAN
STM32MP151	1	1	0	0
STM32MP153	2	1	0	2
STM32MP157	2	1	1	2



(source: Viacheslav Gromov)

Membres et structure de la famille MP1

Le **tableau 1** donne un aperçu de tous les représentants de la famille MP1. Ils se distinguent surtout par la configuration du noyau ainsi que par la présence d'une unité graphique (GPU). Le membre le plus complet, le *STM32MP157C*, possède les trois cœurs (les siamois ARM-Cortex-A7 avec leur cache commun L2, et l'ARM-Cortex-M4), une GPU-3D avec interface série (DSI), une unité de cryptage et une interface FD-CAN (**fig. 1**).

Du fait de la présence du cœur A7 sur le silicium, la fabrication de la puce complète, y compris les cœurs M4, n'est possible qu'en technologie CMOS, ce qui contraint à en exclure les grosses mémoires. Il y a suffisamment de possibilités de connecter de la mémoire RAM ou Flash par (LP)DDR ou Dual-Mode-Quad-SPI ou SDMMC. On peut connecter jusqu'à 1 Go de RAM externe et une mémoire Flash dont la taille n'est limitée que par l'espace adressable.

La technologie des zones de confiance (*TrustZone*) de ARM procure, grâce au contrôleur associé, des domaines sécurisés dans la mémoire, qui ne peuvent être lus ou écrits que par des parties de programmes spécifiées. Il y a d'autres domaines, comme les contrôleurs d'alimentation et d'horloge, qui bénéficient de cette technologie. Grâce à l'archi-

ture ARM-NEON-SIMD, les coeurs A7 ont accès à un jeu d'instructions étendu spécifique aux fonctions telles que les calculs Quad-MAC, p. ex. pour les applications multimédias.

L'horloge du bus interne de 64 bits AXI (*Advanced eXtensible Interface*) des noyaux A7 et des mémoires atteint 266 MHz. Le noyau M4 est connecté au bus matriciel AHB (*Advanced High-performance Bus*) à 209 MHz, et la plupart des périphériques aux bus APB (*Advanced Peripheral Bus*). Par la matrice de liaison AXI-AHB, il est possible d'assigner les périphériques, entre autres, à des domaines particuliers du noyau ou au domaine de sécurité A7, seulement accessible à des parties de programme particulières (fig. 2). Les domaines de mémoire banalisés bénéficient d'une fonction comparable.

À sa fréquence d'horloge maximale de 533 MHz, la GPU-3D *Vivante* traite un flux de 133 mégapixels par seconde, ce qui lui permet de supporter non seulement le très répandu *OpenGL 2.0* dans sa version embarquée, mais aussi *OpenVG 1.1* ou *EGL 1.4* et d'autres. La résolution des interfaces graphiques associées pour afficheurs LCD-TFT ou DSI atteint 1366 x 768 pixels (*Wide eXtended Graphics Array, WXGA*) à 60 images par seconde.

Par (M)DMA et les 288 interruptions ou événements globaux (partagés), de nombreux processus peuvent s'exécuter pratiquement indépendamment et sans la participation de coeurs particuliers. Dans le large domaine du traitement numérique du signal (DSP), on notera l'intérêt du *Digital Filter for Sigma Delta Modulators* (DFSDM) qui peut travailler avec des valeurs brutes de convertisseurs analogique-numérique de 24 bits. On peut ainsi connecter directement un microphone MEMS avec son convertisseur sigma/delta.

Passionnantes sont aussi l'interface *Consumer-HDMI* (d'après la spécification *CEC 1.4*), l'interface caméra d'une largeur de 14 bits et d'une vitesse de 140 Mo/s, l'identifiant unique à 96 bits avec ses coordonnées de galette et autres, et le capteur de température à sortie en fréquence (-40 à +125 °C).

Il s'y ajoute bien entendu quelques organes de sécurité, p. ex. ceux qui peuvent fournir des codages AES-256, réputés «sûrs» (*Cryptocontroller + Hash-Elements*). Le générateur de nombres aléatoires peut ainsi

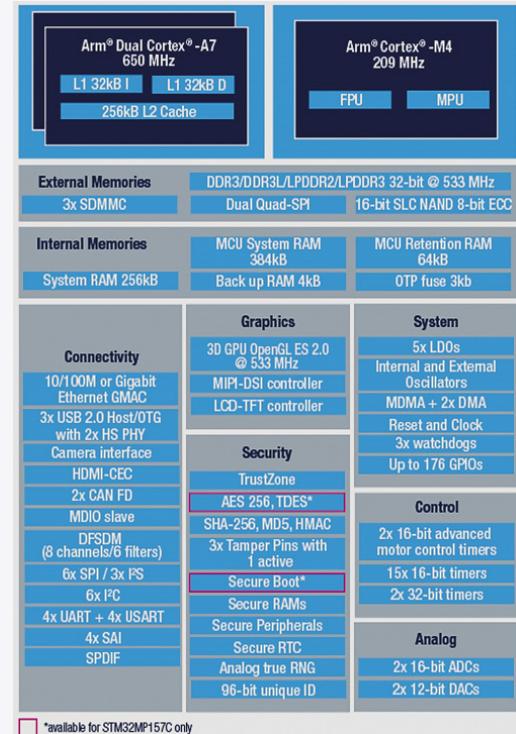


Figure 1. Vue d'ensemble des organes de calcul y compris la liste sommaire des périphériques, de la sécurité et des interfaces du plus important membre de la famille, le STM32MP157 (source : ST).

produire, à partir de *semences* analogiques, quatre nombres de 32 bits réputés aléatoires en au plus 213 cycles d'horloge. Le *tamper*, connu au moins depuis le STM32H7, peut mémoriser en toute sécurité 32 registres de sauvegarde dans tous les modes de fonctionnement et les effacer immédiatement à la demande en cas d'événements particuliers comme l'interruption d'un circuit entre broches spécifiées.

La communication entre les deux mondes de coeurs peut s'effectuer par l'*Inter Processor Communication Controller* (IPCC) en mode simplex

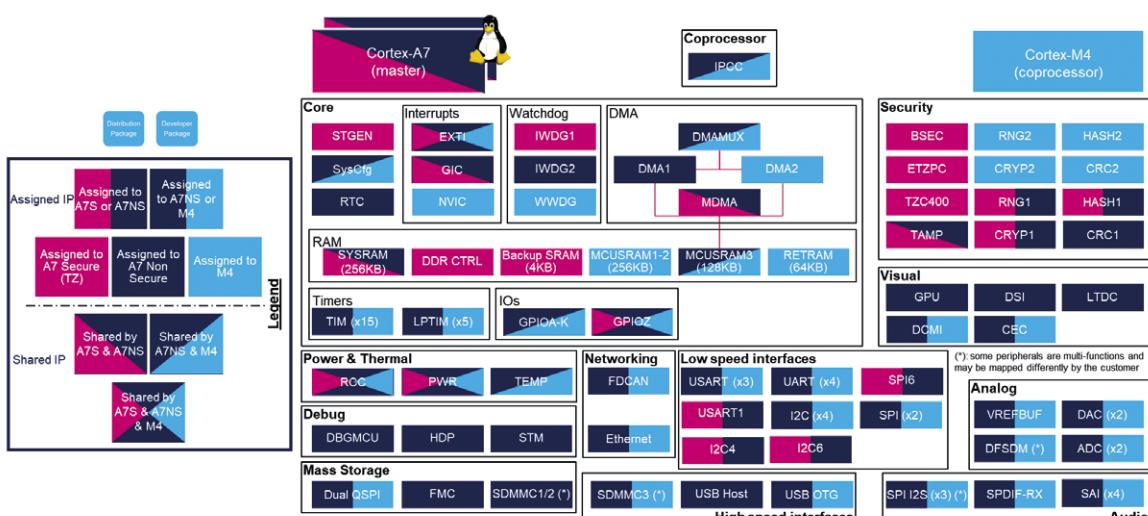


Figure 2. Les différentes possibilités d'affectation aux coeurs et à la SecureZone des périphériques, des zones mémoire et autres fonctions internes sont ici distinguées par des couleurs (source : ST).

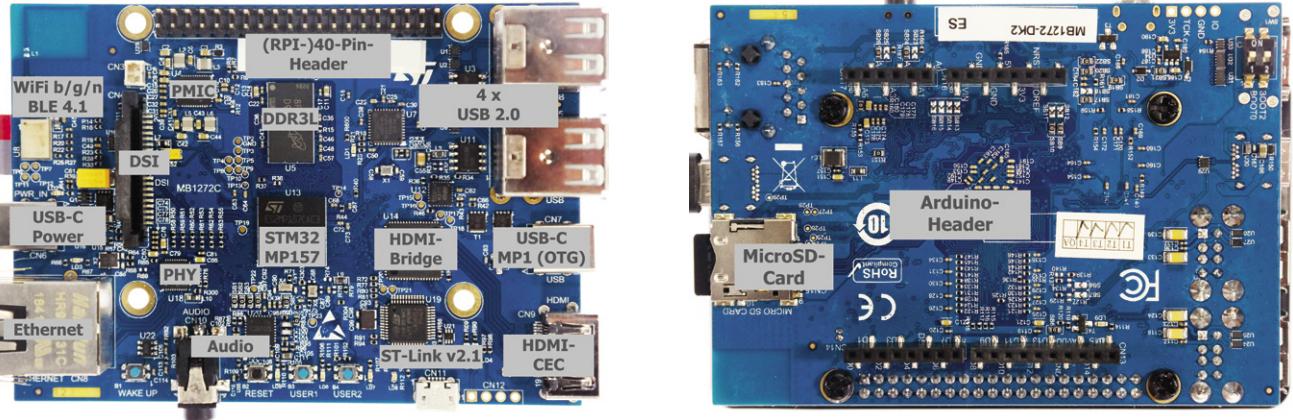


Figure 3. Les deux faces de la carte des STM32MP157C-DK2 avec identification des composants essentiels, mais sans l'afficheur (source : Viacheslav Gromov).

ou half-duplex par deux canaux, ou par l'intermédiaire d'une zone de mémoire partagée SRAM (*Shared Memory*, AXI: 256 ko), ce qui permet l'échange interne de données par un système de boîtes-aux-lettres. On dispose de nombreux moyens de débogage, invasifs ou non (JTAD/ SWD), ainsi que de possibilités de traçage (broches du TPIU), y compris les interfaces associées.

Cet article ne se prête pas à une description exhaustive de la pléthora de périphériques, mais il existe une riche documentation. Un *Reference Manual* de plus de 4000 pages (!), une feuille de caractéristiques condensée sur 260 pages et des notes d'applications logicielles, p. ex. la logique d'interaction avec l'USB du chargeur d'amorçage ; tout est téléchargeable [1]. On retrouve beaucoup des informations mentionnées ici sous la forme d'un cours en ligne [2] et, présentées de manière plus pratique, sur la page officielle ST-YouTube.

L'alimentation du STM32MP1, des coeurs et des périphériques analogiques requiert la présence de tensions variées sur de multiples broches, ces tensions variant aussi selon le mode d'économie d'énergie utilisé ! Ne vous tracassez pas, c'est le boulot du STPMIC1, une puce de gestion de la consommation (*Power Management IC, PMIC*), connectée aux broches d'alimentation du MP1, ainsi qu'à d'autres broches, dont une de r.-à-z. (*reset*), une de réveil (*wake-up*) et une interface I²C. Cette puce peut aussi fournir l'alimentation d'une mémoire DDR. Selon le mode d'économie d'énergie actif, la puissance consommée par le MP1 peut passer d'un maximum de 487 mW (les trois coeurs, plus tous les périphériques, plus la GPU), à 92 mW si seul le M4 est actif, et à quelques µW dans le mode stop/standby. Le prix unitaire du MP1 se situe entre 5 et 10 €, selon la version et la quantité.

LE MÊME SILICIUM, PAS LE MÊME PRIX

Si l'on achète un petit modèle du MP1, cela n'implique pas que la puce ne contient que les coeurs et autres fonctions mentionnés dans le tableau. Pour la fabrication en masse de ces familles de puces, il arrive que toutes les fonctions soient coulées dans le silicium, mais que seule la configuration payée soit testée et donc disponible pour l'utilisateur. Un concepteur aventureux, prêt à braver tous les dangers (comme preuve de son ambition) pourrait s'attaquer au déverrouillage (s'il est possible par logiciel) de ces fonctions (à ses risques et périls, bien sûr !).

Carte économique

Avec de tels processeurs multicœurs hétérogènes (PMH), les temps sont révolus où l'on pouvait faire fonctionner un microcontrôleur rien qu'en le montant sur une platine de prototypage ou un adaptateur CMS, ne serait qu'à cause des exigences liées à la connexion de mémoire DDR. Cela rend d'autant plus intéressantes les cartes de développement à prix réduit pour faire ses premiers pas.

La carte de développement STM32MP157C-DK2 en tête de cet article est bien documentée [3]. Elle coûte ±90 €. Sans afficheur tactile (désinence -DK1) il en coûte ±60 € (prix de vente recommandé: 99 \$ et 69 \$). La -DK1 comprend un MP157A sans cryptage et sans la puce WiFi. La carte d'évaluation STM32MP157A/C-EV1, nettement plus puissante, coûte ±350 €.

La carte -DK2 devrait largement suffire pour les premiers pas. De par sa taille et son équipement, dont deux prises USB doubles et un connecteur GPIO, on la prendrait au premier regard pour un Raspberry Pi (ce qui n'est pas fortuit). L'afficheur tactile à contrôleur I²C intégré connecté à l'interface DSI possède une résolution native de 480 x 800 pixels pour une diagonale de 4 pouces. La STPMIC1 pour l'alimentation est également présente. Le contrôleur PMH dispose d'un demi-Go de mémoire DRAM (4 gigabits) sur son interface DDR3L habituelle. Une carte micro-SD (>= 2 Go, classe >6) pour la mémoire Flash doit être acquise en plus. À l'avant, on trouve un codec audio connecté par SAI (*Serial Audio Interface*) et I²C, avec son convertisseur N/A et son amplificateur, ensuite un UART-BLE 4.1 et un SDIO-WLAN sur un module Murata. Au verso, il y a des rangées de connecteurs femelles compatibles Arduino Uno V3 pour la périphérie. Notez que du point de vue de ses GPIO, le MP1 est en principe un système 3,3 V, mais qui tolère des tensions «externes» de 5 V. Les autres périphériques matériels sont brièvement présentés sur la fig. 3.

La carte peut être alimentée par un connecteur USB-C, mais il faut 3 A. Pour commencer, lors des premiers pas, on peut se contenter des alimentations de téléphone portable 2 A, si on supporte le témoin d'alimentation qui clignote rouge en permanence. L'autre connecteur USB-C est directement relié au MP1 et peut ensuite fonctionner en mode OTP. Le débogueur intégré ST-Link possède l'interface Micro-USB habituelle. La prise Ethernet devrait présenter un intérêt dès le début (liaison SSH). Pour l'embarquement dans des projets personnels, sans gros effort de conception autour de la MPU, il existe de nombreuses solutions à systèmes sur un module. Ceux-ci sont adaptables individuellement aux variantes les plus diverses, de fournisseurs comme Phytel ou bytes at work, que vous pouvez emboîter ou souder sur votre application personnelle.

CubeMX, EDI et autres outils

Comme de coutume dans l'écosystème ST, on dispose du configurateur de code GUI *STM32CubeMX* auquel s'ajoute un paquet MP1 comprenant les bibliothèques *HAL* et *LL* [4]. On a tout pour configurer les périphériques, les caractéristiques de base du système et la distribution des signaux d'horloge. Dans ce contexte, il faut porter une attention particulière à l'affectation des éléments aux coeurs et à la zone de sécurité de la partie A7 (**fig. 4**).

Avec les fichiers de programmes ainsi produits, on peut poursuivre avec des EDI habituels comme Keil ou IAR, et finir avec la compilation et le transfert du code. Certains tutoriels montrent comment utiliser l'*EDI System Workbench for STM32* (abrégé: *SW4STM32*), l'outil de ST développé avec *ac6-tools*.

Mentionnons encore les outils suivants, intégrés à CubeMX, qu'on peut utiliser avec le MP1 pendant la phase de développement :

- le *Power Consumption Calculator* évalue la consommation électrique d'un projet donné dans les différents modes d'économie d'énergie. Quelques exemples d'applications servent de référence.
- La *DDR Tool Suite* permet de configurer et d'optimiser la connexion de la mémoire DDR, puis de la tester (test de stress p. ex.). Mots clés pour connasseurs : *Deskew*, *DQS Gating* ou *Eye Centering*. Après une optimisation réussie, on peut stocker les données de configuration sur le MP1 dans le FSBL (donc le plus souvent dans le chargeur d'amorçage sécurisé *TF-A*). Cet outil est censé faciliter la tâche aux concepteurs du matériel.

Bien entendu, pour les tests de la DDR, le MP1 doit être connecté à l'environnement du Cube et les broches d'amorçage doivent être mises dans l'état correct pour que le micrologiciel nécessaire soit chargé dans le SSBL (voir ci-dessous).

Lacunes et pièges de la GPU

Un aspect critiquable du point de vue des développeurs est que la plus grande partie des bibliothèques internes de la GPU n'est accessible qu'après la conclusion d'un accord de non-divulgation avec le fabricant de Vivante, *VeriSilicon*. Voilà qui est inconnu chez d'autres types de GPU. La transparence du logiciel est une exigence légitime de l'utilisateur (final), particulièrement dans le cas d'applications critiques. ST prendra-t-il cet aspect en compte ?

Pour la GPU, il vient s'ajouter au *vCompiler* basé CL quelques outils d'analyse et d'optimisation :

- on peut laisser tourner *vProfiler* sur le MP1 pour enregistrer les performances de la GPU en temps réel et exploiter ensuite les résultats sur un PC au moyen de *vAnalyzer*, fonction par fonction.
- avec *vShader*, les thèmes de rendu comme les ombres des éléments 3D affichés peuvent être générés et préanalysés. *vShader* dispose d'une interface graphique utilisateur et propose un choix de formes 3D de base.
- avec *vTexture* on (dé)comprime des fichiers images aux formats *dxt* et *etc*.

Ces outils se trouvent dans le paquet *GPU Toolkit* [6].

Sur la page de produit du MP1 on trouvera aussi le *GPU Application Programming Manual PM0263*, qui donne des conseils pour une gestion optimale de la GPU pour des débits de trames acceptables, et le moyen d'en tenir les CPU voisines à l'écart si possible. Ces conseils

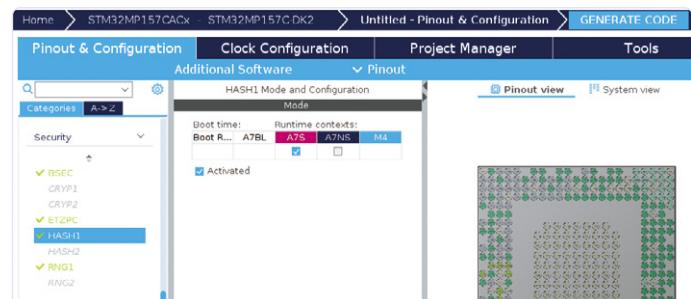


Figure 4. Exemplaire : l'affectation simple de l'élément de hachage au cœur M4 ou M7 (S: zone sécurisée/NS: zone non sécurisée) dans CubeMX (saisie d'écran : Viacheslav Gromov).

portent sur l'usage des textures, des *Vertex Buffer Objects* (VBO), et comprennent des avertissements sur l'utilisation de certaines instructions GL et de commandes d'effacements partiels.

La troïka du logiciel

L'offre de systèmes d'exploitation pour le (ou les) cœur(s) comprend deux variantes, l'habituel OpenSTLinux, donc une adaptation de Yocto, et une distribution d'Android. La **figure 5** montre l'interaction des trois domaines de logiciel y compris une possible séquence d'amorçage avec un noyau Linux. Dans la zone de sécurité (*Secure-Zone*), l'environnement de sécurité OP-TEE utilise le plus souvent les propriétés de la zone de confiance (*TrustZone*). L'environnement sécurisé cible y est démarré directement par le chargeur sécurisé d'amorçage de premier niveau (*First Stage Bootloader*, *FSBL*), alors que le noyau Linux utilise encore le chargeur habituel U-boot.

Notez le chemin en pointillé du chargeur d'amorçage de second niveau (*Second Stage Bootloader*, *SSBL*), donc, ici, U-boot. L'expérience montre que, de cette manière, on peut charger le microgiciel dans M4 et le démarrer environ 600 ms avant que le noyau Linux soit définitivement activé. Bien entendu, selon l'application, on peut faire tourner sur le cœur M4 soit des applications purement *bare-metal* (sans OS), soit des moniteurs en temps réel, parmi lesquels le très répandu FreeRTOS est bien supporté par l'environnement Cube. La séquence d'amorçage décrite n'est qu'un exemple, car il est bien entendu possible, pendant le développement, de charger du microgiciel dans le M4 indépendamment du A7 avec le débogueur JTAG/SWD.

Du point de vue du M4, il n'est pas nécessaire d'insister sur les biblio-

RUMEURS : D'AUTRES STM32MP

À l'expo *Embedded World 2020*, des variantes MP1-(D/-F) ont été présentées avec une fréquence d'horloge maximum de 800 MHz.

Selon des rumeurs, un modèle MP2 devrait apparaître dans quelques années, avec des coeurs A plus gros et plus rapides (GHz) avec plus de mémoire et un interfaçage plus riche, un noyau M plus efficace et une GPU plus trapue, avec une accélération des calculs d'apprentissage machine. D'ici là on s'attend à l'arrivée d'une MPU A7 native.

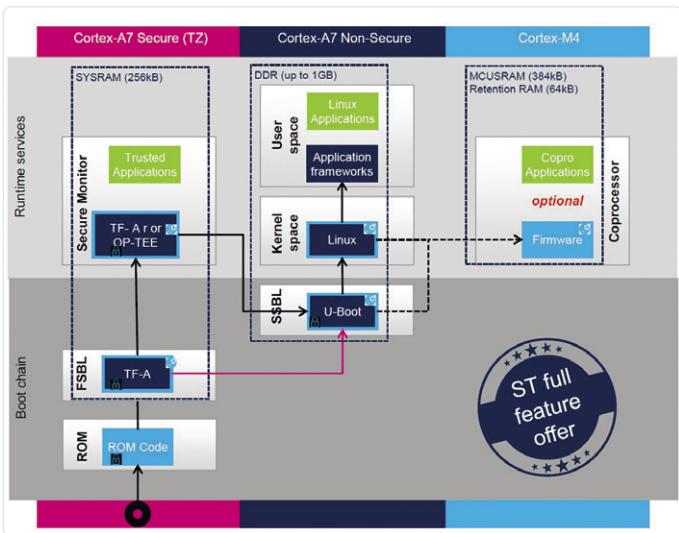


Figure 5. La troïka : à gauche, la zone sécurisée, au milieu, le système d'exploitation applicatif (ici Linux) et à droite, le M4 en temps réel. Une logique d'amorçage possible avec *Calls* et *Loads* est représentée sous forme d'organigramme approximatif (source : ST).

thèques disponibles HAL, LL et autres, mais du point de vue de A7 ou de Linux, cela devient passionnant. L'accès aux unités (périphériques) associées passe par des pilotes mentionnés sur la figure 6. Par ailleurs, il existe bien entendu du logiciel intermédiaire (*middleware*) tel que *Wayland-Weston* pour des applications GPU ou, en version standard, le logiciel de base comme *apt*, *dpkg*, *python* et *minicom*.

Sur le Wiki en ligne [7] avec son domaine de développement, on peut non seulement télécharger les paquets de OpenSTLinux-Starter et -Developer, ainsi que des paquets de distributions avec leurs SDK associés, mais aussi observer des étapes de transferts sur carte

microSD par le CubeProgrammer, des premiers pas de programmation, des exemples et des «comment faire» pour certains composants logiciels comme le GTK pour les applications graphiques.

Mentionnons encore le paquet mis à disposition par ST pour la suite intégrée *AWS IoT Greengrass*, bien connue dans le monde de l'IoT, pour le prétraitement des données «à la marge» et la connexion au nuage. Notons enfin *X-LINUX-AI-CV*, une extension d'intelligence artificielle, qui, associée au framework Python-3 *TensorFlow Lite*, apporte, entre autres, au niveau applicatif, des fonctions de reconnaissance d'objets et de classement d'images par réseaux de neurones artificiels.

Premières activités ludiques

Indépendamment de l'image chargeable du système d'exploitation, vous devriez voir sur la carte SD le programme de démo et quelques applications, ainsi que quelques démos graphiques (comme le *Videoplayer* accompagné de quelques appareils audio connectés par Bluetooth) et un affichage de performance sur un moniteur nommé *netdata* accessible sur le réseau local par une adresse IP.

Pour obtenir un accès rapide à la carte par le système chargeable que vous venez d'installer sur la carte SD, vous pouvez vous connecter à l'aide de votre système Linux-PC (virtualisé) par l'interface série de la liaison ST intégrée avec des outils comme *minicom* et voir et utiliser la console du système MP1-Linux. Une alternative serait une liaison SSH. Depuis la console vous pouvez redémarrer la carte et, au cours du redémarrage, appuyer sur une touche utilisateur quelconque pour accéder à la console U-Boot, sur laquelle la commande

`$ ums 0 mmc 0`

demande à U-Boot de rendre le système de fichiers accessible sur USB-OTG (On-The-Go). Au moyen de commandes telles que

`$ mount | grep userfs`

vous pouvez alors examiner plus précisément les différents domaines du système de fichiers sur le calculateur Linux connecté à l'OTG et éventuellement modifier les démos et autres exemples d'applications.

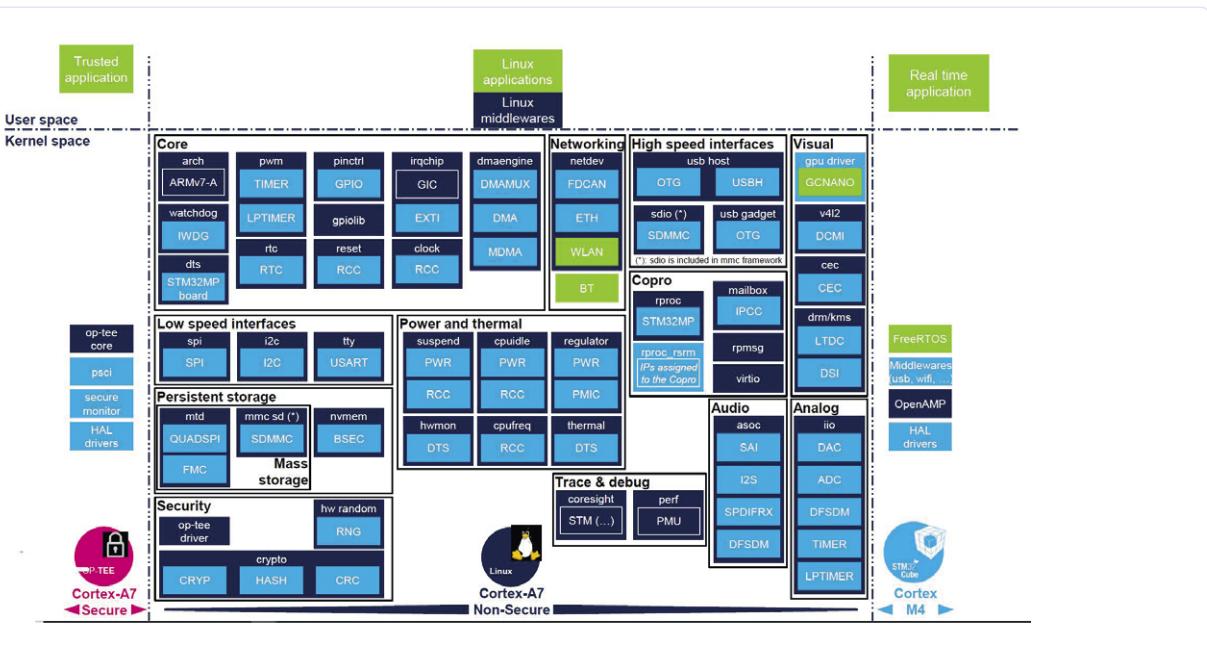


Figure 6. Les pilotes de «périphériques» dans les trois zones sous Linux dans la partie applicative du PMH. Les bibliothèques indiquées en bleu clair sont fournies par ST, celles en bleu foncé ont été développées/supportées par/avec la communauté (source : ST).

Listage 1. Fragment du programme de démo montrant la simplicité d'utilisation de la bibliothèque graphique GTK. On y reconnaît la structure de base de la barre d'infos de pages et une icône.

```
def create_page_icon(self):
    page_main = Gtk.HBox(False, 0)
    page_main.set_border_width(0)

    # create a grid of icon
    icon_grid = Gtk.Grid(column_homogeneous=True, row_homogeneous=True)
    icon_grid.set_column_spacing(20)
    icon_grid.set_row_spacing(20)

    # STM32MP1 Logo and info area
    logo_info_area = _load_image_Box(self, «%s/pictures/ST11249_Module_STM32MP1_alpha.png« % DEMO_
        PATH, «%s/pictures/ST13340_Info_white.png» % DEMO_PATH, self.board_name, -1, 160)
    rgba = Gdk.RGBA(0.31, 0.32, 0.31, 1.0)
    logo_info_area.override_background_color(0,rgba)

    # Button: Netdata icon
    eventBox_webser = _load_image_eventBox(self, «%s/pictures/netdata-icon-192x192.png« %
DEMO_PATH,
        «netdata», «perf monitor», -1, self.icon_size)
    eventBox_webser.connect(«button_release_event», self.wifi_hotspot_event)
    eventBox_webser.connect(«button_press_event», self.highlight_eventBox)

    ...

```

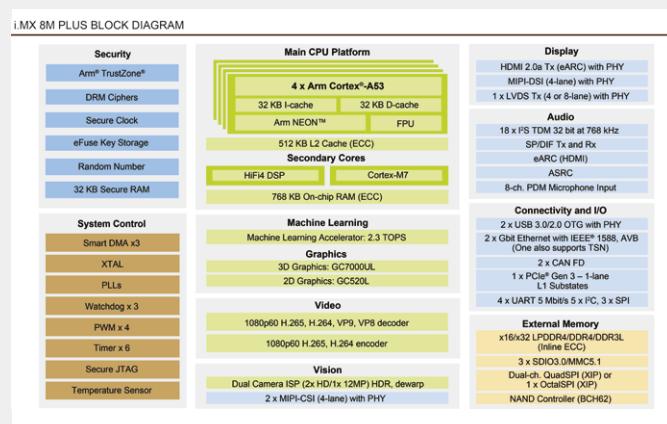
EXEMPLES D'AUTRES PMH EMBARQUÉS

Les apparences sont trompeuses, le MP1 n'est pas tout seul sur le marché, qui a déjà proposé maints PMH embarqués. Il en existe d'innombrables, variant en configuration et en puissance au gré des applications.

Le monde de l'embarqué proche des consommateurs est fermé au sommet par les PMH habituellement utilisés dans les téléphones portables (si l'on excepte des applications particulières comme la conduite autonome). Un exemple courant est la série *Galaxy-S10* qui incorpore en Europe l'*Exynos 9820 Octacore* avec deux coeurs développés par *Samsung* (qui s'appellent M4, mais n'ont rien à voir avec ARM-Cortex), deux coeurs A75 et quatre coeurs A55.

À un niveau plus modeste, on trouve le *RT1170* de NXP. Il est composé d'un cœur M7, d'un cœur M4 et d'une GPU-2D. Ce n'est donc pas un processeur comme les autres, mais une MCU spéciale avec une mémoire Flash externe. Sa vitesse de 1 GHz (sic!) impressionne, ses périphériques et sa sécurité sont respectables.

Avec la famille Sitara-AM57x, TI se lance sur le marché haut de gamme. Les principales applications concernent le multimédia et le *Machine Learning* (ML). Dans sa configuration maximum, ce PMH possède deux coeurs A15, deux coprocesseurs DSP C66x, deux coeurs doubles M4, une GPU-3D double, un



accélérateur vidéo, etc. Avec tous ces auxiliaires, une résolution de 4K n'est pas hors de portée.

Le PMH présenté à l'expo CES2020, qui ressemble le plus au futur STM32MP2, est le *i.MX 8M Plus* de NXP (cf [diagramme](#)). En dehors de ses deux coeurs A53 et de son cœur M7, il possède deux unités graphiques (2D/3D) un accélérateur KNN (*k-Nearest Neighbor algorithm*), une double connexion de caméras, une TrustZone et d'autres éléments de sécurité.

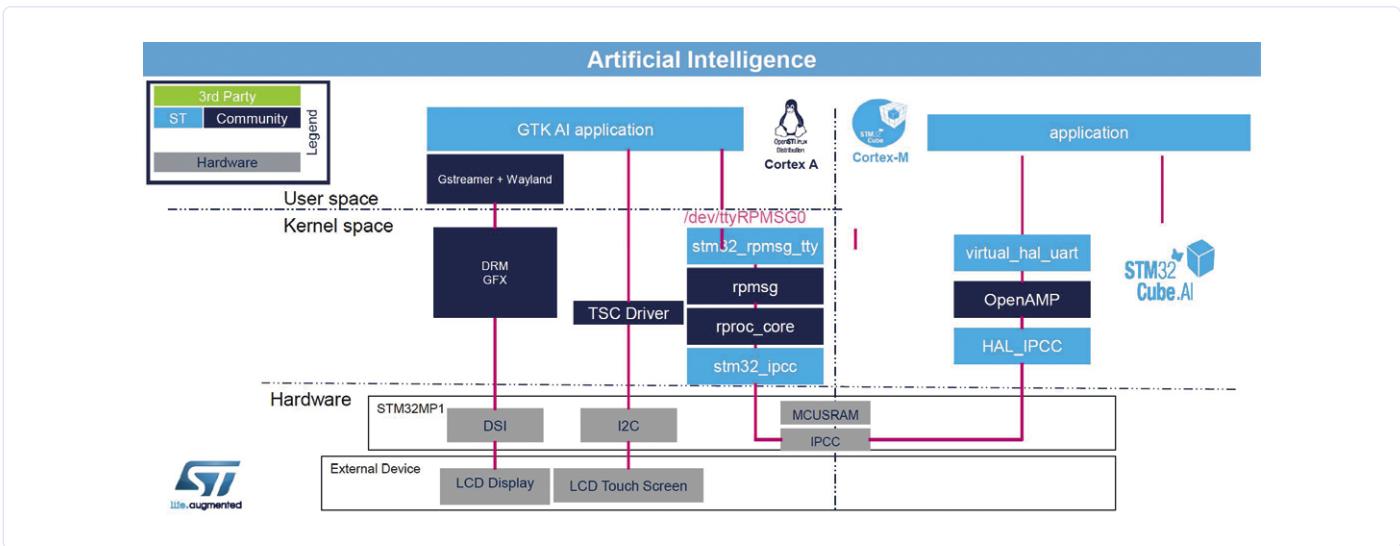


Figure 7. Le travail d'une équipe de rêve : l'application IA d'écriture manuscrite sur le M4 dialogue avec le RPMsg puis l'application graphique GTK sur les coeurs A7 et la GPU par l'intermédiaire de l'OpenAMP et de l'IPCC-HAL (source : ST).

Pour le transfert et l'exécution de programmes M4, on peut passer depuis la vue Linux par les sous-répertoires de `/sys/class/remoteproc/remoteproc0`.

L'exemple de la figure 7 montre un aspect possible d'un programme terminé qui participe aux deux mondes des coeurs du PMH. Ce programme KI est l'une des applications de la démo et reconnaît sur l'afficheur tactile des lettres et des chiffres tracés avec le doigt et exécute l'application éventuellement associée. Le réseau neuronal entraîné et porté par CubeAI réside dans le domaine M4, alors que toute l'interface graphique personne-machine s'exécute sur A7 ou l'unité graphique. Le déroulement des interactions, y compris avec l'interface IPCC (*Inter-Processor Communication Controller*) et la SRAM partagée par le renommé framework OpenAMP est bien observable. L'interface utilisateur sur l'afficheur peut être réalisée très rapidement avec la bibliothèque GTK (listage 1). Il existe bien entendu d'autres frameworks graphiques pour les fonctions les plus diverses, comme *GStreamer* ou *Wayland/Weston*, même *QT* peut être utilisé comme niveau intermédiaire.

Que vous soyez débutant, reconverti ou connaisseur de Cortex-M ou

de Linux : avec ce petit tour d'horizon du marché et de la documentation, vous venez de bénéficier d'une petite poussée dans le monde nouveau pour vous des processeurs multicœurs hétérogènes. Vous en êtes maintenant au point où nous pouvons vous abandonner sans mauvaise conscience : vous êtes sur la bonne voie ! ↗

(200072-04 VF Helmut Muller)



@ WWW.ELEKTOR.FR

- **Pyboard D-series with STM32F722 and WiFi/BT**
www.elektor.fr/pyboard-d-series-with-stm32f722-and-wifi-bt
- **Livre Programming with STM32 Nucleo Boards**
www.elektor.fr/programming-with-stm32-nucleo-boards
- **STM32 Nucleo L476RG Board**
www.elektor.fr/stm32-nucleo-l476rg-board

LIENS

- [1] **page de produit de la famille MPU :**
www.st.com/en/microcontrollers-microprocessors/stm32mp1-series.html
- [2] **STM32MP1 - tutoriel en ligne :**
www.st.com/content/st_com/en/support/learning/stm32-education/stm32-online-training/stm32mp1-online-training.html
- [3] **Kit Discovery DK2 avec STM32MP157C :** <https://bit.ly/37HXOF8>
- [4] **paquet STM32CubeMP1 :** <https://bit.ly/2UaBhwx>
- [5] **STM32CubeIDE :** www.st.com/en/development-tools/stm32cubeide.html
- [6] **STM32MP157 GPU Toolkit :** www.st.com/en/development-tools/stm32mp157gputk.html
- [7] **Wiki/Development Zone avec instructions de téléchargement d'OS :** <https://wiki.st.com/stm32mpu>