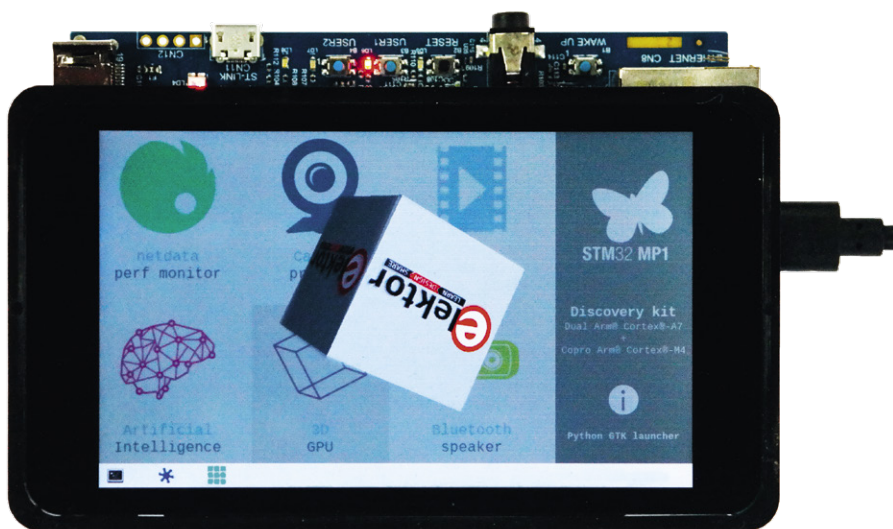


M4 + 2× A7 + GPU: een ongelijk droomteam

de nieuwe
STM32MP1-SoC voor
het zwaardere werk

Viacheslav Gromov (leser@gromov.de)

Door trends als (I)IoT, cryptografie en steeds complexere mens-machine-interfaces (HMI) komt zelfs een ARM-cortex M7 inmiddels aan zijn grenzen. Daarom brengen de bekende fabrikanten van microcontrollers al enige tijd zogenaamde heterogene multi-core-processoren (HMP's) op de markt, met grote applicatie-kernen voor HMI's en kleine kernen voor real-time toepassingen in één behuizing. We werpen hier een blik op de eerste HMP's van STMicroelectronics, die bijzonder interessant zijn qua prijs en documentatie. De belangrijkste voordelen van deze MP1-familie zijn energie-efficiëntie, een 3D GPU, Linux- en Android-ondersteuning en meer veiligheid, ondermeer door een interne verdeling in veiligheidszones.



(Bron: Viacheslav Gromov)

Dit veel te korte artikel (over de meeste onderwerpen raak je nooit uitgepraat) is alleen bedoeld als een eerste kennismaking met de HMP-wereld. Het is vooral bedoeld voor gevorderde ontwikkelaars die met ARM Cortex-M werken, en embedded Linux-ontwikkelaars die zich misschien al intensief met de Raspberry Pi (RPI) hebben beziggehouden.

Tabel 1. De verschillende kenmerken van de MP1-familie. Bij elk type is er een A-versie zonder en een C-versie met geïntegreerde HW-crypto-eenheid.

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

De MP1-familie: leden en structuur

In **tabel 1** ziet u een overzicht van alle leden van de MP1-familie. Ze verschillen vooral in de kernconfiguratie en in de aanwezigheid van een GPU. De meest uitgebreide versie, de *STM32MP157C*, heeft alle drie de kernen (ARM-Cortex-A7 als Siamese tweeling met gedeelde L2-cache, en de ARM-Cortex-M4), een 3D GPU met DSI, een crypto-eenheid en een FD-CAN-interface (**figuur 1**).

Ten gevolge van de A7-kern op de wafer kan de gehele chip inclusief de M4-kernen alleen in pure CMOS-technologie worden vervaardigd, wat betekent dat er op de chip geen groot geheugen kan zitten. Die functie kan worden uitbesteed door bijvoorbeeld RAM en flash aan te sluiten via (LP)DDR en dual-mode Quad-SPI of SDMMC. Het is mogelijk om tot 1 GB externe RAM aan te sluiten, alsmede flash-geheugen dat alleen begrensd is door de maximale adresruimte.

De TrustZone-technologie van ARM creëert veilige gebieden in het geheugen, die alleen door bepaalde programmaonderdelen kunnen worden gelezen of geschreven. Er zijn ook gebieden zoals de voedings- en klokbesturing die onder deze technologie vallen. De A7-kernen hebben dankzij de ARM-NEON-SIMD-architectuur een uitgebreide instructieset, speciaal voor functies zoals Quad-MAC-berekeningen,

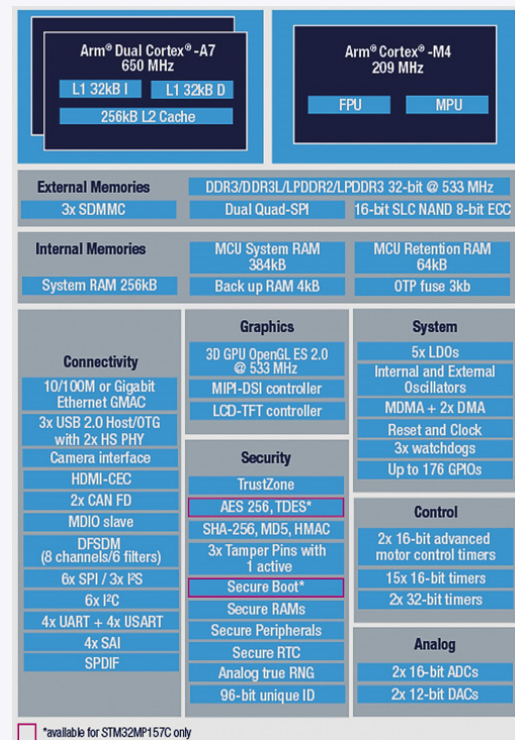
bijvoorbeeld in multimediatoeepassingen met beeld en/of geluid.

De A7-kernen en de geheugenaansluitingen bevinden zich op de interne 64-bits AXI-bus die tot 266 MHz kan worden geklokt. De M4 is verbonden met de 209 MHz AHB-busmatrix en de meeste periferie zit op APB-bussen. Met behulp van de AXI- en AHB-verbindingsmatrix kan de periferie ook worden toegewezen aan bepaalde kerngebieden of het A7-veiligheidsbereik, die alleen door als veilig aangemerkte programma-componenten kunnen worden bediend (figuur 2). Hetzelfde geldt voor de volledig geüniformeerde geheugenbereiken.

De Vivante 3D GPU haalt een pixelstream van 133 MP per seconde dankzij de maximale kloksnelheid van 533 MHz. Hij ondersteunt niet alleen de veelgebruikte *OpenGL 2.0* in de embedded versie, maar ook onder andere *OpenVG 1.1* en *EGL 1.4*. De bijbehorende grafische interfaces voor LCD TFT's of DSI ondersteunen resoluties van maximaal 1366 x 768 pixels (WXGA) bij 60 fps.

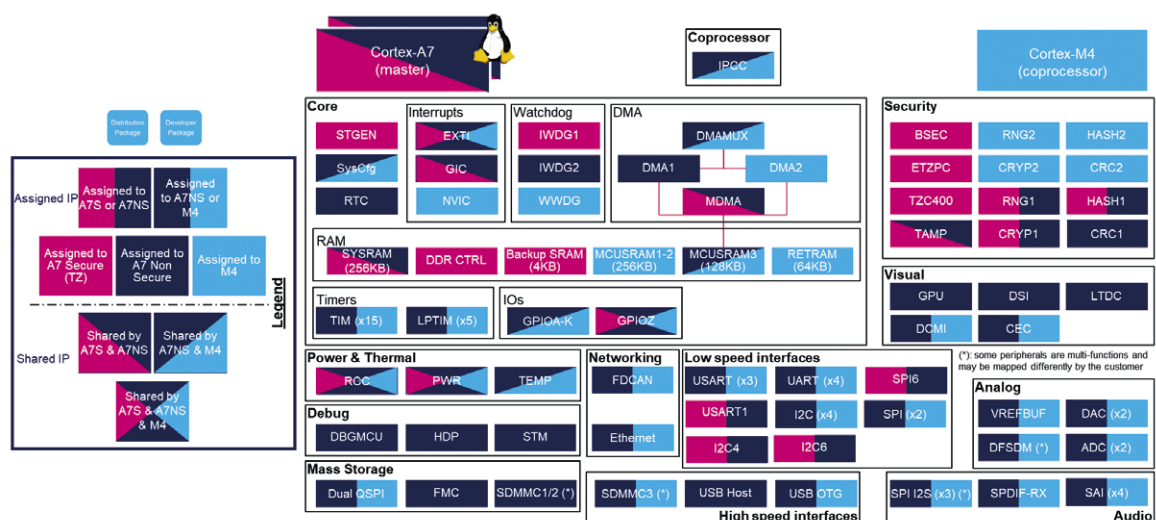
(M)DMA's en 288 (gedeelde) globale interrupts of events maken het mogelijk om veel processen zo onafhankelijk mogelijk te laten verlopen, zonder tussenkomst van de rekenkernen. Op het brede terrein van DSP is *Digital Filter for Sigma Delta Modulators* (DFSDM) interessant. Dit kan overweg met 24-bits (raw) ADC-waarden. Zo kan bijvoorbeeld een MEMS-microfoon met zijn sigma/delta-omzetter rechtstreeks worden aangesloten.

Ook interessant zijn de consumer-HDMI-interface (volgens de *CEC 1.4*-specificatie), de tot 14 bits brede en 140 MB/s snelle camera-interface, de unieke 96 bits ID met wafer-coördinaten en dergelijke en de digitale temperatuursensor met frequentie-uitgang (-40...+125 °C). Daarnaast zijn er natuurlijk enkele *secure*-componenten, die bijvoorbeeld de *AES-256* coderingen kunnen leveren die als 'veilig' worden beschouwd (*cryptocontroller*- en *hash*-elementen). De random-generator kan op analoge wijze (analoog gegenereerde *seed*) vier zo willekeurig mogelijke 32bit-waarden genereren na ten minste 213 klokcycli, en de *Tamper*, bekend van de STM32H7, kan veilig 32 backup-registers in alle bedrijfsmodi opslaan en indien gewenst onmiddellijk wissen als zich bepaalde gebeurtenissen voordoen, zoals een onderbreking tussen bepaalde aansluitpennen.

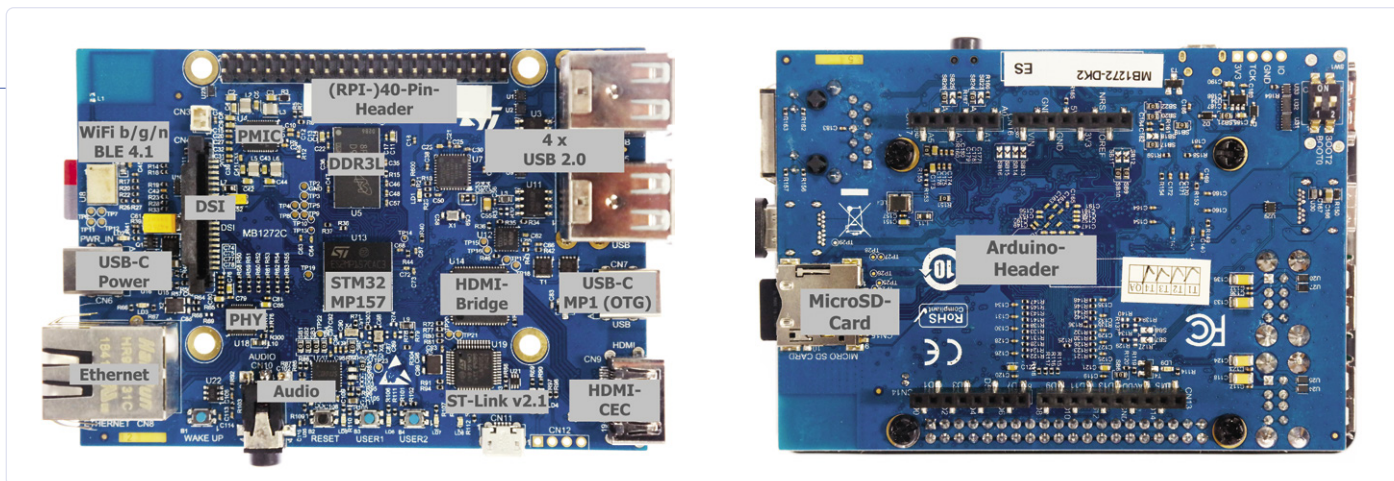


Figuur 1. De structuur van de rekenkernen met de belangrijkste periferie, beveiliging en interfaces voor het grootste lid van de familie, de STM32MP157 (bron: ST).

De communicatie tussen de twee kern-'werelden' kan plaatsvinden via de interne *Inter Processor Communication Controller* (IPCC) in simplex- of half-duplex-modus via twee kanalen en/of via een gedeeld SRAM-geheugen (*Shared Memory*, AXI: 256 KB). Zo kunt u met een mailbox-systeem voor interne gegevensuitwisseling werken. Er zijn veel invasieve en niet-invasieve debug- (JTAD/SWD) of trace-mogelijkheden (trace-pennen van de TPIU) met bijbehorende interfaces.



Figuur 2. De individuele toewijzingsmogelijkheden aan de kernen of de beveiligde zone van de periferie, de geheugengebieden en andere interne functionaliteit is hier met kleuren weergegeven (bron: ST).



Figuur 3. Beide zijden van het STM32MP157C-DK2-board met de belangrijkste elementen, maar zonder display (bron: Viacheslav Gromov).

De uitgebreide periferie kan niet (zelfs niet bij benadering) volledig worden beschreven in dit artikel, daarvoor zult u de uitgebreide documentatie moeten raadplegen. Naast het meer dan 4000 pagina's (!) tellende *Reference Manual* zijn er ook een datasheet van 260 pagina's en (software) application notes, bijvoorbeeld over USB-bootloader procedures, die allemaal te downloaden zijn van de productpagina [1]. Veel van de hier genoemde informatie is terug te vinden in de vorm van een online MP1-training [2] en op de officiële YouTube-pagina van ST. Voor de STM32MP1, voor de voeding van de kernen en de analoge periferie, moeten niet alleen verschillende voedingsspanningen worden aangesloten op vele pinnen, maar de spanningen variëren ook afhankelijk van de ingestelde energiebesparingsmodus! Om tijdens het ontwerp zo min mogelijk met dit onderwerp bezig te hoeven zijn, is er de STPMIC1: een *Power Management IC* (PMIC), dat naast de voedingsspanningen met de MP1 verbonden is via een I²C-interface en andere lijnen zoals reset en wakeup. Dit IC kan ook een DDR-geheugen voeden. Het vermogen van de MP1 kan van de maximale 487 mW (drie kernen plus alle randapparatuur en ingeschakelde GPU) worden verminderd met de energiebesparende modus, waarin alleen de M4 wakker is (92 mW), en met de stop/standby-modus, waarin slechts enkele microwatts worden gebruikt.

De prijs van de MP1-chip zelf ligt overigens tussen de € 5 en € 10 €, afhankelijk van het aantal en de uitvoering.

Een goedkoop board

De tijden dat we gewoon een µC op een prototype-print of zelfs een SMD-adaptor konden zetten om ermee te experimenteren zijn voorbij.

HETZELFDE SILICIUM, ANDERE PRIJS?

Als u een kleinere MP1-variant koopt, hoeft dat niet per se te betekenen dat alleen de kernen en andere functies die in de tabel zijn vermeld, beschikbaar zijn op de *die*. In de massaproductie kan het gebeuren dat, hoewel alle functies in silicium worden 'gegoten', alleen de betaalde configuratie wordt getest en zo wordt vrijgegeven voor de ontwikkelaar. Avontuurlijke ontwikkelaars kunnen misschien zulke componenten (softwarematig?) ontsluiten (natuurlijk op eigen risico!).

Met deze HMP's zal dat niet lukken, al was het maar vanwege de zware eisen die gelden voor de DDR-verbinding met het geheugen. Dat maakt low-cost-ontwikkel-boards des te interessanter voor de eerste experimenten.

Het board in de kopfoto is door ST goed gedocumenteerd voor een kennismaking met de MP1. Hij heet *STM32MP157C-DK2* [3] en kost ruim € 90. Zonder het getoonde aanraakscherm (type -DK1) kost hij slechts € 60 € (aanbevolen verkoopprijzen: \$ 99 en \$ 69). De -DK1 heeft een MP157A zonder crypto-eenheid en ook de draadloze interface is bij dat type niet aanwezig. Het veel uitgebreidere evaluatieboard *STM32MP157A/C-EV1* kost hier ongeveer € 350.

Het DK2-ontwikkelboard zou ruim voldoende moeten zijn voor de eerste ontwikkelingsstappen. Op het eerste gezicht lijkt het een RPi qua formaat en uitrusting: twee dubbele USB-aansluitingen en een GPIO-connector. Het aanraakscherm met geïntegreerde I²C-interface voor de touchcontroller, dat is aangesloten op de DSI-connector, heeft een resolutie van 480 x 800 pixels en een diagonaal van 4 inch. De STPMIC1 voor de stroomvoorziening is ook aan boord. Er is een halve gigabyte DRAM (4 GBit) beschikbaar voor de HMP via de normale DDR3L-verbinding, en er is ruimte voor een MicroSD-kaart (>= 2 GB, klasse >6) als extra flash-geheugen nodig is. Aan de voorzijde vinden we een audiocodec met DAC en versterker (aangesloten via SAI en I²C) en een UART-BLE 4.1 en SDIO-WLAN op een Murata-module. Aan de achterkant zitten Arduino UNO V3-compatibele connectoren voor shields. Let er wel op dat de MP1 normaal gesproken een 3,3V-systeem is, hoewel de GPIO's wel 5 V 'van buitenaf' tolereren. **Figuur 3** geeft een klein overzicht van de andere periferie.

Het board kan worden gevoed via een USB-C-aansluiting, hoewel eigenlijk een voedingseenheid nodig is die 3 A kan leveren. In het begin kunt u voor niet-kritische experimenten wel gebruik maken van een gewone smartphone-voeding die 2 A kan leveren, als u het niet erg vindt om continu tegen de rood knipperende power-LED aan te kijken, die zegt dat u daarmee de regels overtreedt.

De andere USB-C-poort is rechtstreeks met de MP1 verbonden en kan later ook in OTP-modus worden gebruikt. De geïntegreerde ST-Link-debugger heeft een gewone micro-USB-connector. De Ethernet-aansluiting kan al bij de eerste experimenten interessant zijn (SSH-verbinding). Er zijn allerlei, vaak individueel aanpasbare, systeem-op-module oplossingen verkrijgbaar van fabrikanten zoals *Phytec* en *bytes at work*, waarmee u de MPU in eigen projecten kunt toepassen zonder zelf veel hardware te hoeven ontwikkelen.

CubeMX, IDE's en aanvullende tools

Zoals gebruikelijk in het ST-ecosysteem is de GUI-codeconfigurator STM32CubeMX beschikbaar met een toegevoegd MP1-pakket met (HAL- en LL-) bibliotheken [4]. Daar kunt u de periferie, de basiseigenschappen van het systeem en de verdeling van het kloksignaal configureren. Daarbij is vooral de toewijzing van de elementen aan de kernen en de beveiligde zone van het A7-deel belangrijk (figuur 4). Met de gegenereerde programmabestanden kunt u verder gaan in de gebruikelijke IDE's zoals Keil of IAR, om de code te compileren en op het systeem te zetten. Er is ook lesmateriaal waarin u kunt zien hoe u ST's eigen, met ac6-tools ontwikkelde *System Workbench voor STM32-IDE* (kort: SW4STM32) kunt gebruiken. De huidige versie is waarschijnlijk de op Eclipse gebaseerde *STM32CubeIDE* [5], die de CubeMX combineert met de Atollic TrueStudio en de MP1 ondersteunt sinds begin 2020.

In CubeMX zijn nog twee belangrijke tools geïntegreerd die u in de ontwikkelingsfase met de MP1 kunt gebruiken:

- Met de *Power Consumption Calculator* kunt u het stroomverbruik in verschillende energiebesparende modi doorrekenen. Er zijn ook enkele voorbeeldtoepassingen die als referentie kunnen worden gebruikt.
- Met de *DDR Tool Suite* kan de DDR-interface van het geheugen worden getest (bijvoorbeeld als stresstest), geconfigureerd en geoptimaliseerd. Sleutelwoorden voor experts zijn *Deskew*, *DQS Gating* of *Eye Centering*. Na succesvolle optimalisatie kunnen de configuratiegegevens op de MP1 worden opgeslagen in de *FSBL* (d.w.z. meestal in de *TF-A Secure Bootloader*). Met dit tool wil men de hardware-ontwikkelaar zoveel mogelijk van deze taak ontlasten. Voor de DDR-tests moet de MP1 natuurlijk worden aangesloten op de Cube-omgeving en moeten de opstartpenen in de juiste toestand worden gebracht, zodat de benodigde firmware in de *SSBL* kan worden geladen (zie verderop).

Tekortkomingen van de GPU

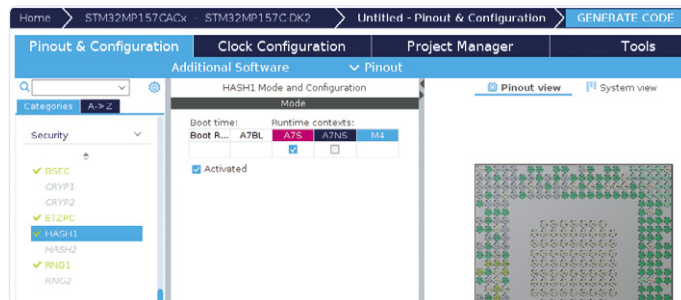
Een punt van kritiek vanuit het oogpunt van een ontwikkelaar is dat de meeste interne GPU-bibliotheken pas kunnen worden bekeken nadat u een NDA-overeenkomst bent aangegaan met *VeriSilicon*, de fabrikant van Vivante. Dat is bij andere GPU-types niet gebruikelijk. Transparante software kan essentieel zijn voor de (eind)gebruiker, zeker voor zeer kritische toepassingen. Het is te hopen dat ST hier bij verdere ontwikkelingen aandacht aan gaat besteden.

Voor de GPU zijn er naast de op CL gebaseerde vCompiler enkele tools voor analyse en optimalisatie:

- U kunt de *vProfiler* op de MP1 draaien om de prestaties van de GPU in real time op te nemen en deze vervolgens op de PC te evalueren met behulp van de *vAnalyzer*, ook uitgesplitst per functie.
- Met de *vShader* kunt u rendering-thema's zoals schaduwen van weergegeven 3D-elementen bekijken en vooraf analyseren. *vShader* heeft een GUI en biedt een aantal 3D-basisvormen om uit te kiezen.
- Met *vTexture* kunt u beeldbestanden (de)comprimeren van en naar *dxt*- en *etc*-formaat.

Deze tools zijn te vinden in het *GPU Toolkit*-pakket onder [6].

Op de MP1-productpagina kunt u ook het *GPU Application Programming Manual PM0263* vinden, met tips over hoe u de GPU maximaal



Figuur 4. Voorbeeld: eenvoudige toewijzing van het hash-element aan de M4- of M7-kern (S: secure / NS: non-secure zone) in CubeMX (screenshot: Viacheslav Gromov).

kunt benutten bij een aanvaardbare framerate, om de andere CPU's zo veel mogelijk te ontlasten. Het gaat daarbij om de juiste behandeling van texturen, *Vertex Buffer Objects* (VBO's), en ook om waarschuwingen tegen het gebruik van bepaalde GL-commando's en gedeeltelijke wis-instructies.

De software-trojka

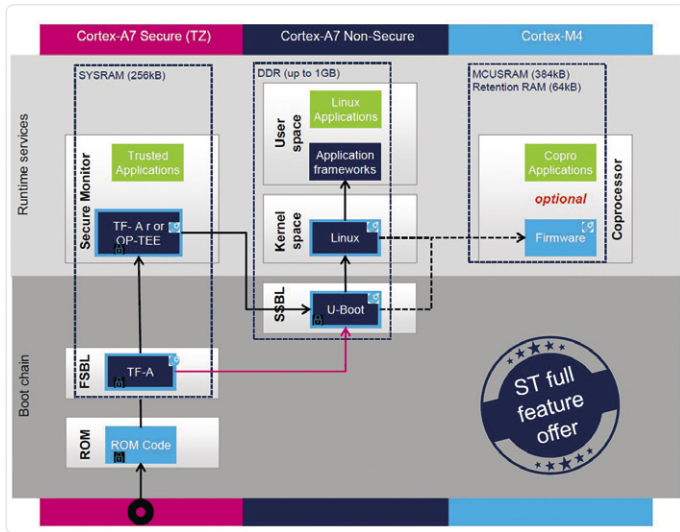
Op de applicatiekern(en) worden voornamelijk twee besturingssystemen aangeboden, de gewone OpenSTLinux als Yocto-aanpassing en als alternatief een Android-distributie. In figuur 5 ziet u de typische interactie van de drie onderdelen van de software, inclusief een mogelijke bootchain met een Linux-kernel. In de beveiligde zone maakt de OP-TEE-beveiligingsomgeving meestal gebruik van de TrustZone-eigenschappen. Hier start de beveiligde target-omgeving rechtstreeks via de beveiligde *First Stage Boot Loader* (FSBL), terwijl de Linux-kernel nog de gebruikelijke U-Boot-bootloader nodig heeft.

Let vooral op het gestippelde pad van de *Second Stage Bootloader* (SSBL, in dit geval dus U-Boot). De praktijk leert dat de M4 zo ongeveer 600 ms voordat de Linux-kernel actief wordt, kan worden voorzien van firmware en gestart. Uiteraard kan de M4-kern, afhankelijk van de toepassing, zowel bare metal-oplossingen als de bekende real time-besturingssystemen draaien. Het veelgebruikte FreeRTOS wordt goed ondersteund door de Cube-omgeving. De beschreven opstartketen is slechts één voorbeeld van het gebruik, want bij het ontwikkelen kan de M4 natuurlijk onafhankelijk van de A7 worden geladen met de JTAG/SWD-debugger.

BIJ GERUCHTE VERNOMEN:

MEER STM32MP'S

Tussen het schrijven en ter perse gaan van dit artikel zal naar verluidt een MP1 worden geïntroduceerd met een iets hogere maximale kloksnelheid en misschien full-HD grafische ondersteuning. En volgens onbevestigde geruchten zal over enkele jaren een MP2 worden uitgebracht met grotere GHz-A kernen (en dus meer geheugen- en interface-mogelijkheden), een efficiëntere M-kern, en een nog grotere GPU met snellere mogelijkheden voor machine-leren (met name KNN's). En ook een pure A7-MPU kan worden verwacht.



Figuur 5. De trojka: links de beveiligde zone, in het midden het besturingssysteem (hier Linux) en rechts de M4 in real time. De mogelijke bootvolgorde is hier weergegeven als een ruwe flowchart (bron: ST).

Op de bekende beschikbare HAL-, LL- en andere bibliotheken hoeven we vanuit het M4-perspectief niet nader in te gaan, maar vanuit het A7- of Linux-perspectief is het wel interessant. De toegang tot de eventueel toegewezen (periferie-)elementen loopt via de in **figuur 6** genoemde drivers. Daarnaast is er natuurlijk middleware zoals *Wayland-Weston* voor GPU-toepassingen en de gebruikelijke basismiddleware zoals *apt*, *dpkg*, *python* en *minicom*.

Overigens kunt u in het development-gebied van de online-wiki [7] niet alleen de OpenSTLinux start-, ontwikkel- en distributiepakketten en de bijbehorende SDK krijgen: daarnaast vindt u daar de stappen voor

het overzetten van de CubeProgrammer naar de MicroSD-kaart, een inleiding in het ontwikkelen van programma's, voorbeelden en "How to's" voor bepaalde softwarecomponenten, zoals de GTK voor graphics. ST stelt ook een interessant pakket voor het in de IoT-wereld bekende *AWS IoT Greengrass* ter beschikking, voor data-voorverwerking "at the edge" en de verbinding met de cloud. En dan is er nog een Linux-AI uitbreidingspakket onder de naam *X-LINUX-AI-CV*, dat onder andere objectherkenning en beeldclassificatie met kunstmatige neurale netwerken (KNN's) op applicatieniveau mogelijk maakt met behulp van het *TensorFlow Lite* Python-3 framework.

Eerste experimenten

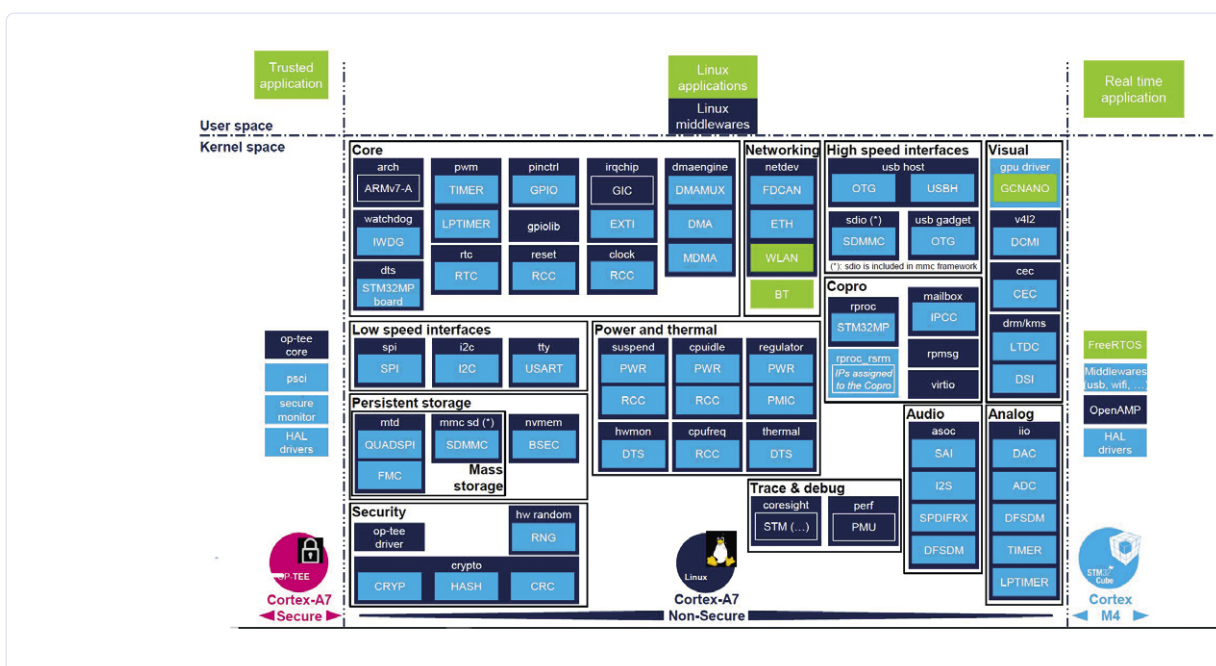
Afhankelijk van het opstartbare OS op de microSD-kaart, zult u daar een demoprogramma met enkele apps zien, enkele grafische demo's (zoals de videospeler met de mogelijkheid om via Bluetooth te verbinden met audioapparaten) en een prestatieweergave onder de naam *netdata*, die op een IP-adres in het lokale WLAN-netwerk is te bereiken. Om snel toegang tot het board te krijgen met uw geflashte microSD-kaart, kunt u uw (virtuele) Linux-PC gebruiken om via de seriële poort van de geïntegreerde ST-Link verbinding te maken met tools zoals *minicom* om zo de terminal van het MP1 Linux-systeem te zien en te bedienen. Ook een SSH-verbinding is mogelijk.

Vanaf de terminal kunt u het board opnieuw opstarten en de U-Boot-terminal betreden door tijdens het opstarten een willekeurige gebruikers-toets in te drukken. Geef daar het commando

```
$ums 0 mmc 0
```

zodat U-Boot het bestandssysteem op USB OTG vrijgeeft. Daarna kunt u vanaf de Linux-computer die is verbonden met OTG met commando's als

```
$mount | grep userfs
```



Figuur 6. De 'periferie'-drivers in alle drie de zones onder Linux op het HMP-gedeelte. De lichtblauw gemarkeerde bibliotheken worden geleverd door ST, de donkerblauwe bibliotheken zijn ontwikkeld door/met de community (bron: ST).

Listing 1. Gedeelte van het demoprogramma om de eenvoudige bediening van de GTK grafische bibliotheek te illustreren. Hier ziet u de basisstructuur van de infobalk aan de zijkant en een pictogram.

```
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_webserv = _load_image_eventBox(self, „%s/pictures/netdata-icon-192x192.png“ %
DEMO_PATH,
        "netdata", "perf monitor", -1, self.icon_size)
    eventBox_webserv.connect(„button_release_event“, self.wifi_hotspot_event)
    eventBox_webserv.connect(„button_press_event“, self.highlight_eventBox)
    ...
```

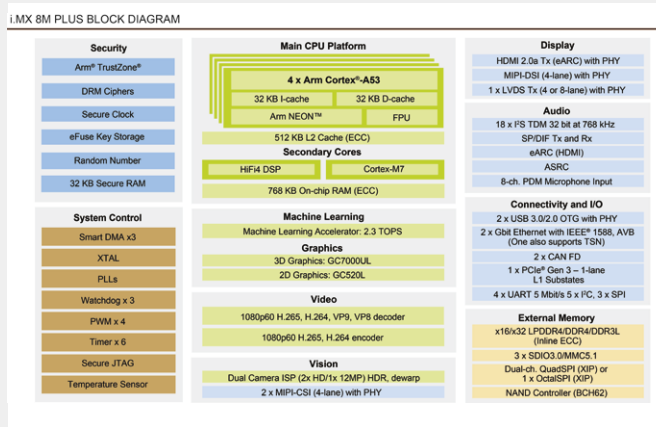
VOORBEELDEN VAN ANDERE EMBEDDED-HMP'S

De MP1 staat niet alleen op de huidige processormarkt, er zijn ook al andere embedded-HMP's gesignaleerd, met allerlei toepassingsgerichte specificaties.

Vanuit de consument gezien is de top van de embedded wereld te vinden in smartphone HMP's (afgezien van speciale toepassingen zoals autonoom rijden). Een actueel voorbeeld is de Exynos 9820 Octacore die in Europa in de Galaxy S10-serie wordt gebruikt, met twee kernen die door Samsung zelf zijn ontwikkeld (ze heten M4, maar hebben niets te maken met ARM-cortex!), twee A75- en vier A55-kernen.

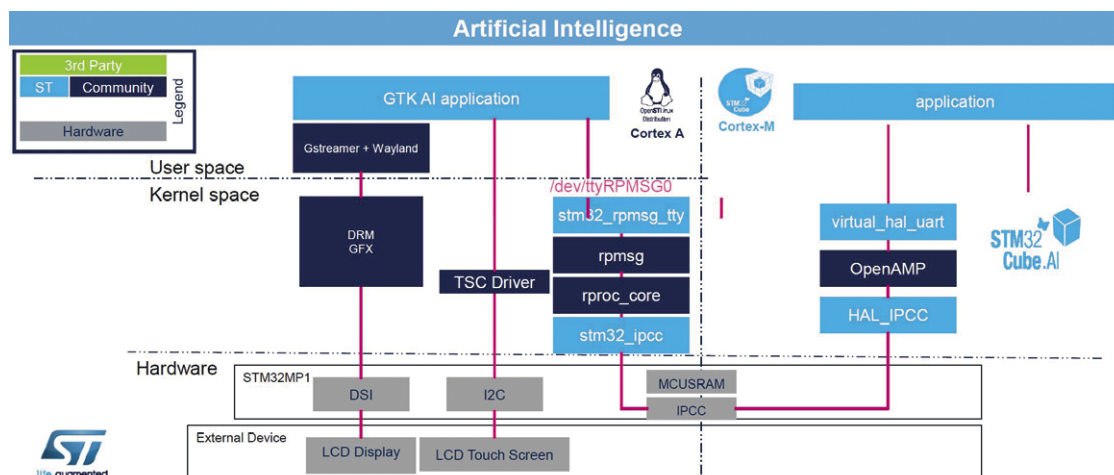
De RT1170 van NXP hoort tot het lagere prestatiesegment. Hij bestaat uit een M7- en een M4 kern en een 2D-GPU. Strikt genomen is het geen processor zoals de andere, maar een dubbele speciale µC met een extern flash-geheugen. Toch heeft hij een maximale klokfrequentie van 1 GHz (sic!) en volwaardige periferie en beveiliging.

TI betreedt de high-performance markt met onder andere de Sitara AM57x-familie. De multimedia- en ML-industrie worden genoemd als de belangrijkste toepassingen. Deze HMP heeft in de maximale configuratie twee A15-kernen, twee DSP-coprocessoren C66x, twee dubbele M4-kernen, een dubbele 3D-GPU, videooversnellers enzovoort. Een 4K-resolutie is met



deze resources niet onbereikbaar.

De HMP die het meest lijkt op de mogelijke verdere ontwikkeling van de STM32MP2 is gepresenteerd op CES2020: de i.MX 8M Plus van NXP (zie blokschema). Naast de vier A53-kernen en een M7-kern heeft hij twee grafische eenheden (2D/3D), een KNN-versneller, een dubbele camera-interface en ook TrustZone en andere speciale beveiligingsfuncties.



Figuur 7. Het werk van een droomteam: de AI touch-handschrietherkennings-applicatie op de M4 is via de *OpenAMP* of de *IPCC-HAL* verbonden met de *RPMsg* en de grafische *GTK*-applicatie op de A7-kernen en de GPU (bron: ST).

de verschillende gebieden van het bestandssysteem inspecteren en bijvoorbeeld de demo- of voorbeeldtoepassingen aanpassen. Voor het overdragen en uitvoeren van M4-programma's uit Linux kunt u vanuit de A7-Linux-view naar de submappen van `/sys/class/remoteproc/remoteproc0/` gaan.

Het voorbeeld in **figuur 7** laat zien hoe een programma dat gebruik maakt van beide kernwerelden van de HMP, eruit zou kunnen zien. Dit AI-programma is een van de apps in de demo en herkent letters en tekens die u met een vinger op het aanraakscherm tekent en voert indien nodig de bijbehorende applicaties uit. Het getrainde en door CubeAI geporteerde neurale netwerk bevindt zich in het M4-gebied, terwijl de gehele grafische HMI natuurlijk op de A7 of de GPU draait. Ook de interactie met de genoemde IPCC-interface en het gedeelde SRAM via het bekende *OpenAMP*-framework is goed te zien. De gebruikersinterface op het display kan met behulp van de GTK-bibliotheek zeer snel worden geïmplementeerd, zoals blijkt uit **listing 1**. Natuurlijk zijn er ook andere grafische frameworks voor verschillende functies zoals *GStreamer* of *Wayland/Weston*, en zelfs *QT* kan als meta-laag worden gebruikt.

Of u nu een nieuwkomer bent, een overstapper, of een Cortex-M of Linux-expert: met dit kleine markt- en documentatieoverzicht hebben we u een klein duwtje gegeven in de richting van de HMP-wereld die u nog niet kent. Nu is het punt bereikt waarop we u met een zuiver geweten uw reis in de HMP-wereld kunnen laten voortzetten! 

200072-02



IN DE STORE

- **Pyboard D-serie met STM32F722 en WiFi/BT:**
www.elektor.nl/pyboard-d-series-with-stm32f722-and-wifi-bt
- **Boek "Programming with STM32 Nucleo Boards":**
www.elektor.nl/programming-with-stm32-nucleo-boardsSTM32
- **Nucleo L476RG Board:**
www.elektor.nl/stm32-nucleo-l476rg-board

WEBLINKS

- [1] **Productpagina van de MPU-familie:** www.st.com/en/microcontrollers-microprocessors/stm32mp1-series.html
- [2] **STM32MP1 online-training:**
www.st.com/content/st_com/en/support/learning/stm32-education/stm32-online-training/stm32mp1-online-training.html
- [3] **Discovery Kit DK2 met STM32MP157C:** <https://bit.ly/37HXOF8>
- [4] **STM32CubeMP1-pakket:** <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 met OS-downloadinstructies:** wiki.st.com/stm32mpu