



Q&A

1. Gdzie można uzyskać slajdy i video link z dzisiejszej prezentacji?

Pełne nagranie ze szkolenia jest dostępne na stronie NXP, link został wysłany do wszystkich uczestników po zakończeniu szkolenia.

Filmy z nagraniem odcinkami szkolenia są także dostępne na [kanale YT](#) firmy SoMLabs.

2. Czy można udostępnić folder *excercise* poza maszyną wirtualną?

Wyekstrahowany z obrazu maszyny wirtualnej katalog *excercise* opublikowaliśmy na FTP pod adresem: <http://ftp.somlabs.com/Trainings/Hands-on-Linux-Academy-2020/ExcercisesDirectory-Linux-Workshop-10-2020.zip>

3. Czym jest layer/meta-layer w Yocto?

Opis zawarty w warstwie (*layer*) dostarcza instrukcji budowania określonego zestawu komponentów (tzw. receptury), konfiguracji sprzętu itp. Ponadto warstwy łączą receptury, które są w jakiś sposób powiązane, np. *meta-somlabs* dostarcza receptury związane z obsługą modułów charakterystycznych dla wersji systemu SoMLabs, a *meta-imx* zawiera opisy “wspierające” procesory NXP.

Temat jest dość rozbudowany, szczegóły znajdują się w dokumentacji Yocto:

<https://www.yoctoproject.org/docs/2.5/overview-manual/overview-manual.html>

<https://www.yoctoproject.org/docs/1.8/dev-manual/dev-manual.html>

4. Dlaczego akurat Yocto a nie Buildroot, który jest znacznie prostszy w użyciu?

Są dwa powody:

1. Yocto jest natywnie wspierane przez NXP, dzięki czemu w tym wariantcie systemu znajduje się duża liczba bibliotek i driverów z certyfikatami przemysłowymi, co ma duże znaczenie dla naszych dotychczasowych klientów.

2. Większość naszych dotychczasowych klientów samodzielnie zdecydowało się na Yocto i oczekuje od nas wspierania swoich działań.

5. Jakie kodeki audio są możliwe do zastosowania z iMX8Mmini?

Nie ma żadnych ograniczeń sprzętowych po stronie MPU. Procesor wyposażono w konfigurowalny 8-kanałowy interfejs PDM oraz 5 interfejsów SAI (*Synchronous Audio Interface*), które obsługują wiele różnych protokołów/formatów wymiany danych z kodekami (m.in. I2S, AC97, TDM oraz kodekowe interfejsy używane w DSP). Do naszych projektów wybraliśmy kodek NAU88C22YG firmy Nuvoton.

6. Jak dodać oprogramowania które ma być obecne w kompilowanym obrazie, np. Busybox, wersja biblioteki C itp.?

Pakiety do instalacji są definiowane m. in. w zmiennej `IMAGE_INSTALL` - tak samo jak przykładowa aplikacja na warsztatach. Dodatkowo pakiety podstawowe (jak np. biblioteka C) są zwykle definiowane w warstwach *meta-openembedded* i *meta-imx*. Szczegółów dotyczących ich wersji najlepiej szukać w źródłach tych warstw.

7. Czy dostępne są publicznie schematy/projekty płytek z modułami somlabs?

Dokumentacje opracowanych i produkowanych przez nas carrier boardów udostępniamy, także w postaci projektu przygotowanego w Altium Designerze, naszym klientom realizującym projekty przemysłowe.

8. Czy i gdzie jest dostępna zalecana przez SoMLabs receptura konfiguracji systemu Yocto dla modułów VisionSOM-8Mmini?

Konfiguracja modułów VisionSOM jest w omawianej na warsztatach warstwie *meta-somlabs*: <https://github.com/SoMLabs/imx-meta-somlabs/tree/zeus>

9. Czy firma SoMLabs przygotowuje i udostępni Debiana dla VisionSOM-8Mmini?

Tak, taki system będzie dostępny na wiki.somlabs.com w grudniu 2020.

10. Czy SoMLabs zajmuje się przygotowaniem systemów dostosowanych do potrzeb klientów?

Tak, świadczymy takie usługi naszym klientom.

11. Czy do i.MX8MMini można podłączyć wyświetlacz z interfejsem HDMI?

Do MPU nie można podłączyć bezpośrednio wyświetlacza z interfejsem HDMI. Dostępny jest konwerter MIPI->HDMI firmy NXP: <https://www.nxp.com/design/development-boards/i-mx-evaluation-and-development-boards/i-mx-8-series-accessory-boards:i.MX8-ACCESSORY-BOARDS>
Firma SoMLabs opracowała i produkuje konwerter MIPI->HDMI/LVDS, który można zastosować w przypadku korzystania z wyświetlaczy z interfejsem HDMI lub LVDS
<https://somlabs.com/product/mipi-to-lvds-hdmi-converter/>

12. Jak modyfikować istniejące już wpisy w *device tree*? Czy za pomocą *devtool* jestem w stanie zmienić już dodany wpis na inny i w jaki sposób? (Część już jest dodana ze statusem *disabled* jak na przykład MIPI i CSI).

Modyfikację wszystkich źródeł, w tym *device tree* można zrobić na przykład tak jak w ćwiczeniu dotyczącym konfiguracji interfejsów UART/I2C/SPI.

13. Jakie narzędzia polecacie Państwo do tworzenia aplikacji „okienkowych”. Interesuje mnie również wyświetlanie strumieni video.

W naszych projektach używamy obecnie GTK+ oraz *gststreamer*.

14. Czy można udostępnić pliki do ćwiczeń poza maszyną wirtualną? Yocto skompilowałem natywnie, chciałbym mieć źródła.

Wykestrahowany z obrazu maszyny wirtualnej katalog *excercises* opublikowaliśmy na FTP pod adresem: <http://ftp.somlabs.com/Trainings/Hands-on-Linux-Academy-2020/ExcercisesDirectory-Linux-Workshop-10-2020.zip>

15. Jak podłączyć JTAG na starszej wersji był dostępny (chodzi o moduł VisionSOM-RT)?

Używany podczas warsztatów zestaw VisionCB-8M-STD ma wbudowany debugger JTAG J-Link firmy Segger. W zestawie VisionCB-RT-STD zastosowaliśmy standardowe złącza IDC20 dla JTAG oraz ETM, do których można dołączyć dowolny interfejs typu Segger J-Link, ARM uLink itp.

16. Co z interfejsem I2S? Co z audio przez I2S, czy są wspierane kodeki?

Procesor i.MX8Mmini jest wyposażony w 8-kanłowy interfejs PDM oraz 5 interfejsów SAI, które mogą pracować jako I2S. Konfiguracja tych interfejsów wymaga odpowiednich zapisów w *device tree*, pomocne w tym może być bezpłatne narzędzie NXP - [iMXPinTool](#).

17. Czy złącze PiHat jest w 100% zgodne z PI?

Dołączenie linii GPIO procesora do złącza dla hatów Raspberry Pi jest takie samo, jak w przypadku pierwowzoru. Jedyną różnicą są linie IS_SD i ID_SC – na naszym carrier boardzie nie zostały one dołączone do MPU.

18. Czy istnieje jakiś edytor plików device tree wspomagający ich tworzenie?

Jest narzędzie, które częściowo realizuje to zadanie – jest to bezpłatny konfigurator multiplexerów IO w MPU firmy NXP – [iMXPinTool](#).

19. Czy ładowanie kodu do rdzenia Cortex-M4 nie da się zrealizować w trakcie działania Linuxa? Np wyładowanie modułu jądra, załadowanie kody na odpowiednią partycję i załadowanie modułu jądra w trakcie pracy OS?

W przygotowanym przez NXP BSP dla i.MX8Mmini program dla rdzenia Cortex-M4 jest ładowany przez u-boota poleceniem *bootaux* (źródła: https://source.codeaurora.org/external/imx/uboot-imx/tree/arch/arm/mach-imx/imx_bootaux.c?h=imx_v2020.04_5.4.47_2.2.0). Takie samo polecenie można wywołać w sterowniku w jądrze Linuxa.

20. Czy wyświetlacz RGB z zestawu z i.MX6ULL pasuje też do i.MX8Mmini?

Mikroprocesor i.MX8Mmini jest wyposażony w nowoczesny, różnicowy interfejs wyświetlacza – MIPI-DSI. Bez użycia konwertera interfejsów, wyświetlacz z interfejsem równoległym RGB nie da się dołączyć do i.MX8Mmini.

21. Czy można wymieniać kod Cortex-M4 z poziomu Linuxa działającego na SOMie?

W przygotowanym przez NXP BSP dla i.MX8Mmini program dla rdzenia Cortex-M4 jest ładowany przez u-boota poleceniem *bootaux* (źródła: https://source.codeaurora.org/external/imx/uboot-imx/tree/arch/arm/mach-imx/imx_bootaux.c?h=imx_v2020.04_5.4.47_2.2.0). Takie samo polecenie można wywołać w sterowniku w jądrze Linuxa.

22. Czy wyjście "Composite video" jest dużym problemem dla IMX8M Mini?

Procesor jest wyposażony w różnicowy interfejs wyświetlacza MIPI-DSI. Za pomocą zewnętrznych konwerterów sprzętowych można „przejąć” z MIPI-DSI na praktycznie dowolny inny format, w tym *composite video*.

Przykładowo: firma SoMLabs opracowała i produkuje konwerter MIPI->HDMI/LVDS, który można zastosować w przypadku korzystania z wyświetlaczy z interfejsem HDMI lub LVDS <https://somlabs.com/product/mipi-to-lvds-hdmi-converter/>

23. W środowisku Delphi od jakiegoś czasu można tworzyć aplikacje nie tylko na Windows, ale również na Androida, MacOS i Linux Ubuntu. Osobiście to przetestowałem. Czy mają Państwo wiedzę czy uda się uruchomić na imx8Mmini Yocto aplikacje wytworzone w Delphi?

Nie znamy tego narzędzia i nie testowaliśmy poprawności działania aplikacji generowanych za jego pomocą.

24. Dzień dobry, chciałbym zapytać czy SoMLabs planuje udostępnienie modułów SOM z procesorem i.MX 8MPlus. Jeśli tak to w jakim terminie?

Tak, taki moduł wprowadzimy do oferty handlowej w pierwszym kwartale 2021 roku.

25. Jak developować i debugować (JTAG) aplikacje na Cortexie?

Odpowiedź na to pytanie wymaga pełnowymiarowego szkolenia, osobno w odniesieniu do: rdzenia aplikacyjnych Cortex-A i *realtime* Cortex-M. Tymczasem możemy polecić materiały przygotowane przez firmę Segger (producenta interfejsów do programowania-debugowania *via* JTAG/SWD/ETM), które są dostępne bezpłatnie na stronie wiki.segger.com.

26. Czy rpmsg to jedyna forma komunikacji między rdzeniami?

Nie, jest możliwość wymiany danych pomiędzy domenami za pomocą pamięci współdzielonej, ale dotychczas nie testowaliśmy tego rozwiązania.

27. Czy zawsze jest tak że pogram na Cortex-M4 nie może się sam uruchomić, tylko zawsze przez u-boota?

W i.MX8Mmini tak.

28. Czy przewidywane są carrier boardy z obsługą audio, ale nie przez pwm, tylko w lepszej jakości 44k/16 bit minimum? Bez audio dzisiejsze aplikacje multimedialne są raczej niewykonalne. Przydało by się przynajmniej wejście/wyjście stereo i mikrofon mono.

Tak, dla i.MX8Mmini przygotowaliśmy projekt zaawansowanego carrier boarda, wyposażonego m.in. w stereofoniczny kodek NAU88C22YG firmy Nuvoton. Jest to 24-bitowy konwerter ADC/DAC zintegrowany ze wzmacniaczem mocy (1W/8Ω), wzmacniaczem słuchawkowym, podsystemem DSP, przedwzmacniaczem mikrofonowym itp.

Do procesora i.MX8MMini można ponadto bezpośrednio dołączyć do 8 mikrofonów cyfrowych lub innych urządzeń audio z wyjściami PDM.

29. Czy którykolwiek z HAT z DAC lub audio jest wspierany przez system (hifiberry, WM8961 itp.)?

W doborze podzespołów używanych w naszych projektach i ich wsparciu technicznym kierujemy się przede wszystkim gwarancją ich długotrwałej dostępności, czego dość często nie uwzględniają firmy oferujące swoje rozwiązania na rynku komercyjnym. Przykładem jest właśnie kodek WM8961, który miał rynkową premierę w 2009 roku a od 2012 nie jest oferowany w sieciach dystrybucji.

Biorąc pod uwagę ukierunkowanie działań SoMLabs na rynki przemysłowe, skupiamy się wspieraniu rozwiązań akceptowalnych w przemyśle.

30. Czy dostępne są materiały ułatwiające projektowanie własnych carrier-boardów?

Takie materiały są dostępne w postaci:

- Bezpłatnych bibliotek dla Altium Designera, zawierających symbol schematowy, footprint oraz model 3D każdego z produkowanych przez nas SOM-ów. Biblioteki są dostępne na stronie www.somlabs.com w zakładce *Documentation* na podstronach z opisami SOM-ów.
- Dokumentacji w formacie Altium Designer do opracowanych i produkowanych przez SoMLabs carrier boardów. Dokumentacje nie są dostępne bezpośrednio na stronie, udostępniamy je klientom na żądanie po kontakcie e-mailowym.

31. Czy istnieje na tę płytkę biblioteka podobna do WiringPi z Raspberry Pi?

Takie rozwiązanie nie jest nam znane, natomiast portowanie WiringPi na inne platformy niż RPi, ze względu na decyzje autora tego oprogramowania, nie jest obecnie możliwe.

32. Jak szybko można przełączać stan pinu za pomocą żądań REST (HTTP)?

Jest to zależne od wielu czynników, m.in. opóźnień transmisji danych w sieci, prędkości obsługi zapytań przez framework, ponadto prędkości operacji plikowych w Linuksie. Bez specjalnych optymalizacji, przy bezpośrednim połączeniu komputera-klienta do serwera zaimplementowanego na module VisionSOM-8Mmini w sposób pokazany podczas szkolenia, możliwe jest uzyskanie częstotliwości do kilku Hz.

33. Kiedy będzie się można zgłosić po środowisko do uruchomienia bezpośrednio pod Linuxem bez maszyny wirtualnej

Konfiguracja środowiska była omówiona w pierwszym odcinku szkolenia. Szczegóły można znaleźć również na naszym *wiki* i *githubie*:

<https://github.com/SoMLabs/imx-meta-somlabs/tree/zeus>

https://wiki.somlabs.com/index.php/VisionSOM_imx-meta-somlabs

34. Czy prawdą jest, że i.MX6 ma być wycofywany z produkcji? Do kiedy jest planowana jego produkcja?

Zakładamy, że pytanie dotyczy i.MX6ULL, który został użyty w module SoMLabs VisionSOM-6ULL. Procesor ten pojawił się oficjalnie na rynku w październiku 2016 roku, gwarantowany przez NXP czas jego dostępności wynosi 15 lat od tej daty. Firma NXP udostępnia deklaracje dostępności na swojej stronie internetowej [pod adresem](#).

35. Czy będzie możliwość zadawania dodatkowych pytań do ćwiczeń po zakończeniu tego cyklu/po własnoręcznym przerobieniu ćwiczeń?

Będziemy się starali odpowiedzieć na wszystkie pytania, prosimy o kontakt przez formularz na stronie <http://www.somlabs.com>.

36. Czy na wasze produkty jest planowane portowanie CODESYS?

CODESYS jest bardzo rozbudowanym środowiskiem programistycznym z rynku automatyki przemysłowej, składającym się z wielu modułów „tematycznych”. Jako firma zajmująca się produkcją sprzętu nie mamy możliwości portowania wszystkich interesujących, zwłaszcza tak rozbudowanych, środowisk na nasze platformy sprzętowe. Możemy natomiast pomóc w przygotowaniu carrier boarda, który będzie funkcjonalnym odpowiednikiem PLC.

37. Czy system RTOS MQX jest jeszcze oferowany?

System MQX jest wspierany przez NXP, szczegóły można znaleźć pod linkiem: https://www.nxp.com/design/software/embedded-software/mqx-software-solutions:MQX_HOME

38. Jaką maksymalną liczbę interfejsów SPI, I2C, CAN i UART można jednocześnie używać w i.MX8Mmini?

Zgodnie ze specyfikacją, procesor i.MX8Mmini jest wyposażony w 4 interfejsy I2C, 3 interfejsy uSDHC, 4 interfejsy UART oraz 3 interfejsy SPI. Nie ma wbudowanego CAN, możliwa jest natomiast jego tania implementacja na zewnątrz.

Co do liczby jednocześnie aktywnych interfejsów – jest to zależne od konfiguracji multiplexerów IO, w czym pomocne jest bezpłatne narzędzie NXP - [iMXPinTool](#). Za pomocą tego narzędzia można dobrać i zweryfikować optymalne konfiguracje interfejsów komunikacyjnych.

39. Czy J-Link wbudowany w carrier board VisionCB-8M-STD jest licencjonowany?

Tak, dzięki współpracy SoMLabs z firmą Segger interfejsy J-Link implementowane w naszych carrier boardach mają pełne licencje, własne oryginalne numery seryjne i mogą być używane w pracach komercyjnych.

40. Czy jest dostępne jakieś darmowe środowisko do programowania Cortex-M4 w i.MX8Mmini?

NXP oferuje dla mikrokontrolerów Cortex-M bezpłatne środowisko MCUXpresso (IDE+ConfigTool), które można pobrać ze strony producenta.

41. Jaka jest kolejność startu rdzeni Cortex-A53 i Cortex-M4 w i.MX8Mmini?

Jako pierwsza startuje domena aplikacyjna, której bootloader ładuje do pamięci RAM oprogramowanie wykonywane przez rdzeń Cortex-M4.

42. Czy Python Flask umożliwia korzystanie z innych zasobów MPU niż GPIO – mam na myśli układy z SPI lub I2C?

Tak, wszystkie interfejsy komunikacyjne procesora są dostępne dla aplikacji napisanych w Python Flask.

43. Czy moduł VisionSOM z i.MX8Mmini ma wyprowadzenia zgodne z wcześniejszym modelem VisionSOM-6ULL?

Format mechaniczny wszystkich modułów VisionSOM jest taki sam – SODIMM200, ale układ wyprowadzeń modułów VisionSOM-8Mmini jest inny niż VisionSOM-6ULL. Kompatybilności nie udało się zachować, m.in. ze względu na charakterystykę napięciową interfejsu Ethernet i kilku innych wyprowadzeń IO, należących do różnych domen napięciowych MPU.

44. Czy projektor DLP, którego użycie pokazaliście na jednym z filmów na kanale youtube, można bezpośrednio dołączyć do i.MX8Mmini?

Opracowany przez nas projekt demonstracyjny, który przedstawiliśmy na youtube, bazuje na projektorze DLPDLR2000EVM. Jest on wyposażony w interfejs równoległy RGB, który jest bezpośrednio obsługiwany m.in. przez nasze moduły VisionSOM-RT oraz VisionSOM-6ULL, których procesory są wyposażone w interfejsy równoległe. Procesor i.MX8Mmini nie ma takiego interfejsu, konieczne jest więc użycie modułu DLP z interfejsem MIPI-DSI.

45. Jaki procesor używacie w modułach SoM podczas szkolenia? Na filmie ze szkolenia nie widać zamontowanego radiatora, czy nie jest potrzebny?

Domyślnie moduły VisionSOM-8Mmini są wyposażane w procesory 4-rdzeniowe (4xCortex-A53 @1.8GHz + Cortex-M4F) w komercyjnej wersji temperaturowej. Eksperymenty w naszym laboratorium potwierdziły informacje przygotowane przez firmę NXP w notcie aplikacyjnej AN12410. Wskazują one na to, że procesory i.MX8MMini są energooszczędne w realnych aplikacjach i nie wydzielają nadmiernych ilości ciepła, co pozwala w wielu aplikacjach zrezygnować ze stosowania elementów wspomagających odprowadzanie ciepła.

Warto zwrócić uwagę, że procesory i.MX8MMini wyposażono w mechanizm minimalizujący wydzielanie ciepła, który dostosowuje częstotliwość taktowania rdzeni do wymogów aplikacji (im większa potrzebna moc obliczeniowa, tym wyższa częstotliwość taktowania CPU).

Ponadto i.MX8MMini wyposażono w wewnętrzny czujnik temperatury, który dostarcza systemowi operacyjnemu informacji o aktualnej temperaturze struktury półprzewodnikowej (T_j), dzięki czemu możliwe jest automatyczne obniżanie częstotliwości taktowania w przypadku zbyt wysokiej temperatury struktury.

Jak duże ma znaczenie dla czasu bezawaryjnej pracy procesora ma minimalizowanie temperatury struktury, firma NXP informuje użytkowników za pomocą noty aplikacyjnej AN12468.

Działanie wymienionych mechanizmów w praktyce można przeanalizować na przygotowanym przez nas [filmie](#).