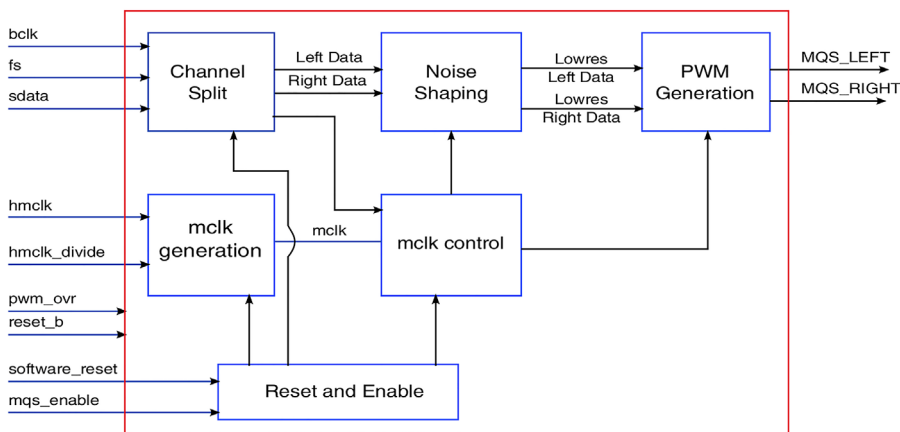


Strumieniowy odtwarzacz audio na i.MX6ULL

W systemach mikroprocesorowych często zachodzi potrzeba porozumienia się z użytkownikiem nie tylko za pomocą obrazu i dotyku, ale również za pomocą dźwięku. Nie zawsze musi to być dźwięk o najwyższej jakości, szerokiej dynamice i pozbawiony zniekształceń. Do zwrócenia uwagi użytkownika na wyświetlany na ekranie komunikat, udźwiękowania zapisanego w pamięci urządzenia filmiku instruktażowego, odtworzenia mowy lub muzyki z niewielkich, wbudowanych głośniczków, w zupełności wystarczy jakość porównywalna z radioodbiornikiem lub kasetą magnetofonową. Do takich zastosowań można wykorzystać tani układ wzmacniacza audio i wbudowany w procesory i.MX6 przetwornik MQS.

Procesory z rodziny i.MX, w tym procesor i.MX6ULL, do którego odnosi się ten artykuł, wyposażone są w liczne sprzętowe peryferia obsługi dźwięku:

- **SAI** – interfejs zgodny z I²S/AC97/TDM,
- **ASRC** – synchronizuje i konwertuje sygnały o różnej częstotliwości próbkowania,



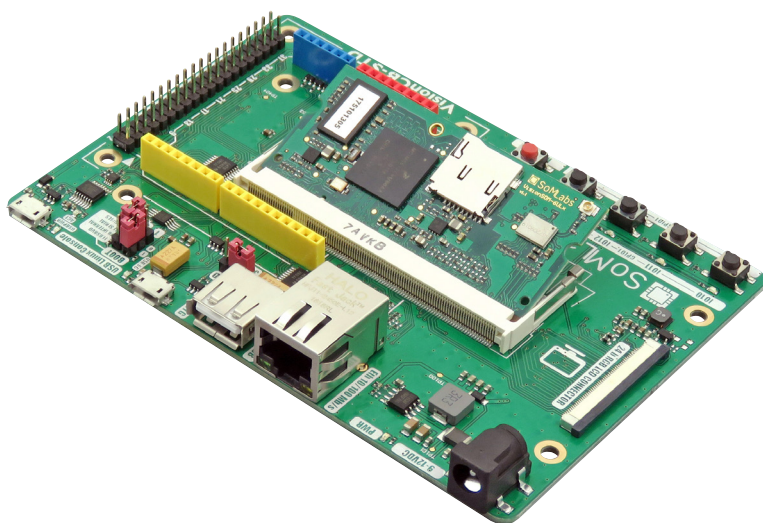
Rysunek 1. Schemat blokowy MQS

- **SPDIF** – nadajnik i odbiornik S/PDIF,
- **MQS** – stereofoniczny przetwornik DAC typu PWM, którego schemat blokowy pokazano na **rysunku 1**.

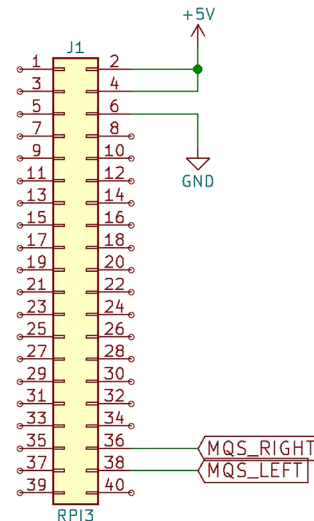
W prezentowanym projekcie pokażemy sposób użycia przetwornika C/A MQS, który jest 2-kanalowym, 1-bitowym przetwornikiem typu sigma-delta.

Strumień próbek nadchodzący z magistrali I²S jest rozdzielany na lewy i prawy kanał, następnie sygnały w obu kanałach są nadpróbkowane (z wielokrotniane). Ze względu na małą rozdzielczość generatora PWM dla zwielokrotnionych próbek zastosowano technikę *noise shaping*. Technika ta polega na modulacji wartości najmniej znaczących bitów każdej próbki tak, aby po uśrednieniu przez zewnętrzny filtr dolnoprzepustowy odzyskać część dynamiki sygnału. Na jedną próbkę sygnału wejściowego przypada zatem wiele okresów PWM, każdy o nieznacznie innym stopniu wypełnienia.

Wewnątrz procesora i.MX6ULL blok MQS jest podłączony do jednego z wyjść SAI. Nie ma więc potrzeby osobnego konfigurowania w sterowniku buforów i kolejek odtwarzania, wystarczy skonfigurować SAI tak, jak miałyby to miejsce w przypadku podłączenia zewnętrznego kodeka audio do magistrali I²S. Rejestry konfiguracyjne MQS pozwalają jedynie ustawić dzielnik zegara *mclk* i liczbę cykli *mclk* przypadającą na jeden okres PWM.



Fotografia 2. Komputer SoMLabs VisionCB-STD z modułem VisionSOM-6ULL



Rysunek 3. Rozmieszczenie sygnałów wyjściowych MQS na złączu RPi3 na płycie VisionCB

W domyślnej konfiguracji częstotliwość PWM wynosi 768 kHz. Każdej próbie sygnału wejściowego o częstotliwości próbkowania równej 48 kHz odpowiada szesnastoscie okresów PWM.

Połączenia

Uruchomienie MQS zaprezentujemy na przykładzie modułu VisionSOM-6ULL i płyty bazowej VisionCB-STD firmy SoMLabs (fotografia 2).

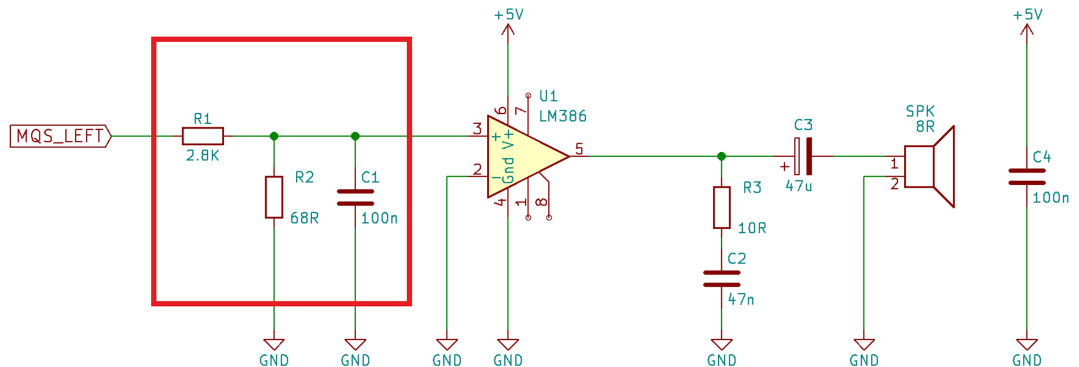
Wyjścia MQS_LEFT i MQS_RIGHT to jedyne sygnały przetwornika dostępne na zewnątrz układu (na zewnątrz procesora, nie na zewnątrz bloku MQS). Z dokumentacji dowiemy się, że można je wyprowadzić na fizycznych padach o nazwach GPIO1_IO01/00, JTAG_TDI/TDO lub LCD_DATA23/22. Wykorzystamy parę padów JTAG_TDI i JTAG_TDO, aby uzyskać dźwięk stereo.

Obydwa sygnały wyprowadzone są na pinach 36 i 38 złącza zgodnego z Raspberry Pi 3 na płycie VisionCB-STD (rysunek 3).

Filtr i wzmacniacz

Schemat na rysunku 4 przedstawia przykładowy, prosty jednokanałowy wzmacniacz, oparty na popularnym układzie LM386 i jego referencyjnej aplikacji. Wzmacniacz pracuje ze wzmocnieniem około 20×. Dzielnik R1–R2 ogranicza amplitudę sygnału wejściowego, aby nie przesterować wejścia wzmacniacza – na wyjściach MQS występuje przebieg PWM o wartości międzyszczytowej ok. 3,3 V.

Ze względu na to, że w prezentowanym rozwiązaniu zastosowaliśmy przetwornik jednobitowy sigma-delta, pomiędzy wyjściem



Rysunek 4. Schemat elektryczny jednego kanału wzmacniacza mocy z filtrem dolnoprzepustowym (czerwona ramka)

mikroprocesora a wejściem wzmacniacza powinien znaleźć się filtr, który wytlumi ze zmodulowanego sygnału częstotliwość PWM (768 kHz) i jej harmoniczne.

Prosty filtr dolnoprzepustowy RC zrealizowano na dzielniku R1–R2 oraz kondensatorze C1 (zaznaczony czerwonym prostokątem na rysunku 4). Częstotliwość –3 dB filtra dolnoprzepustowego obliczamy jak przy połączeniu równoległym rezystorów R1 i R2, wynosi ona ok. 23 kHz. Ze względu na wysoką częstotliwość PWM w stosunku do pasma sygnału audio, filtr pierwszego rzędu można uznać za wystarczający.

W zależności od zastosowanego głośnika i napięcia zasilającego wzmacniacz konieczna może okazać się zmiana wartości elementów R1, R2 i C1, aby dopasować poziom głośności. Dokumentacja układu LM386 opisuje sposób regulacji wzmocnienia przez dołączenie szeregowo kondensatora 10 μF i potencjometru między pinami 1 i 8 wzmacniacza.

Jeżeli chcemy regulować głośność za pomocą systemowego miksera w Linuxie,

najlepszym rozwiązaniem będzie zastosowanie wzmacniacza wyposażonego w magistralę sterującą. Niektóre programy odtwarzające oraz system dźwięku PulseAudio w systemie Linux pozwalają regulować głośność przez programowe przeskalowanie próbek przed zapisaniem ich do bufora odtwarzania. Takie rozwiązanie nie wymaga zmian sprzętowych, jednak może wiązać się z pogorszeniem jakości dźwięku.

Na rysunku 5 przedstawiono zrzut ekranu z oscyloskopu w trybie 'zoom', który ilustruje przebiegi:

- sonda kanału czwartego (przebieg zielony) jest podłączona do wyjścia MQS_LEFT,
- sonda kanału trzeciego (przebieg różowy) jest podłączona do wyjścia wzmacniacza mocy.

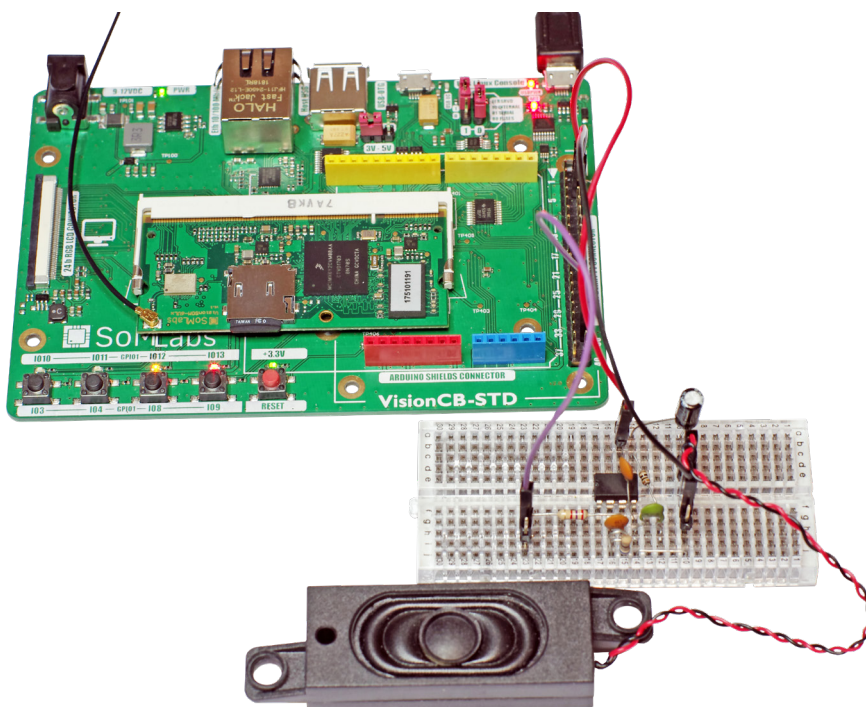
Resztki sygnału PWM są dostrzegalne w sygnale wyjściowym, jednak jego amplituda jest znacznie wytłumiona

Oprogramowanie

Firma SoMLabs na swojej Wiki udostępnia do pobrania materiały pozwalające uruchomić system Debian GNU/Linux w wersji 9 'Stretch'.

Do realizacji projektu potrzebne będą dwa pliki:

- obraz karty pamięci (*debian-stretch-visionsom-6ull.img.xz*) – z niego startuje mikroprocesor,



REKLAMA

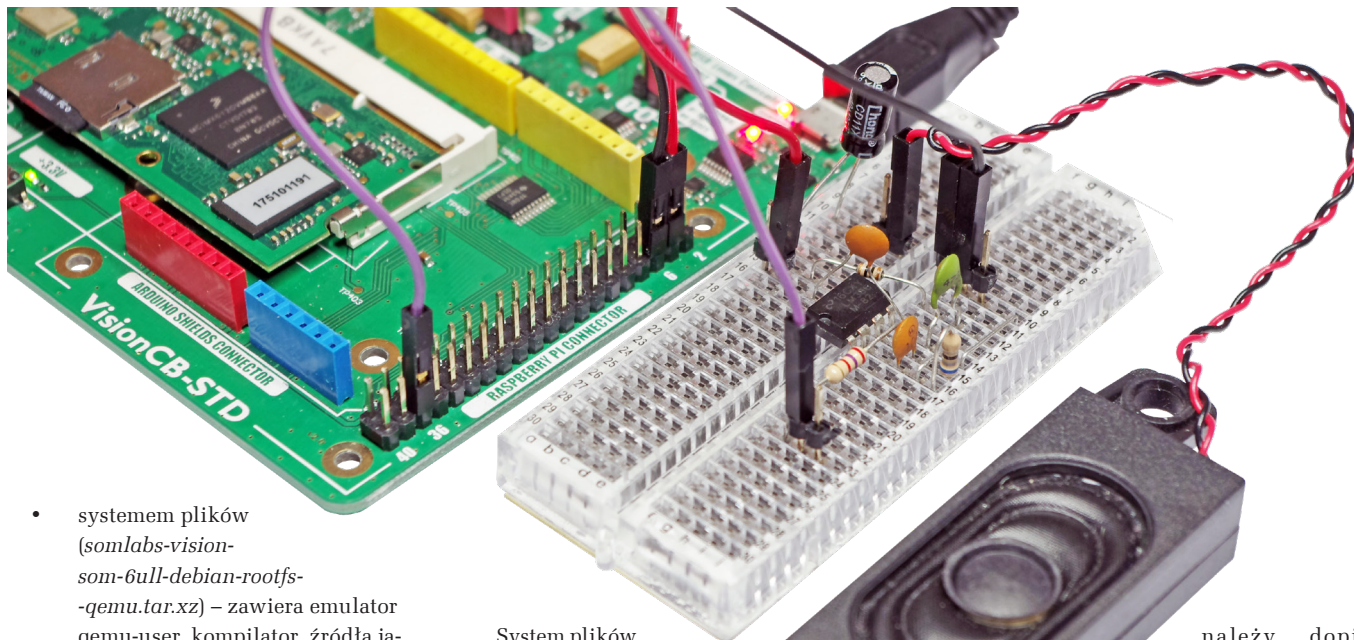
Specjalistyczne szkolenia dla elektroników i automatyków



TECHDAYS

techdays@techdays.pl
TECHDAYS.PL

CERTYFIKOWANY PARTNER SZKOLENIOWY
ST
life.augmented



- systemem plików (*somlabs-vision-som-6ull-debian-rootfs-qemu.tar.xz*) – zawiera emulator *qemu-user*, kompilator, źródła jądra oraz pliki *devicetree* dla platformy VisionSOM.

Sposób uruchomienia systemu Debian na platformie VisionSOM przy wykorzystaniu obrazu karty pamięci opisano w serii artykułów *QuickStart* na SoMLabs Wiki.

Jądro Linuxa, które jest w obrazie karty pamięci, zawiera już sterownik dla MQS. Nie ma więc konieczności jego ponownej kompilacji. Jeżeli dysponujemy własnym obrazem systemu i kompilowaliśmy jądro samodzielnie, wówczas warto upewnić się, że opcja *SND_SOC_IMX_MQS* jest aktywna [=y].

W konfiguratorze *menuconfig* (rysunek 6) wybranej opcji odpowiada etykieta „SoC Audio support for i.MX boards with MQS”. Znajdziemy ją w „Device Drivers/Sound architecture/ALSA for SoC audio support/SoC Audio for Freescale CPUs”.

Kod źródłowy sterownika MQS jest zawarty w plikach *sound/soc/codecs/fsl_mqs.c* oraz *sound/soc/fsl/imx_mqs.c*.

System plików ściągnięty z SoMLabs Wiki posłuży nam do wygenerowania nowego drzewa urządzeń (*devicetree*), które podczas rozruchu informuje jądro, jakie peryferia są dostępne w systemie, w jaki sposób są przyłączone i którego sterownika użyć do ich obsługi.

Domyślnym plikiem *devicetree* dla zestawu VisionSOM-6ULL i VisionCB-STD jest *somlabs-visionsom-6ull.dts* w katalogu */home/developer/source/somlabs-dts-1.0/*. Na **listingu 1** przedstawiono ustawienia, które

należy dopisać do *devicetree*. W ustawieniach multipleksowania wyprwadzeń procesora (*pinmux*) należy określić właściwą funkcję i parametry pinów, do których podłączymy wzmacniacz mocy. Konfigurację pokazano w **listingu 2**. Jeżeli piny *JTAG_TDI* i *JTAG_TDO* są już

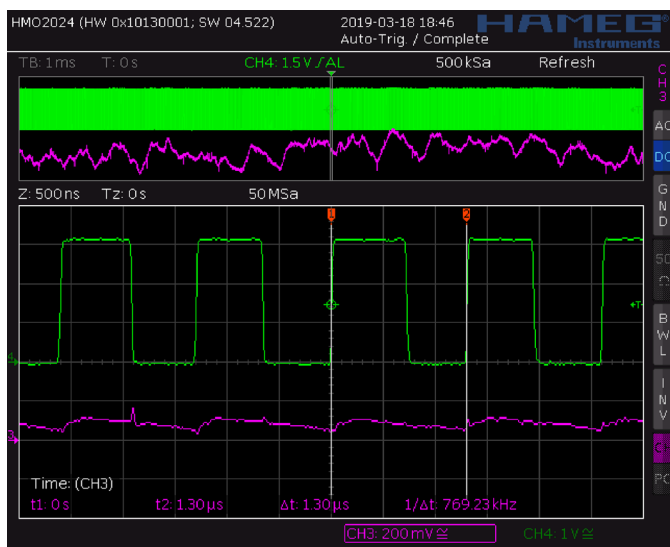
Listing 1. Modyfikacje niezbędne do wprowadzenia w domyślne *devicetree*

```

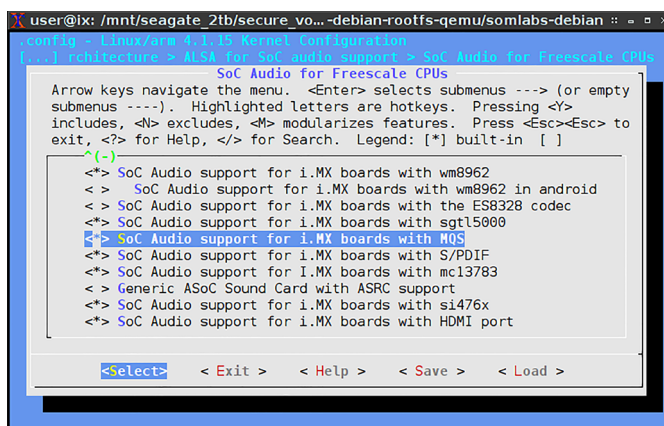
{
    sound-mqs {
        compatible = „fsl,imx-audio-mqs”;
        model = „mqs-audio”;
        cpu-dai = <&sai1>;
        asrc-controller = <&asrc>;
        audio-codec = <&mqs>;
    };
}

&sai1 {
    assigned-clocks = <&clks IMX6UL_CLK_SAI1_SEL>,
    <&clks IMX6UL_CLK_SAI1>;
    assigned-clock-parents = <&clks IMX6UL_CLK_PLL4_AUDIO_DIV>;
    assigned-clock-rates = <0>, <24576000>;
    status = „okay”;
};

&mqs {
    pinctrl-names = „default”;
    pinctrl-0 = <&pinctrl_mqs>;
    clocks = <&clks IMX6UL_CLK_SAI1>;
    clock-names = „mclk”;
    status = „okay”;
};
    
```



Rysunek 5. Zrzut ekranu z oscyloskopu z przebiegami: zielony – wyjście *MQS_LEFT*, różowy – wyjście wzmacniacza mocy



Rysunek 6. Okno konfiguratora *menuconfig*

Listing 2. Niezbędne modyfikacje konfiguracji pinmux w domyślnym devicetree

```

&iomuxc
{
    /* ... */
    imx6ul-evk
    {
        pinctrl_hog_1: hoggrp-1
        {
            fsl,pins = <
            /* ... */
            /* MX6UL_PAD_JTAG_TDI__GPIO1_I013          0x17099 /* LED 2 */
            /* MX6UL_PAD_JTAG_TDO__GPIO1_I012          0x17099 /* LED 3 */
            /* ... */
            pinctrl_mqs: mqsgrp
            {
                fsl,pins = <
                MX6UL_PAD_JTAG_TDI__MQS_LEFT          0x11088
                MX6UL_PAD_JTAG_TDO__MQS_RIGHT         0x11088
            }
        }
    }
};

```

skonfigurowane w innej części *devicetree* (np. w *pinctrl_hog_1*), odpowiednie linie należy usunąć lub opatrzyć znakiem komentarza, aby uniknąć konfliktu podczas uruchamiania systemu. Firma NXP udostępnia narzędzie i.MX Pins Tool, które ułatwia wprowadzanie większych zmian w konfiguracji multipleksowania pinów mikroprocesora.

Po wprowadzeniu zmian plik *devicetree* skompilujemy poleceniem *make*. Kompilację przeprowadzimy w środowisku *chroot*

z zainstalowanym emulatorem, a więc przy użyciu natywnego kompilatora dla architektury ARM:

```

sudo chroot somlabs-debian/
cd home/developer/source/
somalabs-dts-1.0/
make somlabs-visionsom-6ull.dtb

```

Gotowe pliki *dts* i *dtb* zostały udostępnione na stronie <http://co.rru.pt/somalabs/mqs/>.

Nasze nowe, zmodyfikowane, a następnie skompilowane drzewo urządzeń – *somalabs-visionsom-6ull.dtb* – powinno

znaleźć się w katalogu */boot* na karcie pamięci.

Można teraz uruchomić system i przejść do testowania dźwięku z MQS. Proponuję wykorzystać do tego celu *mpg123* – program do odtwarzania dźwięku sterowany z wiersza poleceń. W systemie Debian program *mpg123* dostępny jest w repozytoriach *apt*:

```

root@somalabs:~# apt-get install
mpg123

```

Program *mpg123* odtwarza zarówno muzykę w formacie MP3 z plików znajdujących się na karcie pamięci, jak i odbiera internetowe stacje radiowe nadające strumień MP3 w standardzie *ShoutCast* (rysunek 7):

```

root@somalabs:~# mpg123 test.mp3
root@somalabs:~# mpg123 -@ http://
ant-waw-01.cdn.eurozet.pl:8602/
listen.pls

```

Poziom głośność regulujemy, wpisując w konsoli znaki '+' i '-' w trakcie odtwarzania; 'h' wyświetla pomoc a 'q' – kończy działanie programu.

Podsumowanie

Wbudowany w mikroprocesory i.MX6ULL interfejs audio MQS pozwala niewielkim kosztem, łatwo wzbogacić projektowane przez nas urządzenie w dźwięk. Rozwiązanie jest zgodne z większością oprogramowania w systemie Linux, a dzięki dostępowi do repozytoriów Debiana, po podłączeniu wyświetlacza łatwo stworzymy prawdziwie multimedialny interfejs użytkownika.

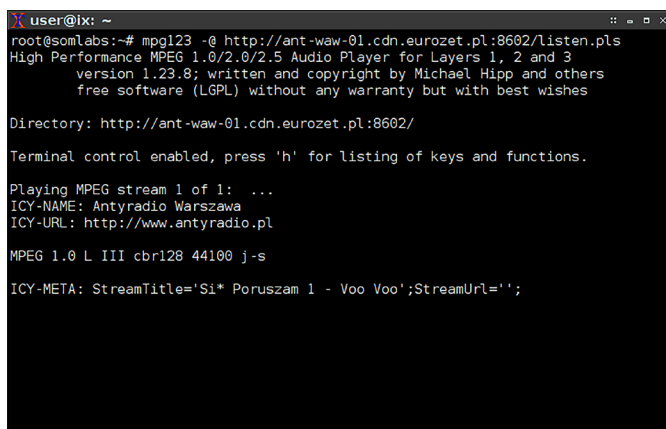
Wnikliwych Czytelników zachęcam do eksperymentów z ustawieniami MQS, filtrami i wzmacniaczami. W kolejnym artykule zaproponujemy coś dla miłośników muzyki – podłączymy do i.MX6ULL wysokiej jakości zewnętrzny przetwornik i niebanalny wzmacniacz.

Witek Ewert

witek.ewert@nxp.com

Bibliografia:

1. i.MX6ULL Reference Manual (IMX6ULLRM) 4.1.1 Muxing Options
2. <http://bit.ly/2UO4db9>
3. <http://bit.ly/2Ofz4uU>
4. <http://bit.ly/2Wjt6w3>
5. i.MX Pins Tool <http://bit.ly/2WeOgeG>



Rysunek 7. Widok okna konsoli systemowej podczas odtwarzania internetowej wersji Antyradio

REKLAMA

KITy AVT

O KITach AVT
przeczytasz również
na Facebooku

<http://bit.ly/2BjVMN7>

KITy AVT na wideo <http://bit.ly/2ScLZTy>

Karta umożliwia sterowanie przekaźnikami
poprzez sieć Internet.

www.sklep.avt.pl