

Jak zastąpić wyświetlacz LCD monitorem HDMI w systemie embedded i.MX6ULL z Linkusem

Współcześni klienci przyzwyczajeni do kolorowych interfejsów smartfonów coraz częściej domagają się od producentów systemów embedded, aby nawet najprostsze urządzenia były wyposażone w kolorowy interfejs graficzny. Tego typu rozwiązania najwygodniej jest realizować z wykorzystaniem tanich komputerów SOM z procesorami aplikacyjnymi działającymi pod kontrolą systemu Linux. W przypadku niewielkich wolumenów okazuje się, że ceny paneli LCD sprzedawanych jako oddzielne komponenty są znacznie wyższe niż fabryczny monitor z interfejsem HDMI. Istnieją układy SOC, które mają zintegrowany interfejs HDMI, ale są to najczęściej bardzo rozbudowane układy wielordzeniowe, dosyć drogie.

Najbardziej popularnym interfejsem graficznym w systemach embedded jest równoległy interfejs RGB, do którego możemy dołączać bezpośrednio popularne panele TFT-LCD. Dysponując nawet najprostszym SOC-em istnieje możliwość konwersji sygnału RGB na sygnał HDMI za pomocą specjalnych układów – nadajników HDMI. W niniejszym artykule na przykładzie komputera VisionSOM (procesor NXP i.MX6ULL) polskiej firmy SoMLabs, pokażemy, w jaki sposób dołączyć monitor HDMI do interfejsu równoległego RGB-LCD.

Interfejs RGB, jest nieskomplikowanym interfejsem cyfrowym służącym do przesyłania obrazu pomiędzy układem graficznym, a wyświetlaczem, który składa się z równoległych

linii R, G oraz B (najczęściej 8-bitowych) przesyłających informację o kolorach używając linii CLK, DE oraz sygnałów synchronizacji poziomej HSYNC i pionowej VSYNC

(rysunek 1). Typowy sposób dołączenia wyświetlacza LCD-TFT firmy Powertip (SL-TFT7-TP-800-480) do systemu embedded za pomocą interfejsu równoległego pokazano na fotografii 2.

Informację o intensywności świecenia poszczególnych pikseli przesyłane są w takt linii CLK na zboczu narastającym oraz opadającym (DDR). Dodatkowo linie HSYNC oraz VSYNC informują o początku nowej linii oraz początku nowej ramki obrazu. Konwersję sygnału RGB na sygnał HDMI umożliwiają wyspecjalizowane nadajniki HDMI, których bardzo popularnym przedstawicielem jest układ ADV7513 produkowany przez Analog Devices (rysunek 3).

Układ ten umożliwia nadawanie sygnału HDMI o maksymalnej rozdzielczości 1920×1080@60Hz (Full HD), oraz zapewnia wsparcie dla przesyłania dźwięku do monitora z częstotliwością próbkowania do 192 kHz. Sygnał video dostarczany jest za pomocą wspomnianego wcześniej interfejsu RGB, natomiast sygnał audio dostarczany jest za pomocą I²S lub SPDIF. Oprócz

sygnałów A/V, układ posiada magistralę I²C służącą do konfiguracji parametrów układu, oraz sygnał INT umożliwiający zgłoszenie przerwania np. po włożeniu wtyczki HDMI. Po stronie wyjściowej oprócz różnicowego interfejsu HDMI do dyspozycji mamy kontroler CEC oraz kontroler magistrali I²C umożliwiający odczytanie parametrów EDID monitora. Sposób dołączenia układu ADV7513 do modułu VisionSOM przedstawiono na **rysunku 4** (rozwiązanie zastosowane na płycie bazowej VisionCB-IND-HDMI – **fotografia 5**). Linie RGB, HSYNC, VSYNC, CLK interfejsu równoległego są dołączone bezpośrednio linii IO interfejsu kontrolera LCDIF, linia SPDIF dołączona jest do kontrolera SPDIF, linia INT dołączona jest do dowolnej linii GPIO w tym przypadku UART1-CTS (GPIO1.18), natomiast linie sterujące I²C dołączono do kontrolera #I2C2.

Sterownik dla systemu Linux

Pomimo iż układ ADV7513 pełni rolę konwertera, do jego prawidłowej pracy potrzebny jest sterownik pracujący w przestrzeni jądra, którego zadaniem jest:

- odczytanie za pomocą EDID parametrów konfiguracyjnych monitora,
- wybór odpowiedniej rozdzielczości obrazu wspieranej przez monitor,
- wysłanie danych konfiguracyjnych do układu nadajnika.

W jądrze *mainline* począwszy od wersji 4.10 istnieje gotowy sterownik dla podsystemu KMS dla wybranego układu. Niestety sterowniki graficzne dostarczane przez firmę NXP wraz z jądrem dla układu i.MX6ULL nie wykorzystują podsystemu KMS. W związku z tym konieczne stało się opracowanie odrębnego sterownika współpracującego ze sterownikiem *mxsfb*. Zaprezentowany w artykule sterownik dla uproszczenia umożliwia wyświetlanie obrazu o stałej rozdzielczości, którą możemy konfigurować za pomocą pliku opisu sprzętu *Device Tree*.

W OPISIE DRIVERA HDMI MOGĄ WYSTĘPOWAĆ ODWOŁANIA DO UKŁADU ADV7511 FIRMY ANALOG DEVICES. UŻYTY W VISIONCB-IND-HDMI UKŁAD ADV7513 JEST JEDNĄ Z JEGO ZGODNYCH WERSJI.

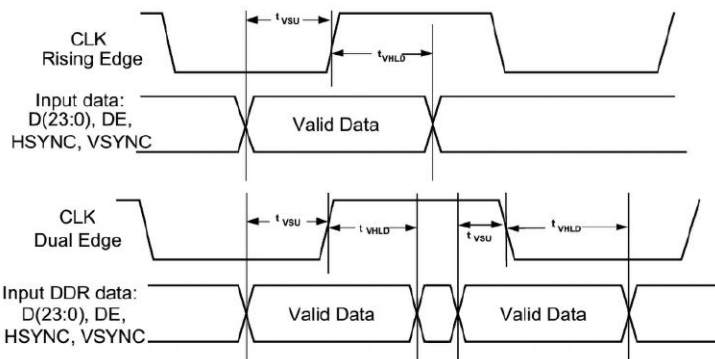
Kod źródłowy sterownika znajduje się w pliku *adv7511-fbdev.c*, natomiast pliki *adv7511.h* oraz *adv7511_default_regs.h* zawierają podstawowe stałe oraz tablice konfiguracyjne rejestrów układu. Diagram sekwencji (**rysunek 6**) przedstawia sposób interakcji podsystemów jądra Linux ze sterownikiem konwertera.

Sterownik rozpoczyna działanie od wywołania funkcji *adv7511_probe*, która jest wywołana przez podsystem sterowników jądra w momencie odczytania obecności wpisu odnoszącego się ADV7513 w *Device Tree*. Diagram aktywności dla wspomnianej funkcji przedstawiono na **rysunku 7**.

Na początku jest konfigurowany kontroler interfejsu I²C, następnie poprzez odczyt rejestru identyfikacyjnego sprawdzamy, czy do wskazanej magistrali dołączony jest układ konwertera. W przypadku niemożności wykrycia układu zgłaszany jest błąd i sterownik kończy działanie. Jeżeli układ zostanie wykryty poprawnie to odczytywane są wszystkie parametry konfiguracyjne z *Device Tree*,

a następnie do rejestrów układu ADV7513 zapisywane są wartości zgodne z wcześniej odczytanymi parametrami, a następnie uruchamiany jest układ nadajnika HDMI.

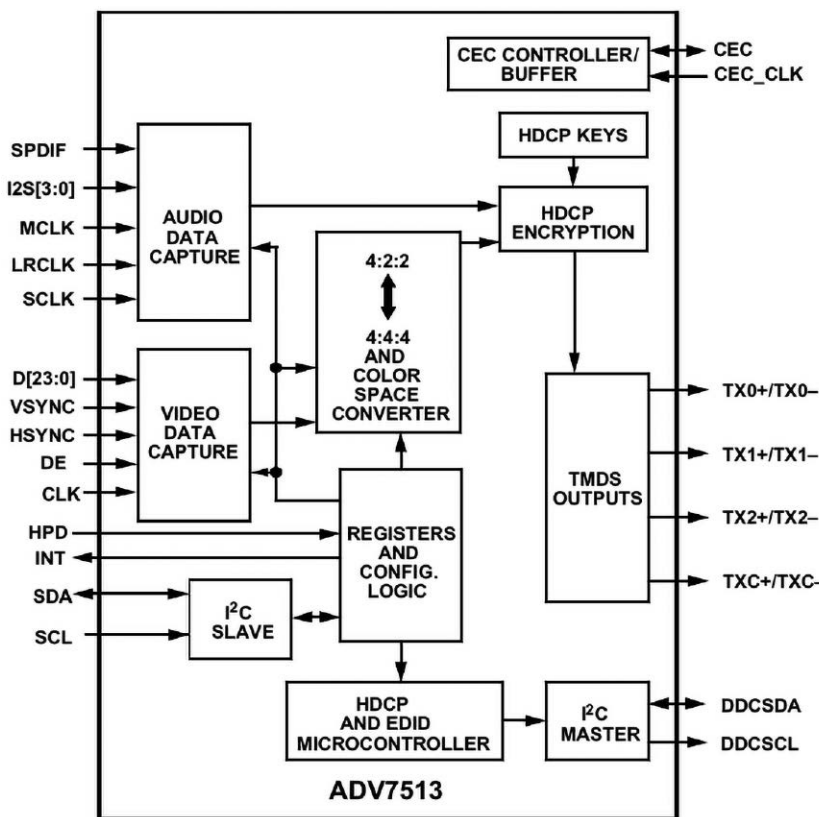
Po zakończeniu konfiguracji sprzętu, rejestrowany jest *callback* w podsystemie FB, którego zadaniem jest reagowanie na zdarzenia pochodzące od tego podsystemu. Od tego momentu konwerter jest gotowy do pracy.



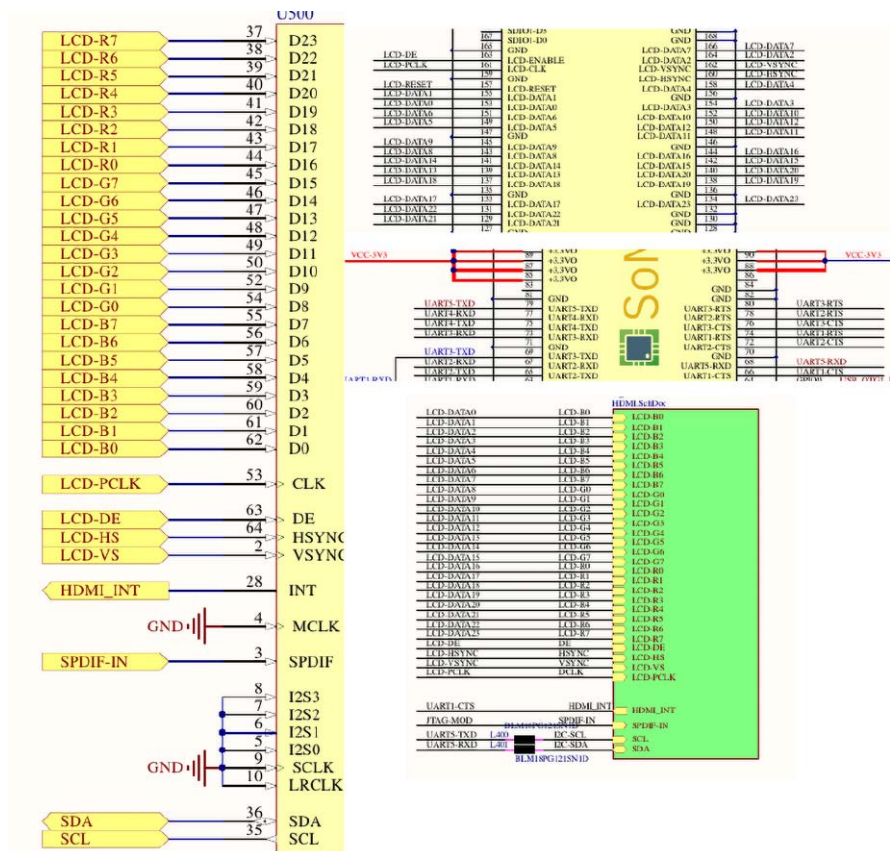
Rysunek 1. Przebiegi charakteryzujące pracę równoległego interfejsu graficznego używanego powszechnie w systemach embedded



Fotografia 2. Przykładowy sposób dołączenia wyświetlacza LCD-TFT firmy Powertip (SL-TFT-7-TP-800-480) do płyty bazowej komputera embedded



Rysunek 3. Schemat blokowy układu ADV7513



Rysunek 4. Schemat ilustujący sposób dołączenia układu ADV7513 do 24-bitowego interfejsu graficznego

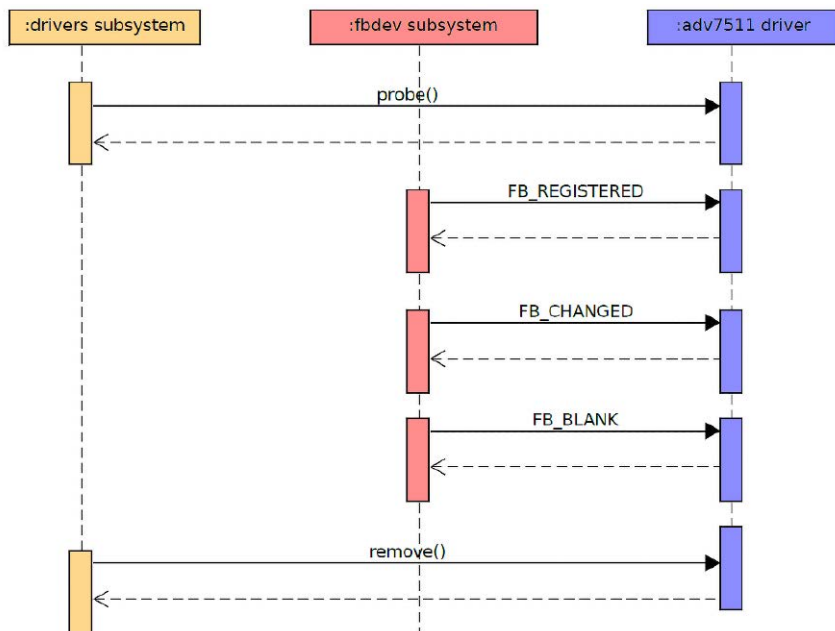
Funkcja `adv7511_fb_event` wywoływana jest w reakcji na zdarzenia od podsystemu FB, i wywoływana jest w momencie wystąpienia następujących zdarzeń:

- `FB_EVENT_FB_REGISTERED`: Zdarzenie rejestracji, wywoływana jednorazowo przez podsystem podczas inicjalizacji. Tutaj może być wykonana dodatkowa inicjalizacja związana ze sterownikiem,

- `FB_EVENT_MODE_CHANGE`: Zmiana parametrów VIDEO bufora FB,
- `FB_EVENT_BLANK`: Żądanie wyświetlenia lub wygaszenia wyświetlania obrazu.

Kod funkcji obsługującej zdarzenia od podsystemu FB przedstawiono na **listingu 1**.

Zdarzenie `FB_EVENT_FB_REGISTERED` dokonuje jednorazowej inicjalizacji sterownika związanej z FB w naszym przypadku



Rysunek 6. Diagram sekwencji prezentujący interakcję podsystemów jądra Linux ze sterownikiem konwertera ADV7513



Fotografia 5. Wygląd płytki bazowej VisionCB-IND-HDMI, na której był testowany driver opisany w artykule

przypisujemy jedynie kontekst sterownika do odpowiedniej struktury oraz wyświetlamy logo.

Zdarzenie `FB_EVENT_MODE_CHANGE` nie jest przez nasz sterownik obsługiwane, ponieważ w podstawowej zapewniamy jedynie wyświetlanie obrazu o ściśle określonej (za pomocą *Device Tree*) rozdzielczości.

Zdarzenie `FB_EVENT_BLANK`, obsługiwane jest poprzez włączenie lub wyłączenie wysyłania sygnału HDMI przez układ konwertera.

Kompilacja jądra oraz przygotowanie Device Tree

Przygotowanie toolchaina dla komputera VisionSOM zostało szczegółowo opisane w EP 3/2018, w artykule „Emulator konsoli NES w systemie Linux na komputerze VisionSOM”. Na tej podstawie zakładamy, że użytkownik ma już w pełni działające środowisko. Jeśli dopiero zaczynamy przygodę z VisionSOM należy zapoznać się z wcześniej wspomnianą publikacją.

Po pobraniu pliku archiwum `adv7511-fbdev-1.0.tgz` należy go rozpakować w katalogu `home/developer`, a następnie uruchomić środowisko: `sudo ./somlabs-debian/chttoolchain su - developer cd source`

Następnie należy rozpakować archiwum, oraz skopiować konfigurację jądra z włączonym wsparciem dla sterownika obrazu: `tar xf adv7511-fbdev-1.0.tar.gz cp adv7511-fbdev-1.0/config_video_hdmi_out kernel/linux rel_imx_4.1.15_2.1.0_ga/.config`

W kolejnym kroku należy zaaplikować łątkę na *Device Tree* włączającą interfejs video oraz dodatkową łątkę zawierającą opis konfiguracji układu ADV7513:

```
patch -d somlabs-dts-1.0 < adv7511-video/somlabs-visionsom-6ull-fix-dts.patch
patch -d somlabs-dts-1.0 < adv7511-video/somlabs-visionsom-6ull-video.patch
```

```
Listing 1. Funkcja obsługująca zdarzenia od podsystemu FB
static int adv7511_fb_event(struct notifier_block *self, unsigned long val, void *data)
{
    struct fb_event *event = data;
    struct fb_info *fbi = event->info;
    struct adv7511 *state = container_of(self, struct adv7511, nb);

    switch (val) {
    case FB_EVENT_FB_REGISTERED:
        dev_dbg(fbi->dev, "FB_EVENT_FB_REGISTERED\n");
        if (state->fbi == NULL)
            state->fbi = fbi;
        fb_show_logo(fbi, 0);
        break;
    case FB_EVENT_MODE_CHANGE:
        dev_dbg(fbi->dev, "FB_EVENT_MODE_CHANGE\n");
        break;
    case FB_EVENT_BLANK:
        if (*(int *)event->data == FB_BLANK_UNBLANK) {
            dev_dbg(fbi->dev, "FB_BLANK_UNBLANK\n");
            adv7511_power_on(state);
        } else {
            dev_dbg(fbi->dev, "FB_BLANK_BLANK\n");
            adv7511_power_off(state);
        }
        break;
    }
    return 0;
}
```

Po zaaplikowaniu łatek dla *Device Tree* możemy przystąpić do kompilacji nowej wersji jądra. W tym celu przechodzimy do katalogu *kernel* oraz wydajemy polecenie, którego zadaniem jest skompilowanie jądra:

```
cd kernel
make deb-pkg -j8 CFLAGS=-fno-pic
```

Następnym krokiem jest powrót do katalogu ze źródłami oraz skompilowanie pliku opisu sprzętu *Device Tree*, co realizujemy w następujący sposób:

```
cd ~/source/somlabs-dts-1.0/
make
```

Po skompilowaniu *Device Tree* oraz jądra pozostało nam skompilowanie sterownika obsługującego układ ADV7513:

```
cd ~/source/adv7511-video
make -j8 CFLAGS=-fno-pic
```

Mamy w zasadzie wszystko, co jest niezbędne do pracy, musimy jeszcze przygotowane przez nas pliki, wgrać na kartę SD. Przed tą czynnością należy opuścić środowisko *chroot*, za pomocą polecenia *exit*, przejść do katalog *home/developer*, a następnie skopiować nowe jądro, sterownik dla konwertera, oraz plik opisu sprzętu *Device Tree* na kartę:



Fotografia 8. System testowy i efekt jego działania

```
cp ~/source/somlabs-dts-1.0/
somlabs-visionsom-6ull.dtb
<sdcard_path>/boot
mkdir -p <sdcard_path>/lib/
modules/4.1.15/kernel/drivers/
misc/
cp ~/source/ adv7511-fbdev-1.0/
adv7511-fbdev.ko /<sdcard_path>/
lib/modules/4.1.15/kernel/drivers/
misc/
echo adv7511_fbdev >> <sdcard_
path>/etc/modules-load.d/modules.
conf
cp source/kernel/linux-
image-4.1.15_4.1.15-*_armhf.deb
<sdcard_path>/home/developer
```

Po zainstalowaniu wszystkich potrzebnych plików należy dokonać ostatnich czynności instalacyjno-konfiguracyjnych bezpośrednio na konsoli VisionCB-IND-HDMI. W tym celu płytkę należy podłączyć do sieci za pomocą gniazdka RJ-45, włączyć zasilanie oraz za pomocą konsoli szeregowej RS232 (vCOM) zalogować się do *shella*. Po zalogowaniu należy zainstalować nową paczkę *deb* zawierającą jądro:

```
dpkg -i ~/home/developer/inux-
image-4.1.15_4.1.15-*_armhf.deb
depmod -a
```

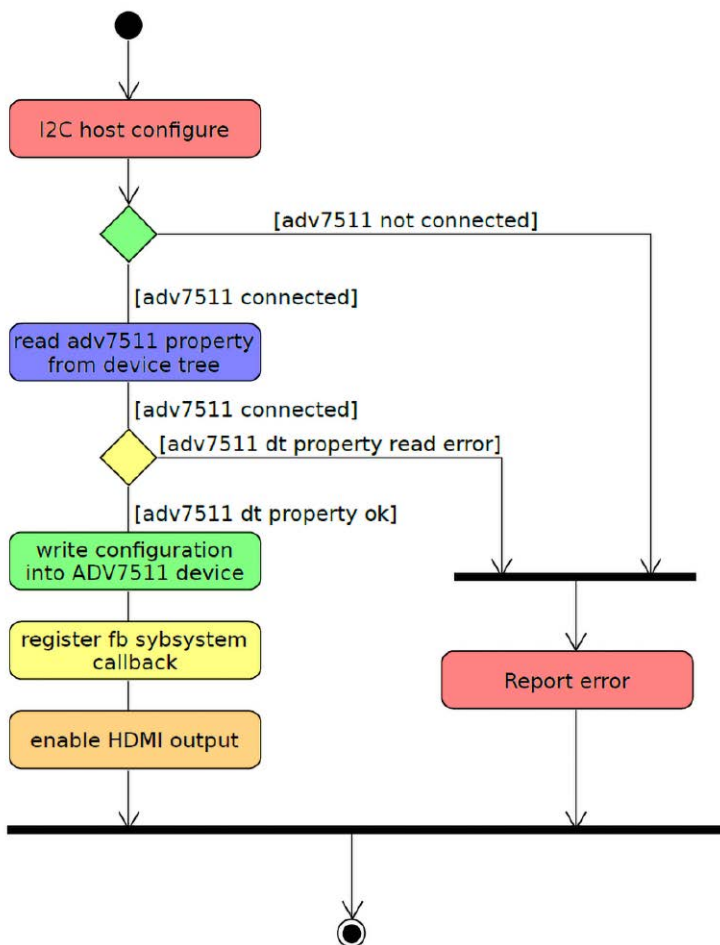
Następnie instalujemy wybrane środowisko graficzne. Autor poleca lekkie i konfigurowalne środowisko XFCE:

```
apt update
apt install xfce4-desktop
```

Instalacja środowiska graficznego w zależności od szybkości Internetu oraz klasy karty SD może zająć stosunkowo długi czas. Kolejnym krokiem jest dodanie standardowego użytkownika (nie *roota*), za pomocą polecenia *adduser*.

Po wykonaniu powyższych czynności system jest w zasadzie gotowy do pracy. Pozostało jedynie dołączenie do złącz USB-A myszki oraz klawiatury, a następnie ponowne uruchomienie systemu (polecenie *reboot*). Gdy system uruchomi się ponownie na monitorze powinien ukazać się graficzny ekran logowania (fotografia 8). Korzystając z wcześniej utworzonego konta użytkownika możemy zalogować się do pulpitu XFCE.

Lucjan Bryndza, EP
 emsyslabs.com
 support@emsyslabs.com



Rysunek 7. Diagram aktywności funkcji *adv7511_probe*