

Mateusz MUCHA¹

SYSTEM WYSOKIEJ ROZDZIELCZOŚCI OPARTY O KONTROLER RASPBERRY PI

W artykule przedstawiono system wysokiej rozdzielczości oparty o kontroler Raspberry Pi przeznaczony do prezentacji danych. System zaprojektowany został jako niekosztowna alternatywa dla już istniejących rozwiązań. W przeprowadzonych badaniach w porównaniu do innych urządzeń pod względem wydajności wypada lepiej. Jest on też znacznie bardziej ekonomiczny. Praca opisuje system prezentacji danych, program sterujący, jego funkcjonowanie oraz predefiniowane szablony. Opisany został wygląd przykładowego szablonu oraz różne typy informacji prezentowanych przez system. Raspberry Pi wraz z aplikacją stanowi końcowy, wykonawczy element całego systemu, w skład którego wchodzi również panel administracyjny oraz baza danych.

Słowa kluczowe: Raspberry Pi, HMP, QML

1. Wprowadzenie

Systemy wysokiej rozdzielczości są przeznaczone dla rozwiązań biznesowych opierających się na prezentacji jak największej ilości informacji w formie multimedialnej na ekranie telewizora. Do prawidłowej oraz wydajnej pracy potrzebują odpowiedniej platformy sprzętowej. Taką platformą okazało się urządzenie Raspberry Pi, będące wydajnym mikrokomputerem wyświetlającym obraz w wysokiej rozdzielczości. Urządzenie o wymiarach karty kredytowej oraz wadze 45 g jest w stanie wyświetlać obraz w rozdzielczości 1920x1080 pikseli przy zachowaniu płynności 30 klatek na sekundę. Aplikacja współpracująca z urządzeniem w pełni wykorzystuje jego możliwości. Urządzenie posiada również wiele scenariuszy oraz trybów pracy. Szatę graficzną opisują szablony wyglądu.

Podobne rozwiązanie skupiające się na prezentacji danych zostało zaprezentowane w pracy [4]. Rozwiązaniem tego typu dysponuje również firma SpinetiX. Opracowała ona zamknięty system prezentacji danych oparty o urządzenia HMP130 i HMP200 [2,3]. Jak podaje producent, urządzenia te mają

¹ Mateusz Mucha, Politechnika Rzeszowska, Krasne 10A, 36-007 Krasne, tel. 501252243, email: mateusz.mucha@wp.pl

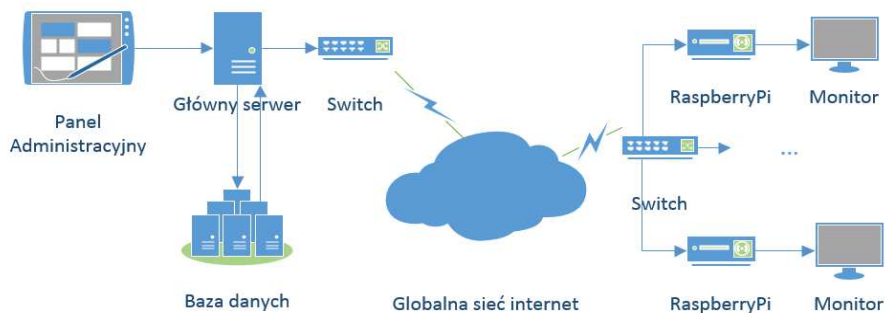
możliwość wyświetlania obrazu FullHD, przy zachowaniu wysokiej częstotliwości odświeżania. W przeciwieństwie do Raspberry Pi urządzenia te są wielkości domowego routera (135x86mm). Zostały stworzone wyłącznie do prezentacji danych i nie pozwalają na ingerencję w oprogramowanie.

Brak możliwości podłączenia klawiatury uniemożliwia bezpośrednie zarządzanie kontrolerem co zmusza do użycia odpowiedniego oprogramowania. Jest to dość poważna wada, ponieważ urządzenie HMP nie może być dowolnie konfigurowane oraz zamyka możliwość rozwoju rozwiązania. Odwrotnie jest w przypadku Raspberry Pi, który jako komputer może być dowolnie konfigurowany.

Kontroler jest wyposażony w układ Broadcom BCM2835 typu *System on a chip* [1]. Oznacza to, że procesor zawiera układ graficzny, pamięć RAM oraz cyfrowy przetwornik sygnału zamknięty w jednym układzie scalonym. Głównym elementem zestawu jest procesor ARM111196JZFS domyślnie taktowany częstotliwością 700MHz. Dzięki temu jest w stanie obsłużyć typowe systemy operacyjne, w tym Linux. Kolejnym komponentem jest układ graficzny VideoCore IV CPU, pozwalający wyświetlić obraz w wysokiej rozdzielczości. W zależności od modułu, Raspberry Pi dysponuje pamięcią RAM 256MB lub 512MB. Dane przechowywane są na karcie SD. Układ Broadcom BCM2835 wraz z układem VideoCore IV GPU, są dość powszechnie stosowaną architekturą sprzętową. Ze względu na wydajność, układ graficzny może obsługiwać bibliotekę OpenCV. Testy takiego układu przedstawia m.in. artykuł [5]. Raspberry Pi został porównany pod względem wydajności do komputera klasy PC Shuttle XS36V, o wymiarach 20x16x3,6 cm [9].

2. Opis systemu prezentacji danych

System pokazany na rys.1 można podzielić na dwie części. Taki podział pozwala na globalne rozproszenie, co zwiększa zasięg.



Rys. 1. Schemat systemu

Fig. 1. System diagram

System został zaprojektowany do pracy w sieci lokalnej LAN, jednak jego konfiguracja pozwala na wykorzystanie globalnej sieci Internet i przeniesienie administracji np. do tzw. „chmury”. Tabela 1 przedstawia krótki opis obydwu części.

Tabela 1. Opis części systemu wysokiej rozdzielczości

Table 1. Description of the system high definition

Część 1	Zarządzanie wyświetlaną treścią. Jest to pewnego rodzaju panel administracyjny pozwalający na konfigurację urządzenia bądź urządzeń oraz ustawienia związane z administrowaniem, takie jak przypisywanie ról administracyjnych czy blokowanie dostępu do określonych funkcjonalności. Dodatkowo, w skład wchodzi również serwer bazy danych Microsoft SQL pozwalający na wydajne składowanie danych oraz związane z tym operacje.
Część 2	Część druga odpowiada za wyświetlanie już przesłanej informacji i jest typowo wykonawcza. W jej skład wchodzi urządzenie Raspberry Pi otrzymujące informacje z centralnego serwera oraz urządzenie wyświetlające komunikaty w formie multimedialnej. Zazwyczaj jest to telewizor bądź monitor.

Komunikacja w systemie odbywa się z wykorzystaniem *webserwisów* za pośrednictwem protokołu SOAP. Takie rozwiązanie zwiększa wydajność oraz zapewnia stabilność komunikacji [6], pozwalając na zastosowanie systemu w skali globalnej. Serwer z kontrolerem wymieniają informacje zgodnie z odpowiednim scenariuszami działania. Raspberry Pi może wyświetlać informacje również w trybie *offline*. Wystarczy, że wcześniej informacje zostały pobrane w zwykłym trybie pracy.

System dysponuje predefiniowanymi szablonami wyglądu. Przykładowy szablon przedstawiono na rys 2. Każdy szablon zawiera różne typy informacji wyświetlające inny rodzaj komunikatów. System prezentuje informacje typu:

- Film, prezentacja wideo
- Zdjęcie, pokaz slajdów
- Statyczny pasek informacyjny
- Dynamiczny pasek informacyjny wyświetlający dane z kanału RSS
- *Widgety* HTML pozwalające pokazać dowolne strony.



Rys. 2. Szablon ułożenia poszczególnych elementów

Fig. 2. Template arrangement of individual elements

3. Funkcjonowanie systemu

System został zaprojektowany w celu wyświetlania informacji w formie multimedialnej. By dodać odpowiedni komunikat należy wykonać poniższe kroki:

1) Zarejestrować urządzenia Raspberry Pi jako element wyświetlający informacje. Od tego momentu można zarządzać urządzeniem lub grupą urządzeń.

2) Określić układ prezentowanych informacji, które będą wyświetlane. Zostało przygotowane 8 predefiniowanych schematów, które można odpowiednio modyfikować. Nie ma możliwości tworzenia personalizowanych schematów.

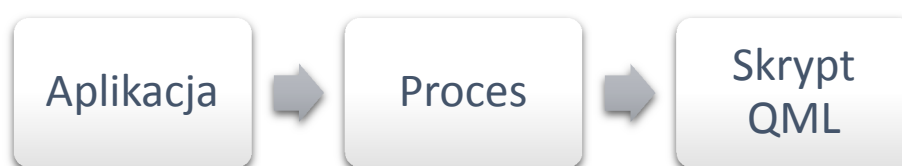
3) Określić zawartości wybranego szablonu. Należy pamiętać, że wszystkie szablony posiadają różnego typu komponenty, z którego każdy przedstawia inny typ informacji.

4) Zdefiniować kolejkę komunikatów, czyli kolejność pokazywania różnych komunikatów dla danego kanału. Każdy z komunikatów ma własny priorytet, co pozwala określić częstotliwość wyświetlania informacji. Dla kanałów statycznych, nie posiadających kolejki, definiuje się jednorazowo źródło danych.

5) Jeżeli dany kanał tego wymaga, zdefiniować listę plików dla kolejki. Przykładowo, definiuje listę filmów reklamowych wyświetlanych po sobie. Każdy komunikat musi mieć informację, priorytet oraz treść w postaci załącznika. Zatwierdzenie takiej konfiguracji oznacza wstawienie jej do kolejki aktualizacji dla danego kontrolera. Urządzenie cyklicznie odpytuje serwer o nową kolejkę komunikatów bądź nowy wygląd. Jeżeli któreś z tych ustawień uległo zmianie, urządzenie automatycznie dostosowuje się do nowych ustawień, do momentu pojawienia się nowych konfiguracji.

4. Program sterujący

Python jako język wysokiego oraz niskiego poziomu okazał się właściwym rozwiązaniem dla głównej aplikacji. Brak potrzeby kompilacji kodu spowodował zwiększenie wydajności Raspberry Pi [7]. Program główny wykorzystuje w wysokim stopniu środowisko Linux. Graficzne szablony komunikatów zostały napisane w języku QML, będącym częścią środowiska Qt, dedykowanym na urządzenia wbudowane oraz cechującym się wysoką prędkością działania [8]. Główne zastosowanie język QML znalazł w telefonach Nokia pracujących w systemie Symbian. Pliki filmowe są odtwarzane za pomocą programu OMX Player, przeznaczonego specjalnie na platformę Raspberry Pi. Takie rozwiązanie zwiększyło wydajność całego systemu oraz przeniosło najbardziej wymagającą część, jaką jest wyświetlanie plików filmowych w jakości fullHD na układ VideoCore IV GPU, zaprojektowany właśnie dla takich operacji.



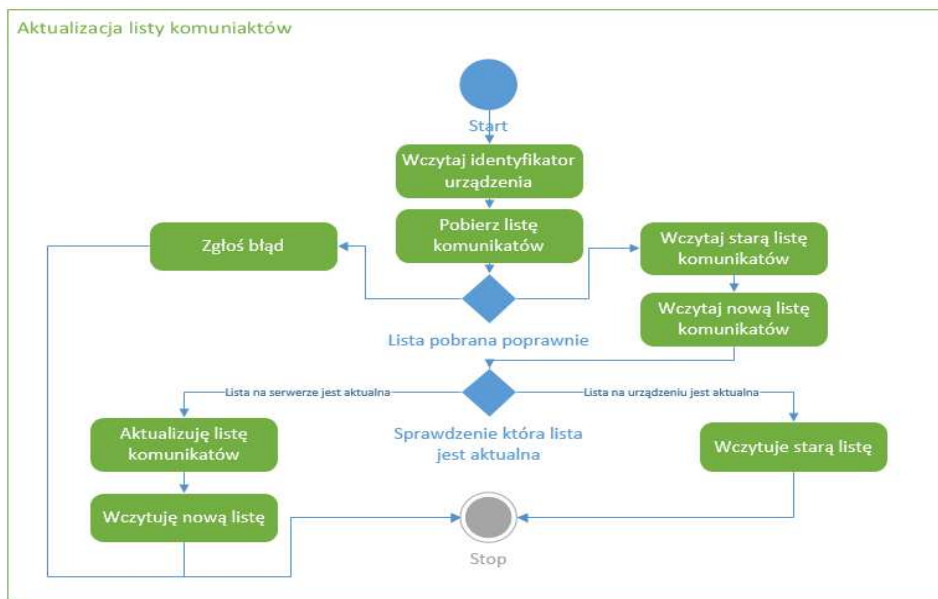
Rys. 3. Schemat uruchomienia procesów graficznych

Fig. 3. Scheme run graphical processes

Aplikacja sterująca pośredniczy pomiędzy panelem administracyjnym a środowiskiem systemu Linux. Wszystkie akcje odbywają się na zasadzie tworzenia nowych procesów w określonej kolejności. Przebieg ten prezentuje rys. 3. Procesy odpowiedzialne za wyświetlanie grafiki tworzą złudzenie istnienia warstw. Przy odpowiedniej konfiguracji uzyskuje się efekt niezauważalnych przejść pomiędzy odpowiednimi scenami. Dodatkowo takie rozwiązanie pozwala całą logikę odpowiedzialną za „życie procesu” przenieść

na system operacyjny, a działanie aplikacji opiera się na powoływaniu nowego procesu z odpowiednimi parametrami oraz na zakończeniu jego działania.

Funkcjonowanie programu sterującego opiera się na odpowiednich scenariuszach działania. Aplikacja posiada zaimplementowane rozwiązania pozwalające odpowiednio zachować się w każdej sytuacji. Rysunek 4 przedstawia algorytm postępowania w przypadku aktualizacji komunikatów. Kluczowym aspektem jest identyfikator. Dzięki niemu jest możliwe rozpoznanie konkretnego urządzenia w sieci. Po identyfikacji zostaje sprawdzona globalna kolejka komunikatów, czy dla danego urządzenia nastąpiły modyfikacje, też nie. Jeżeli nie było zmian, to aplikacja działa dalej. Jeżeli natomiast, wystąpiła zmiana, to zostaje pobrana nowa kolejka komunikatów i posortowana według kolejności wyświetlania. Jeżeli podczas tego procesu wystąpi błąd, to automatycznie zostaje zapisany w globalnym systemie, a urządzenie wchodzi w serwisowy tryb pracy.



Rys. 4. Diagram przedstawia aktualizację listy komunikatów

Fig. 4. This diagram shows the update of the list of messages

Tabela 2 przedstawia przykładowy scenariusz uruchomienia aplikacji. Należy założyć, że wcześniej została zdefiniowana kolejka komunikatów, a konfiguracja wyglądu nie została zmieniona.

Tabela 2. Przykładowy scenariusz uruchomienia aplikacji

Table 2. An example scenario to launch the application

Krok	Opis
1	Pobieranie listy potrzebnych załączników
2	Sprawdzanie listy czy nie znajdują się w niej większe pliki które należy pobrać. Jeżeli tak, to aplikacja pobiera je z centralnego serwera
3	Docelowe uruchamianie grafiki
4	Uruchamianie procesu odpowiedzialnego za tło
5	Ustawianie wygenerowanej grafiki jako tła ekranu
6	Uruchamianie procesów odpowiedzialnych za wyświetlanie komunikatów

Wszystkie typy komunikatów za wyjątkiem pliku filmowego posiadają następujące pliki sterujące:

1) Główny plik pobierający parametry startowe od głównego procesu sterującego oraz przekazujący je do odpowiedniego pliku opisującego grafikę.

2) Plik QML opisujący grafikę, zdefiniowany dla odpowiedniego typu komunikatu.

Odtwarzanie pliku filmowego jest możliwe za pomocą powołania procesu *OMX playera* z odpowiednimi ustawieniami konfiguracji. Aplikacja posiada scenariusze, które opisują zachowanie w przypadku braku dostępu do Internetu, serwisu urządzenia, czy pobraniu konfiguracji i listy komunikatów jednocześnie.

5. Badania i porównania

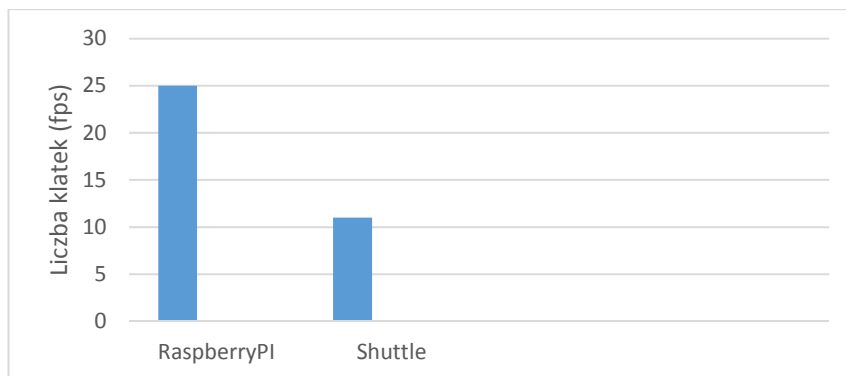
System wysokiej rozdzielczości jest rozwiązaniem prototypowym. Jego działanie przedstawia rys. 5. Widać na nim obraz w rozdzielczości FullHD oraz dodatkowe informacje. Układ ekranu został zdefiniowany za pomocą odpowiedniego szablonu w skład którego wchodzi statyczny pasek informacyjny umieszczony w górnej części szablonu i dolny pasek informacyjny przedstawiający dane z kanału RRS pobierane w czasie rzeczywistym. Pasek prezentuje dane w formie animowanej. W prawej części szablonu znajduje się *widget* pogody pobierający informacje dla określonej lokalizacji z serwisów pogodowych. W centralnej części widać plik multimedialny dostosowany wymiarami do odpowiedniego miejsca w szablonie. Każdy z wyżej wymienionych komponentów działa niezależnie, co świadczy o wielowątkowość aplikacji.



Rys. 5. Wygląd działającego systemu komunikatów

Fig. 5. Appearance acting messaging system

Jednym z głównych elementów systemu wysokiej rozdzielczości jest aplikacja zaprojektowana tak by działać w środowisku Linux. Dzięki temu możliwe jest uruchomienie jej na *dowolnym* urządzeniu. Kolejne badanie pokaże porównanie urządzenia Raspberry Pi oraz Shuttle pod względem wydajności. Rysunek 6 przedstawia zestawienie *liczby* klatek na sekundę obu tych urządzeń. Z zestawienia wynika iż mimo niższych osiągnięć Raspberry Pi wypadł lepiej. Jest to spowodowane posiadaniem sprzętowego kodeka h.264/MPEG-4 AVC [1]. Dzięki niemu możliwe jest *przekształcenie strumienia danych w sygnał audio i wideo*. Podobne testy zostały przeprowadzone w pracy magisterskiej Klient Systemu Komunikatów OPTeam[10].



Rys. 6. Porównanie wydajności urządzeń

Fig. 6. Comparison of equipment performance

6. Podsumowanie

Pokazano że kontroler Raspberry Pi nadaje się do rozwiązań związanych z prezentacją obrazu. Wbudowane komponenty sprzętowe takie jak procesor Broadcom BCM2835 pozwalają na wydajne wyświetlanie obrazu przy zachowaniu płynności. Istnieją również inne systemy tego typu takie jak urządzenia HMP130 oraz HMP200 firmy SpinetiX , ale ich koszt jest znacznie wyższy. Zastosowanie różnych szablonów graficznych oraz płynność odtwarzania filmów w wysokiej rozdzielczości tworzą z Raspberry Pi wydajną platformę sprzętową. Cały system dzięki swojej strukturze może pracować w środowisku rozproszonym, co zwiększa zasięg działania. Zastosowanie bazy danych Microsoft SQL poprawia wydajność systemu i pozwala na wykorzystywanie materiałów filmowych zajmujących kilkaset megabajtów. Raspberry Pi jest propozycją rozwiązania do prezentacji danych. Stworzenie systemu wysokiej rozdzielczości było kontynuacją już istniejącego rozwiązania opartego na PC, tyle że bazującego na innej platformie programowej. Nowy system jest znacznie tańszy. System wysokiej rozdzielczości oparty o mikrokontroler Raspberry Pi może być alternatywą dla rozwiązania opartego o zwykłe komputery klasy PC. Ponieważ Raspberry Pi w znacznym stopniu różni się cenowo od urządzeń mających działanie identycznego systemu, zostało przetestowane również na komputerze ShuttleXS36V. Ponieważ aplikacja działa w środowisku Linux, niezbędne były narzędzia programistyczne takie jak edytor Eclipse do sprawnego pisania kodu, czy QtCreator do sprawniej edycji plików QML. Zastosowanie języka Python dało prostą możliwość edycji kodu źródłowego projektu bez jego kompilacji. Poza tym, jako język wysokiego poziomu, Python zwiększa wydajność całego systemu.

Literatura

- [1] Eben Upton: Raspberry Pi User Guide Wiley 2013
- [2] Spinetix: Product Brief, SpinetiX HMP130
http://www.spinetix.com/sites/default/files/documentation/SpinetiX_ProductBrief_HMP130_English.pdf
- [3] Spinetix: Product Brief, SpinetiX HMP200,
http://www.spinetix.com/sites/default/files/documentation/SpinetiX_ProductBrief_HMP200_English.pdf
- [4] Bonganay, A.C.D. ; Magno, J.C. ; Marcellana, A.G. ; Morante, J.M.E. ; Perez, N.G.: Automated electric meter reading and monitoring system using ZigBee-integrated raspberry Pi single board computer via Modbus, Electrical, Electronics and Computer Science (SCEECS), 2014 IEEE Students' Conference onDOI:
- [5] Abaya, W.F. ; Basa, J. ; Sy, M. ; Abad, A.C. ; Dadios, E.P.: Low cost smart security camera with night vision capability using Raspberry Pi and OpenCV, Humanoid,

- Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM), 2014 International Conference on
- [6] Kwong Yuen Lai ; Thi Khoi Anh Phan ; Tari, Z. Efficient SOAP binding for mobile Web services Local Computer Networks, 2005. 30th Anniversary.
- [7] Allen B. Downey: Think Python, O,Reilly 2014.
- [8] Paimen, L. ; Pohjalainen, P. : Case study: QML for the web, Web Systems Evolution (WSE), 2011 13th IEEE International Symposium on
- [9] Slim PC System: XS 36V-703 , Shuttle Computer Handels GmbH, Product Specification, www.shuttle.com
- [10] Mateusz Mucha: praca magisterska Klient Systemu Komunikatów OPTeam, 2014

HIGH-RESOLUTION SYSTEM BASED ON THE RASPBERRY PI CONTROLLER

S u m m a r y

The paper presents high-resolution system based on Raspberry Pi controller developer for data presentation. The system has been designed as an inexpensive alternative to existing systems. The device is a final, executive part of the whole system, which includes administration panel and database. The system has various kinds of data presentation channels including channel video or multimedia RRS bar. Application of the resulting works on the principle of integration with Linux, so, a certain part of the functionality is transferred to the operating system. Python is main language in which the application has been created. Auxiliary language is a language describing graphics QML which greatly increases the efficiency of the system. FullHD videos are run through, OMX player program takes full advantage of the device Raspberry Pi.

Keywords: Raspberry Pi, HMP, QML

DOI: 10.7862/re.2016.2

Tekst złożono w redakcji: styczeń 2016

Przyjęto do druku: marzec 2016