

DOI: 10.5604/01.3001.0010.4840

## PORÓWNANIE ROZWIĄZAŃ INTELIGENTNEGO BUDYNKU NA WYBRANYCH PLATFORMACH SPRZĘTOWYCH

Piotr Pańczyk, Jakub Smolka

Politechnika Lubelska, Wydział Elektrotechniki i Informatyki, Instytut Informatyki

**Streszczenie.** Niniejszy artykuł prezentuje porównanie dwóch platform sprzętowych pod kątem możliwości implementacji systemu inteligentnego domu: minikomputera Raspberry Pi i mikrokontrolera AVR. Analiza porównawcza została wykonana na podstawie dwóch niezależnych, autorskich implementacji o tych samych założeniach. Wyniki zaprezentowano w postaci zestawień tabelarycznych i opisów. We wnioskach wskazano korzyści i potencjalne problemy wynikające z zastosowania obu platform.

**Słowa kluczowe:** inteligentne domy, minikomputer, komputer jednopłytkowy, Raspberry Pi, mikrokontroler, AVR

### SMART HOME IMPLEMENTATION COMPARISON ON DIFFERENT HARDWARE PLATFORMS

**Abstract.** This article presents a comparison of the two hardware platforms in terms of Smart Home implementation: Raspberry Pi minicomputer and AVR microcontroller. The comparative analysis was done on the basis of two independent, author's implementations with the same aims. The results are presented in tables and descriptions. The conclusions indicate the benefits and potential problems of using both platforms.

**Keywords:** Smart Home, minicomputer, single-board computer, Raspberry Pi, microcontroller, AVR

### Wstęp

W ostatnich latach systemy inteligentnych budynków stają się coraz bardziej popularne. Stosowane są zarówno w biurach czy budynkach użyteczności publicznej, jak i w prywatnych domach czy mieszkaniach. Wynika to z coraz większej dostępności urządzeń i podzespołów elektronicznych oraz ich malejących cen. Fakt ten przyciąga także hobbystów, którzy mogą zaimplementować coraz więcej funkcjonalności samodzielnie.

Najczęściej decydującym czynnikiem przy wyborze systemu inteligentnego domu jest cena, a wśród amatorów również prostota implementacji. Systemy wbudowane coraz częściej oparte są nie na samodzielnych mikrokontrolerach, ale gotowych układach elektronicznych o szerokich możliwościach, często z systemem operacyjnym Linux. Umożliwia to ich programowanie w językach wysokopoziomowych, z zastosowaniem nowoczesnych technologii i gotowych rozwiązań, podnosi natomiast cenę samych urządzeń. Pojawia się więc pytanie, kiedy warto inwestować w bardziej złożone układy, a kiedy wystarczy użyć prostego i taniego mikrokontrolera.

Niniejszy artykuł poświęcony jest porównaniu dwóch platform z grupy stosunkowo tanich, konsumenckich urządzeń: mikrokontrolera AVR [5], którego główną zaletą jest niska cena, oraz minikomputera Raspberry Pi [4, 7], który w zamian za dużo wyższą cenę umożliwia łatwiejszą, wysokopoziomową implementację. Są to platformy na tyle odmienne, że brakuje artykułów je porównujących. Dokonane zestawienie ma na celu wskazanie wad i zalet obu platform w kontekście budowy systemu inteligentnego domu.

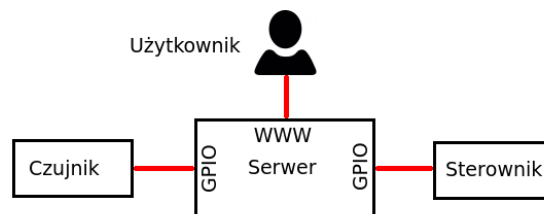
Artykuł skupia się na porównaniu ilości pracy koniecznej do wykonania w celu zaimplementowania systemu na obu platformach oraz wydajności, niezawodności, zużyciu energii, a częściowo również skalowalności wykonanych, autorskich rozwiązań.

### 1. Systemy inteligentnych budynków

System inteligentnego domu składa się zwykle ze swojego rodzaju serwera (jednostki sterującej) oraz kontrolowanych urządzeń – czujników i sterowników. Czujniki, jako urządzenia wejściowe (np. termometr), dostarczają do systemu dane potrzebne do podjęcia decyzji o sterowaniu urządzeniami wyjściowymi (np. ogrzewaniem). Podłączone urządzenia mogą być zarówno analogowe, jak i cyfrowe, często będące dodatkowym sterownikiem, wymagającym osobnego oprogramowania. Możliwe jest też sterowanie ręczne – z poziomu manipulatora (fizycznego urządzenia), aplikacji bądź strony web. Każdy z tych interfejsów pozwala również na konfigurację systemu zgodnie z oczekiwaniami użytkownika.

Połączenie pomiędzy serwerem a poszczególnymi urządzeniami może być zrealizowane na wiele sposobów. Ogólnie można je podzielić na przewodowe i bezprzewodowe. Drugi przypadek wymaga najczęściej programowania użytych sterowników cyfrowych. Pierwszy jest o wiele łatwiejszy, często sterownikiem urządzenia wyjściowego może być urządzenie analogowe kontrolowane przez sam serwer – przykładowo, sterowanie zasilaniem można zrealizować za pomocą przełącznika, wykorzystując pojedynczy pin GPIO jednostki sterującej.

Poglądowy schemat systemu przedstawiony jest na rysunku 1.



Rys. 1. Poglądowy schemat systemu inteligentnego budynku

Artykuł nie obejmuje programowania dodatkowych sterowników, daje natomiast ogólny pogląd na tę tematykę – sterowniki te różnią się od serwera jedynie tym, że obsługują jedno, a nie wiele urządzeń, a zamiast interfejsu użytkownika muszą posiadać wyłącznie interfejs służący do komunikacji z serwerem.

### 2. AVR

AVR [1, 2] (rys. 1) jest ośmiobitowym mikrokontrolerem produkowanym przez firmę Atmel z jednostką arytmetyczno-logiczną w architekturze RISC. Opisywana w artykule implementacja została wykonana i przetestowana na modelach ATmega328P oraz ATmega32A. Model 32A [13] może być taktowany zegarem do 16 MHz. Posiada pamięć flash o rozmiarze 32 KB oraz pamięć EEPROM o rozmiarze 1 KB. Urządzenia peryferyjne można dołączać do 32 pinów GPIO (niektóre z nich mogą działać też np. jako pin PWM, interfejs UART, I<sup>2</sup>C czy SPI). Pamięć ulotna (RAM) ma pojemność 2 KB. Koszt wynosi mniej niż 10 zł.

Mikrokontrolery z rodziny ATmega charakteryzują się prostotą programowania w stosunku do innych mikrokontrolerów. Kod źródłowy może być napisany m.in. w języku Assembler, Bascom oraz C [1, 2] (wykorzystany w przedstawionym rozwiązaniu).

Mimo, że istnieje kilka prostych, ale wielowątkowych systemów operacyjnych na tę platformę, korzysta się z nich rzadko – w zdecydowanej większości zastosowań mikrokontrolery

AVR realizują krótkie zadania, które mogą być w całości obsługiwane w jednym wątku oraz w procedurach obsługi przerwań. Poza obsługą zdarzeń – przerwań wewnętrznych i zewnętrznych, zwykle nie ma konieczności przełączania kontekstu pracy procesora.

Dostępność bibliotek jest ograniczona. Najczęściej stosuje się biblioteki typu Open Source, służące do obsługi konkretnego modułu elektronicznego dołączonego do mikrokontrolera (np. moduł Ethernet [9]).

W kontekście serwera systemu inteligentnego budynku, funkcjonalność serwera WWW trzeba niestety zaimplementować samodzielnie [16]. Ze względu na ograniczoną ilość pamięci, generowanie złożonych, dynamicznych stron internetowych jest zadaniem prawie niewykonalnym. Podobnie wygląda kwestia przechowywania konfiguracji systemu – zamiast bazy danych konieczne jest wykorzystanie zmiennych przechowywanych w pamięci EEPROM. Alternatywnie, serwerem WWW i bazy danych może być niezależne urządzenie komunikujące się z mikrokontrolerem za pomocą jednego z interfejsów UART, I<sup>2</sup>C lub SPI.

AVR jest jedynie mikrokontrolerem, do działania wymaga kilku dodatkowych elementów elektronicznych, co zwiększa nakład pracy niezbędnej do uruchomienia systemu.



Rys. 2. Mikrokontroler AVR

### 3. Raspberry Pi

Raspberry Pi [3, 6] (rys. 2) jest jednopłytkowym minikomputerem opartym o procesor ARM. Implementacja opisana w niniejszym artykule wykorzystuje model 2B [11]. Posiada on procesor ARM7 taktowany zegarem 900 MHz, pamięć RAM o rozmiarze 1 GB i 28 pinów GPIO (nie licząc pinów, na które na stałe wyprowadzone jest napięcie zasilające i masa; niektóre piny mogą działać też jako interfejsy UART, I<sup>2</sup>C czy SPI). Pamięć nieulotna dołączana jest w postaci karty pamięci MicroSD. Komputer posiada zintegrowany kontroler Ethernet, gniazdo HDMI oraz złącza do dedykowanej kamery oraz wyświetlacza dotykowego. W bardziej złożonych implementacjach może więc służyć nie tylko jako serwer, ale również manipulator do obsługi systemu. Koszt urządzenia to około 180 zł.

Poza systemem Microsoft Windows 10 IoT Core, urządzenie umożliwia uruchomienie prawie dowolnej dystrybucji systemu Linux, co czyni go pełnoprawnym komputerem. Przekłada się to na możliwość programowania w praktycznie dowolnym języku, w tym językach wysokiego poziomu, jak Java czy Python. Możliwe jest wykorzystanie większości otwartoźródłowych bibliotek dla systemu Linux/Unix.

Komputer bez problemu radzi sobie z uruchamianiem standardowych serwerów, takich jak Apache, nginx, MySQL, czy Tomcat. W kontekście porównywanych systemów jest to bardzo duże ułatwienie przy tworzeniu interfejsu użytkownika.



Rys. 3. Raspberry Pi

### 4. AVR i Raspberry Pi – porównanie teoretyczne

Porównanie najważniejszych parametrów i cech minikomputera Raspberry Pi i mikrokontrolera AVR przedstawia tabela 1.

Tabela 1. Porównanie cech AVR i Raspberry Pi [11–13]

Parametr	AVR Atmega 32a	Raspberry Pi 2B
Częstotliwość zegara	1 MHz, max 16 MHz	900 MHz
Pamięć RAM	0,002 MB	1024 MB
Pamięć flash	0,032 MB	MicroSD (do 64 GB)
Ilość pinów GPIO	32	26
Alternatywne funkcje GPIO	UART, I <sup>2</sup> C, SPI, PWM, wej. analogowe, wej. przerwań zewnętrznych	UART, I <sup>2</sup> C, SPI, JTAG, PWM
System operacyjny	Brak	Linux (Ubuntu MATE 16.04 LTS)
Przechowywanie danych	Pamięć EEPROM	Bazy danych, pliki
Serwer WWW	Brak	Apache
Kontroler Ethernet	Moduł SPI [9]	Zintegrowany
Cena (orientacyjna)	8 zł + ok. 20-30 zł za niezbędne elementy elektroniczne	180 zł

Niekwestionowaną zaletą mikrokontrolera AVR w stosunku do Raspberry Pi jest cena. W przypadku realizacji prostych zadań mikrokontroler cechuje też większa stabilność działania – w systemie operacyjnym (a więc np. na platformie Raspberry Pi) istnieje wiele złożonych procesów, które w niektórych sytuacjach mogą spowodować niestabilne działanie lub całkowitą blokadę funkcjonalności. Dodatkowo AVR jest bardziej odporny na awarie programowe. Najprostszym przykładem jest uszkodzenie systemu plików na karcie pamięci Raspberry Pi spowodowane zanikiem zasilania – mikrokontroler w takiej sytuacji po prostu przerywa działanie, a po ponownym podłączeniu zasilania wczytuje ostatnio zapisany stan i działa dalej. Niewielkie znaczenie ekonomiczne ma również niższe zużycie energii przez mikrokontroler, jednakże obie platformy cechują się wysoką oszczędnością energii (pobierana moc nie przekracza 5W).

Na tym w zasadzie teoretyczne zalety zastosowania procesora AVR się kończą, zwłaszcza z punktu widzenia amatora. Implementacja systemu inteligentnego domu na platformie Raspberry Pi zajmuje z pewnością znacznie mniej czasu – większość funkcjonalności niezbędnej do działania systemu w sieci oraz interakcji z użytkownikiem (stos TCP/IP, serwery WWW i baz danych) dostarczane są razem z systemem operacyjnym. W gestii programisty leży jedynie implementacja obsługi urządzeń podłączonych do portów GPIO i stworzenie strony sieci web w niemalże dowolnej technologii. Serwery i biblioteki są wspierane przez społeczność Open Source, co wpływa na szybsze wykrywanie i usuwanie ewentualnych usterek, poprawę niestabilnego działania.

Porównując aspekty związane z programowaniem niskopoziomowym, warto zwrócić uwagę na system przerwań. Obie platformy umożliwiają dostęp do programowania procedur obsługi przerwań zewnętrznych, co znacznie ułatwia i optymalizuje obsługę czujników. Niestety większość mikrokontrolerów AVR, w tym ATmega32A, posiada tylko 2-3 wejścia mogące być bezpośrednim źródłem przerwań [13]. Nowsze generacje, w tym ATmega328P, posiadają system przerwań PCINT (*Pin Change Interrupt*) – w tym przypadku źródłem przerwania może być zmiana stanu dowolnego pinu mikrokontrolera, jednak weryfikacja, jaka zmiana i na którym pinie wystąpiła leży w gestii programisty [12]. Biblioteka wiringPi dla Raspberry Pi pozwala na przypisanie dowolnej funkcji do dowolnej zmiany stanu dowolnego pinu [8].

### 5. Założenia porównywanych systemów

- Wykorzystanie wyłącznie technologii przewodowych.
- Użycie czujnika binarnego (czujnik ruchu symulowany przez przycisk) i numerycznego (termometr).

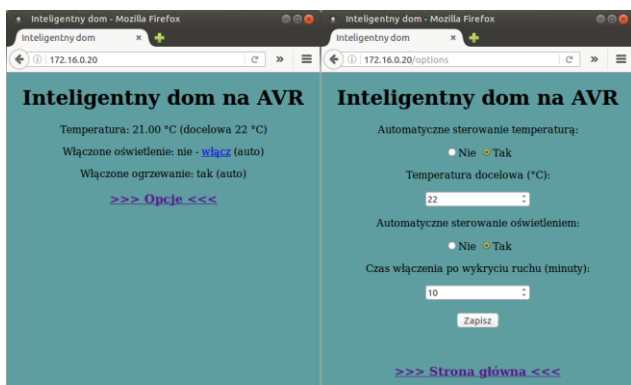
- Użycie sterowników binarnych (diody symulujące stan oświetlenia i ogrzewania).
- Sterowanie automatyczne (ogrzewaniem na podstawie wskazań termometru, oświetleniem – włączenie przy wykryciu ruchu).
- Sterowanie ręczne przez interfejs web.
- Dostępność konfiguracji (włączenie i wyłączenie automatycznego sterowania urządzeniem, docelowa temperatura, czas świecenia po wykryciu ruchu).
- Serwer inteligentnego domu jest jednocześnie serwerem WWW i sterownikiem urządzeń.

Zdecydowano się na wykorzystanie różnych rodzajów termometrów. Do mikrokontrolera AVR podłączony został analogowy układ LM35 [15], Raspberry Pi korzysta natomiast z cyfrowego termometru DS18B20 [14] dołączonego za pomocą magistrali 1-wire. Obsługa magistrali realizowana jest przez jeden z pinów GPIO za pomocą modułu w1-gpio do jądra systemu operacyjnego. Dodatkowo uruchomione zostało oprogramowanie oserver i owfs [10], tworzące pamięć tymczasową (cache) dla pomiaru i udostępniającego go w postaci wirtualnego pliku.

Układ z mikrokontrolerem AVR został dodatkowo wyposażony w wyświetlacz LCD prezentujący aktualny stan systemu.

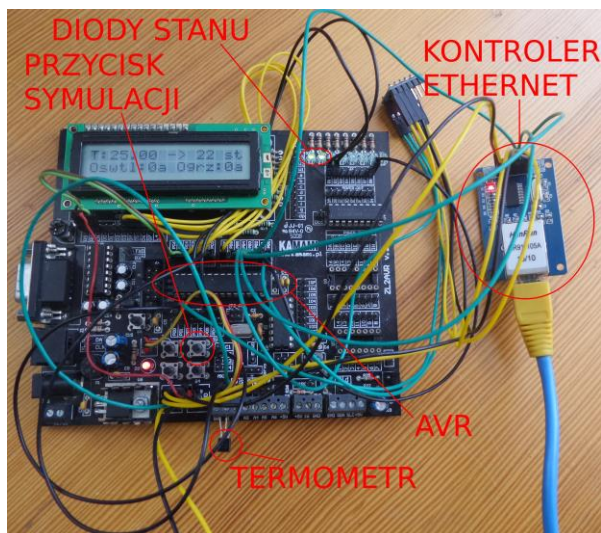
## 6. Wyniki

System został z powodzeniem zaimplementowany na obu platformach. Prosty interfejs użytkownika przedstawiony jest na rysunku 4.

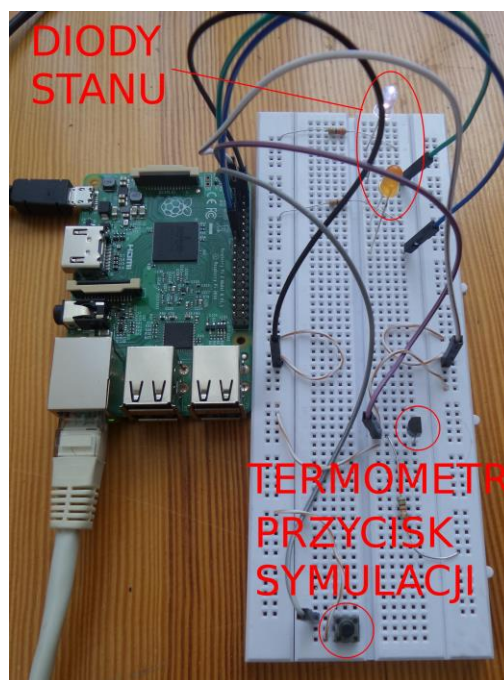


Rys. 4. Interfejs użytkownika

Zdjęcia wykonanych układów elektronicznych przedstawiają rysunki 5 (AVR) i 6 (Raspberry Pi).



Rys. 5. Układ z mikrokontrolerem AVR i modulem Ethernet ENC28J60



Rys. 6. Układ z minikomputerem Raspberry Pi

Interfejs użytkownika systemu działającego na Raspberry Pi wygląda identycznie jak na rysunku 4, różni się tylko nazwa platformy w nagłówku. Oba interfejsy zostały stworzone z wykorzystaniem języka HTML5. Generowanie strony na mikrokontrolerze odbywa się bezpośrednio w programie napisanym w języku C, na Raspberry Pi wykorzystano w tym celu język PHP.

Analizując podstawowe parametry porównywanych platform (częstotliwość zegara procesora, ilość pamięci RAM) oraz dostępność gotowego oprogramowania mogłoby się wydawać, że – nie licząc ceny – Raspberry Pi jest platformą o wiele bardziej atrakcyjną. Wyniki (szczegółowo zaprezentowane w tabeli 2) pokazują jednak, że jest zupełnie inaczej. Implementacja systemu na mikrokontrolerze wymaga wprawdzie wiedzy z zakresu programowania niskopoziomowego i sieci IP, jednakże w przypadku komputera konieczne jest dla odmiany stworzenie dwóch oddzielnych aplikacji i zapewnienie wymiany danych pomiędzy nimi. Jedna aplikacja musi być odpowiedzialna za automatyczną obsługę warstwy fizycznej, druga natomiast za interakcję z użytkownikiem poprzez stronę web. Jako warstwę pośredniczącą w wymianie danych zastosowano bazę danych MySQL. Powoduje to niestety znaczne opóźnienia w reagowaniu na zdarzenia. Brak bezpośredniego dostępu do warstwy fizycznej z poziomu aplikacji PHP również ma wpływ na szybkość działania – odczyt wirtualnych plików z temperaturą [10] oraz wykonywanie poleceń w powłoce systemowej w celu sterowania lub sprawdzenia stanu urządzeń [8] dodatkowo spowalnia generowanie strony. Rozwiązanie optymalne (np. komunikacja między aplikacjami przy użyciu gniazd sieciowych) wymagałoby znacznie większego nakładu pracy niż w przypadku mikrokontrolera. Wykorzystanie bazy danych skutkuje za to znacznym zwiększeniem skalowalności systemu.

Kwestiami nieomówionymi w niniejszym artykule są bezpieczeństwo i skalowalność. O ile sterowanie systemem wyłącznie z sieci lokalnej, pod warunkiem dobrego zabezpieczenia samej sieci, może odbywać się bez haseł i szyfrowania transmisji, o tyle udostępnienie interfejsu web w sieci globalnej bezwzględnie wymaga zastosowania zabezpieczeń. Na platformie Raspberry Pi jest to zagadnienie o podobnym stopniu skomplikowania jak w przypadku zwykłych stron internetowych, natomiast na mikrokontrolerach może okazać się o wiele bardziej problematyczne. Algorytmy szyfrujące zużywają moc obliczeniową procesora i są często złożone, co przekłada się nie tylko na szybkość działania, ale też zajętość

pamięci, zarówno programu (flash), jak i RAM. Wykorzystanie w bieżącej implementacji zaledwie 11 z dostępnych 32 kB pamięci flash mikrokontrolera AVR pozwala jednak przypuszczać, że nie jest to zadanie niewykonalne.

Skalowalność systemu, rozumiana jako możliwość dołączania różnych urządzeń ze zmianą wyłącznie konfiguracji, bez konieczności przeprogramowywania systemu, w obu przypadkach jest znacznie trudniejsza. Podstawowym ograniczeniem jest dostępna liczba pinów GPIO. Zakładając jednak, że wystarczy ona do obsługi systemu, stopień skomplikowania aplikacji w obu przypadkach powinien być podobny. Różnica będzie polegać na sposobie przechowywania informacji o podłączonych urządzeniach – na Raspberry Pi do tego celu w zupełności wystarczy baza danych, na mikrokontrolerze konieczne będzie wykorzystanie zmiennych tablicowych przechowywanych w pamięci EEPROM. 1 kB tej pamięci powinien wystarczyć do przechowywania informacji o kilkudziesięciu urządzeniach możliwych do podłączenia do mikrokontrolera, problemem może natomiast być implementacja interfejsu użytkownika – strony umożliwiające dynamiczną konfigurację mogą mieć duży rozmiar, wykraczający poza możliwości pamięci flash, a przede wszystkim RAM. Jednym z możliwych rozwiązań jest takie zaprogramowanie mikrokontrolera, aby treść strony była generowana i przesyłana do klienta w mniejszych fragmentach.

Tabela 2. Porównanie zaimplementowanego systemu na AVR i Raspberry Pi

Parametr	Mikrokontroler	Minikomputer
Rozmiar aplikacji	11 kB	19 kB
Zajętość pamięci RAM	2 kB	12 MB (aplikacja i serwer WWW), ok. 180 MB z systemem operacyjnym
Czas generowania strony WWW	80 ms	70 ms (temperatura w cache), 300 ms (odczyt temperatury)
Czas wykonywania akcji ręcznej	80 ms	Od 60 ms do kilku sekund
Ilość linii kodu w implementacji (bez bibliotek i wykorzystanych projektów)	450	400
Pobierana moc	0,85 W (80% przez moduł Ethernet)	1,6 – 2,5 W

## 7. Wnioski

Minikomputer Raspberry Pi i mikrokontroler AVR są platformami skrajnie różnymi, jednakże obie umożliwiają implementację prostego lub średnio złożonego systemu inteligentnego budynku. Mikrokontroler jest znacznie tańszy, ale jego programowanie wymaga większej wiedzy, dlatego amatorzy zwykle wybierają Raspberry Pi. Rozwiązaniem pośrednim dla amatorów może być wykorzystanie platformy Arduino, która bazuje na AVR, jednak oferuje łatwiejsze metody programowania, większe wsparcie i szerszą dostępność bibliotek oraz gotowych projektów.

W profesjonalnych zastosowaniach mikrokontroler wydaje się być bardziej uzasadniony niż komputer typu Raspberry Pi. O ile system nie jest zbyt skomplikowany, w zupełności wystarczy AVR. W bardziej złożonych przypadkach zamiast procesorów z tej rodziny można wykorzystać bardziej zaawansowany układ, taki jak np. mikrokontrolery ARM. AVR i inne proste układy idealnie za to sprawdzą się jako sterowniki urządzeń, komunikujące się tylko z serwerem, nie bezpośrednio z użytkownikiem. W tego typu rozproszonych systemach również zastosowanie Raspberry Pi znajduje swoje uzasadnienie – może ono być wykorzystane jako serwer, który komunikuje się z innymi

urządzeniami, w tym mikrokontrolerami sterującymi konkretnym sprzętem domowym.

Zastosowanie Raspberry Pi jako niezależnego sterownika pojedynczego urządzenia powodowałoby, że koszty systemu byłyby zwykle nie do przyjęcia. Amatorzy muszą więc albo zorganizować system tak, aby wszystkie urządzenia były kontrolowane bezpośrednio przez Raspberry Pi, albo wykorzystać również mikrokontrolery lub wspomnianą już platformę Arduino.

Kwestiami wartymi przeanalizowania w przyszłości są:

- implementacja zabezpieczeń (hasła, szyfrowanie),
- skalowalność systemów.

## Literatura

- [1] Baranowski R.: Mikrokontrolery AVR ATmega w praktyce. BTC, Warszawa 2005.
- [2] Doliński J.: Mikrokontrolery AVR w praktyce. BTC, Warszawa 2008.
- [3] Monk S.: Raspberry Pi. Receptury. Helion, Gliwice 2014.
- [4] Skalski M.: System do zarządzania automatyką domową z wykorzystaniem terminalu mobilnego i platformy Raspberry Pi – praca inżynierska, AGH, Kraków 2013.
- [5] Spale J.: Home Automation with a low-cost AVR-based Board. IFAC-PapersOnLine 48(4)/2015, 398–403.
- [6] Upton E., Halfacree G.: Raspberry Pi. Przewodnik użytkownika. Helion, Gliwice 2015.
- [7] Vujović V., Maksimović M.: Raspberry Pi as a Sensor Web node for home automation. Computers & Electrical Engineering 44/2015, 153–171.
- [8] Dokumentacja biblioteki GPIO dla Raspberry Pi, <http://wiringpi.com/> (available 9.06.2016).
- [9] Dokumentacja modułu Ethernet ENC28J60, <http://ww1.microchip.com/downloads/en/DeviceDoc/39662c.pdf> (available 30.04.2016).
- [10] Dokumentacja oprogramowania owfs i owserver, <http://owfs.org/> (available 10.06.2016).
- [11] Dokumentacja Raspberry Pi, <https://www.raspberrypi.org/documentation/> (available 30.04.2016).
- [12] Dokumentacja techniczna mikrokontrolera ATmega328P, [http://www.atmel.com/images/Atmel-8271-8-bit-AVR-Microcontroller-ATmega48A-48PA-88A-88PA-168A-168PA-328-328P\\_datasheet\\_Complete.pdf](http://www.atmel.com/images/Atmel-8271-8-bit-AVR-Microcontroller-ATmega48A-48PA-88A-88PA-168A-168PA-328-328P_datasheet_Complete.pdf) (available 10.06.2016).
- [13] Dokumentacja techniczna mikrokontrolera ATmega32A, [http://www.atmel.com/images/atmel-8155-8-bit-microcontroller-avr-atmega32a\\_datasheet.pdf](http://www.atmel.com/images/atmel-8155-8-bit-microcontroller-avr-atmega32a_datasheet.pdf) (available 30.04.2016).
- [14] Dokumentacja termometru Dallas DS18B20, <https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf> (available 10.06.2016).
- [15] Dokumentacja termometru LM35, <http://www.ti.com/lit/ds/symlink/lm35.pdf> (available 10.06.2016).
- [16] Projekt z implementacją stosu TCP/IP dla AVR, <http://www.tuxgraphics.org/electronics/200611/embedded-webserver.shtml> (available 9.06.2016).

### Inż. Piotr Pańczyk

e-mail: pmpancyk@gmail.com

Student Informatyki II stopnia na Politechnice Lubelskiej, specjalności Aplikacje internetowe. Od 2014 roku pracuje jako Specjalista ds. Systemów IT w Asseco Business Solutions, gdzie zajmuje się rozwojem i utrzymaniem usług działających w systemie Linux. Interesuje się też szeroko pojętą elektroniką i telekomunikacją, systemami operacyjnymi, sieciami komputerowymi.



### Dr inż. Jakub Smółka

e-mail: jakub.smolka@pollub.pl

Pracownik Instytutu Informatyki, Wydziału Elektrotechniki i Informatyki Politechniki Lubelskiej. Stopień magistra otrzymał na Politechnice Lubelskiej, tytuł doktora na Politechnice Śląskiej. Obszar działalności naukowej obejmuje przetwarzanie danych motion capture, przetwarzanie i kompresję cyfrowych obrazów.



otrzymano/received: 12.06.2016

przyjęto do druku/accepted: 01.06.2017