

Analiza możliwości wykorzystania technologii IoT w systemach inteligentnych domów

Arkadiusz Bęben*, Piotr Kopniak

Politechnika Lubelska, Instytut Informatyki, Nadbystrzycka 36B, 20-618 Lublin, Polska

Streszczenie. Wiele urządzeń wyposażenia domowego dostępnych na rynku, jest wyposażona w podzespoły umożliwiające odbieranie i generowanie danych oraz moduły komunikacji pozwalające na połączenie z Internetem. Dzięki temu urządzenia mogą być łączone w systemy inteligentnych domów. Analiza przedstawia poszczególne możliwości urządzeń, które mogą pracować w takich systemach. Przedstawione są urządzenia dostępne na rynku oraz wykonany system inteligentnego domu pracujący w sieci Wi-Fi oraz Z-Wave.

Słowa kluczowe: Internet of Things; inteligentny dom; inteligentne rzeczy

* Autor do korespondencji.

Adres e-mail: arekbeben@gmail.com

Analysis of the application possibilities of IoT technology in smart home systems

Arkadiusz Bęben*, Piotr Kopniak

^a Institute of Computer Science, Lublin University of Technology, Nadbystrzycka 36B, 20-618 Lublin, Poland

Abstract. Many home appliances available on the market are equipped with components to receive and generate data and communication modules allow connection to the Internet. As a result, the devices can be combined to smart homes systems. The analysis presents the individual capabilities of devices that can work in such systems. Presented are devices available on the market and made a smart home system working in Wi-Fi and Z-Wave networks.

Keywords: Internet of Things; smart home; smart things

* Corresponding author.

E-mail address: arekbeben@gmail.com

1. Wstęp

W ostatnich latach postęp technologiczny umożliwił wprowadzenie na rynek urządzeń wyposażonych w różnego rodzaju czujniki oraz moduły sieciowe pozwalające na komunikację z innymi urządzeniami, a w następstwie podłączenie ich do Internetu. Konsekwencją takiej sytuacji jest rosnąca liczba inteligentnych urządzeń w pomieszczeniach mieszkalnych. Obecnie coraz więcej wykonywanych czynności domowych, jest wspierane poprzez zastosowanie w nich urządzeń posiadających połączenie internetowe.

1.1. Internet of Things

Internet rzeczy (ang. IoT – Internet of Things) jest koncepcją informatyczną, powstałą w wyniku potrzeby określenia urządzeń codziennego użytku, które potrafią generować, przesyłać lub przechowywać dane, bez konieczności ingerencji użytkownika. Cechą charakterystyczną przedmiotów przynależnych do tej idei, jest podłączenie do sieci komputerowej. Oczywiście nie wyklucza to wykorzystania innych sposobów komunikacji, zarówno z użytkownikiem końcowym, jak i z innymi urządzeniami wchodzącymi w skład poszczególnych instalacji. Główną zaletą takiego rozwiązania jest możliwość śledzenia przez użytkownika bieżącego stanu oraz pomiarów czynionych

przez takie urządzenia, a także wydawania poleceń mających wpływ na obiekty, które są pod kontrolą tych urządzeń.

Celem wyposażenia pomieszczeń w przedmioty korzystające z rozwiązań zawartych w omawianej koncepcji, jest usprawnienie codziennych czynności, wykonywanych przez osoby przebywające w zasięgu ich działania. Ustalenie takiego priorytetu IoT powoduje, że urządzenia wykorzystujące daną ideę obejmują bardzo szerokie spektrum obszarów życia. Wpływ technologii wchodzących w skład pojęcia Internetu rzeczy jest coraz częściej zauważalny i rozpoczyna swoje działanie już od systemów wspierających wykonywanie codziennych obowiązków w mieszkaniach i gospodarstwach domowych. Kolejne miasta wprowadzają urządzenia IoT do systemów miejskiego monitoringu, czy wczesnego wykrywania zagrożeń [5]. Wielkie przedsiębiorstwa odpowiedzialne za produkcję energii elektrycznej, wspierają działania odpowiedniego rozpraszania energii, poprzez zastosowanie urządzeń IoT monitorujących bieżące zużycie energii w danych regionach. Czynności handlowe, związane z logistyką, czy magazynowaniem towarów, również są obsługiwane przez szereg urządzeń podłączonych do sieci [6].

1.2. Inteligentny domy

Rozwój technologii w ostatnich latach przyczynił się znacząco do modyfikacji podejścia w kwestii projektowaniu budynków. Duży wpływ na to miał spadek cen urządzeń elektronicznych i rozwój kolejnych sposobów komunikacji i przesyłania danych. Dzięki temu, iż kolejne urządzenia gospodarstwa domowego, zaczęły być wyposażane w obsługę nowych rozwiązań technologicznych, takich jak wszelkiego rodzaju czujniki, przełączniki, czy moduły umożliwiające podłączenie do sieci telekomunikacyjnej; zaczęto projektować specjalne systemy, w obrębie których wszelkie podmioty, mogą być obsługiwane z poziomu jednego urządzenia sterującego. w przypadku urządzeń IoT takim punktem sterującym najczęściej jest smartfon lub tablet lokatora budynku. Możliwości jakie dają nowoczesne urządzenia zaczęto uwzględniać już na poziomie projektowania budynków i traktować jako integralną część nowoczesnego budownictwa. Takie podejście do projektowania budynków przybrało określenie inteligentnego domu (ang. smart home) [1].

Dzięki implementacji jednego z dostępnych na rynku systemów inteligentnych domów, użytkownik dostaje kontrolę nad wieloma urządzeniami, z których każde może być odpowiedzialne za obsługę odrębnych czynności. Wyposażenie kolejnych przedmiotów gospodarstwa domowego w obsługę nowych technologii sprawiło, że dobrze znane nam urządzenia otrzymały nowe funkcjonalności. To z kolei przełożyło się na większe spersonalizowanie obecnego stanu pomieszczenia (poprzez np. ustawienie temperatury lub poziomu oświetlenia), a w dalszej kolejności na wzrost komfortu jego lokatora. Jednak zastosowanie urządzeń IoT ma również inne pozytywne skutki. Wyposażenie budynku w urządzenia monitorujące jego stan, takie jak przykładowo czujniki dymu, wody, otwarcia okien i drzwi, wpływają bezpośrednio na wzrost bezpieczeństwa mieszkańców, a w połączeniu z modułem pozwalającym zaalarmować odpowiednie służby (policja, straż pożarna) skracają czas reakcji w sytuacji zagrożenia zdrowia i życia lokatorów.

Nie bez znaczenia jest również kwestia wykorzystania energii elektrycznej przez użytkowników danego pomieszczenia. Dzięki zautomatyzowaniu pracy urządzeń i instalacji elektrycznych, prąd zużywany jest tylko wtedy gdy rzeczywiście jest niezbędny do wykonywania domowych czynności. Szczególnie ważnym aspektem jest zarządzanie oświetleniem. System dzięki czujnikom ruchu oraz natężenia światła, potrafi utrzymać prawidłowe oświetlenie pomieszczenia, a w przypadku gdy nie wykryje żadnych osób lub naturalne światło okaże się wystarczające – wyłączy oświetlenie zasilane energią elektryczną. Pomocne również są harmonogramy, w których użytkownik może określić, kiedy w domu nikogo nie ma (np. podczas czasu pracy poza domem), dzięki czemu system wyłącza niepotrzebne urządzenia elektryczne. Takie działania powodują zmniejszenie zużycia energii, co finalnie prowadzi do oszczędności przy płaceniu rachunków.

2. Sposoby komunikacji urządzeń IoT

Urządzenia IoT wykorzystują różne sposoby komunikacji bezprzewodowej. Obecnie dostępne są protokoły przygotowane specjalnie z myślą o urządzeniach przesyłających małe ilości danych w krótkich odstępach czasu. Systemy inteligentnych domów opierają swoje działanie najczęściej o konkretny protokół przesyłania danych pomiędzy połączonymi podmiotami. Cechą charakterystyczną koncepcji IoT jest to, że systemy wykorzystujące to rozwiązanie składają się głównie z wielu bezprzewodowych sieci czujnikowych WSN (ang. Wireless sensor network) i urządzeń korzystających z połączeń radiowych RFID (ang. Radio-frequency identification). Paradygmat WSN definiuje sieć złożoną z wielu czujników potrafiących komunikować się ze sobą poprzez łączność radiową. Urządzenia RFID składają się przeważnie z układu scalonego z pewnymi możliwościami obliczeniowymi i anteny służącej do komunikacji [3].

2.1. Technologie połączeń

Podstawowym sposobem komunikacji urządzeń IoT z Internetem jest przeważnie łączność Wi-Fi. Jest to spowodowane przede wszystkim popularnością technologii i faktem, że w przypadku jej wykorzystania, urządzeniem dostępowym do Internetu może być domowy router. Jednak nie jest to najbardziej optymalny sposób komunikacji dla urządzeń IoT, gdyż charakteryzuje się dużą szybkością przesyłania danych, która nie jest konieczna w urządzeniach generujących je w małych ilościach. Ponadto moduł Wi-Fi posiada największe zapotrzebowanie na energię w porównaniu do przedstawionych dalej technologii, co jest ważną kwestią w przypadku urządzeń posiadających zasilanie z baterii.

Kolejnym sposobem komunikacji między urządzeniami może być wykorzystanie technologii Bluetooth. Charakterystyka przesyłania danych opierająca się na krótkim zasięgu, małej konsumpcji energii elektrycznej i małej szybkości sprawia, że jej wykorzystanie idealnie wpasowuje się w sieć urządzeń obejmujących jedno pomieszczenie. Minusem jest konieczność posiadania centrali służącej jako punkt dostępu do Internetu, a plusem możliwość połączenia z urządzeniem mobilnym.

Następnym protokołem komunikacji bezprzewodowej, który został stworzony głównie na potrzeby urządzeń IoT jest Zigbee. Wykorzystuje on pasma sieci 2,4 GHz, 915 MHz lub 868 MHz, przesyłając dane na odległość do 20 metrów i tworząc sieci w topologii siatki, co sprawia, że każde z urządzeń odbiera i przekazuje dane z innych znajdujących się w sieci. Dzięki takiemu rozwiązaniu z łatwością możliwe jest pokrycie większych powierzchni, czy też wielu pięter mieszkania. Charakterystyczną cechą technologii jest niskie zużycie energii [2].

Bezpośrednim konkurentem dla protokołu Zigbee jest pracująca na pasmach 868.42 MHz i 908.42 MHz, Z-Wave. Charakteryzuje się ona również działaniem w topologii siatki, jednak oferuje mniejsze możliwości jeżeli chodzi

o maksymalną liczbę urządzeń podłączonych w jednej sieci. Dla Z-Wave wynosi ona 232, a Zigbee potrafi utrzymać połączenie pomiędzy aż 65 000 urządzeń. Z-Wave jest natomiast lepszy pod względem maksymalnej odległości pomiędzy urządzeniami, która może wynosić nawet do 100 metrów [4].

2.2. Oprogramowanie sterujące

Aspektem, od którego zależy wydajność, ale przede wszystkim funkcjonalność sieci zbudowanej z urządzeń IoT jest oprogramowanie. Urządzenia dostępne na rynku, coraz częściej otrzymują możliwość obsługi wybranych systemów inteligentnych domów, lub są wręcz projektowane i produkowane z myślą o konkretnym systemie. Nie oznacza to, że nie występują również przyrządy posiadające autorskie oprogramowanie z przygotowanymi specjalnie aplikacjami, jednak wadą takiego rozwiązania jest brak możliwości reakcji na dane generowane przez inne urządzenia. Najpopularniejszymi systemami są te wykorzystujące systemy urządzeń mobilnych, takie jak Google Home od Google, czy HomeKit od Apple, które to firmy posiadają duopol na rynku systemów smartfonów (Android i iOS). Kolejnym ze zjawisk, które występują na rynku systemów inteligentnych domów, jest obsługiwane urządzeń, które najczęściej zostały wyprodukowane przez producenta oprogramowania – przeważnie aplikacji mobilnej. Do takich systemów zalicza się przykładowo Mi Home od Xiaomi, czy FIBARO Home Center od Fibaro. Oddzielną kategorią systemów inteligentnych domów, jest oprogramowanie dostępne na licencji open source, takie jak OpenHAB, Domoticz, czy Home Assistant. z racji udziału społeczności w tworzeniu takich systemów, obsługują one coraz więcej urządzeń i funkcjonalności. Ciekawym rozwiązaniem, pozwalającym na łączenie urządzeń różnych producentów jest również IFTTT. Jest to serwis internetowy, który pozwala na obsługę i tworzenie własnych apletów, czyli prostych poleceń wykonujących się w koncepcji „IfThis Then That”, co można przetłumaczyć na „jeśli to, to to”. Pozwala ona dodać zależności pomiędzy aplikacjami obsługiwanymi przez serwis i wykonać polecenia na podstawie ich powiadomień, czyli wybrać zdarzenie wywołane przez jedno z urządzeń i gdy ono wystąpi wykonać ustaloną akcję.

3. Cel i plan badań

Podstawą technologii IoT jest generowanie, przesyłanie i przetwarzanie odebranych danych przez urządzenia, dlatego wykorzystanie możliwości zostało przedstawione na przykładzie logów z funkcjonalnego systemu inteligentnego domu. Logi pobrane z systemu ukazują przepływ danych, przetwarzanych w systemie oraz w konsekwencji, zachowanie poszczególnych urządzeń, w zależności od odebranych danych i przygotowanego wcześniej oprogramowania. Celem badań było wykazanie możliwości technologii Internet of Things poprzez zachowanie poszczególnych urządzeń, które obecnie są dostępne na rynku i sprawdzenie, czy system inteligentnego domu, może zostać wykonany w prosty sposób, przy wykorzystaniu kilku urządzeń i oprogramowania typu open source. Ponadto został przedstawiony i zweryfikowany proces przygotowania i pracy

takiego systemu z wykorzystaniem protokołu sieciowego Z-Wave. Teżą postawioną w niniejszej pracy jest stwierdzenie:

Urządzenia IoT dostępne na rynku nadają się do wykorzystania w systemach inteligentnych domów.

Badania doprowadziły do zweryfikowania postawionych hipotez:

Hipoteza 1: Dostępne urządzenia IoT można włączyć do systemu inteligentnego domu dzięki ogólnie dostępnemu oprogramowaniu.

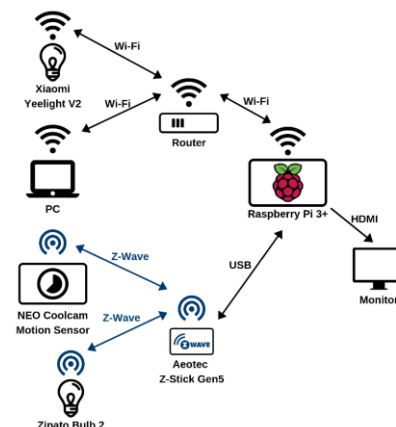
Hipoteza 2: Możliwe jest oprogramowanie działania poszczególnych urządzeń IoT działających w systemie inteligentnego domu.

Hipoteza 3: Dzięki stosowanym standardom IoT urządzenia w systemie inteligentnego domu mogą ze sobą współpracować.

W celu weryfikacji postawionych założeń, został przygotowany system inteligentnego domu, który łączy się z urządzeniami poprzez sieć Wi-Fi oraz sieć wykorzystującą protokół Z-Wave.

3.1. System inteligentnego domu

Model systemu inteligentnego domu składa się z kilku urządzeń działających w dwóch sieciach bezprzewodowych. Podstawowym urządzeniem jest centrala Raspberry Pi 3+ z zainstalowanym systemem kontrolującym pracę urządzeń. Centrala posiada moduł sieci Wi-Fi, jak i moduł sieciowy Aeotec Z-Stick Gen5, umożliwiający stworzenie sieci bezprzewodowej w technologii Z-Wave. w systemie jako urządzenia przeznaczone do sterowania, zastosowano dwie żarówki połączone w dwóch różnych sieciach bezprzewodowych. Zipato Bulb 2 obsługuje połączenie bezprzewodowe korzystające z protokołu Z-Wave, a Xiaomi Yeelight V2 w technologii Wi-Fi. Urządzeniem posiadającym czujnik trzech właściwości (temperatura, natężenie światła, ruch) jest NEO Coolcam Motion Sensor i to na podstawie odczytów tego czujnika wykonywane były akcje pozostałych urządzeń.



Rys. 1. Schemat ideowy połączeń pomiędzy urządzeniami

Do instalacji na wcześniej przedstawionej konfiguracji sprzętowej został wybrany system inteligentnego budynku Domoticz, udostępniane na zasadzie licencji open source. Instalacja systemu, wymagała instalacji na urządzeniu Raspberry Pi systemu Raspbian, a następnie z konsoli systemowej, wpisania komendy pobierającej i instalującej Domoticz. Głównym celem systemu inteligentnego domu jest zautomatyzowanie czynności. Domoticz opiera proces automatyzacji o wydarzenia (ang. events) i skrypty (ang. scripts). Każde wydarzenie ma własną nazwę, metodę skryptu i typ, który określa kiedy wydarzenie ma być uruchamiane. Typ wydarzenia może być określony jako zależny od urządzenia (uruchamiany podczas zmiany danych przesyłanych z urządzenia), zabezpieczenia (zmiana uzbrojenia systemu) lub czasu (wydarzenie uruchamiane cyklicznie, z określonym odstępem czasowym). Każde wydarzenie dodatkowo ma określoną wartość określającą, jego stan aktywności (aktywne/ nie aktywne). Skrypty przeznaczone do wykorzystania w wydarzeniach mogą być przygotowane za pomocą edytora Blockly lub języków skryptowych takich jak LUA, dzVents, Perl czy PHP.

4. Konfiguracja systemu Domoticz

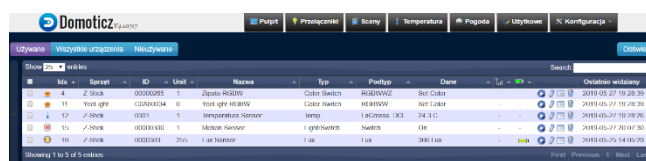
Aby zainstalować Domoticz należało posłużyć się instalacją systemu Raspbian i z poziomu konsoli systemowej należało wpisać komendę:

```
sudo curl -L install.domoticz.com | bash
```

Proces instalacji przebiegł bezproblemowo, a po jej zakończeniu system został zrestartowany. Wtedy możliwe było sprawdzenie jaki adres sieciowy został przypisany do urządzenia. Następnym krokiem było wpisanie tego adresu w przeglądarce Google Chrome, na systemie Windows 10, który był zainstalowany na komputerze podłączonym do tej samej sieci Wi-Fi co Raspberry Pi, w celu połączenia z witryną internetową do zarządzania udostępnioną pod tym adresem poprzez zainstalowany serwis Domoticz. Dzięki temu było możliwe zarządzanie systemem Domoticz z poziomu systemu internetowego.

W celu dodania urządzeń sieci Z-Wave, należało w sekcji „Sprzęt” dodać moduł Aeotec Z-Stick Gen5 jako urządzenie typu Open Z-Wave USB, a następnie włączyć skanowanie urządzeń. w trakcie skanowania należało nacisnąć przycisk na czujniku NEO Coolcam Motion Sensor, w celu włączenia do sieci Z-Wave. Żarówka Zipato Bulb 2 została wykryta automatycznie, bez konieczności wykonywania jakichkolwiek akcji. Żarówkę Xiaomi Yeelight V2, pracującą w sieci Wi-Fi podłączono do systemu poprzez dodanie w sekcji „Sprzęt” wirtualnego kontrolera Yellight LED. Następnie należało w aplikacji Yeelight służącej do kontrolowania pracy urządzenia, a zainstalowanej na urządzeniu mobilnym z systemem Android, włączyć możliwość sterowania żarówką poprzez sieć LAN. Po wykonaniu tej czynności, włączono skanowanie na kontrolerze w systemie Domoticz, który odnalazł urządzenie YeeLight RBGW.

W sekcji „Urządzenia” należało włączyć podmioty, które zostały dodane do systemu. Obydwie żarówki zostały sklasyfikowane jako urządzenia tego samego typu, z funkcjonalnością, pozwalającą zarządzać natężeniem oraz kolorem emitowanego światła. Podstawową różnicą w przypadku obu urządzeń była sekcja określona jako „Sprzęt”, gdzie w przypadku połączenia poprzez sieć Wi-Fi została wskazana nazwa urządzenia, a dla Zipato Bulb 2 widniała nazwa modułu sieciowego, dzięki któremu możliwe było połączenie. Różnicę da się również zobaczyć w sekcji „Podtyp” gdzie dla urządzenia Yellight wskazany jest RGBWW, czyli kontrola emitowanego światła w palecie barw RGB oraz światła chłodnego białego i ciepłego białego. w przypadku drugiej żarówki, podtyp określono jako RGBWWZ, co pozwala na kontrolę w paletach jak w poprzednim wypadku oraz mieszania palety odpowiedzialnej za kolory RGB i palety zakresów światła białego. Czujnik był pokazany w systemie jako trzy oddzielne moduły służące do pomiaru temperatury, ruchu oraz natężenia światła.



Rys. 2. Urządzenia systemu Domoticz

W celu przeprowadzenia przypadków testowych, należało włączyć urządzenia:

- YeeLight RBGW – kontrola żarówki Yellight
- Zipato RGBW – kontrola żarówki Zipato
- Lux Sensor – odczyt czujnika wartości natężenia światła
- Motion Sensor – odczyt czujnika ruchu
- Temperature Sensor – odczyt czujnika temperatury

5. Przypadki testowe

Przypadki testowe obejmowały instalację i konfigurację systemu z wykorzystaniem opisanych urządzeń, oraz próbę uzależnienia inteligentnych żarówek od czujnika natężenia światła, czujnika ruchu oraz wartości mierzonej temperatury. Pierwszym etapem konfiguracji było sprawdzenie, czy żarówki połączone poprzez sieć bezprzewodową Wi-Fi i Z-Wavesą w stanie regulować natężenie oświetlenia w zależności od czujnika ruchu. w drugim przypadku została przeprowadzona konfiguracja wykorzystująca czujnik natężenia oświetlenia, który po zmianie wartości, wymusza na systemie zmianę mocy świecenia żarówek. Trzeci przypadek obejmuje regulację koloru światła żarówki, w zależności od zmierzonej wartości temperatury.

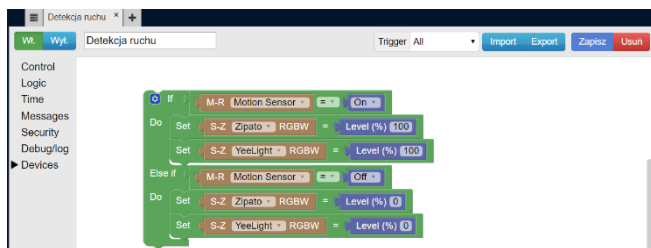
Podstawową funkcjonalnością NEO Coolcam Motion Sensor, jest wykrywanie ruchu. Aby umożliwić prawidłową pracę czujnika ruchu, należało ustawić parametry konfiguracyjne urządzenia:

Tabela 1. Ustawione parametry konfiguracyjne urządzenia NEO Coolcam Motion Sensor

Numer parametru	Nazwa	Wartość
1	Motion detection sensitivity	High Sensivity
2	Motion detection ON time	30
3	Basic Set Level	255
4	Motion detection function	Enable
6	Motion detection blind time	5
10	Motion detection LED indicator	Enable

Aby uzyskać pewność, że nawet najmniejszy ruch użytkownika zostanie wykryty, ustawiono parametr numer 1 jako wysoką czułość. Parametrem numer 2 określono, że urządzenia włączone poprzez pomiar czujnika ruchu mają pozostać włączone przez 30 sekund. w parametrze numer 6 ustawiono 5 sekund jako czas po jakim czujnik zacznie reagować na kolejny ruch w jego obszarze, od momentu wykrycia poprzedniego ruchu. Dodatkowo w parametrze numer 10 włączono sygnalizację diodą LED po wykryciu ruchu przez czujnik.

W przypadku testowym „Detekcja ruchu” uzależniono włączenie żarówek od wykrycia ruchu przez sensor. Pierwszy przypadek powoduje włączenie oświetlenia po wykryciu ruchu, a następnie jego wyłączenie po zakończeniu aktywności czujnika. Drugi przypadek „Detekcja ruchu (2 minuty)” powoduje włączenie oświetlenia po wykryciu ruchu na czas 2 minut. Obydwa przypadki zostały obsługane przez stworzenie wydarzeń ze skryptami edytora Blockly.



Rys. 3. Skrypt wydarzenia „Detekcja ruchu”



Rys. 4. Skrypt wydarzenia „Detekcja ruchu (2 minuty)”

Drugą z funkcjonalności urządzenia NEO Coolcam Motion Sensor jest pomiar natężenia światła. w celu zapewnienia poprawnego działania sensora ustawiono następujące parametry:

Tabela 2. Ustawione parametry konfiguracyjne urządzenia NEO Coolcam Motion Sensor

Numer parametru	Nazwa	Wartość
7	Illumination reporting interval	180
8	Illuminationfunction	Enable
9	Illuminationreport threshold	5

Parametr numer 7, ustawiono 180 sekund, jako czas jaki musi upłynąć aby urządzenie przesłało zmierzoną wartość natężenia światła, w przypadku gdy nie odnotowano zmiany większej, niż wartość parametru numer 9. w parametrze 9 ustawiono wartość 5 lux, jako różnicę od poprzedniego pomiaru, która powoduje przesłanie nowego pomiaru do kontrolera.

W pierwszym przypadku testowym „Regulacja oświetlenia” regulowano intensywność emitowanego światła żarówek, w zależności od wartości zmierzonej przez czujnik natężenia oświetlenia. Gdy wartość zmierzona wynosiła poniżej 10 lux, to żarówki świeciły z maksymalną mocą. Dla zmierzonych wartości pomiędzy 10 lux a 100 lux, moc żarówek była ustawiana na 50%. w przypadku zmierzenia ponad 100 lux, system wyłączał oświetlenie. Wraz z każdą zmianą wartości odczytanej z urządzenia Lux Sensor, system zapisywał odebraną wartość.



Rys. 5. Skrypt wydarzenia „Regulacja oświetlenia”

Drugi przypadek testowy „Automatyczna regulacja oświetlenia” pokazuje w jaki sposób możliwe jest wykonanie automatycznej regulacji oświetlenia, do ustalonej wartości natężenia światła. Tym razem skrypt do wydarzenia został przygotowany w języku skryptowym dzVents. Zadaniem skryptu jest regulowanie mocy świecenia żarówek w ten sposób, aby uzyskać wartość oświetlenia w pomieszczeniu zadeklarowaną w zmiennej LuxValue.


```

Wł. Wyt. Automatyczna regulacja ośi
1 return {
2   on = {
3     devices = { 'Lux Sensor' }
4   },
5   execute = function(domoticz, item, triggerInfo)
6     local LuxValue = 50
7     local GranicaBledu = 5
8     local LuxDevice = domoticz.devices('Lux Sensor')
9     local LuxLevel = LuxDevice.lux
10    local ZipatoDevice = domoticz.devices('Zipato RGBW')
11    local ZipatoLevel = ZipatoDevice.level
12    local YeeLightDevice = domoticz.devices('YeeLight RGBW')
13    local YeeLightLevel = YeeLightDevice.level
14
15    if LuxLevel + GranicaBledu < LuxValue then
16      ZipatoDevice.dimTo(ZipatoLevel + 10)
17      YeeLightDevice.dimTo(ZipatoLevel + 10)
18    end
19    if LuxLevel - GranicaBledu > LuxValue then
20      ZipatoDevice.dimTo(ZipatoLevel - 10)
21      YeeLightDevice.dimTo(ZipatoLevel - 10)
22    end
23  end
24 }
    
```

Rys. 6. Skrypt wydarzenia „Automatyczna regulacja oświetlenia”

Skrypt w linii 3 deklaruje urządzenie, którego zmiana wartości ma powodować wykonanie kodu. Tym urządzeniem jest Lux Sensor, czyli czujnik natężenia światła. Następnie deklarowane są zmienne:

- LuxValue – wartość oświetlenia jaką chcemy uzyskać w pomieszczeniu
- GranicaBledu – różnica pomiędzy pomiarem czujnika oświetlenia a docelową wartością (LuxValue), jaka nie spowoduje wywołania skryptu
- LuxDevice – zmienna przechowująca odwołanie do urządzenia Lux Sensor
- LuxLevel – zmienna przechowująca wartość zmierzoną przez urządzenie Lux Sensor
- ZipatoDevice – zmienna przechowująca odwołanie do urządzenia Zipato Bulb 2
- ZipatoLevel – zmienna przechowująca wartość mocy oświetlenia emitowanej przez żarówkę Zipato Bulb 2
- YeeLightDevice – zmienna przechowująca odwołanie do urządzenia Xiaomi Yeelight V2
- YeeLightLevel – zmienna przechowująca wartość mocy oświetlenia emitowanej przez żarówkę Xiaomi Yeelight V2

W ostatniej części skryptu wywoływane są instrukcje warunkowe „if”, które sprawdzają czy obecny pomiar czujnika natężenia światła, jest mniejszy bądź większy od docelowej wartości i odpowiednio zwiększają bądź redukują moc świecenia żarówek.

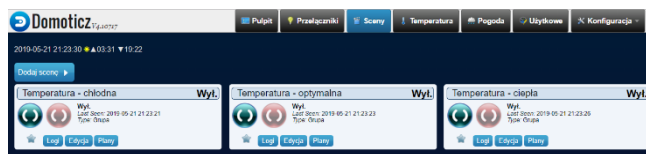
Pomiary temperatury, w przeciwieństwie do poprzednich wartości mierzonych przez urządzenie, nie posiadają możliwości konfiguracji.

W przypadku testowym posłużono się żarówkami jako urządzeniem informującym o obecnej temperaturze w pomieszczeniu. Temperatury skategoryzowano i podzielono na 3 zakresy:

- mniej niż 20°C – temperatura chłodna

- od 20°C do 25°C – temperatura optymalna
- więcej niż 25°C – temperatura ciepła

Aby możliwe było ustawienie kolorów emitowanych przez żarówki, utworzono trzy sceny. w każdej scenie należało dodać dwie żarówki i ustawić ich wartości w kodzie heksadecymalnym, przedstawione w Tabeli 3.



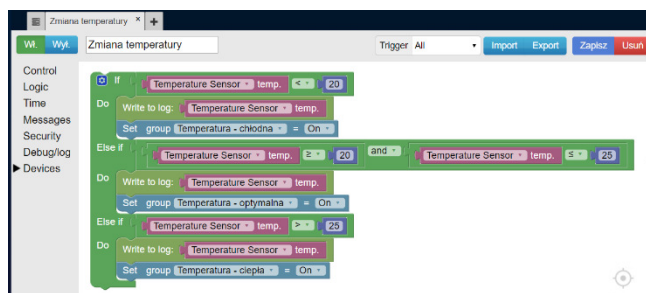
Rys. 7. Sceny w systemie Domoticz

Aktywacja każdej ze scen powoduje włączenie i ustawienie określonych kolorów na żarówkach:

Tabela 3. Kolory żarówek ustawione dla danej sceny

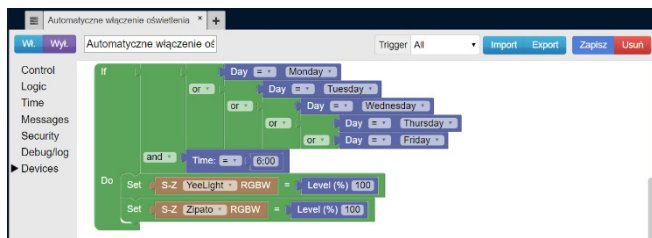
Nazwa sceny	Kod Hex	Wartość RGB	Kolor
Temperatura – chłodna	0100FF	1, 0, 255	
Temperatura – optymalna	2CFF00	44, 255, 0	
Temperatura – ciepła	FF000C	255, 0, 12	

Za aktywację scen odpowiada wydarzenie „Zmiana temperatury” ze skryptem w edytorze Blockly. Skrypt zmienia stan konkretnej sceny w zależności od zmierzonej temperatury.



Rys. 8. Skrypt wydarzenia „Zmiana temperatury”

Edytor Blockly dostępny z poziomu systemu Domoticz, pozwala również na ustawienie zależności innych urządzeń od obecnego czasu. w utworzonym wydarzeniu „Automatyczne włączanie oświetlenia” posłużono się funkcjonalnością, która włącza oświetlenie na maksymalną wartość, o godzinie 6:00, w dniach od poniedziałku do piątku.



Rys. 9. Skrypt wydarzenia „Automatyczne włączenie oświetlenia”

6. Wyniki

Instalacja systemu Domoticz na urządzeniu Raspberry Pi 3+, nie sprawiła problemów i nie stworzyła nieprzewidzianych sytuacji. Wszystkie kroki zostały wykonane według instrukcji instalacji, dostępnej na stronie internetowej oprogramowania. Dodanie urządzeń i utworzenie połączeń zarówno w sieci Wi-Fi, jak i Z-Wave było możliwe dzięki intuicyjnemu interfejsowi systemu inteligentnego budynku i przebiegło bezproblemowo.

W pierwszym przypadku „Detekcja ruchu” system włączał oświetlenie, w momencie wykrycia ruchu przez czujnik. System po wykryciu zmiany stanu czujnika ruchu Motion Sensor informuje o uaktywnieniu wydarzenia i zmianie stanu żarówek. System zmienia poziom oświetlenia komendą ustawiając wartość Level: 99. Po około 30 sekundach, czyli wartości ustawionej w konfiguracji czujnika ruchu, system aktywuje ponownie wydarzenie wyłączając oświetlenie (Level: 0).

```
20:36:10.869 (Z-Stick) Light/Switch (Motion Sensor)
20:36:10.370 EventSystem: Event triggered: Detekcja ruchu_1
20:36:10.373 OpenZWave: Domoticz has send a Switch command!, Level: 99, NodeID: 2 (0x02)
20:36:10.388 (Z-Stick) Color Switch (Zipato RGBW)
20:36:10.941 (Z-Stick) Light/Switch (Level)
20:36:11.008 (YeeLight) Color Switch (YeeLight RGBW)
20:36:41.999 (Z-Stick) Light/Switch (Motion Sensor)
20:36:42.001 EventSystem: Event triggered: Detekcja ruchu_2
20:36:42.026 OpenZWave: Domoticz has send a Switch command!, Level: 0, NodeID: 2 (0x02)
20:36:42.040 (Z-Stick) Color Switch (Zipato RGBW)
20:36:42.113 (YeeLight) Color Switch (YeeLight RGBW)
```

Rys. 10. Logi systemu Domoticz - aktywacja wydarzenia "Detekcja ruchu"

Kolejny przypadek „Detekcja ruchu (2 minuty)” posiadała podobną sekwencję wydarzeń, jednak tym razem komenda aktywująca skrypt wyłączający oświetlenie wysyłana jest dopiero po około 2 minutach od momentu wykrycia ruchu, czyli zgodnie z przygotowanym skrypcem wydarzenia.

```
20:39:47.706 Light/Switch (Motion Sensor)
20:39:47.707 (Z-Stick) EventSystem: Event triggered: Detekcja ruchu (2 minuty)
20:39:47.737 OpenZWave: Domoticz has send a Switch command!, Level: 49, NodeID: 2 (0x02)
20:39:47.758 (Z-Stick) Color Switch (Zipato RGBW)
20:39:47.804 (Z-Stick) Light/Switch (Level)
20:39:47.885 (YeeLight) Color Switch (YeeLight RGBW)
20:40:18.809 (Z-Stick) Light/Switch (Motion Sensor)
20:41:47.717 OpenZWave: Domoticz has send a Switch command!, Level: 0, NodeID: 2 (0x02)
20:41:47.731 (Z-Stick) Color Switch (Zipato RGBW)
20:41:47.801 (YeeLight) Color Switch (YeeLight RGBW)
```

Rys. 11. Logi systemu Domoticz - aktywacja wydarzenia "Detekcja ruchu (2 minuty)"

Przypadek „Regulacja oświetlenia” pozwolił na zmianę poziomu mocy świecenia żarówek. System po odebraniu nowej wartości z czujnika Lux, informuje, że wydarzenie zostało uruchomione. Wysyła komendę „Switch command” z wartością „Level” określającą zakres mocy, w skali 0-100.

Następnie informuje, że urządzenia Zipato RGBW i YeeLight RGBW zmieniły swój stan. Sekwencja powtarza się trzykrotnie, gdzie w zależności od poziomu zmierzonego przez czujnik Lux, zostaje uruchomiona inna instrukcja warunkowa. Za każdym razem po zmianie wartości Lux Sensor, system zapisuje odczytaną wartość komendą „Status”.

```
21:03:29.084 (Z-Stick) Lux (Lux Sensor)
21:03:29.086 EventSystem: Event triggered: Regulacja oświetlenia_1
21:03:29.118 OpenZWave: Domoticz has send a Switch command!, Level: 99, NodeID: 2 (0x02)
21:03:29.140 (Z-Stick) Color Switch (Zipato RGBW)
21:03:29.224 (Z-Stick) Light/Switch (Level)
21:03:29.268 (YeeLight) Color Switch (YeeLight RGBW)
21:03:29.086 Status: 2
21:03:34.756 (Z-Stick) Lux (Lux Sensor)
21:03:34.757 EventSystem: Event triggered: Regulacja oświetlenia_2
21:03:34.759 OpenZWave: Domoticz has send a Switch command!, Level: 50, NodeID: 2 (0x02)
21:03:34.772 (Z-Stick) Color Switch (Zipato RGBW)
21:03:34.931 (YeeLight) Color Switch (YeeLight RGBW)
21:03:34.758 Status: 14
21:03:40.495 (Z-Stick) Lux (Lux Sensor)
21:03:40.497 EventSystem: Event triggered: Regulacja oświetlenia_3
21:03:40.502 OpenZWave: Domoticz has send a Switch command!, Level: 0, NodeID: 2 (0x02)
21:03:40.521 (Z-Stick) Color Switch (Zipato RGBW)
21:03:40.594 (YeeLight) Color Switch (YeeLight RGBW)
21:03:40.497 Status: 132
```

Rys. 12. Logi systemu Domoticz - aktywacja wydarzenia "Regulacja oświetlenia"

Przypadek „Automatyczna regulacja oświetlenia” reguluje poziom świecenia żarówek za każdym razem o 10%. System w omawianym przypadkusekwencyjnie odczytywał nowe wartości z czujnika natężenia oświetlenia, oraz wykonywał skrypt, zwiększając moc światła emitowanego przez żarówki. Finalnie po zmierzeniu wartości, która mieściła się w granicy błędów wartości docelowej, system wykonał skrypt, nie zmieniając pracy żarówek, co pozwoliło na ustabilizowaniu oświetlenia pomieszczenia na wymaganym poziomie.

Po odebraniu nowej wartości z czujnika temperatury, uruchomiony został skrypt wydarzenia „Zmiana temperatury”. w logach system zapisał aktywowanie sceny „Temperatura – optymalna”, oraz obu żarówek. Do żarówek zostało wysłane polecenie ustawienia mocy emitowania światła przez urządzenia (Level) oraz ustalenia koloru, podane w pięciu wartościach (Red, Green, Blue, White, cWhite). System zapisał również temperaturę zmierzoną przez urządzenie Temperature Sensor.

W funkcjonalnościach systemu niezależnych od wartości odczytanych przez czujniki została przedstawiona sytuacja z włączeniem oświetlenia o konkretnej godzinie ustawionej w skrypcie „Automatyczne włączenie oświetlenia”. Skrypt został zaprogramowany tak, aby aktywować urządzenia tylko w dni robocze tj. od poniedziałku do piątku.

```

21:15:33.939 (Z-Stick) Lux (Lux Sensor)
21:15:40.073 OpenZWave: Domoticz has send a Switch command!, Level: 10, NodeID: 2
(0x02)
21:15:40.081 (Z-Stick) Color Switch (Zipato RGBW)
21:15:40.204 (YeeLight) Color Switch (YeeLight RGBW)
21:15:40.055 Status: dzVents: Info: Handling events for: "Lux Sensor", value: "0"
21:15:40.055 Status: dzVents: Info: ----- Start internal script: Automatyczna
regulacja oświetlenia: Device: "Lux Sensor (Z-Stick)", Index: 19
21:15:40.057 Status: dzVents: Info: ----- Finished Automatyczna regulacja
oświetlenia
21:15:40.058 Status: EventSystem: Script event triggered:
/home/pi/domoticz/dzVents/ runtime/dzVents.lua
21:15:49.769 (Z-Stick) Lux (Lux Sensor)
21:15:49.904 OpenZWave: Domoticz has send a Switch command!, Level: 20, NodeID: 2
(0x02)
21:15:49.950 (Z-Stick) Color Switch (Zipato RGBW)
21:15:49.884 Status: dzVents: Info: Handling events for: "Lux Sensor", value: "21"
21:15:49.884 Status: dzVents: Info: ----- Start internal script: Automatyczna
regulacja oświetlenia: Device: "Lux Sensor (Z-Stick)", Index: 19
21:15:49.886 Status: dzVents: Info: ----- Finished Automatyczna regulacja
oświetlenia
21:15:49.887 Status: EventSystem: Script event triggered:
/home/pi/domoticz/dzVents/ runtime/dzVents.lua
21:15:50.021 (YeeLight) Color Switch (YeeLight RGBW)
21:16:02.446 (Z-Stick) Lux (Lux Sensor)
21:16:02.507 OpenZWave: Domoticz has send a Switch command!, Level: 30, NodeID: 2
(0x02)
21:16:02.612 (Z-Stick) Color Switch (Zipato RGBW)
21:16:02.722 (YeeLight) Color Switch (YeeLight RGBW)
21:16:02.591 Status: dzVents: Info: Handling events for: "Lux Sensor", value: "27"
21:16:02.591 Status: dzVents: Info: ----- Start internal script: Automatyczna
regulacja oświetlenia: Device: "Lux Sensor (Z-Stick)", Index: 19
21:16:02.593 Status: dzVents: Info: ----- Finished Automatyczna regulacja
oświetlenia
21:16:02.594 Status: EventSystem: Script event triggered:
/home/pi/domoticz/dzVents/ runtime/dzVents . lua
21:16:03.139 (Z-Stick) Lux (Lux Sensor)
21:16:05.336 OpenZWave: Domoticz has send a Switch command!, Level: 40, NodeID: 2
(0x02)
21:16:05.340 (Z-Stick) Color Switch (Zipato RGBW)
21:16:02.451 (YeeLight) Color Switch (YeeLight RGBW)
21:16:09.300 Status: dzVents: Info: Handling events for: "Lux Sensor", value: "35"
21:16:09.300 Status: dzVents: Info: ----- Start internal script: Automatyczna
regulacja oświetlenia: Device: "Lux Sensor (Z-Stick)", Index: 19
21:16:09.302 Status: dzVents: Info: ----- Finished Automatyczna regulacja
oświetlenia
21:16:09.303 Status: EventSystem: Script event triggered:
/home/pi/domoticz/dzVents/ runtime/dzVents . lua
21:16:13.743 (Z-Stick) Lux (Lux Sensor)
21:16:18.860 Status: dzVents: Info: Handling events for: "Lux Sensor", value: "47"
21:16:18.860 Status: dzVents: Info: ----- Start internal script: Automatyczna
regulacja oświetlenia: Device: "Lux Sensor (Z-Stick)", Index: 19
21:16:18.861 Status: dzVents: Info: ----- Finished Automatyczna regulacja
oświetlenia

```

Rys. 13. Logi systemu Domoticz - aktywacja wydarzenia "Automatyczna regulacja oświetlenia"

```

20:54:56.281 (Z-Stick) Temp (Temperature Sensor)
20:54:56.284 EventSystem: Event triggered: Zmiana temperatury 2
20:54:56.313 Activating Scene/Group: [Temperatura - optymalna]
20:54:56.314 Activating Scene/Group Device: Zipato RGBW (On)
20:54:56.314 OpenZWave: Domoticz has send a Switch command!, Level: 99, NodeID: 2
(0x02)
20:54:56.315 Red: 044, Green:255, Blue:000, white:000, cwhite:000
20:54:56.330 (Z-Stick) Color Switch (Zipato RGBW)
20:54:56.379 (Z-Stick) Light/Switch (Level)
20:54:56.381 Activating Scene/Group Device: YeeLight RGBW (On)
20:54:56.551 (YeeLight) Color Switch (YeeLight RGBW)
20:54:56.284 Status: 24.5

```

Rys. 14. Logi systemu Domoticz - aktywacja wydarzenia "Zmiana temperatury"

```

06:00:00.004 EventSystem: Event triggered: Automatyczne włączenie oświetlenia_1
06:00:00.152 (YeeLight) Color Switch (YeeLight RGBW)
06:00:00.152 OpenZWave: Domoticz has send a Switch command!, Level: 99, NodeID: 2
(0x02)
06:00:00.170 (Z-Stick) Color Switch (Zipato RGBW)
06:00:00.239 (Z-Stick) Light/Switch (Level)

```

Rys. 15. Logi systemu Domoticz - aktywacja wydarzenia "Automatyczne włączenie oświetlenia"

7. Wnioski

Uzależnienie świecenia żarówek od czujnika ruchu umożliwia pomoc lokatorowi budynku w sytuacji, gdy nie ma możliwości fizycznego naciśnięcia przełącznika, np. gdy lokator niesie coś w rękach. Przypadki obsługiwały sytuacje, w których światło włączało się po wykryciu ruchu i zostało utrzymywane przez 30 sekund, czyli wartość ustawioną w parametryzacji czujnika ruchu, bądź włączane na precyzyjnie określony czas 2 minut. Obie sytuacje pozwalają na zmniejszenie zużycia energii elektrycznej, gdyż światło jest gaszone w krótkim czasie po wykryciu ostatniej aktywności w zasięgu sensora.

Funkcjonalności związane z czujnikiem natężenia światła pozwoliły na odpowiednie oświetlenie pomieszczenia, w czasie, gdy naturalne światło słoneczne nie jest wystarczające. w pierwszym przypadku „Regulacja oświetlenia” proste przełączanie w zależności od zmierzonej wartości może być wykorzystane w sytuacji, gdy na czujnik natężenia światła nie będą miały wpływu podłączone urządzenia oświetlające. Żarówki zmieniające poziom mocy oświetlenia, w pomieszczeniu, w którym znajduje się czujnik, mogłyby zakłócić jego pomiary i powodować włączanie niechcianych skryptów, co finalnie prowadziłyby do pracy urządzeń niezgodnie z przeznaczeniem. Drugi przypadek „Automatyczna regulacja oświetlenia” pokazuje możliwość automatycznej regulacji oświetlenia do zadeklarowanego poziomu w sytuacji, gdy czujnik natężenia światła znajduje się w tym samym pomieszczeniu co sterowane żarówki. Tym razem jednak należało użyć bardziej zaawansowanego języka skryptowego dzVents.

Zmiana koloru emitowanego światła w zależności od temperatury, ma charakter informacyjny i może być przydatna w sytuacjach, gdzie przykładowo, temperatura w pomieszczeniu musi być utrzymana w pewnym zakresie, a nie ma możliwości ustawienia jej konkretnej wartości w systemie ogrzewania.

Przypadki testowe pozwoliły na ukazanie kluczowego aspektu możliwego do zrealizowania dzięki technologii IoT w systemach inteligentnych domów, czyli automatyzacji czynności wykonywanych przez urządzenia. Automatyzacja możliwa była dzięki przechowywaniu utworzonych przez użytkownika skryptów i przetwarzaniu ich w zależności od wartości odebranych danych, pochodzących z urządzeń znajdujących się w systemie.

Wykonany system inteligentnego domu pozwolił potwierdzić wszystkie sformułowane początkowo hipotezy. Dzięki bezproblemowemu podłączeniu wszystkich urządzeń, do systemu Domoticz udostępnionemu na licencji open source, udało się potwierdzić tezę stwierdzającą, że „Dostępne urządzenia IoT można włączyć do systemu inteligentnego domu dzięki ogólnie dostępnemu oprogramowaniu”. Stworzenie oprogramowania w edytorze Blockly oraz przy wykorzystaniu języka skryptowego dzVents potwierdziło, że „Możliwe jest oprogramowanie działania poszczególnych urządzeń IoT działających w systemie inteligentnego domu”. Wyniki przeprowadzonych przypadków testowych, w których żarówki były uzależnione od odczytów czujników ruchu, temperatury i natężenia światła, pozwalają potwierdzić hipotezę, która twierdziła, że „Dzięki stosowanym standardom IoT urządzenia w systemie inteligentnego domu mogą ze sobą współpracować.”

Udowodnienie postawionych hipotez oraz stworzenie działającego systemu inteligentnego domu pozwalają potwierdzić sformułowaną początkowo tezę, która twierdzi, że „Urządzenia IoT dostępne na rynku nadają się do wykorzystania w systemach inteligentnych domów”.

Praca analityczna nie objęła wykorzystania połączeń innych niż poprzez Wi-Fi i protokół Z-Wave. Następne badania

mogłyby objąć pozostałe sposoby komunikacji pomiędzy urządzeniami wykorzystującymi technologię Internet of Things. Ponadto można by zbadać możliwość sterowania systemem spoza sieci wewnętrznej, tak aby możliwa była kontrola oraz odczyt wartości z urządzeń, podczas nieobecności domowników. Szczególną uwagę należałoby objąć wtedy obszar powiadomień alarmowych w systemach inteligentnych domów.

Literatura

- [1] Augusto J., Nugent C., Designing Smart Homes: The Role of Artificial Intelligence, 2006.
- [2] Deshpande R., Deshmukh T., Puranik M., Sagar P., a novel approach to design Smart Home Architecture in Energy Efficient way based on IOT-A Survey, 2018.
- [3] Fouladi B., Ghanoun S., Security Evaluation of the Z-Wave Wireless Protocol, 2013.
- [4] Stojkoska B., Trivodaliev T., a review of Internet of Things for smart home: Challenges and solutions, 2016.
- [5] Talari S., Shafie-khah M., Siano P., Loia V, Tommasetti A., Catalão J., a Review of Smart Cities Based on the Internet of Things Concept, 2017.
- [6] Wan j., Humar I., Zhang D., Industrial IoT Technologies and Applications: International Conference, Industrial IoT 2016, GuangZhou, China, March 25-26, 2016, Revised Selected Papers, 2016.