

WYŻSZA SZKOŁA BANKOWA W POZNANIU
Wydział Finansów i Bankowości

Karol Krasuski

**System automatyki domowej z użyciem układu Raspberry Pi
i platformy Domoticz**

Praca magisterska

**Promotor
prof. nadzw. dr hab.
Grzegorz Pawłowski**

Poznań 2021

Spis treści

Rozdział 1 Wstęp	3
1.1. Szerszy kontekst problematyki pracy	3
1.2. Uzasadnienie podjęcia tematyki pracy	3
1.3. Cel pracy i hipoteza pracy	4
1.4. Zakres	5
1.5. Metody i techniki badawcze zastosowana w pracy	5
1.6. Struktura pracy	6
1.7. Charakterystyka źródeł i literatura	6
Rozdział 2 Charakterystyka rozwiązań opartych na IoT	7
2.1. Czym jest IoT?	7
2.2. IoT spotykane na co dzień	8
2.3. Prognozy rozwoju dziedziny IoT	9
2.4. Standardy komunikacji w IoT	15
2.5. Aspekty bezpieczeństwa w IoT	19
2.6. Przegląd komercyjnych rozwiązań Smart Home	24
2.7. Przegląd wolnego oprogramowania do Smart Home	36
2.8. Podsumowanie	41
Rozdział 3 Opis funkcjonalności Raspberry Pi	43
3.1. Specyfikacja produktu	43
3.2. Wspierane systemy operacyjne	44
3.3. Możliwości rozszerzeń	46
3.4. Istniejące projekty	47
3.5. Pomoc techniczna i społeczność	48
Rozdział 4 Prezentacja środowiska Domoticz	50
4.1. Opis produktu	50
4.2. Wymagania systemowe i sprzętowe	51
4.3. Kompatybilność z urządzeniami IoT	52
4.4. Pomoc techniczna i społeczność	52
Rozdział 5 Praktyczne zastosowanie konceptu rozwiązania Smart Home w Domoticz	53
5.1. Koncepcja rozwiązania i opis funkcjonalności	53
5.2. Przygotowanie sprzętowe	53
5.3. Konfiguracja początkowa	54
5.4. Dostępne pluginy	57
5.5. zigbee2mqtt-plugin	57
5.6. Pluginy do monitorowania urządzeń i sieci:	62
5.7. Dołączanie i konfiguracja kolejnych urządzeń	66
5.8. Prezentacja funkcjonalności	69
5.9. Przykłady Automatyzacji:	74
5.10. Możliwości rozwoju projektu	79
Rozdział 6 Zakończenie	80
Bibliografia	83
Spis tabel	86
Spis rysunków (ilustracji)	86

Rozdział 1

Wstęp

1.1. Szerszy kontekst problematyki pracy

Automatyka domowa to niewątpliwie szybko rozwijająca się gałąź w dziedzinie Internetu Rzeczy. Rosnąca popularność tego typu rozwiązań jest elementem napędzającym kolejne projekty na otwartych licencjach. Na rynku komercyjnym istnieją już dobrze zaprojektowane systemy automatyki domowej np. Fibaro Google Home, jednak nadal wejściowy próg cenowy dla tych specjalistycznych rozwiązań cały czas jest dość wysoki. Mając alternatywę w postaci rozwiązania na bazie darmowego oprogramowania i mikrokomputera Raspberry pi, jesteśmy w stanie stworzyć system automatyki domowej o podobnej funkcjonalności jednocześnie wykorzystując mniejszy budżet. Problem jednak tkwi w tym, że aby opisany wyżej scenariusz się ziścił, musimy wszystkie elementy systemu samodzielnie zmontować i odpowiednio oprogramować. Jest to jednocześnie duża zaleta, ponieważ dzięki temu mamy pełną kontrolę nad naszym systemem i możemy go dowolnie modyfikować. Należy jednak zwrócić uwagę na fakt, że to niewątpliwie wymagające zadanie, do którego realizacji niezbędna jest gruntowna wiedza z zakresu programowania, czujników i ich komunikacji, a także umiejętność przedstawiania treści w odpowiedni sposób. Duża część tych zadań automatycznie potrafi wykonać oprogramowanie Domoticz, które jest tematem głównym tej pracy magisterskiej. Domoticz pozwala na stosunkowo łatwe zarządzanie elementami automatyki domowej, oferując tym samym ogrom możliwości dostrajania każdego z elementów. W pracy postaram się opisać najważniejsze elementy tego oprogramowania oraz konkretne scenariusze użycia systemu w codziennym życiu. Efektem końcowym oprócz merytorycznej wiedzy o technikach automatyzacji domowej będzie gotowy do użytku system, który znajdzie swoje zastosowanie w każdym domu.

1.2. Uzasadnienie podjęcia tematyki pracy

Nowatorskie rozwiązania z zakresu automatyki domowej często spotykają się z zachwytem użytkowników, którzy do tej pory nie korzystali z korzyści, jakie daje Internet

of Things. Prognozy wzrostu oceniają, że w najbliższych latach automatyka domowa będzie intensywnie rozwijana, dlatego jest to jak najbardziej temat wart uwagi. Ludzie często szukają sposobów na ułatwienie sobie codziennych obowiązków, ciesząc się, że nie muszą o pamiętać o wielu rzeczach. Wiedząc, że są sposoby na automatyzację rutynowych i nudnych obowiązków, mogą wieść bezstresowe życie. Oprócz tego, możliwość wpływania na elementy należące do systemu praktycznie z każdego miejsca na ziemi z dostępem do Internetu daje poczucie spokoju. Szczególnie wtedy, gdy nie możemy być fizycznie na miejscu. Jest to niewątpliwie pożądaną cechą w dzisiejszym świecie, w którym króluje pośpiech. Właśnie dzięki argumentom podanymi powyżej Automatyka domowa to duże pole do komercjalizacji, które przyszłościowo może przynieść duży zysk twórcom tych technologii. Powstanie tej pracy ma na celu przedstawić możliwości związane z automatyką domową, jednocześnie nakładając nacisk na to, że ta technologia jest dostępna na wyciągnięcie ręki, bez ponoszenia dużych kosztów a samo przygotowanie środowiska nie musi być żmudną pracą tworzoną od podstaw. Praca dodatkowo może pełnić funkcję szablonu dla projektów z zakresu automatyki domowej szczególnie w przypadku korzystania z oprogramowania Domoticz. Warto też zwrócić uwagę na fakt, iż na ten moment w polskim Internecie ilość technicznych materiałów na temat użytkowania platformy Domoticz jest niewielka. Dodatkowo problemy poruszone w tej pracy mają na celu uporządkowanie wiedzy na temat tej platformy, dzięki czemu praca ta może służyć jako podręcznik użytkownika w języku polskim.

1.3. Cel pracy i hipoteza pracy

Celami pracy są:

- omówienie istniejących na rynku rozwiązań z zakresu automatyki domowej oraz porównanie ich możliwości do tańszych alternatyw,
- zaprezentowanie propozycji budżetowej jako alternatywa dla komercyjnych rozwiązań automatyki domowej np. Fibaro, Google Home, z możliwością zaawansowanej rozbudowy i konfiguracji przy niewielkim nakładzie środków pieniężnych,
- przedstawienie możliwości i zalet mikrokomputera Raspberry Pi,
- praktyczne przedstawienie sposobów na wykorzystanie mikrokomputera Raspberry pi jako HUB dla urządzeń Internetu Rzeczy (lokalna chmura) oraz serwera HTTP z dostępem zdalnym,

- przygotowanie scenariusza użycia systemu automatyki domowej w codziennym życiu (studium przypadku),
- zaproponowanie kierunku rozwijania projektu automatyki domowej o kolejne elementy.

1.4. Zakres

Merytoryczny (rzeczowy)

Praca dotyczy istniejących komercyjnych rozwiązań z dziedziny Smart Home ¹ oraz ich alternatyw opartych na licencjach otwartych. Przedmiotem analizy są aktualne rozwiązania na rynku oraz ocena możliwości stworzenia systemu Inteligentnego budynku o zbliżonej funkcjonalności, bazując na wolnym oprogramowaniu i tańszych zamiennikach sprzętowych. Dodatkowo praca skupia się na zobrazowaniu potencjalnych korzyści płynących z automatyzacji gospodarstw domowych w obecnych czasach przy użyciu urządzeń IoT.

Czasowy

Praca magisterska powstała na podstawie analizy ówczesnie istniejących rozwiązań w zakresie rozwiązań stosowanych w IoT² w okresie 01.09.2020-01.05.2021 oraz ogólnodostępnych w tym czasie rozwiązań sprzętowych w zakresie Inteligentnych budynków.

Terytorialny

Przedmiot pracy dotyczy terenów Unii Europejskiej, głównie ze względu na miejsce wykonania prototypu i dostępność do sieci Internet i poszczególnych części, jednak nie wyklucza się wykorzystywania metod i technik opisanych w tej pracy na terenie całego świata.

1.5. Metody i techniki badawcze zastosowana w pracy

Praca powstała w oparciu o metodę obserwacji z podziałem na część analityczną, której celem jest przedstawienie dostępnych rozwiązań a także jednocześnie zweryfikowanie problemu i potrzeb, a także metodą Case Study polegającą na

¹ Smart Home – Określenie inteligentnego budynku używane w dziedzinie Internetu Rzeczy i przez producentów sprzętu z nim związanych.

² IoT – Internet rzeczy, skrót od ang. Internet of Things

zaplanowaniu i zaprojektowaniu rozwiązania spełniającego zaobserwowane wcześniej wymagania

1.6. Struktura pracy

Praca składa z czterech rozdziałów i podzielona jest na część teoretyczną i praktyczną w następujący sposób

- ❖ Wstęp,
- ❖ Część Teoretyczna
 - Charakterystyka i przegląd rozwiązań opartych na IoT,
 - Omówienie funkcjonalności Raspberry,
 - Prezentacja środowiska Domoticz,
- ❖ Część Praktyczna
 - Praktyczne zastosowanie konceptu rozwiązania Smart Home
- ❖ Zakończenie,
- ❖ Bibliografia,
- ❖ Spis ilustracji i tabel.

1.7. Charakterystyka źródeł i literatura

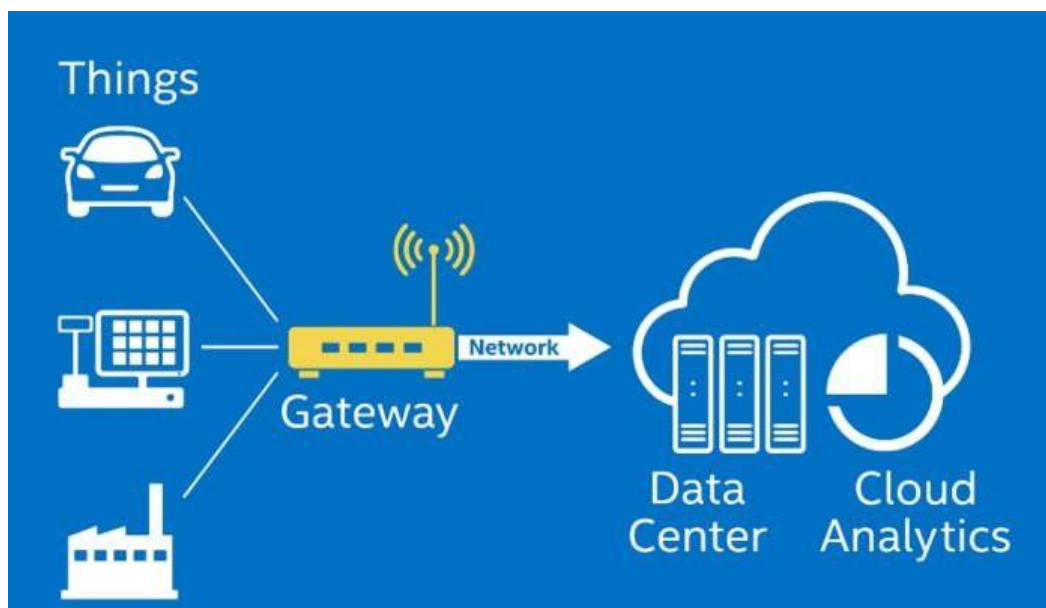
Ze względu na tematykę pracy, która obejmuje zagadnienia stosunkowo młode (dziedzina IoT) zdecydowana większość pozycji ma formę w postaci elektronicznej. Największa część literatury możemy znaleźć w sieci Internet. Nawet pozycje zwarte z tej dziedziny mają nierzadko wydania wyłącznie elektroniczne. Oprócz czasopism naukowych, w pracy znajdują się odnośniki do projektów o zbliżonej tematyce oraz oficjalnych stron poświęconym zagadnieniom związanych z przedmiotem pracy (Domoticz i Raspberry PI). Dodatkowo literaturę uzupełniają fora internetowe i blogi osób pasjonujących się tematyką IoT, dzielących się swoją wiedzą i doświadczeniem w Internecie.

Rozdział 2

Charakterystyka rozwiązań opartych na IoT

2.1. Czym jest IoT?

Internet rzeczy (IoT) - to obszerny zbiór technologii tj. urządzeń i oprogramowania, którego głównym celem jest zbieranie informacji z otaczającego go ekosystemu. Jest to zestaw narzędzi i przypadków użycia, który nie posiada jednoznacznej i jasnej definicji. Mimo to, wszystkie urządzenia systemu IoT powinny być podłączone do systemu łączności tak aby mogły komunikować się między sobą lub przekazywać zebrane dane do jednostek je gromadzących np. bramy (agregatora danych), chmury lokalnej czy chmura publicznej. Udostępnianie danych innym urządzeniom to warunek konieczny do zakwalifikowania urządzenia do systemu IoT, jednak mimo to urządzenia te mogą występować w różnych formach np. jako proste czujniki, elementy maszyn w fabryce lub gotowe urządzenia domowe (żarówki, przełączniki).



Rysunek 2.1-1 Podział elementów systemu IoT. Źródło: <https://www.inferenz.ai/services/artificial-intelligence-in-iot/>

Systemy IoT można podzielić na 3 główne elementy [1]:

Urządzenia (ang. device, things): Najmniejsza jednostka w IoT która ma za zadanie zbierać dane z otoczenia w zależności od jego przeznaczenia i możliwości. Posiada system komunikacji z urządzeniem nadzorczym kumulującym dane lub bezpośrednio do sieci

Internet. Każde urządzenie powinno charakteryzować się zbiorem podstawowych informacji takich jak:

- identyfikator,
- model lub wersja,
- numer seryjny,
- data produkcji

Brama urządzeń IoT (ang. gateway): Urządzenie lub system agregujący dane z urządzeń IoT która pełni rolę pośrednika w komunikacji pomiędzy końcowymi urządzeniami a siecią Internetową lub konkretną aplikacją czy usługą, z którą to brama potrafi się komunikować.

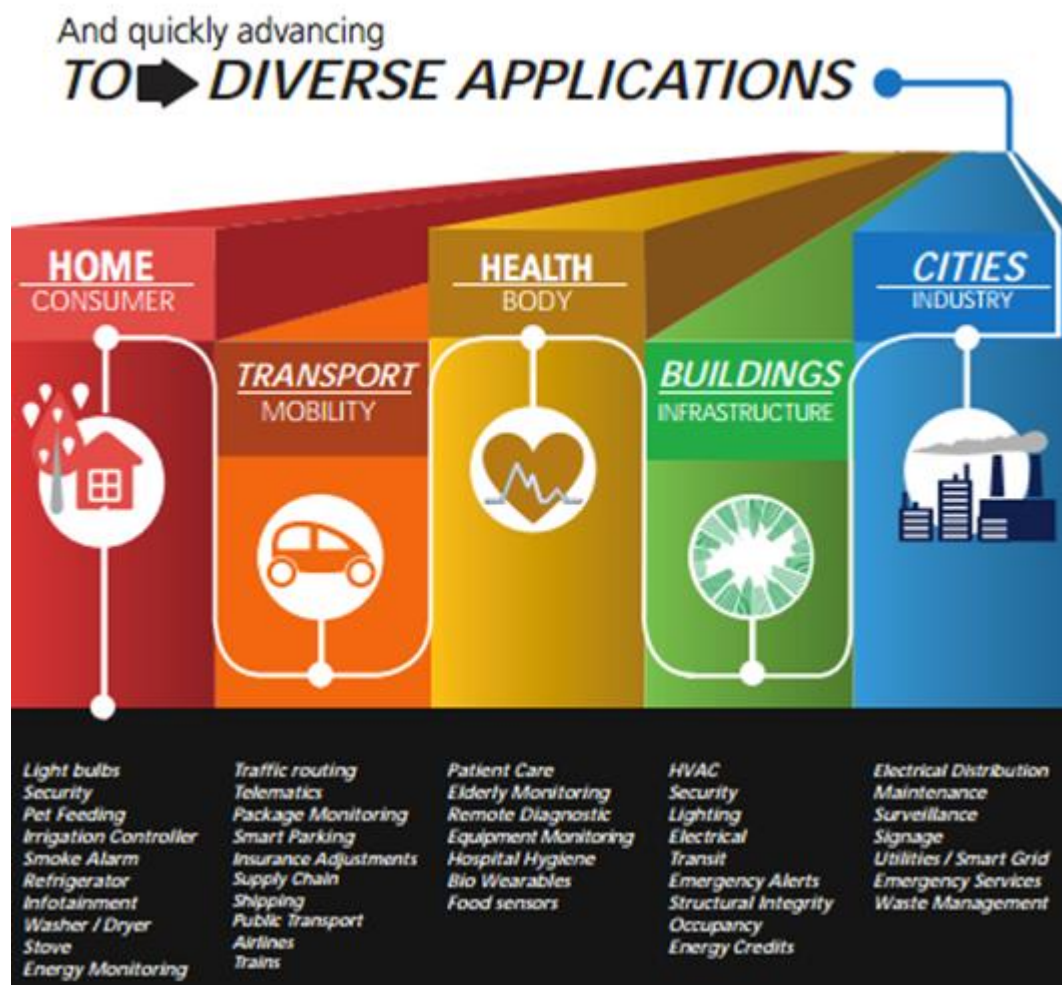
Chmura obliczeniowa (ang. cloud): Zbiór zasobów (komputerów) udostępniający usługi bezpośrednio w sieci Internet potrafiąca gromadzić i przetwarzać dane w zależności od sposobu jej zaprogramowania.

Dane zebrane przez urządzenia często służą do przewidywania i automatyzacji następstw wynikających ze zmieniających się danych np. w przypadku pogody (spadek temperatury) pomogą wybrać odpowiedni moment na włączenie/wyłączenie systemu ogrzewania, tak aby zachować stałą temperaturę. Oprócz tego dane zebrane z otoczenia pozwalają tworzyć modele matematyczne które wymagają zasilenia odpowiednią dawką danych do analizy po to by mogły poprawnie działać i przewidywać następstwa wynikające z wahań poszczególnych wartości.

2.2. IoT spotykane na co dzień

Zastosowanie IoT jest z powodzeniem wykorzystywane w wielu gałęziach przemysłu czy działalnościach oferujących usługi komercyjne (np. transport, przemysł), jednak oprócz tego wdrożenie IoT choć w częściowym stopniu jest możliwe w praktycznie każdej dziedzinie życia człowieka.

W codziennym życiu szczególnie często IoT spotykane jest w np. sporcie [2], medycynie czy lokalach mieszkalnych tzw. SmartHome. W każdej z branż IoT przynosi wymierne korzyści i wpływa na ułatwienie codziennych zadań powtarzających się na cyklicznie. Poniższa grafika dobrze ilustruje zakres dziedzin mogących być polem do rozwoju dla dziedziny Internetu Rzeczy.



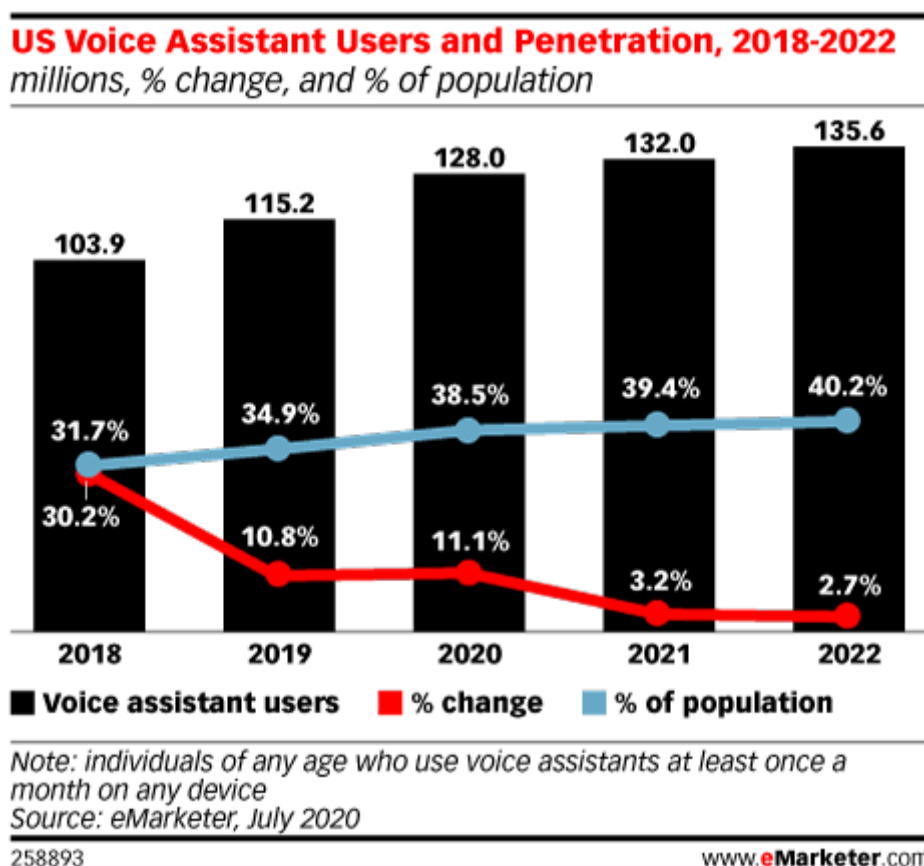
Rysunek 2.2-1 Podział elementów systemu IoT. Źródło: <https://www.inferenz.ai/services/artificial-intelligence-in-iot/>

2.3. Prognozy rozwoju dziedziny IoT

Prognozy dla potrzeb domowych

W dzisiejszych czasach nie jest zaskakujące, gdy ktoś mówi do telefonu nie prowadząc wcale rozmowy z drugim człowiekiem. Ludzie korzystają z asystentów głosowych pytając ich o pogodę i dzięki wejściu z nimi w konwersację oszczędzają czas na poszukiwaniu informacji samodzielnie. Dodatkowo nikogo też nie powinno dziwić poproszenie asystenta głosowego o obniżenie temperatury w samochodzie lub domu, czy zapytanie o własne plany na dziś. Aktualnie ludzie stają się coraz bardziej wygodni tym samym coraz więcej roli powierzają urządzeniom inteligentnym. Firma Gartner szacuje, że do końca 2018 30% interakcji z technologią miało odbywać się poprzez rozmowy z inteligentnymi

maszynami³. Jak podaje Forbes, w 2018 roku jeden na sześciu dorosłych w Stanach Zjednoczonych posiada inteligentny głośnik lub urządzenie z asystentem głosowym⁴ – Można uznać, że trend cały czas można obserwować i aktualnie wynik byłby lepszy z korzyścią dla technologii IoT co potwierdza poniższa grafika, gdzie według tych danych w 2020 roku 128mln osób w Stanach Zjednoczonych (ok. 38,5% populacji) korzysta z asystenta głosowego przynajmniej raz w miesiącu:



Rysunek 2.3-1 Przewidywania liczby użytkowników asystentów głosowych wg. eMarketer. Źródło: www.emarketer.com

Można tutaj dodatkowo rozważyć podział na kilka kategorii urządzeń z aplikacją do zarządzania głosowego (asystentów) są to min:

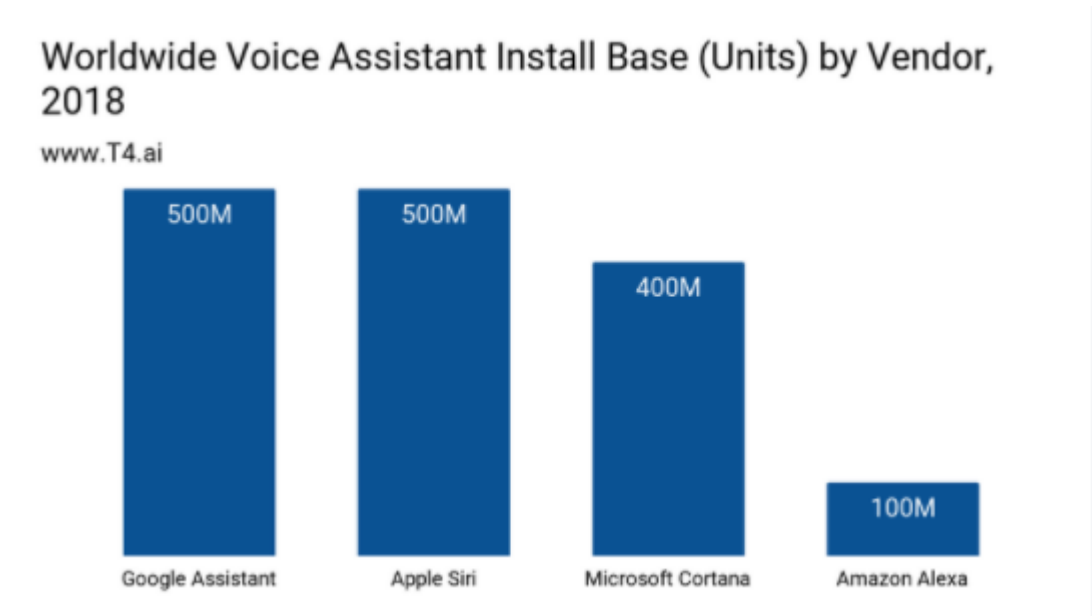
- aplikacje preinstalowane przez producentów smartfonów: Asystent Google (Android), Apple Siri (iOS), Samsung Bixby (Android),
- systemy operacyjne: Apple Siri MacOS, Microsoft Cortana (Windows),

³ Analiza Gartner nt. interakcji użytkowników z urządzeń konsumenckich wykorzystującymi asystentów głosowych: <https://www.gartner.com/en/documents/3021226/market-trends-voice-as-a-ui-on-consumer-devices-what-do->

⁴ Artykuł Forbes o przyszłości związanej z interaktywną technologią: <https://www.forbes.com/sites/danielnewman/2018/08/22/voice-interface-technology-the-future-of-business/?sh=16fdc96a316a>

- urządzenia dedykowane (np. głośniki inteligentne) – Amazon Alexa, Apple Siri, Google Asystent.

Pokrycie rynkowe przez poszczególnych producentów wygląda różnie w zależności od przeprowadzonego badania, natomiast pewnym jest, że ilość urządzeń z możliwością uruchomienia asystenta głosowego przekroczyła już w 2018 Miliard urządzeń:



Rysunek 2.3-2 Udział urządzeń z asystentem głosowym w rynku w 2018 roku. Źródło: T4.ai

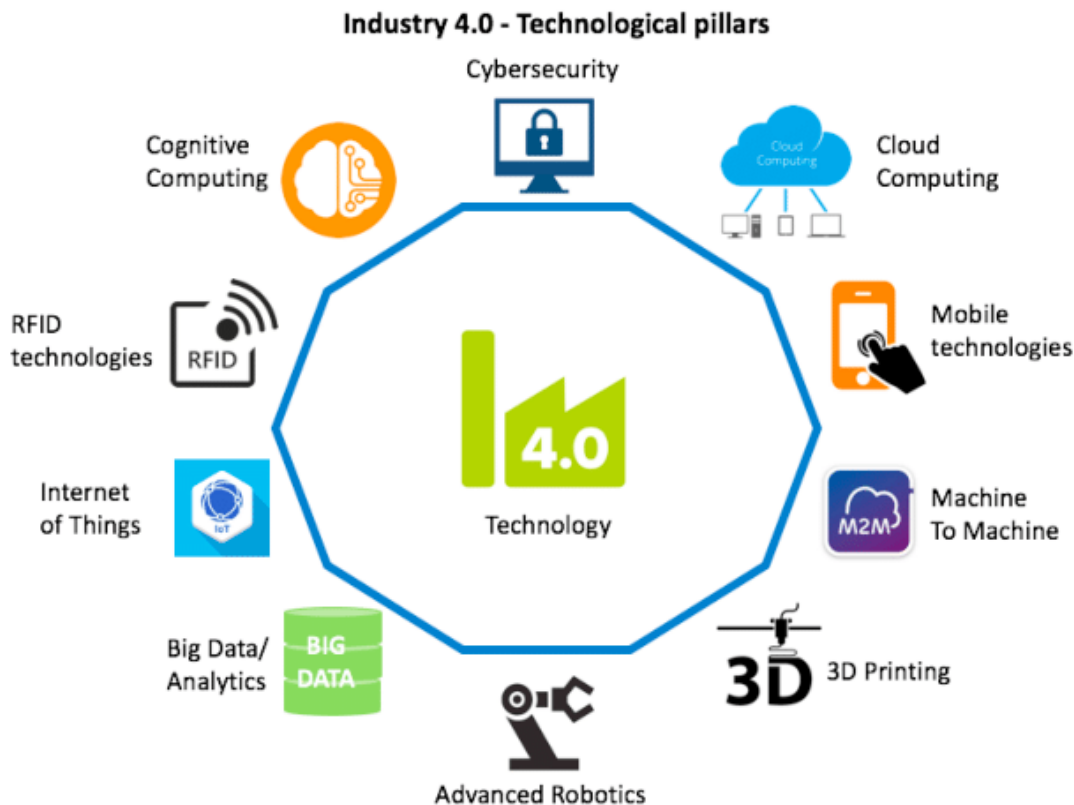
Jednym z argumentów idących za rosnącą popularnością rozwiązań IoT jest znaczące wsparcie korporacji takich jak Amazon, Microsoft, Apple czy Google. Są to giganci, którzy swoje wersje asystentów głosowych udostępniają za darmo razem ze sprzętem lub oprogramowaniem dzięki temu dają prawie że darmową przepustkę do świata IoT [3]. Często pierwsza konfrontacja z inteligentnymi urządzeniami zaczyna się od asystenta głosowego i skutkuje większym zainteresowaniem nowinkami technologicznymi i zarażeniem się pasją do urządzeń IoT. Asystenci głosowi popularnych firm często umożliwiają połączenie do swojej chmury czujników kompatybilnych ze swoim API⁵ i pozwalają na podstawowe sterowanie tymi urządzeniami co również sprzyja popularności SmartHome.

Prognozy dla potrzeb biznesowych i przemysłowych

IoT to wciąż dość nowy rynek - ale taki, który ma spory potencjał. Podlega szybkiemu rozwojowi i obejmuje niezwykle różnorodne technologie przydatne w

⁵ application programming interface, API – zbiór reguł ściśle opisujący, w jaki sposób programy lub podprogramy komunikują się ze sobą.

procesach biznesowych: od czujników, przez sztuczną inteligencję i uczenie maszynowe po robotykę i drony wykorzystywane w przemyśle. Dziedzina IoT jest jednym z filarów stojąca za czwartą rewolucją przemysłową. Robi to napędzając wzrost produktywności, zmieniając modele biznesowe i redefiniując sposób, w jaki organizacje realizują swoje usługi i wychodzą naprzeciw do swoich klientów i partnerów



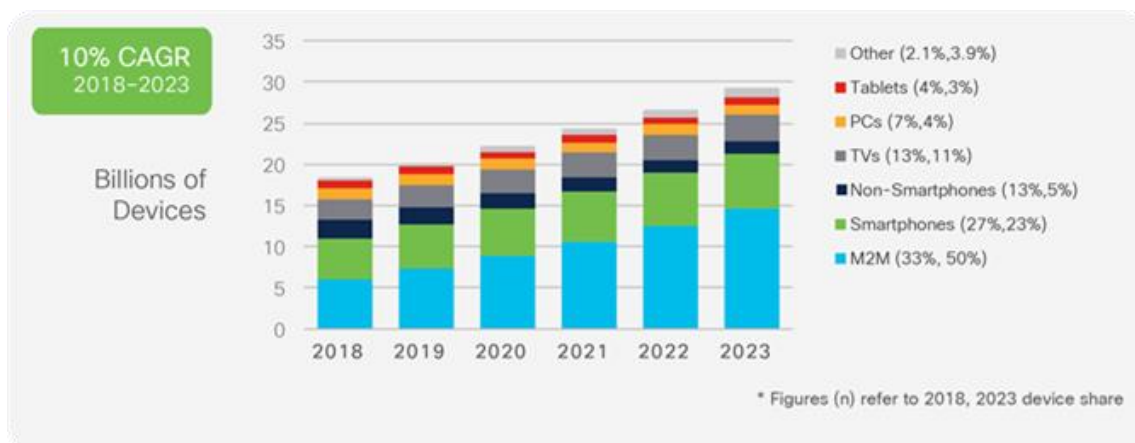
Rysunek 2.3-3 Rewolucja przemysłowa 4.0 Źródło: aie-internship.com

Warto zauważyć, że w obecnych czasach coraz więcej dużych korporacji interesuje się rynkiem IoT dostrzegając korzyści za tym płynące i wdraża kolejne produkty i usługi oparte na IoT i jego elementami.

Osobną dziedziną obok IoT jest IIoT (Industrial Internet of Things) która pokrywa aspekty związane z kompleksową obsługą inteligentnych fabryk. Magazynów i centr dystrybucyjnych. IIoT jest szczególnie ważną gałęzią IoT, bo to tutaj zwykle inwestowane są największe pieniądze, po to, aby z końcem inwestycji czerpać wymierne korzyści dzięki zautomatyzowanej infrastrukturę i świadomym podejmowaniu strategicznych decyzji opartych o dane zbierane w procesach IoT. Kluczową rolę w IIoT jest niezawodność, prędkość komunikacji i dostęp do Internetu urządzeń agregujących w związku z tym korporacje związane z tworzeniem urządzeń sieciowych takie jak Cisco, Huawei a także te związane z telekomunikacją Ericson, Nokia, Qualcomm prześcigają się w coraz to nowych

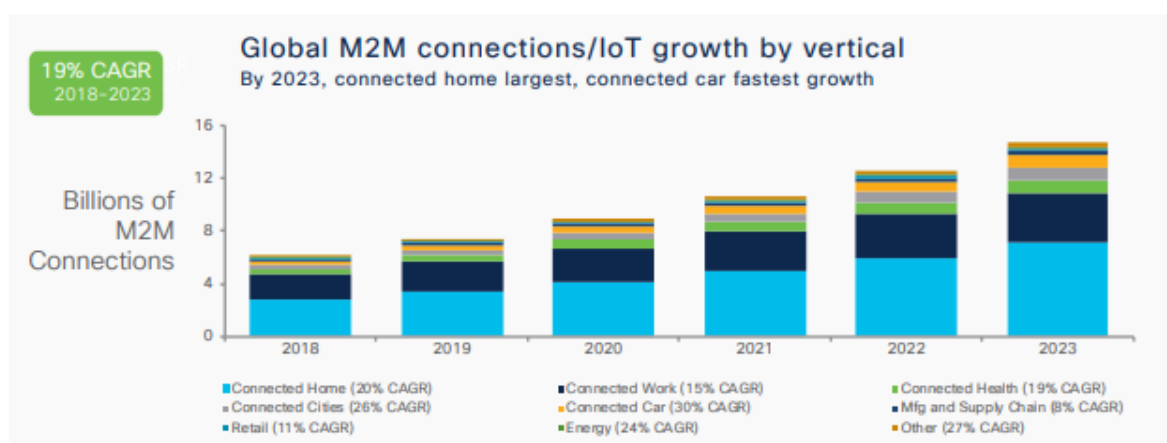
rozwiązaniach komunikacyjnych wdrażając przy tym nowe standardy komunikacji bezprzewodowej takie jak np. 5G[4], tak bardzo potrzebne w środowiskach o dużym zagęszczeniu czujników i mikrokontrolerów agregujących dane. Jak podaje Cisco [5], liczba urządzeń podłączonych do sieci Internet do 2023 roku będzie ponad trzykrotnie większa niż liczba ludzi na całym świecie. Cisco szacuje się, że około 2/3 ludności ma jakikolwiek dostęp do Internetu a raport obejmuje stacjonarne łącza szerokopasmowe, Wi-Fi i sieci mobilne (3G, 4G, 5G). Według prognozy do 2023 r. przypadąć będzie 3,6 urządzenia podłączonego do sieci na każdego człowieka na Ziemi. W porównaniu do 2018 roku było to 2,4 urządzeń na osobę. Firma Cisco przewiduje, że do 2023 roku będzie 29,3 miliarda urządzeń podłączonych do sieci, w porównaniu z 18,4 miliarda w 2018 roku. Na wykresie widać, że największą część w 2018 roku (33%) w 2023 (50%) ruchu w sieci ma odbywać się w trybie M2M⁶. Prognoza ta daje nadzieje na rozwój w dziedzinie IoT, ponieważ właśnie tam najczęściej komunikacja odbywa się bez interakcji człowieka tj. urządzenia już raz skonfigurowane potrafią komunikować się między sobą i wymieniać dane wtedy, kiedy jest potrzeba ich aktualizacji. Kolejnym rzeczą, którą warto tutaj zauważyć jest fakt, że komputery stacjonarne w 2018 roku stanowiły 7% urządzeń a w 2023 przewiduje się, że będzie to już tylko 4%. Dzieje się tak że ludzie aktualnie stawiają na mobilność i nie chcą być podłączeni do Internetu wyłącznie w miejscu swojego zamieszkania a wręcz przeciwnie, cenią sobie mobilność. Dlatego też na wykresie widnieje trend, że aktualnie około 40% to mobilne urządzenia przenośne Laptop, smartfon, tablet. Wszystkie te czynniki sprzyjają wcześniej wspomnianej wysokowydajnej sieci bezprzewodowej 5G, ponieważ jest to jedna z niewielu technologii, która można zapewnić dostęp do Internetu bez względu na lokalizację geograficzną na tak dużą skalę.

⁶ M2M – (ang. Machine to machine) – Skrócona nazwa zjawiska komunikowania się urządzeń między sobą np. w dziedzinie IoT bez udziału pośredniego człowieka.



Rysunek 2.3-4 Prognoza podłączonych urządzeń do Internetu: Coroczny raport cisco nt. Internetu 2020 Źródło: Ciscopress.com

Biorąc pod lupę same urządzenia działające w trybie M2M firma Cisco przewiduje że największe wzrosty CAGR ⁷ ma osiągnąć branża motoryzacyjna która to wykorzystuje coraz więcej elektroniki przy produkcji pojazdów (30% wzrostu) inteligentne miasta (26% wzrostu) które również za pomocą czujników będą w stanie za pomocą badać otaczającą rzeczywistość, ale także rynek automatyki (20% wzrostu) domowej/biurowej tj. wszelkiego rodzaju czujniki, kamery i systemy bezpieczeństwa W Każdej z tych dziedzin IoT może odegrać znaczną rolę a na pewno jej elementy będą obecne w modułach ponieważ zbieranie danych i ich analiza to kluczowy element mierzenia się z potrzebami rynku i innowacji.



Rysunek 2.3-5 Wzrost podłączonych urządzeń komunikujących się w trybie M2M - Podział na kategorie Źródło: Ciscopress.com

⁷ CAGR – (ang. compound annual growth rate) Skumulowany roczny wskaźnik wzrostu – wskaźnik wykorzystywany do obliczeń średniego rocznego wzrostu pewnej wielkości w badanym okresie

2.4. Standardy komunikacji w IoT

Standardy komunikacji na odległość można podzielić na kilka typów [6] w zależności od ich:

- zasięgu,
- poboru mocy,
- przeznaczenia,
- warstwy w modelu OSI ⁸.

Sieci o małej mocy i krótkim zasięgu

Sieci o małej mocy i krótkim zasięgu doskonale nadają się do domów, biur i innych małych środowisk. Zwykle potrzebują tylko małych baterii i zazwyczaj elementy wykorzystujące te protokoły są tanie w zakupie. W ramach tej kategorii można wyróżnić standardy min.:

Bluetooth (IEEE 802.15.1)

Standard do szybkiego przesyłania danych, Bluetooth potrafi nie tylko przesyłać sygnały głosowe, ale też dane na odległość przeważnie do 10 metrów. Działający w zakresie fal radiowych: od 2.402 GHz do 2.480 GHz

NFC (rozszerzenie ISO/IEC 14443)

Zestaw protokołów komunikacyjnych do komunikacji między dwoma urządzeniami elektronicznymi na odległość 4 cm (1/2 cala) lub mniejszą. NFC oferuje wolne połączenie z prostą konfiguracją, której można użyć do inicjowania bardziej wydajnych połączeń bezprzewodowych. Pasywne nośniki danych np. NFC tagi uaktywniają się dopiero po wykryciu odbiornika które je zasili.

Wi-Fi (IEEE 802.11)

Zbiór standardów które są powszechnie używane w lokalnych sieci komputerowych i w celu bezprzewodowego dostępu do Internetu. Charakteryzuje się niskim kosztem obsługi jednak ma ograniczony zasięg i wymaga całodobowego zasilania energią elektryczną.

⁸ Model OSI (ang. *ISO Open Systems Interconnection Reference Model*) – Model przedstawiający jako zbiór zasad komunikacji urządzeń sieciowych. Składa się z siedmiu warstw, w którym to każda warstwa ma połączenie z poprzednią

Z-Wave

Protokół komunikacji bezprzewodowej wykorzystywane przede wszystkim do automatyki domowej. Z-Wave to sieć typu mesh która wykorzystuje niskoprężowe fale radiowych, aby komunikować się pomiędzy urządzeniami. W Europie pracuje w częstotliwości 868,42 MHz

Zigbee (IEEE 802.15.4)

Zestawu protokołów które z założenia mają być prostsze i tańsze niż inne bezprzewodowe sieci np. takie jak Bluetooth lub bardziej ogólne sieci bezprzewodowe, takie jak Wi-Fi. Podobnie jak Z-Wave jest to standard głównie spotykany w automatyce domowej. Standard został bardziej szczegółowo opisany w rozdziale trzecim.

Sieci rozległe o małej mocy

Sieci LPWAN umożliwiają komunikację na co najmniej 500 metrów oraz mają niewielkie zapotrzebowanie na energię elektryczną. Typowe przykłady sieci LPWAN⁹ to:

4G LTE IoT

To sieć bezprzewodowa o dużej szerokości pasma i małych opóźnieniach aktualnie najbardziej popularna metoda komunikacji urządzeń bezprzewodowych. Oferuje prędkość do 300Mb/s i pojemność na poziomie miliona urządzeń na 500km² dlatego dla wielu rozwiązań są dobrym wyborem dla scenariuszy IoT, które wymagają informacji lub aktualizacji w czasie rzeczywistym.

5G IoT

Następca standardu 4G który wciąż znajduje się w trakcie rozwoju, oczekuje się, że umożliwi dalszą innowację w IoT, zapewniając znacznie szybsze prędkości pobierania i oferując większą pojemność (do miliona urządzeń na 1km²) oraz opóźnienia na poziomie ~1ms [8]

Cat-0

Sieć o niewielkiej przepustowości i niskim zakresie częstotliwości opierająca się na standardzie LTE która stanowi fundament Cat-M, technologię, która zastąpi 2G. Stworzona z myślą o IoT i komunikacji machine to machine

⁹ LPWAN (ang. low-power wide-area network) – rodzaj bezprzewodowej sieci rozległej o małej mocy

Cat-1

Standard komórkowy IoT który ostatecznie zastąpi 3G. Sieci Cat-1 są łatwe w konfiguracji i stanowią doskonałe rozwiązanie dla aplikacji komunikatorów głosowych lub urządzeń multimedialnych zapewniając prędkość do 10Mbit/s, ale jednocześnie zapewniając znacznie mniejsze opóźnienia i redundancje połączenia w postaci dwóch anten [9]

LoRaWAN

Standard składający się z bramek sieciowych, serwerów sieciowych i serwerów aplikacji oraz urządzeń końcowych zasilanych bateryjnie potrafiących w stanie uśpienia pracować na jednej baterii kilka lat. Maksymalne zasięg urządzeń tzn. odległość pomiędzy urządzeniami a stacjami bazowymi może wynosić 10–15 km. Prędkość przesyłania w tym standardzie waha się w granicach między 0,3 kb/s a 37,5 kb/s

NB-IoT

kategoria standardów komunikacji bezprzewodowej o niskim poborze energii, w skład której wchodzi min. LTE Cat-M1 Cat-M2 Maksymalny transfer danych wynosi do 1 Mb/s co pozwala np. na obsługę aplikacji głosowych Voice Over LTE (VoLTE). Nadaje się również do zastosowań mobilnych i przenośnych (np. telematyka, zarządzanie flotą).

Sigfox

Nazwa standardu pochodzącego od Francuskiego operatora dostawcy sieci i urządzeń IoT Sigfox korzysta z technologii radiowej Ultra Narrow Band (UNB) i częstotliwości 868 MHz w Europie. UNB pracuje z relatywnie wąskimi sygnałami radiowymi. Głównymi atutami tego standardu jest możliwość przesyłania wielu sygnałów jednocześnie na duży dystans oraz niewielkie zużycie energii ze względu na komunikację, w której większość czynności wykonuje stacja bazowa a czujniki odpytywane są na żądanie [7]

Podział w tej kategorii można również wyróżnić na kategorie takie jak np. „prędkość czy przykłady użycia

Tabela 2.4-1: Podział LPWAN – Źródło: Opracowanie własne.

Maksymalna prędkość przesyłu	< 5 Mbit/s	> 5 Mbit/s, < 500 kbit/s	> 500 kbit/s
Przykładowe standardy	4G,5G	3G,2G, LTE-M	LoRa, SIGFOX, NB-IOT

czy pasmo wymaga licencjonowania	Tak	Tak	Tak/Nie
Typowe przykłady użycia	smartfony, urządzenia multimedialne, kamery, samochody,	czujniki automatyki, inteligentne bramki sms, czujniki aktywności fizycznej/życiowej, inteligentne budynki,	elementy "smart city": inteligentne parkingi, inteligentne oświetlenia uliczne, mierniki zużycia energii/gazu/wody

Podział komunikacji ze względu na warstwy modelu OSI

Warstwa aplikacji

Warstwa aplikacji pełni rolę interfejsu pomiędzy użytkownikiem a urządzeniem w ramach danego protokołu IoT. Przykładowe protokoły to:

- zaawansowany protokół kolejowania wiadomości (AMQP),
- ograniczony protokół aplikacji (CoAP),
- usługa dystrybucji danych (DDS),
- transport telemetryczny kolejki wiadomości (MQTT).

Warstwa transportowa

W warstwa transportowa umożliwia i zabezpiecza przesyłanie danych podczas ich przemieszczania się między warstwami. Dominującymi protokołami w tej warstwie są:

- protokół kontroli transmisji (TCP),
- protokół data gramów użytkownika (UDP).

Warstwa sieci

Warstwa sieciowa protokołu IoT pomaga poszczególnym urządzeniom komunikować się z urządzeniami spoza własnej podsieci we współpracy z routerem. Odpowiadają za to standardy min.

- IP,
- 6LoWPAN.

Warstwa łącza danych

Warstwa danych jest częścią protokołu IoT, który przesyła dane w ramach architektury systemu lub tej samej podsięci, identyfikując i korygując błędy znalezione w warstwie fizycznej. Charakterystyczne protokoły w tej warstwie:

- IEEE 802.15.4 (Zigbee, 6LoWPAN),
- LPWAN.

Warstwa fizyczna

Warstwa fizyczna to kanał komunikacyjny między urządzeniami w określonym środowisku. Może być to komunikacja przewodowa lub bezprzewodowa, jednak najczęściej używane standardy w tej warstwie to:

- Bluetooth Low Energy (BLE),
- Ethernet,
- Long Term Evolution (LTE),
- komunikacja bliskiego zasięgu (NFC),
- komunikacja w sieci energetycznej (PLC),
- identyfikacja radiowa (RFID),
- Wi-Fi / 802.11,
- Z-Wave,
- Zigbee.

2.5. Aspekty bezpieczeństwa w IoT

Zagrożenia i dobre praktyki

Ze względu na szerokie zastosowanie elementów IoT w życiu prywatnym i służbowym należy zwrócić uwagę na aspekty bezpieczeństwa związane z udostępnianiem danych przez urządzenia IoT, ale także ich dostępność w sieci Internet.

Warto zwrócić uwagę na fakt, że urządzenia IoT często wchodzi w interakcję z innymi maszynami bez naszej interwencji. Często urządzenia te nieustannie komunikują się z Internetem, np. lodówka wysyłająca aktualizację żywności w środku lub pojazd przekazujący wiadomości do mechanika informujące o poziomie oleju, ale to tylko wierzchołek góry lodowej. Obecnie smartfony są naszpikowane czujnikami, do których ma dostęp mnoga liczba aplikacji i to za naszym przyzwoleniem.

IoT jest wspaniały pod wieloma względami, niestety, technologia jeszcze nie jest na tyle dojrzała i przez to nie jest całkowicie bezpieczna. Całe środowisko IoT, od producentów po

użytkowników, wciąż ma do pokonania wiele wyzwań związanych z bezpieczeństwem IoT [10], min. takich jak:

- normy produkcyjne – brak zgodności ze strony producentów IoT,
- zarządzanie aktualizacjami i sposobem ich wdrażania,
- ograniczone lub kiepskie wsparcie produktów,
- bezpieczeństwo fizyczne urządzeń,
- niewystarczająca wiedza i świadomość użytkowników,
- niedopracowane oprogramowanie,
- niezabezpieczone protokoły przesyłania danych,
- niezabezpieczony dostęp do pamięci masowej,
- podatność na ataki botnet,
- szpiegostwo i podsłuchiwanie,
- złośliwe urządzenia IoT malware & ransomware.

Oczywistym jest, że domowi użytkownicy podejmują inne środki ostrożności niż firmy czerpiące zysk z urządzeń IoT, jednak kilka fundamentalnych zasad powinno zostać zachowane, jeżeli chodzi o bezpieczeństwo w dziedzinie IT

Do podstawowych obowiązków w zakresie ochrony urządzeń IoT przed niepowołanym dostępem jest: [11]

- **Cykliczna aktualizacja oprogramowania układowego na wszystkich urządzeniach IoT.** Jest to szczególnie ważna czynność, która nie tylko przynosi wymierne korzyści w postaci nowych funkcjonalności, ale też krytyczne łatki bezpieczeństwa, które uniemożliwiają wykorzystywanie podatności we wcześniejszych wersjach.
- **Zmiana domyślnego hasła ustawionego przez producenta.** Wiele urządzeń wystawionych do sieci Internet mimo najnowszej aktualizacji jest podatne na ataki bruteforce / słownikowe które próbują siłą złamać hasło do panelu administracyjnego urządzenia. Pozostawienie domyślnego hasła tylko ułatwia im to zadanie. Warto tutaj też pamiętać o odpowiedniej długości hasła (najlepiej ponad 12 znaków) Dodatkowym zabezpieczeniem, jeżeli to możliwe to zmiana loginu administratora
- **Weryfikacja konieczności dostępu do Internetu** – Warto sprawdzić czy każde urządzenie musi być dostępne z poziomu sieci Internet. Często lepszą praktyką będzie, jeżeli to koncentrator lub agregator urządzeń IoT będzie udostępniony publicznie.

Najczęściej urządzenia / aplikacje specjalizujące się w tej funkcji mają zaimplementowane silne mechanizmy bezpieczeństwa.

- **Separacja urządzeń IoT udostępnionych do sieci Internet**- W momencie, kiedy jest potrzeba udostępnienia urządzeń w Sieci Internet, dobrą praktyką jest przygotowanie osobnej podsieci do tego celu najlepiej takie usługi udostępniać w strefie DMZ¹⁰ spoza której dostęp do sieci wewnętrznej będzie ograniczony lub niemożliwe
- **Jeżeli to możliwe, włączenie logowania alertów bezpieczeństwa** – Taki zabieg powinien odciążyć administratora od ciągłego sprawdzania statusu urządzenia, jednak pomimo istnienia takich mechanizmów dobrą praktyką jest weryfikacja statusu krytycznych urządzeń co jakiś czas ręcznie
- **Cyklicznie sprawdzanie stanu urządzeń IoT** – Dobra praktyka, która może zapobiec utracie danych, konfiguracji, ale zmniejszenie wektorów ataku. Zauważenie nieprawidłowości w działaniu urządzenia powinno zwrócić uwagę administratora i wymusić odpowiednią reakcję
- **Monitorowanie ujawnianych luk w zabezpieczeniach i źródeł informacji o zagrożeniach**

Jeżeli okaże się, że urządzenie lub oprogramowanie na nim zainstalowane ma aktualnie znaną krytyczną podatność np. pozwalającą na zdalne wykonywanie poleceń należy rozważyć odcięcie dostępu do sieci Internet i/lub wdrożenie dodatkowe środków bezpieczeństwa pozwoli to zminimalizować szanse na utratę lub kradzież danych. Często skompromitowane urządzenie może posiadać autoryzację do zarządzania innymi urządzeniami, dlatego ważne jest tutaj zasada” least privilege”¹¹[11]. Aktualna lista podatności aplikacji i systemów operacyjnych można sprawdzić pod adresem: <https://www.exploit-db.com> [12]

Prywatność w IoT

Zauważalnym problemem w dzisiejszym świecie technologii jest niewiedza użytkowników na temat jakie dane są od nich pobierane. Można by przytoczyć tu przykład smartfonów i ich systemów operacyjnych, gdzie do prawidłowego uruchomienia aplikacji wymagany jest szereg uprawnień niekoniecznie zawsze potrzebnych do

¹⁰ DMZ (ang. Demilitarized zone) – Wydzielony obszar sieci komputerowej nienależący do sieci wewnętrznej ukierunkowany pod udostępnienie usług / urządzeń do sieci Internet.

¹¹ Zasada najmniejszego uprzywilejowania (ang. least privilege rule) – Zasada wymagająca, aby na danym poziomie abstrakcji każdy element systemu informatycznego miał dostęp tylko do tych informacji i zasobów, które są niezbędne do spełnienia wyznaczonego mu celu lub zadania

jej poprawnego działania. Aplikacje często za nieumyślnym przyzwoleniem użytkownika zabierają dane diagnostyczne z urządzenia właściciela i jego otoczenia.

Obojętność i brak powszechnej świadomości na temat zasad prywatności w przypadku zbierania danych telemetrycznych jest powodem, dla którego twórcy aplikacji coraz chętniej sięgają do tego typu narzędzi i chętnie te dane wykorzystują do celów komercyjnych. Na szczęście istnieją organy które częściowo kontrolują ten proceder i tworzą rozporządzenia i zasady, na których powinno opierać się zbieranie danych od użytkowników, szczególnie jeżeli mowa tu o danych osobowych. Jednym z ważniejszych dokumentów wdrożonych na terenie Unii Europejskiej przez Komisję Europejską jest: Rozporządzenie Parlamentu Europejskiego i Rady (UE) 2016/679 z dnia 27 kwietnia 2016 r. w sprawie ochrony osób fizycznych w związku z przetwarzaniem danych osobowych i w sprawie swobodnego przepływu takich danych oraz uchylenia dyrektywy 95/46/WE (ogólne rozporządzenie o ochronie danych) Dz. U. UE L 119 4 maja 2016 [2]. Dokument ten przede wszystkim przenosi obowiązek informacyjny na temat zbieranych danych na dostawcę usługi i wymaga od niego poprawne ich przetwarzanie do celów wyłącznie niezbędnych przy realizacji usługi. Polski odpowiednik takiego dokumentu posiadający moc prawną to: Ustawa z dnia 10 maja 2018 r. o ochronie danych osobowych Dz.U. z 2018 r. poz. 1000 [1] Dokument ten również pokrywa aspekty zbierania i przetwarzania danych osobowych, ale też zawiera informacje o ewentualną odpowiedzialność karną w związku nieprzestrzegania wytycznych zawartych w tym dokumencie.

W przypadku IoT osoby postronne mogą bardzo często nie zdawać sobie sprawy, że dane są od nich pobierane, bo obowiązek informacyjny jest tutaj ciężki do wyegzekwowania. Dzieje się tak ponieważ często ma tu miejsce połączenie kilku zestawów informacji w wyniku, którego może powstać chroniona prawnie dana osobowa o której posiadaniu np. przez usługodawcę powinien zostać poinformowany jej właściciel. Dlatego właśnie w przypadku prowadzenia działalności komercyjnej opartej na usługach i urządzeniach IoT warto najpierw zastanowić się nad obowiązkiem informacyjnym, ponieważ w konsekwencji nieprzestrzegania ustalonych przez prawo zasad można spotkać się konsekwencjami finansowo/karnymi

Zasady tworzenia bezpiecznych rozwiązań IoT

Biorąc pod uwagę wcześniej wymienione wytyczne można wyciągnąć wnioski które pomogą tworzyć bezpieczne rozwiązania IoT. Jednocześnie warto wypunktować

kilka charakterystycznych elementów na które warto zwrócić uwagę zawierających się w cyklu życia produktu związanego z IoT:

Analiza platformy urządzenia – Słaba lub źle przemyślana konfiguracja platformy może prowadzić do podatności np. takich jak eskalacja uprawnień.

System operacyjny platformy urządzenia oraz jego właściwości, konfiguracje i funkcje bezpieczeństwa należy zweryfikować pod kątem wymagań bezpieczeństwa informacji określonych w podstawowym zakresie. Dodatkowo należy przeprowadzić weryfikację, aby upewnić się, że wszystkie interfejsy testowe zostały usunięte ze sprzętu.

Weryfikacja sposobu komunikacji – Biorąc na przykład ruch sieciowy (przewodowy lub bezprzewodowy), ruch ten powinien być analizowany pod kątem wszelkich możliwych do przechwycenia, niezaszyfrowanych lub modyfikowalnych danych. Zalecanym podejściem jest wykorzystywanie wyłącznie szyfrowanych protokołów komunikacyjnych w którym powinien zachowany zostać kompromis między wydajnością a bezpieczeństwem. Aby spełnić wymagania dotyczące wydajności, można zastosować lekkie algorytmy szyfrowania. Warto w tym celu śledzić strony internetowe które publikują listę nowych bezpiecznych algorytmów ukierunkowanych na niskie zużycie zasobów np. NIST pod adresem:

<https://csrc.nist.gov/Projects/lightweight-cryptography> [13]

Weryfikacja wymagań bezpieczeństwa funkcjonalnego - Wymagania dotyczące bezpieczeństwa funkcjonalnego wysokiego poziomu powinny zostać sprawdzone skrupulatnie pod kątem różnych przypadków użycia, w tym zostać poddane testom typu subversion (atak na integralność danych) lub fuzzing (atak na aplikację losowymi danymi np. zbioru popularnych ciągów)

Przegląd uprawnień i podatności na wstrzyknięcie danych - Wszystkie granice zaufania na ścieżce sygnału i dostępu do danych powinny zostać zweryfikowana i poddane tzw. iniekcji błędu przy użyciu celowo złośliwych przypadków testowych. Granice zaufania można zweryfikować za pomocą penetracyjnych testów manualnych. Testy te powinny być wykonywane w ciągu całego cyklu życia oprogramowania

Przegląd kodu związanego z zabezpieczeniami - Przeglądy kodu we wczesnej fazie rozwoju aplikacji wymuszają na wczesnym etapie do zaostrożenia polityk bezpieczeństwa. Obszary wrażliwe kodu np. te mające wpływ na bezpieczeństwo, takie jak proces rozruchu, autoryzacja i moduły szyfrowania, powinny przejść przez przegląd kodu przy każdej aktualizacji. Koszt naprawy usterki zabezpieczeń jest znacznie zmniejszony, gdy luka w zabezpieczeniach zostanie odkryta podczas cyklu rozwoju.

Kompleksowy test penetracyjny - kompleksowe testy penetracyjne powinny być przeprowadzane na ścieżce komunikacyjnej, aby zidentyfikować wszelkie luki w interfejsie aplikacji webowej, interfejsie mobilnym i interfejsie chmury tworzonej aplikacji IoT. Testy penetracyjne pozwolą określić poziom bezpieczeństwa rozwiązania IoT dla każdego z jego komponentów. Jeżeli jest to projekt komercyjny warto skorzystać z usług firm zajmujących się audytami bezpieczeństwa aplikacji.

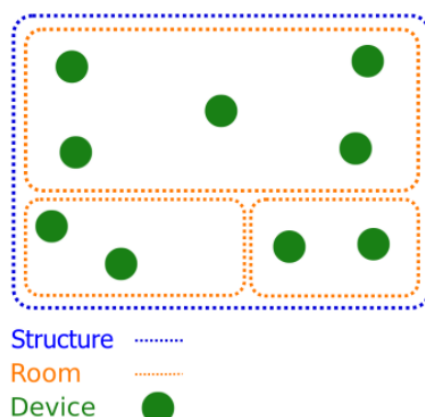
2.6. Przegląd komercyjnych rozwiązań Smart Home

Google Home / Nest Hub

Platforma Google Smart Home umożliwia użytkownikom kontrolowanie urządzeń za pośrednictwem aplikacji Google Home i mechanizmu AI Asystenta Google który jest składową i integratorem usług google. Jak szacuje producent aplikacje dostępne są na ponad miliardzie urządzeń [14], takich jak inteligentne głośniki, telefony, samochody, telewizory, słuchawki, zegarki i wszelkie inne urządzenia kompatybilne z API tej aplikacji. Producent udostępnia swoją platformę za darmo w chmurze, jednak oferuje też dedykowane urządzenia do kontrolowania swoim środowiskiem IoT w ramach tej samej platformy.

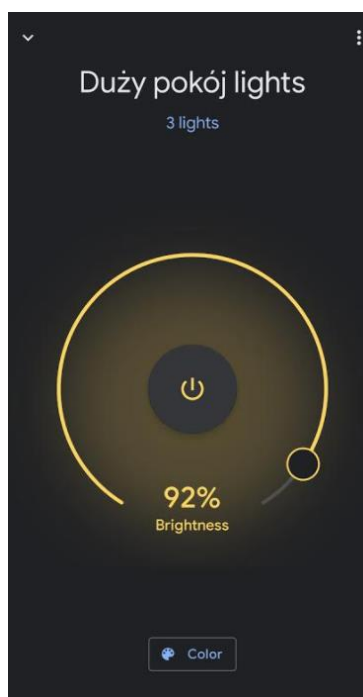
Działanie platformy Google opierają się na Home Graph tj. bazie danych, która przechowuje i dostarcza kontekstowe dane o domu i jego urządzeniach. Home Graph to w zasadzie logiczna mapa budynku. Baza danych Home Graph przechowuje informacje o:

- strukturach (na przykład dom lub biuro – podstawowa jednostka organizacyjna),
- pokojach (na przykład sypialnia lub salon – czyli grupy urządzeń),
- urządzeniach (na przykład głośniki i żarówka – najmniejsza, elementarna jednostka).



*Rysunek 2.6-1 schemat architektury Google Home. Źródło:
<https://developers.google.com/assistant/smarthome/concepts/homegraph>*

Te informacje są dostępne dla Asystenta Google w celu wykonywania żądań użytkowników w oparciu o odpowiedni kontekst np. „zapal światło w jadalni”. Dane stanu urządzeń, takie jak zapalona żarówka, nie są przechowywane przez dłuższy czas - są efemeryczne i są używane tylko na wykresie początkowym. Oprócz podstawowego sterowania stanem urządzeń, jeżeli producent urządzenia kompatybilnego z Google udostępnia API na zasadach wyznaczonych przez producenta możliwe jest dodatkowe sterowanie urządzeniami np. w przypadku oświetlenia jest to możliwość sterowania mocą i kolorem.



Rysunek 2.6-2 Możliwości kontrolowania urządzeń zewnętrznych w Google Home. Źródło Opracowanie własne.

Oprócz sterowania głosowego i aplikację z poziomu smartfonu Google udostępnia dedykowane urządzenie „Nest Hub” które min. Może pełnić funkcje kontrolera dla środowiska IoT, ale też służy jako klienta dla asystenta Google i pozostałych usług, jednocześnie mogąc również pełnić funkcję odtwarzacza multimedialnego.



Rysunek 2.6-3 Google Nest Hub Źródło: https://store.google.com/us/product/nest_hub_2nd

Tabela 2.6-1 Podstawowe parametry Google NestHub Źródło: Opracowanie własne

Wymiary (dł., szer.)	Ekran	Procesor	Zużycie energii	Łączność
17.78cmx11.9cm	7" (1024 x 600)	Quad-core 64-bit 1.9 GHz ARM CPU	15W	<ul style="list-style-type: none">• Wi-Fi 802.11b/g/n/ac (2.4 GHz/5 GHz)• Bluetooth® 5.0

Zakup urządzenia nie jest obligatoryjny, ponieważ wszystkie dane są trzymane w chmurze u producenta. NestHub to po prostu opcja dodatkowego dedykowanego kontrolera do środowiska Google Home i pozostałych usług tej firmy. Aplikacja Google Home pozwala na tworzenie „rutyn” gdzie możemy zdefiniować reakcje urządzeń (lub odpowiedzi asystenta) np. po wypowiedzeniu zadanej komendy głosowej. Rutyny mogą być wywołane również o określonej godzinie. Oprócz natywnych rozwiązań automatyki dobrym integratorem do zadań może być aplikacja IFTT: <https://play.google.com/store/apps/details?id=com.ifttt.ifttt&hl=pl&gl=US> która posiada spory zasób predefiniowanych „rutyn” tworzonych przez użytkowników Aplikacja ta w podłączaniu z usługami Google może sterować nie tylko samą aplikacją, ale również pozostałymi usługami Google.

Pomimo bardzo dużej kompatybilności z urządzeniami firm trzecich sama aplikacja jest zamknięta i mimo że producent udostępnia narzędzia developerskie i integracje ze swoimi usługami to nie pozwala na swobodne rozszerzanie funkcjonalności i wybiórczo wybiera dane które możemy odczytywać z poziomu jej interfejsu. Producent stawia w tym wypadku na łatwość konfiguracji użycia i dużą kompatybilność jednak dzieje się to kosztem okrojonej funkcjonalności i prywatności, gdyż wszystkie dane powiązane z tą usługą są trzymane w chmurze. Będzie to dobry wybór dla osób rozpoczynających przygodę z IoT, ponieważ nie wymaga specjalistycznej wiedzy technicznej a jedynie dostępu do smartfonu i urządzeniami IoT kompatybilnymi z tymi rozwiązaniem

Amazon Alexa Echo

Amazon Alexa jest bezpośrednią konkurencją Google Home i Asystenta Google i oferuje usługi w podobny sposób co jej konkurent i również oferuje aplikacje na smartfony. Amazon także oferuje Asystenta głosowego i kompatybilność z dużą ilością urządzeń firm trzecich. Podobnie jak w przypadku Google home dostępne jest dedykowane urządzenie Alexa echo natomiast jest tutaj dużo większy wybór, jeżeli chodzi o dobór urządzenia. Urządzenia dzielą się na kilka segmentów: inteligentne głośniki mogące pełnić także funkcje intercomu, zestawy multimedialne z wyświetlaczem oraz modele

przystosowane do wideokonferencji. Alexa jest najbardziej popularna w Ameryce jednak jest popularność w innych regionach wzrasta. Niestety nie ma jeszcze dostępnego asystenta w wersji polskojęzycznej. Amazon w z początku zdominował rynek inteligentnych głośników, jednocześnie mając ich najwięcej w swojej ofercie. Poniżej znajduje się zestawienie najpopularniejszych modeli urządzeń dostępnych jako kontrolery smart:

Tabela 2.6-2 Podział urządzeń amazon Echo Źródło: Opracowanie własne.

	Amazon Echo Studio	Amazon Echo (fourth gen)	Amazon Echo Flex	Amazon Echo Show 10	Amazon Echo Spot
Wymiary:	206 x 175 x 175mm, 3.5kg	144 x 144 x 133mm, 970g	72 x 67 x 66mm, 166g	251 x 230 x 172mm, 2560g	32 x 84 x 84mm, 163g
Łączność	Dwuzakresowe Wi-Fi 802.11 a/b/g/n/ac (2.4 and 5GHz), Bluetooth (A2DP), 3.5mm audio in/out, Zigbee	Dwuzakresowe Wi-Fi 802.11 a/b/g/n (2.4 and 5GHz), Bluetooth (A2DP), 3.5mm audio in/out, Zigbee, Sidewalk	Dwuzakresowe Wi-Fi 802.11 a/b/g/n/ac (2.4 and 5GHz), Bluetooth (A2DP), 3.5mm audio in/out	Dwuzakresowe Wi-Fi 802.11 a/b/g/n (2.4 and 5GHz), Bluetooth (A2DP), 3.5mm audio in/out, Zigbee, Sidewalk	Dwuzakresowe Wi-Fi 802.11 a/b/g/n (2.4 and 5GHz), Bluetooth (A2DP), 3.5mm audio in/out
Odtwarzanie i nagrywanie dźwięku	tak	tak	tak	tak	Tak
wbudowana kamera	nie	nie	nie	tak, 13Mpix	Nie
Wbudowany wyświetlacz	nie	nie	nie	ekran dotykowy 10.1"	ekran dotykowy 2.5"

Warto zauważyć, że niektóre z modeli posiadają wsparcie protokołu Zigbee dzięki czemu można sparować urządzenia takie jak wtyczki, przekaźniki czy zamki działające w tym protokole [15]. W przypadku Google home nie można takiego urządzenia podpiąć bezpośrednio i należy do tego celu wykorzystać pośrednika w postaci huba ZigBee.

Podobnie jak poprzedni dostawca Amazon wszystkie dane związane z naszymi urządzeniami trzyma w chmurze, lecz sama aplikacja i dostęp do niej jest za darmo.

Wink

Platforma Wink w zamyśle została zaprojektowana tak, aby była prosta, elastyczna i łatwa w użyciu. Producent oferuje wsparcie z zaufanymi markami, oferując sporo liczbę urządzeń kompatybilnych z tym rozwiązaniem.

Wink Hub/Wink Hub 2 to produkty obligatoryjne do pracy w środowisku Wink, ponieważ są to urządzenia, które pozwala łączyć różne technologie (protokoły) występujące w produktach inteligentnego domu w jednym miejscu. Nowszy model oferuje dodatkowo łączność urządzeń za pomocą protokołu bluetooth, posiada port Ethernet i Wifi 802.11ac oraz charakteryzuje się łatwiejszą obsługą i zautomatyzowanym procesem wykrywania i dołączania urządzeń. Do sparowania urządzeń z kontrolerem należy wykorzystać aplikację Wink. Aplikacja jest jedynym sposobem na komunikację z kontrolerem, ponieważ producent nie udostępnia interfejsu webowego. Kontroler Wink można sparować z asystentem głosowym Alexa lub google Assistant i w ten sposób można czerpać korzyści z obu platform. Do zadań automatyki można używać tzw. „robotów” czyli zadań automatyzacji dzięki prostym algorytmom można zautomatyzować elementy znajdujące się pod kontrolą WinkHub Podobnie jak w przypadków poprzedników wspomagaczem do zadań automatycznych może okazać się aplikacja IFTT która posiada predefiniowane scenariusze automatyzacji gotowe do wykorzystania. Wink do komunikacji może używać sieci lokalnej i WAN.

Od maja 2020 producent wprowadził subskrypcje 5\$ miesięcznie za utrzymanie usług świadczących dla swoich urządzeń, a było to podyktowane problemami finansowymi firmy. Bez aktywnej subskrypcji może okazać się, że sterowanie niektórymi urządzeniami okaże się niemożliwe. Producent zapewnia, że w wypadku braku subskrypcji i dostępu do sieci WAN możliwymi urządzeniami do sterowania z pewnością będą elementy świetlne i zamki elektroniczne. Firma twierdzi, że bardzo poważnie traktuje prywatność i bezpieczeństwo danych użytkowników i regularnie sprawdza aplikację pod kątem bezpieczeństwa, aby upewnić się, że przekraczają one standardy branżowe. W ramach zabezpieczeń możliwe jest przypinanie certyfikatów, szyfrowanie, uwierzytelnianie dwuskładnikowe dla wszystkich administratorów systemu oraz regularne audyty bezpieczeństwa. Według oświadczenia dane zbierane przez firmę ograniczają się głównie

do danych diagnostycznych [16]. Firma prowadzi swój sklep z urządzeniami w pełni wspieranymi przez Wink dostępny pod adresem www.wink.com/products. Podsumowując jest to produkt dla użytkowników posiadających już wiedzę i urządzenia IoT których nie można bezpośrednio dołączyć do Amazon Alexa lub Google home np. takich pracujących z protokołem Z-Wave, ale jednocześnie zależy im na łatwości konfiguracji urządzeń.

Samsung SmartThings



Samsung SmartThings to system dla osób z podstawowym pojęciem o aspektach smart home ceniących sobie łatwość użytkowania i automatyzacji zadań. Oczywiście podobnie jak w przypadku tego typu rozwiązań należy najpierw dodać urządzenia do pomieszczeń i zdefiniować zadania automatyzacji, aby system SmartHome działał poprawnie. Warto zwrócić uwagę na fakt, że w platformie Samsunga znajduje się parę rzeczy na które nie pozwalają wcześniej opisani konkurenci w branży IoT, ponieważ w przeciwieństwie do przeciwników dysponuje szerokim zapleczem technologicznym i doświadczeniu w produkcji elektroniki domowej i użytkowej. Samsung SmartThings obsługuje sporą ilość urządzeń „smart” a nawet sam tworzy kompatybilne rozwiązania w swoich urządzeniach np. Telewizory obsługujące Samsung smart tv, Inteligence lodówki odkurzacze czy klimatyzacje. SmartThings to aplikacja, która stanowi kontroler i sterownik dla podłączonych urządzeń IoT. Nie jest to doskonały system, ale w ramach tej aplikacji możliwe jest tworzenie pewnego rodzaju reguł i algorytmów do sterowania urządzeniami inteligentnymi z poziomu jednej aplikacji. Lista kompatybilnych urządzeń znajduje się pod adresem: <https://www.samsung.com/pl/apps/smartthings/>

Często dedykowane urządzenia oznaczone logiem Samsung mogą być sterowane w dużo większym stopniu niż w przypadku konkurencji np. funkcja pilota w przypadku Samsung smart TV. Dodatkowo podobnie jak inni dostawcy rozwiązań smart home Samsung udostępnia urządzenie pełniące rolę bramy IoT. W tym przypadku mowa tu o urządzeniu: SmartThings Hub. Opcjonalnie Samsung udostępnia wersję wbudowaną w lodówkę „Family Hub fridge” która jest wbudowanym kontrolerem w lodówce z funkcją możliwości instalacji dodatkowych aplikacji, ale także urządzeniem multimedialnym potrafiącym odtwarzać multimedia oraz oferując autorskiego asystenta głosowego „Bixby” listę najważniejszych funkcjonalności można obejrzeć pod adresem:

<https://www.samsung.com/us/explore/family-hub-refrigerator/apps/>

Kontroler SmartThings Hub aktualnie dostępny jest w wersji 3 „V3” i dostępny jest w wersji w przeciwieństwie do poprzedników bez zapasowej baterii oznacza to, że potrzebuje

stały dostęp do prądu. Oprócz tego w kolejnych generacjach urządzenia postawiono większy nacisk na przetwarzanie chmurowe i część urządzeń nie może być już obsługiwana lokalnie np. Kamery D-link [17].

	SmartThings V2 (Old)	SmartThings V3 (New)
Size	4.9 x 4.2 x 1.3 inches	5 x 5 x 1.2 inches
Protocols	Zigbee, Z-Wave, Bluetooth	Zigbee, Z-Wave, Bluetooth
Connectivity	Ethernet	Ethernet or Wireless
Battery Backup		
Under the Hood	1GHz ARM Cortex-A9, 512MB DDR3 RAM, and 4GB FLASH	528 MHz ARM Cortex-A7, 256MB DDR RAM, and 4GB FLASH
Ports	2 USB, 1 Ethernet	1 USB, 1 Ethernet
Camera Support	Local and Cloud	Cloud Only

Rysunek 2.6-4 Porównanie wersji SmartThings hub. Źródło: <https://homealarmreport.com/smart-home/smarthings/>

Hub oferuje standardowe protokoły komunikacji tj. Wifi, Ethernet, Zigbee i Z-Wave, jednak, żeby urządzenia współpracowały z Hubem muszą znaleźć się na liście kompatybilności.

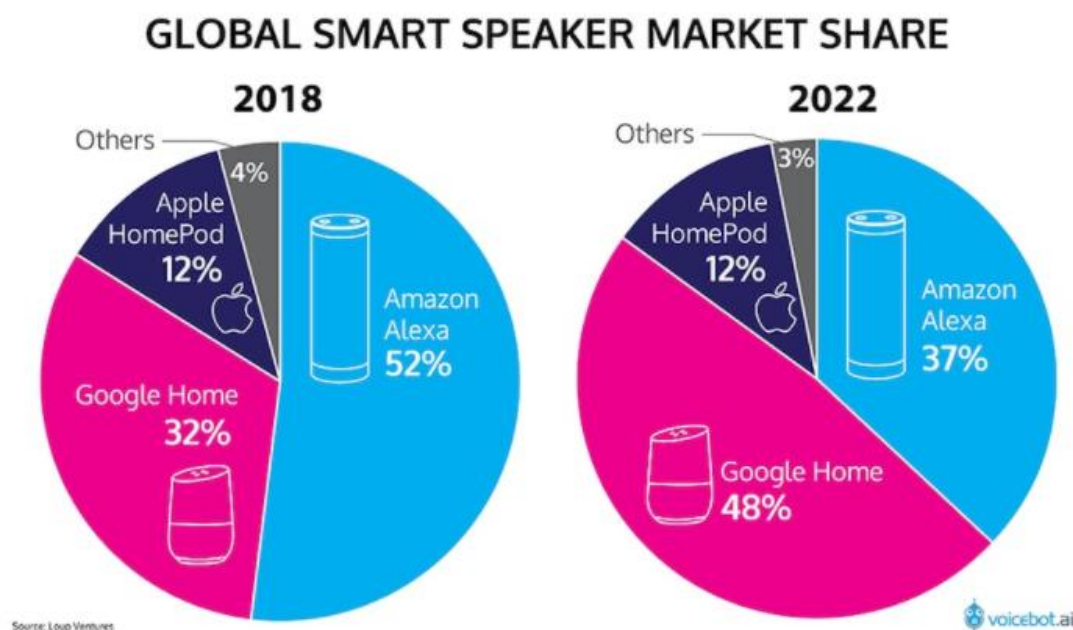
Platforma Samsunga może się okazać dobrym wyborem w szczególności, gdy posiadamy już urządzenia „smart” tego producenta, zależy nam na łatwości konfiguracji i nie potrzebujemy dodawać urządzeń spoza listy kompatybilności.

Apple HomeKit

Firma Apple to czwarty z dużych graczy na rynku smart home oferujący własnego asystenta głosowego oraz szereg urządzeń kompatybilnych w dziedzinie IoT. Będzie to wybór dla fanów marki, ceniących sobie ekosystem Apple i dla ludzi korzystających z usług tej firmy, ponieważ wybór urządzeń jest stosunkowo niewielki. Można go znaleźć na stronie producenta:

<https://www.apple.com/pl/shop/accessories/all/homekit>

Producent oferuje typowe urządzenia smart home takie jak oświetlenie, kamery, systemy nawadniania, inteligentne gniazda elektryczne, przełączniki i czujniki. Jako kontroler urządzeń dodanych w ramach aplikacji do zarządzania urządzeniami IoT może służyć HomePod, czyli głośnik firmy Apple. Komunikacja odbywa się w tym wypadku głosowo przez asystenta głosowego Siri. Niestety nie ma możliwości podłączenia do tego urządzenia urządzeń innych niż tych z systemem iOS co sprawia, że cała platforma staje się jeszcze bardziej hermetyczna. Oprócz dedykowanych usług Apple platforma HomeKit nie oferuje niczego innowacyjnego w dziedzinie smart home w porównaniu do konkurencji, jednak mimo to ze względu na wierność marce nadal rozwiązania tej firmy znajdują się w czołówce, jeżeli chodzi o rozwiązania z pogranicza dziedziny IoT i AI szczególnie jeżeli rozpatrujemy kraj, gdzie jest na tę technologię największy popyt, czyli USA. Według firmy Loup Ventures i jej przewidywaniom Apple kontroluje ok 12% rynku i ten trend się nie zmienia pomimo mniejszej funkcjonalności i otwartości niż u konkurencji

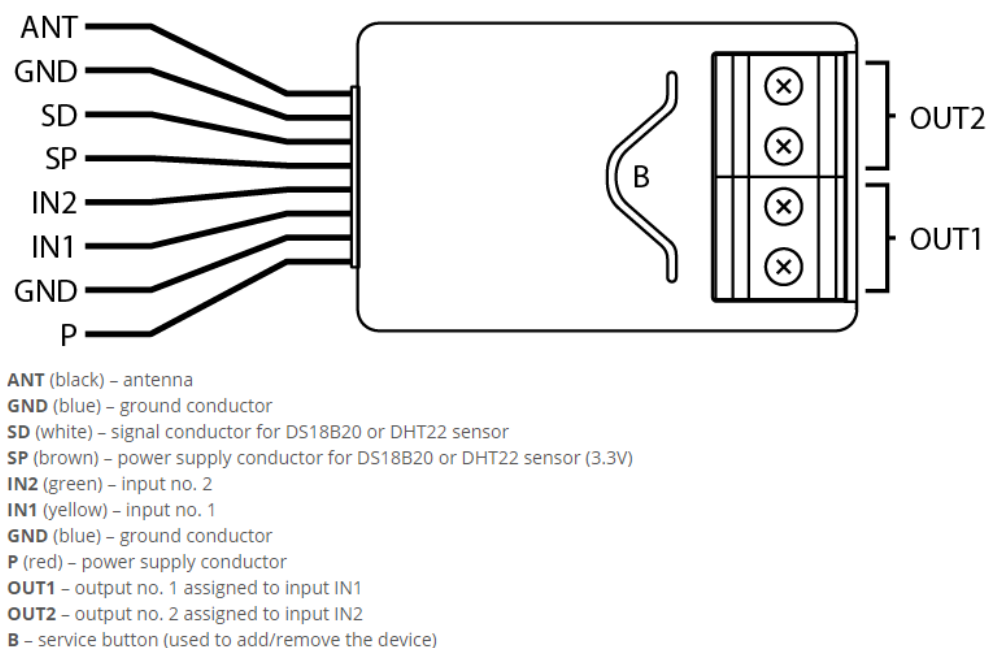


Rysunek 2.6-5 Prognoza udziałów w rynku USA z podziałem na inteligentne głośniki wg. Loup Ventures.
 Źródło: <https://voicebot.ai/>

Fibaro Home Center

Fibaro to polski akcent w dziedzinie smart home. Jest to firma zaczynając swoją działalność jako start-up z okolic Poznania, gdzie powstała pierwsza fabryka w 2011 roku. Obecnie działa na ponad 110 rynkach na całym świecie a do 2016 roku wyprodukowano 665 tysięcy urządzeń [18]. Ostatecznie 100% udziałów zostało sprzedane Włoskiej firmie [19] Nice jednak mogło to dać perspektywy na rozwój, ponieważ firma ta już wcześniej zajmowała się rynkiem smart home a zakup firmy Fibaro napędził postęp technologiczny w branży IoT. W ofercie Fibaro można znaleźć klasyczne urządzenia IoT takie jak czujniki: temperatury, wilgotności, CO, Inteligence oświetlenie, termostaty itp. Na uwagę jednak zasługują kilka autorskich pomysłów takich jak kontroler „Swipe” który pozwala na sterowanie urządzeniami lub wywoływanie komend za pomocą gestów bez dotykania panelu. Urządzenie posiada zestaw dedykowany gestów, ale również pozwala na tworzenie swoich sekwencji (maksymalnie 6). Urządzenie jest kompatybilne z kontrolerami Z-Wave i Z-Wave+ chociaż producent zaleca korzystanie z jednej z centrali dostępnych w ramach produktów Fibaro i aplikacji Home Center. Kolejnym ciekawym produktem jest inteligentny implant „FGBS-222” pozwala on zwiększyć funkcjonalność czujników przewodowych oraz innych urządzeń poprzez dodanie do nich komunikacji sieciowej Z-Wave. Oprócz tego daje możliwość podłączenia czujników binarnych, czujników analogowych, czujników temperatury lub czujników wilgotności i temperatury, aby

przekazać ich wartości do kontrolera Z-Wave. Smart Implant może również sterować urządzeniami poprzez otwieranie i zamykanie styków wyjściowych niezależnie od wejść. Sprawia to, że urządzenie może stanowić most komunikacyjny pomiędzy analogowymi urządzeniami a nowoczesnymi kontrolerami IoT lub aplikacji zarządzającej. Urządzenia posiada dwa wejścia i dwa wyjścia oraz dedykowane porty do sterowania czujnikami.



Rysunek 2.6-6 Rysunek 2-9-6 Schemat połączeń w implancie Fibaro: Źródło:
<https://manuals.fibaro.com/smart-implant/>

Wszystkimi urządzeniami można sterować za pomocą jednej z centrali Fibaro z serii Home center które są jednocześnie kontrolerami Z-Wave innych protokołów bez przewodowych. Najnowsza wersja kontrolera Home Center 3 charakteryzująca się maksymalnym deklarowanym przez producenta zasięgiem do 150m, lepszym procesorem i większą ilością pamięci operacyjnej niż jej poprzednicy, bezprzewodowe standardy komunikacyjne które są obsługiwane przez to urządzenie i zakres ich częstotliwości widnieją poniżej

Komunikacja radiowa			
Protokół	Częstotliwość radiowa		Maks. moc nadawania
Z-Wave (500 series)	EU	868.0-868.6 MHz 869.7-870.0 MHz	+14dBm
	AU, NZ	915.0-928.0 MHz	+5dBm
	CL	902.0-928.0 MHz	+5dBm
	BR	915.0-928.0 MHz	-2dBm
	THA	920.9-923.1 MHz	+7dBm
Wi-Fi (802.11 b/g/n/a/ac)	2.4GHz	2400.0-2483.5 MHz	+20dBm
	5GHz	5150.0-5250.0 MHz	+14dBm
433Mhz	EU, THA	433.05-434.04 Mhz	+10dBm
	AU, NZ, CL, BR	433.05-434.04 Mhz	+14dBm
868Mhz	EU	868.0-869.65 MHz	+10dBm
ZigBee		2400.0-2483.5 MHz	+10dBm
Bluetooth Low Energy		2400.0-2483.5 MHz	+4dBm

Rysunek 2.6-7 Specyfikacja połączeń bezprzewodowych w Fibaro Home center 3 źródło: <https://manuals.fibaro.com/>

Urządzenia fibaro są kompatybilne z wszystkimi z czterech wyżej wymienionymi asystentami głosowymi, dzięki czemu rekompensują brak własnego asystenta głosowego. Produkty fibaro mogą być szczególnie interesujące dla entuzjastów zaawansowanych rozwiązań smart, nie posiadających dużej wiedzy w konfiguracji tego typu urządzeń ponieważ firma ta posiada rozwiniętą siatkę dystrybucyjną w polsce i oferuje montaż i konfigurację urządzeń smart w domu przyszłego klienta przez certyfikowanych instalatorów fibaro.

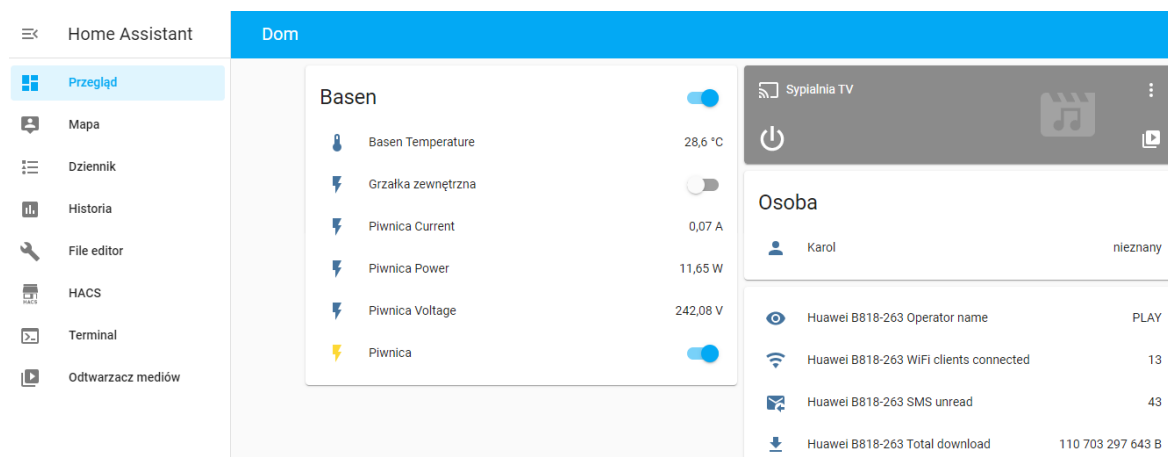
2.7. Przegląd wolnego oprogramowania do Smart Home

Home assistant

Platforma Smart Home na licencji open source wykorzystująca język Python¹² a główne repozytorium kodu znajduje się na platformie GitHub:

<https://github.com/home-assistant/core>

Przykładowy panel użytkownika dostępny z interfejsu Web wygląda jak poniżej:



Rysunek 2.7-1 Interfejs Home Assistant Web. Źródło: Opracowanie własne.

Analizując zasadę działania oprogramowania można zauważyć, że jest ona zbliżona do rozwiązań firm komercyjnych, jednak używanie Home Assistant zapewnia większą elastyczność konfiguracji i nastawiony jest głównie na zarządzanie urządzeniami w sieci lokalnej. Logikę oprogramowania wykorzystaną w oprogramowaniu można podzielić na trzy moduły [20]:

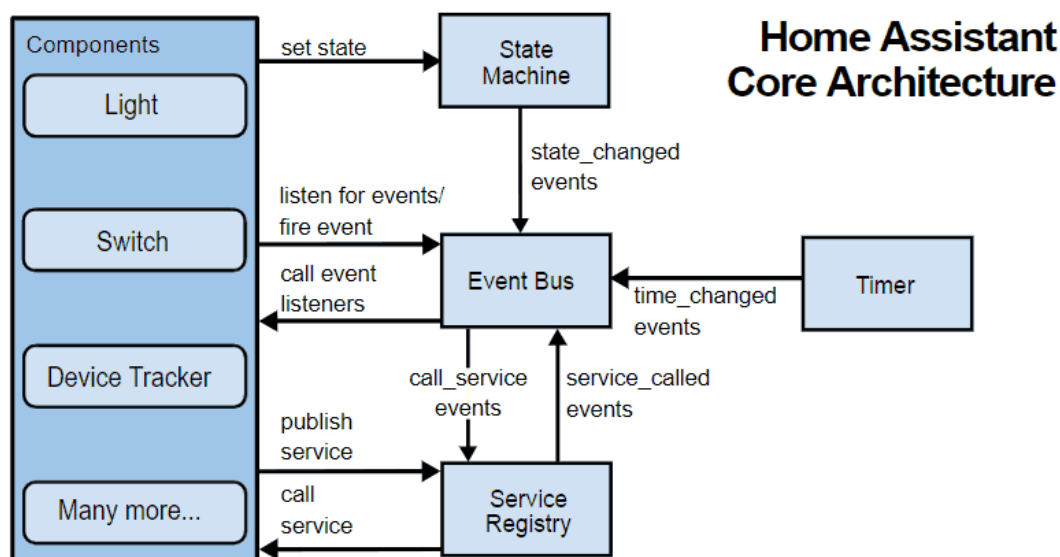
Home Control Odpowiada za zbieranie informacji i kontrolowanie urządzeń.

Home Automation: Wyzwala polecenia w oparciu o konfiguracje użytkownika

Smart Home: uruchamia polecenia na podstawie poprzedniego zachowania

Schemat architektury można znaleźć poniżej.

¹² Według <https://github.com/home-assistant/core> 100% kodu wykorzystuje język Python



Rysunek 2.7-2 Architektura Home Assistant źródło: <https://smarthome.university>

Home Assistant posiada bogatą dokumentację znajdującą się pod adresem:

<https://www.home-assistant.io/getting-started/>

Platforma posiada sporą społeczność, dlatego pierwsze kroki i instalacja środowiska nie powinna sprawić większego problemu dodatkowo istnieją gotowe obrazy dla maszyn wirtualnych popularnych środowisk wirtualizacyjnych takich jak: (VirtualBox, HyperV, KVM) ale również w ramach kontenerów Docker'a¹³ W zależności od typu instalacji mogą być dostępne różne funkcjonalności, dlatego aby w pełni cieszyć się środowiskiem najlepiej zainstalować dedykowany system HASS os.

¹³ Kontener - lekki, niezależny, gotowy do uruchomienia pakiet oprogramowania, który zawiera wszystko, co jest potrzebne do używania aplikacji

Compare Installation Methods				
	OS	Container	Core	Supervised
Automations	✓	✓	✓	✓
Lovelace	✓	✓	✓	✓
Integrations	✓	✓	✓	✓
Blueprints	✓	✓	✓	✓
Uses container	✓	✓	✗	✓
Supervisor	✓	✗	✗	✓
Add-ons	✓	✗	✗	✓
Snapshots	✓	✗	✗	✓
Managed OS	✓	✗	✗	✗

Rysunek 2.7-3 Porównanie typów instalacji i ich funkcjonalności. Źródło: <https://www.home-assistant.io/installation/>

Zarządzanie aktualizacjami stosunkowo proste i odbywa się przez interfejs www, podobnie jak instalacja dodatków czy pluginów która w przypadku pełnej wersji systemu zarządzana jest przez komponent „Supervisor”. Wykorzystywanie dodatków to jedna z kluczowych rzeczy w przypadku tego typu oprogramowania i tutaj również Home Assistant posiada sporą ilość elementów rozszerzających funkcjonalność do wyboru. Dodatki podzielone są na dwie kategorie: oficjalne i dodane przez społeczność dostępne na stronie:

<https://community.home-assistant.io/tag/hassio-repository>.

Do zarządzania dodatkami tworzonymi przez użytkowników warto użyć plugin HACS [24] (ang. Home Assistant Community Store) który synchronizuje dostępne repozytoria w panelu zarządzającym i maksymalnie ułatwia instalacje nowych rozszerzeń. W czerwcu 2021 lista oficjalnie kompatybilnych Home Assistant producentów urządzeń IoT i Serwisów wynosi 1800[21] i dostępna jest pod linkiem:

<https://www.home-assistant.io/integrations/>




Home Assistant ma wiele sposobów na tworzenie i edycję reguł automatyzacji. Pierwszym i najbardziej podstawowym jest użycie YAML¹⁴. YAML jest dobrym sposobem na tworzenie automatyzacji i jest bardzo potężnym narzędziem, ale może sprawiać problemy

¹⁴ YAML – (YAML Ain’t Markup Language) Przyjazny dla człowieka standard serializacji danych dla wszystkich języków programowania.

dla początkujących i jest kilka rzeczy, które ciężiej w nim zaimplementować (np. pętle). Zaletą korzystania z YAML jest to, że możesz łatwo udostępniać pliki konfiguracyjne, bo są one jawne i ustandaryzowane. Dla początkujących użytkowników dostępny jest również edytor automatyzacji, czyli wbudowane narzędzie pozwalające na tworzenie i edycję podstawowych reguł automatyzacji. Jest łatwy w użyciu, ale nadal wymaga dobrego zrozumienia nazw encji i koncepcji wywołań usług.

Node-RED to kolejny sposób zarządzania regułami automatyzacji. Jest to oparte na przepływach narzędzie programistyczne do programowania wizualnego, opracowane pierwotnie przez IBM do łączenia urządzeń sprzętowych, interfejsów API i usług online w ramach Internetu Rzeczy. Jego zalety to: dobra wizualizacja, łatwy w edycji i prosty do wdrożenia.

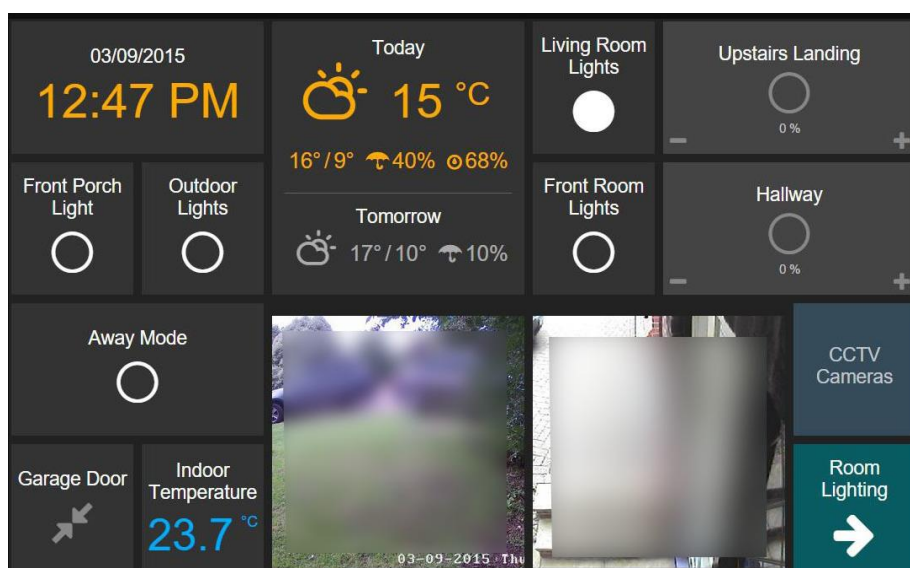
Home assistant udostępnia aplikacje na IOS i Androida która zapewnia podstawową funkcjonalność. Zapewnia on też łatwą integrację z asystentami głosowymi, jednak nie jest to usługa darmowa i kosztuje 5\$ miesięcznie [22]. Rozpatrując miejsce instalacji lokalnego serwera Home Assistant to najczęściej wybierana platforma to Raspberry który może jednocześnie służyć do innych usług, jednak opcjonalnie autorzy oferują dedykowany system i spersonalizowany płytkę ODROID-N2+ pod nazwą Home Assistant blue Cena zestawu do 140\$ a jego specyfikacja znajduje się poniżej

Specifications	
COMPONENTS	FORM FACTOR
PROCESSOR 6-Core Amlogic S922X Processor (ARMv8-A) - Quad-core Cortex-A73 @ 2.2Ghz - Dual-core Cortex-A53 @ 1.8Ghz	WIDTH 10.4 cm/4.1 inch
MEMORY 4GB DDR4	 HEIGHT 3.6 cm/1.4 inch
STORAGE 128GB eMMC Flash included	 DEPTH 9.4 cm/3.7 inch
NETWORKING 1x RJ45 LAN Port 10/100/1000 Mbps	
AUDIO 1x 3.5mm Jack High quality 384Khz/32bit stereo line-out	WEIGHT 292 g/0.64 lb
CONNECTIVITY 4x USB Type-A Ports USB 3.0 1x HDMI Output HDMI 2.0 No Wi-Fi or Bluetooth Support for Z-Wave and Zigbee by external USB adaptor (not included)	
POWER DC 12V/2A Power consumption - Idle: ≈2.2W - Load: ≈6.2W	

Rysunek 2.7-4 Specyfikacja HA Blue Źródło: <https://www.home-assistant.io/blue/>

OpenHAB

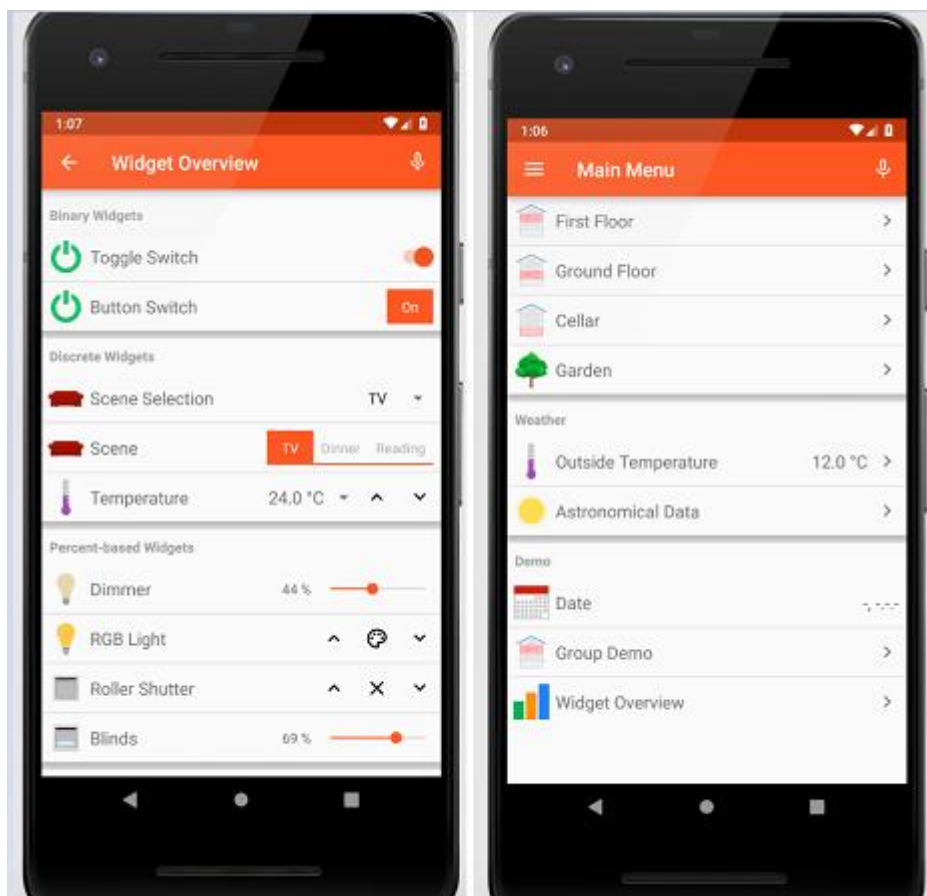
Jest to kolejna platforma automatyki domowej typu open source. Oznacza to, że jest ona niezależny od producentów komercyjnych. Za dodawanie kolejnych integracji odpowiada grupa twórców zwracająca uwagę na głosy społeczności. W rezultacie, jeśli urządzenie jest popularne, prawdopodobnie prędzej czy później będzie obsługiwane przez platformę. Kod źródłowy platformy jest napisany w Javie, co jest bezpośrednią zaletą systemu, ponieważ jest on uniwersalny i nieograniczony do konkretnej architektury systemu, na którym można go uruchomić. Architektura środowiska opiera się min. na powiązaniach, które zapewniają wsparcie dla różnych urządzeń i serwisach z dziedziny inteligentnego domu. Podstawowym sposobem tworzenia i wdrażania automatyzacji jest użycie języka Xtend. Xtend to elastyczny dialekt języka Java, który można kompilować kod źródłowy zgodny z Javą jednak część użytkowników mniej zaznajomiona z językiem twierdzi, że jest on skomplikowany i posiada chaotyczną dokumentację. Opcjonalnie do dyspozycji jest dostępny Blockly, czyli graficzny edytor bloków z silnikiem w JavaScript. Mimo iż wybór tego rozwiązania nie jest oczywisty dla osób zaczynających z systemami automatyki, do zalet tego systemu można zaliczyć dużą stabilność i niska zasobożerność systemu właśnie dlatego system może pracować choćby na Raspberry jako urządzeniu nadzorczym, gdzie dostępny jest dedykowany system „openHABian”. Proces Instalacji systemu jest dobrze opisany i dostępny na stronie projektu dla różnych platform [25]. Główny panel zarządzający jest dostępny w kilku wersjach i w każdym przypadku można go dostosować pod własne potrzeby lub skorzystać z niestandardowych szablonów.



Rysunek 2.7-5 Przykładowy widok OpenHAB Źródło: <https://community.openhab.org/>

Dodawanie urządzeń do tego systemu może być zautomatyzowane przez obsługę auto wykrywania i najpopularniejsze urządzenia można dodawać właśnie w ten sposób natomiast w niektórych przypadkach będą wymagane dodatkowe czynności i instalacja zewnętrznych repozytoriów dla każdego z takich przypadków dostępną są przykłady demonstrujące procedurę dodawania urządzeń [26].

W ramach systemu jest możliwa praca wyłącznie lokalna jak i z pośrednictwem dostawców chmurowych dedykowanych takich jak np. myopenHAB.org (płatny dostęp po uiszczeniu jednorazowej dotacji) lub innych komercyjnych dostawców. OpenHAB oferuje również integracje z najpopularniejszymi asystentami głosowymi a także integracje z serwisem IFTT oferującym zewnętrzny ekosystem automatyzacji. Podobnie jak konkurencja producent udostępnia aplikację mobilną na systemy Android i iOS



Rysunek 2.7-6 Widok aplikacji mobilnej OpenHAB Źródło: [https:// www.openhab.org](https://www.openhab.org)

2.8. Podsumowanie

Powyższe przykłady to zaledwie część dostępnych na rynku rozwiązań z dziedziny automatyki domowej które osiągnęły międzynarodową popularność. Warto zauważyć tendencje która wskazuje, że największe korporacje technologiczne również promują i inwestują w rynek automatyki domowej. Działania tych firm powodują mocny wzrost

ogólnego zainteresowania rynkiem Internetu rzeczy w tym także przyrost twórców niezależnych oferujących alternatywy na licencji opensource. W ogólnym rozrachunku taki stan rzeczy powoduje, że rynek jest bardziej zróżnicowany, a największym beneficjentem takiej zależności jest użytkownik końcowy który możliwość odnalezienia najlepiej dopasowanego do jego potrzeb systemu automatyki.

Rozdział 3

Opis funkcjonalności Raspberry Pi

3.1. Specyfikacja produktu

Raspberry PI to seria mikrokomputerów ciesząca się popularnością wśród entuzjastów technologii informatycznych i elektroniki. Oprócz mocy obliczeniowej komputery z serii Raspberry zyskały swoją sławę poprzez obsługę pinów GPIO (general-purpose input/output). 40-pinowe złącze GPIO wzdłuż górnej krawędzi płyty znajduje się na wszystkich obecnych płytach Raspberry Pi (w niektórych modelach występuje wersja 26-pinowa). Piny GPIO są cyfrowe, co oznacza, że posiadają dwa stany: ON lub OFF. Za ich pomocą można sterować kierunkiem odbioru lub wysyłania prądu (odpowiednio wejścia/wyjścia). Piny GPIO umożliwiają Raspberry Pi kontrolowanie i monitorowanie urządzeń zewnętrznych poprzez podłączenie ich do obwodów elektronicznych. Dzięki temu Pi jest w stanie sterować diodami LED, uruchamiać silniki, badać temperaturę, natężenie światła etc. W trakcie pracy z GPIO przydatna może być interaktywna dokumentacja tego złącza którą można znaleźć na stronie: <https://pinout.xyz/>

Obecnie na rynku dostępna jest czwarta generacja płytki Raspberry jednak pozostałe poprzednie modele również spełniają swoją funkcję a każdy kolejny model to po prostu sprzęt o większej mocy obliczeniowej większej ilości pamięci i obsługujący nowsze standardy łączności i komunikacji dla porównania można podać przykład różnicy modelu Raspberry Pi 4 B i jego wcześniejszą generację Raspberry Pi 3 Model B. Porównanie wszystkich pozostałych wersji można znaleźć na stronie :

<https://socialcompare.com/en/comparison/raspberrypi-models-comparison>

Raspberry Pi 4B vs 3B+



Features/Specs	Raspberry Pi 4 Model B	Raspberry Pi 3 Model B+
Release Date	24th June 2019	14th March 2018
SoC Type (Processor)	Broadcom BCM2711 (with metal cover)	Broadcom BCM2837B0 (with metal cover)
Core Type	Cortex-A72 64-bit (ARMv8)	Cortex-A53 64-bit (ARMv8)
No. of Cores	Quad-Core	
GPU	VideoCore VI	VideoCore IV
Multimedia	H.265 decode (4Kp60) H.264 decode (1080p60) H.264 encode (1080p30) OpenGL ES 1.1, 2.0, 3.0 Graphics	H.264, MPEG-4 decode (1080p30) H.264 encode (1080p30) OpenGL ES 1.1, 2.0 Graphics
CPU Clock	1.5 GHz	1.4 GHz
Memory/OS storage	microSD	
RAM	LPDDR4: 1GB, 2GB, 4GB and 8GB options	LPDDR2 1GB
Ethernet	True Gigabit Ethernet	Gigabit over USB 2.0 (Max 300Mbps)
USB Port	2 x USB 3.0 + 2 x USB 2.0	4 x USB 2.0
HDMI	2 x micro HDMI support Dual Display	1 x full size HDMI
WiFi	802.11 b/g/n/ac (2.4GHz+5GHz & Shielded)	
Bluetooth	5.0 + BLE (Shielded)	4.2 + BLE (Shielded)
Antenna	PCB Antenna (Similar to Rpi Zero W)	
GPIO	40 pins (Fully backwards-compatible with previous boards)	
Operating System	Raspbian (> 24 June 2019)	Raspbian (> March 2018)
Dimension	85mm x 56mm	
Power Input	5V via USB Type C (upto 3A) 5V via GPIO header (upto 3A) Power over Ethernet, requires PoE HAT	5V via USB Micro B (upto 2.5A) 5V via GPIO header (upto 3A) Power over Ethernet, requires PoE HAT

Rysunek 3.1-1 Porównanie płytek RPi 3 i 4. Źródło: <https://smartbitbn.com/product/raspberry-pi-4-model-b-2gb/>

3.2. Wspierane systemy operacyjne

Ze względu na rosnącą popularność minikomputera Raspberry i platformy platform ARM ilość dostępnych systemów operacyjnych rośnie. Zdecydowanie najpopularniejszym systemem jest dedykowana dystrybucja dla Raspberry oparta na Linuxowym Debianie: Raspberry PI OS (dawniej Raspbian). Jest on dostępny na wszystkie modele płytki i dostępny jest w trzech wersjach [23]:

- Raspberry Pi OS Lite – Wersja minimalistyczna bez interfejsu graficznego

- Raspberry Pi OS with desktop – Standardowa wersja z GUI LXDE ¹⁵
- Raspberry Pi OS with desktop and recommended software – Standardowa wersja dodatkowo, posiadająca preinstalowane pakiety polecane przez Pi Foundation.

Cieszący się popularnością Raspberry Pi OS system doczekał się też wersji na PC i Mac jako stabilny i szybki system pod nazwą Raspberry Pi Desktop.

Istnieją też dystrybucje systemów wydawane przez inne podmioty różniące się głównie przeznaczeniem. Wyróżnić je można min. na systemy operacyjne służące jako:

Centrum multimediiów:

- LibreELEC, OSMC – Dedykowane systemy min. dla platformy KODI,
- Volumio - Odtwarzanie i strumieniowanie muzyki:

Platforma dla programistów:

- Windows IoT Core- Platforma programistyczna do prototypowania urządzeń podłączonych do Internetu z wykorzystaniem chmury Microsoft Azure.

Konsole dla graczy:

- RetroPie, Lakka – Systemy operacyjne z wbudowanymi emulatorami konsol do gier.

Serwer plików:

- OpenMediaVault – Jeden z popularniejszych systemów do tworzenia udostępnionych dysków sieciowych.

System dla najmłodszych:

- Kano OS – Przyjazny i prosty system ukierunkowany na edukację dla dzieci.

Narzędzia dla inżynierów sieci

- Alpine Linux – Lekka dystrybucja bez GUI wykorzystywana do projektowania aplikacji sieciowych i bezpieczeństwa za pomocą narzędzi apk,
- Kali Linux – Dystrybucja skupiająca się na aspektach bezpieczeństwa sieciowego i testów penetracyjnych.

Thin Client

- TLXOS – System pozwalający na prace Raspberry jako terminal,
- Linutop – Lekka dystrybucją mogąca służyć za interaktywne banery reklamowe.

Alternatywa dla Raspbian:

¹⁵ GUI (ang. graphical user interface), LXDE – Interfejs graficzny oparty na licencji opensource wymagający niewielką liczbę zasobów do uruchomienia

- Ubuntu: Dystrybucja świetnie znana z architektury PC posiadająca ogromną liczbę użytkowników i bogate wsparcie społeczności,
- RISC OS: System stworzony przez twórców architektury ARM niepowiązany z systemem linux,
- RaspBSD – Alternatywa dla systemów linuksowych oparta otwartym systemie FreeBSD.

3.3. Możliwości rozszerzeń

Wcześniej wspomniane złącza GPIO zapewniają możliwość rozszerzenia funkcjonalności płytki Raspberry Pi.

Połączenia wykonywane za pomocą pinów GPIO, w przeciwieństwie do USB nie są typu „plug and play” i wymagają ostrożności, aby uniknąć błędnego podłączenia okablowania do pinów. GPIO Raspberry PI wykorzystują napięcie 3,3 V i mogą zostać uszkodzone, jeśli zostaną podłączone bezpośrednio do pinów o napięciu 5 V (jak w wielu starszych systemach cyfrowych) bez obwodów konwersji poziomów. Dodatkowo też w przypadku Raspberry możliwa jest komunikacja z chipami lub modułami za pomocą protokołów niskopoziomowych: SPI, I²C lub szeregowym UART. To oznacza, że lista dostępnych urządzeń mogących współpracować z Raspberry jest naprawdę długa, ponieważ przy odpowiednim podłączeniu i oprogramowaniu można podłączyć prawie każdy niskonapięciowy układ elektroniczny lub pośrednio korzystając z płytki rozszerzającej.

Lista sprawdzonych urządzeń rozszerzających znajduje się na Wiki portalu elinux.org: https://elinux.org/RPi_VerifiedPeripherals.

Oprócz tego, występują płytki prototypowe które ułatwiają podłączanie części urządzeń i nie wymagają lutowania złączy. Kolejną kategorią rozpatrywaną w aspekcie możliwości rozszerzeń są gotowe urządzenia działające np. po porcie USB które po podłączeniu i prawidłowym skonfigurowaniu dodają dodatkową funkcjonalność płytce. Np. Karty Wifi, Modemy LTE czy wykorzystany później w tej pracy Hub ZigBee dodający funkcjonalność komunikacji przez protokół ZigBee. Dzięki elastyczności w podłączeniu peryferiów płytka oferuje ogrom możliwości i może zostać wykorzystana w większości projektów z zakresu IT i IoT

3.4. Istniejące projekty

Biorąc pod uwagę mnogość możliwości rozszerzeń płytki Raspberry i spore zainteresowanie społeczności, w Internecie można znaleźć dużą liczbę projektów z przeróżnych dziedzin. Na oficjalnej stronie znajduje się również repozytorium projektów z wykorzystaniem Raspberry PI dostępnych pod adresem:

<https://projects.raspberrypi.org/en/projects>

Poniżej najczęściej spotykane scenariusze wraz z odnośnikami do autorów projektu.

Zdalnie sterowany robot:

- Firma Seeedstudio zajmująca się automatyką w dziedzinie IoT udostępniła projekt ręki robota obsługiwanej przez Raspberry PI 4:

<https://project.seeedstudio.com/SeeedStudio/raspberry-pi-as-robot-arm-controller-with-3d-gesture-shield-edc5b1>

Router i narzędzie do monitorowania sieci:

- Praktyczne wykorzystanie systemu NEMS do monitorowania sieci

<https://www.makeuseof.com/tag/turn-raspberry-pi-network-monitoring-tool/>

Serwer multimedialny:

- Wykorzystanie KODI jako platformy dla kina domowego:

<https://www.makeuseof.com/tag/kodi-raspberry-pi-media-center/>

System automatyki domowej:

- Wykorzystanie platformy OpenHAB:

<https://www.makeuseof.com/tag/getting-started-openhab-home-automation-raspberry-pi/>

Domowy system security:

- Projekt Raspberry pi Foundation z wykorzystaniem wiązki lasera:

<https://projects.raspberrypi.org/en/projects/laser-tripwire>

Konsola do gier:

- Kolejny projekt od Seedstudio, jednak w Internecie można znaleźć wiele wariacji tego projektu:

<https://www.seeedstudio.com/blog/2019/10/16/build-your-own-raspberry-pi-4-retro-game-console-retropie/>

Dysk sieciowy:

- Artykuł branżowego czasopisma poświęcony konfiguracji Raspberry jako serwer plików

<https://magpi.raspberrypi.org/articles/build-a-raspberry-pi-nas>

Sterownik do drukarek 3D

- Kolejny z oficjalnie promowanych projektów tym razem dotyczący sterownika drukarek 3D

<https://projects.raspberrypi.org/en/projects/build-an-octapi>

Inteligente Lustro:

- Domowy gadżet z dostępem do najnowszych informacji ze świata:

<https://www.instructables.com/How-to-Build-a-Raspberry-Pi-Smart-Mirror/>

Stacja pogodowa:

- Popularny projekt do badania parametrów pogody

<https://projects.raspberrypi.org/en/projects/build-your-own-weather-station>

Platforma do uczenia maszynowego:

- Z wykorzystaniem biblioteki TensorFlow

<https://towardsdatascience.com/3-ways-to-install-tensorflow-2-on-raspberry-pi-fe1fa2da9104>

3.5. Pomoc techniczna i społeczność

Grono użytkowników płytki Raspberry to najczęściej społeczność wolontariuszy i młodych ludzi uczących się i tworzących z wykorzystaniem technologii ciekawe projekty. Są to zazwyczaj pasjonaci, którzy gotowi są do pomocy w przypadku rozwiązania problemów na forach tematycznych. Pomysłodawcy platformy zachęcają dzieci do nauki nowych technologii min. Z zakresu IoT tworząc dla nich portale internetowe do nauki zrzeszających nauczycieli i uczniów z całego świata są to min.

- <https://coderdojo.com/> - Klub programistów dedykowany dla dzieci w wieku 7-17 lat,
- <https://codeclub.org/en/> - Globalna sieć edukacyjna dla dzieci w wieku 9–13 lat.

Oprócz tego społeczność często organizuje własne spotkania np. w ramach Raspberry Jam, gdzie spotykają się entuzjaści technologiczni z całego świata we wcześniej określonym miejscu.

Jeżeli chodzi o literaturę nt. projektów i nowości związanych z Raspberry Pi to dostępne są cyklicznie wydawane magazyny poświęcone tematyce IoT i Raspberry min. pod nazwami: The MagPi, HackSpace, Wireframe, Custom PC.

Do dyspozycji są także kursy o przeróżnej tematyce z zakresu IoT które można znaleźć min. korzystając z wyszukiwarki google lub na oficjalnej stronie Raspberry: <https://www.raspberrypi.org/courses/featured>.

Założyciele prowadzą również bloga o tematyce z zakresu IoT gdzie można przeczytać o nowościach w branży: <https://www.raspberrypi.org/blog/>.

Standardowo jak w przypadku tak dużej społeczności dostępne jest też oficjalne forum, gdzie poruszane są tematy techniczne z zakresu Raspberry. W czerwcu 2021 na forum było ponad 1.7 miliona postów i ponad 300 tys. Użytkowników. Forum dostępne jest pod adresem: <https://www.raspberrypi.org/forums/>.

Rozwiązywanie problemów technicznych w przypadku Raspberry również może okazać się proste, ponieważ korzystając z wyszukiwarki Google możemy sprawdzić czy szukany przez nas problem nie wystąpił już u kogoś innego (niekoniecznie na Raspberry, ale też w systemie z rodziny Linux). Tutaj z pomocą przychodzą najczęściej platformy <https://stackoverflow.com/> i <https://unix.stackexchange.com/>

Rozdział 4

Prezentacja środowiska Domoticz

4.1. Opis produktu

Domoticz stanowi otwartą, uniwersalną i otwarto źródłową platformę do zarządzania funkcjami inteligentnego domu. Całe repozytorium znajduje się na platformie GitHub pod adresem: <https://github.com/domoticz/domoticz>. Cechą Domoticz jest łatwość dodawania niestandardowych skryptów w przypadku rozwiązań konkurencyjnych, gdzie najczęściej jest to mocno utrudnione. Charakteryzuje się dużą elastycznością, a dzięki temu, że używa skryptów LUA, zadania automatyzacji są przejrzyste jednocześnie spełniając swoją nie-rzadko skomplikowaną rolę. Można za ich pomocą sterować urządzeniami na podstawie zdarzeń. Podstawowe zadanie automatyzacji takie jak: harmonogramy, wyzwalacze oparte na czasie, wyzwalacze oparte na urządzeniach, a nawet uzbrajanie/rozbrajanie zabezpieczeń działa bezbłędnie i jest dostępna za pomocą kilku kliknięć. Domoticz dobrze poradzi sobie z obsługą integracji z inteligentnymi produktami domowymi najpopularniejszych producentów takimi jak Nest, Philips Hue, Xiaomi oferuje też szeroką kompatybilność z kamerami IP. Kolejną zaletą Domoticz jest to, że wszystkie reguły, skrypty i procedury będą nieprzerwanie kontynuowane nawet wtedy, Internet nie działa. W przypadku komercyjnych rozwiązań możesz się okazać, że bez połączenia z Internetem sterowanie urządzeniami będzie niemożliwe. Istnieje oczywiście możliwość wystawienie usług Domoticz do Internetu, jednak musimy wtedy albo skorzystać z prywatnego dostawcy usług bramy VPN lub udostępnić swoją platformę własnoręcznie.

Domoticz oferuje również aplikacje dostępną z poziomu Play Store, jednak jest podstawowa wersja jest nieco okrojona. Zakup wersji Premium dostępny jest aktualnie za 32.99 zł i jest to wersja wieczysta. Posiadając wersję Premium mamy dostęp do rozszerzonej funkcjonalności min.









- Geo-fencing - możliwość dodawania wyzwalaczy na podstawie lokalizacji urządzeń),
- Wsparcie Android Wear - czyli dostęp do aplikacji np. na Opaskach i zegarkach sportowych,
- Widżety - czyli skrótom dostępne na ekranie głównym w systemach mobilnych również integracja ze skrótami google Home (w wersji android 11).

4.2. Wymagania systemowe i sprzętowe

Aby pomyślnie zainstalować Domoticz na płytce Raspberry należy spełnić poniższe wymagania:

- Raspberry Pi 4, Pi 3 Model B (obsługiwane są również Raspberry Pi 1 Model B (+) i 2 Model B),
- karta microSD minimum 4 GB (zalecany standard: class 10),
- komputer z czytnikiem kart SD do zainstalowania systemu operacyjnego na karcie SD,
- wyświetlacz i kabel HDMI, Klawiatura i mysz USB lub dostęp zdalny np. przez SSH,
- zasilanie micro USB 5V (zalecane zasilanie 2A, 2.5A),
- przewód Ethernet (lub adapter USB WiFi, jeżeli płytka nie ma wbudowanego WiFi).

Producent oferuje instalacje na najpopularniejszych architekturach i systemach operacyjnych min. na Raspberry Pi, czyli architekturze ARM 32bit. Opcjonalnie jest też wersja obrazu w ramach kontenera dla platformy Docker

Stable					
	Arm 32bit (Raspberry/Cubie/...)	2021.1.13191	2021-04-17	15,01 MB	
	Arm 64bit (Cubie/ODroid/...)	2021.1.13191	2021-04-17	15,5 MB	
	Windows	2021.1.13191	2021-04-17	13,12 MB	
	Linux	2021.1.13191	2021-04-17	15,86 MB	

Rysunek 4.2-1 Dostępne obrazy Domoticz Źródło: <https://www.domoticz.com/downloads/>

4.3. Kompatybilność z urządzeniami IoT

Domoticz na swojej stronie Wiki udostępnia listę rekomendowanych urządzeń możliwych do integracji ze swoim systemem. Lista dostępna jest pod adresem:

https://www.domoticz.com/wiki/Raspberry_Pi#Raspberry_Pi_related_hardware

Razem z odnośnikami do sklepu GadgetFreakz który jest partnerem wspierającym rozwiązania Domoticz [27].

Jeżeli urządzenie nie znajduje się na liście kompatybilności nie oznacza to wyłącznie, że jest ono niekompatybilne lub nieprzetestowane przy użyciu oficjalnego build środowiska. Jednakże można skorzystać z dodatkowo dostępnych pluginów oficjalnie wspieranych lub tych rozwijanych przez społeczność które rozszerzają funkcjonalność platformy. Oficjalnie wspierane rozszerzenia z krótkim opisem i odnośnikami do stron autorów znajdują się na stronie: <https://www.domoticz.com/wiki/Plugins>. Dodatkowo korzystając z platformy Raspberry PI jako systemu gospodarza możemy integrować urządzenia bezpośrednio do niego podłączone tzn. wykorzystać porty GPIO podłączając do nich czujniki i urządzenia peryferyjne, aby potem móc sterować nimi centralnie za pomocą panelu zarządzającego dostępnego w środowisku Domoticz.

4.4. Pomoc techniczna i społeczność

Najwięcej artykułów poświęconych Domoticz można znaleźć na wcześniej wspomnianym portalu partnerskim GadgetFreakz [28] Portal prowadzi też swojego bloga, w który omawia zmiany i nowości w oprogramowaniu jednocześnie udostępniając poradniki i konkretne scenariusze użycia platformy Domoticz z przeróżnymi urządzeniami. Oprócz tego Domoticz udostępnia swój własny portal z bazą wiedzy typu Wiki, gdzie znajdują się opisy kompatybilności instrukcje instalacji na najpopularniejszych platformach, ale też wiele innych informacji dotyczących tematów pokrewnych [29]. Domoticz udostępnia też oficjalne forum [30] gdzie można podyskutować na tematy techniczne zaoponować nowe rozwiązania lub poprosić o pomoc w konfiguracji środowiska. Aktualnie na forum jest ponad 200000 postów i ponad 25 tysięcy aktywnych użytkowników. Oprócz tego podobnie jak w przypadku innych rozwiązań warto przeszukać Internet w celu odnalezienia solucji na rozwiązanie problemów, które nierzadko mogą być spowodowane pracą na systemie linuksowym.

Rozdział 5

Praktyczne zastosowanie konceptu rozwiązania

Smart Home w Domoticz

5.1. Koncepcja rozwiązania i opis funkcjonalności

Jako główne założenia prezentacji środowiska Domoticz zostały przyjęte następujące cele:

- uruchomienie platformy Domoticz na Raspberry Pi 2B+ z systemem Raspberry PI OS,
- zainstalowanie uniwersalnej bramki ZigBee z integracją w środowisku Domoticz,
- dodanie przykładowych urządzeń pracujących w protokole ZigBee,
- dodanie przykładowych urządzeń kompatybilnych z platformą,
- dodanie kamery IP,
- dodanie zadań automatyzacji opartych na urządzeniach,
- przygotowanie Sceny urządzeń,
- instalacja pluginów monitorujących,
- utworzenie reguł powiadomień mailowych / typu push,
- konfiguracja aplikacji na systemie android,
- konfiguracja dostępu zdalnego.

Dzięki wyżej wymienionym celom będzie możliwe zdalne sterowanie w sieci LAN i WAN z wykorzystaniem interfejsu Webowego i Aplikacji mobilnej urządzeniami dołączonymi do systemu, ale także pogląd w czasie rzeczywistym na aktualne parametry czujników oraz obraz z kamery. Część zadań zostanie zautomatyzowana dzięki czemu nie będzie potrzeby ręcznego sterowania urządzeniami. Wcześniej zdefiniowane reguły pozwolą na wysyłanie powiadomień o aktualnym stanie urządzeń po spełnieniu warunku lub przekroczeniu zadanego progu wartości.

5.2. Przygotowanie sprzętowe

Na potrzeby projektu zostały przygotowane

- mini komputer Raspberry PI model 2B CPU ARM Cortex-A53 900 MHz, 1GB RAM,

- karta pamięci: SanDisk Extreme microSDHC UHS-I Card (32GB),
- kontroler USB kompatybilny z zigbee2mqtt: ZigBee CC2531,
- urządzenia IoT min.: Zigbee: Tuya NY-386, Xiaomi Yeelight smart bulb.

5.3. Konfiguracja początkowa

- instalacja Raspberry Pi OS na karcie SD np. za pomocą PiImager [34],
- uruchomienie usługi SSH do kontynuowania konfiguracji zdalnie:

Korzystając z konfiguratora wbudowanego w Raspberry Pi (raspi-config)

należy włączyć dostęp zdalny przez SSH, aby to zrobić należy w menu raspi-config przejść do: *Interface Options* -> *SSH* i potwierdzić włączenie usługi

Stan usługi można sprawdzić komendą:

```
root@raspberrypi:/home/pi# systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2021-07-02 20:03:20 CEST; 10min ago
```

Rysunek 5.3-1 Status usług SSH Źródło: Opracowanie własne.

- Aktualizacja listy repozytoriów i instalacja aktualizacji pakietów:

```
root@raspberrypi:/home/pi# apt update && apt upgrade
Stary:1 http://raspbian.raspberrypi.org/raspbian buster InRelease
Stary:2 http://archive.raspberrypi.org/debian buster InRelease
```

Rysunek 5.3-2 Instalacja aktualizacji Źródło: Opracowanie własne.

Instalacja środowiska domoticz

Jedną z metod instalacji udostępnianą przez producenta to pobranie skryptu instalacyjnego.

W takim wypadku instalacja może zostać zrealizowana za pomocą pojedynczego polecenia:

```
curl -L https://install.domoticz.com | bash
```

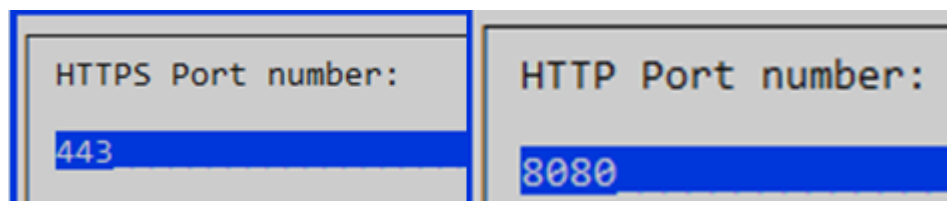
Po przekopiowaniu plików przez skrypt, przechodzimy do kreatora Domoticz i wybieramy serwisy, z których chcemy korzystać. W tym wypadku do wyboru są dwie opcje: http i https:

```
Select Services (press space to select)

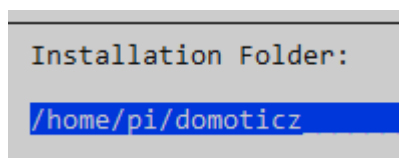
[*] HTTP   Enables HTTP access
[*] HTTPS  Enables HTTPS access
```

Rysunek 5.3-3 Wybór dostępnych serwisów dla Domoticz Źródło: Opracowanie własne

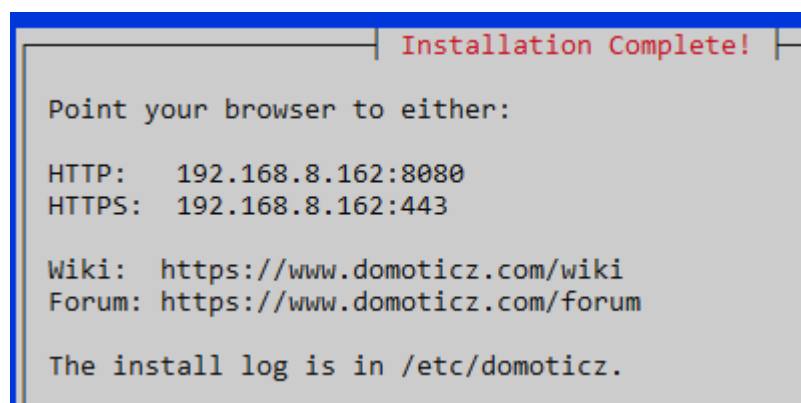
W następnym etapie można skonfigurować porty wcześniej wybranych usług: Domyślnie 8080(http) 443(https),



Rysunek 5.3-4 Konfiguracja portów dla serwisów Domoticz Źródło: Opracowanie własne.
a następnie miejsce instalacji.



Rysunek 5.3-5 Wybór miejsca instalacji Domoticz Źródło: Opracowanie własne
Po pomyślnej instalacji serwer domoticz jest gotowy do pracy:

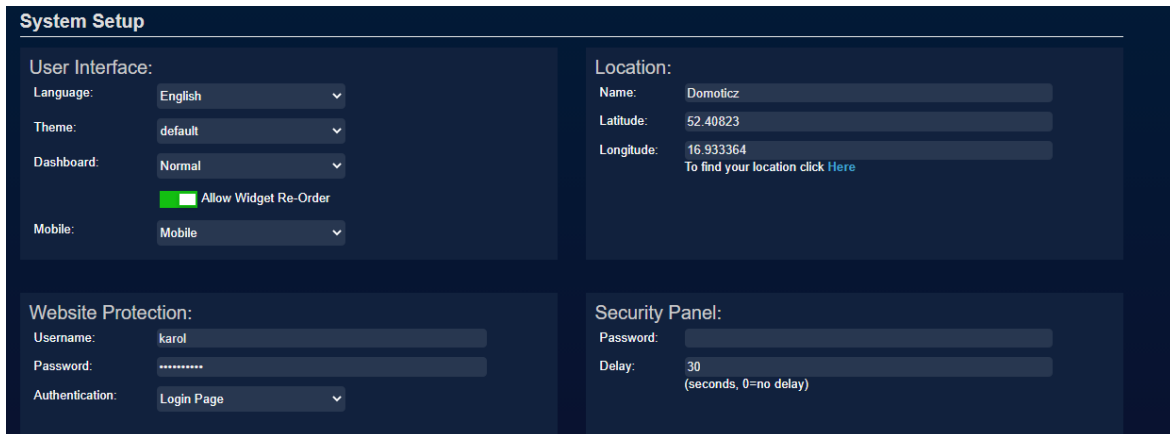


Rysunek 5.3-6 Zakończenie instalacji Źródło: Opracowanie własne.
Przechodząc pod wskazany adres można dostać się do panelu zarządzania domoticz



Rysunek 5.3-7 Strona powitalna Domoticz. Źródło: Opracowanie własne.

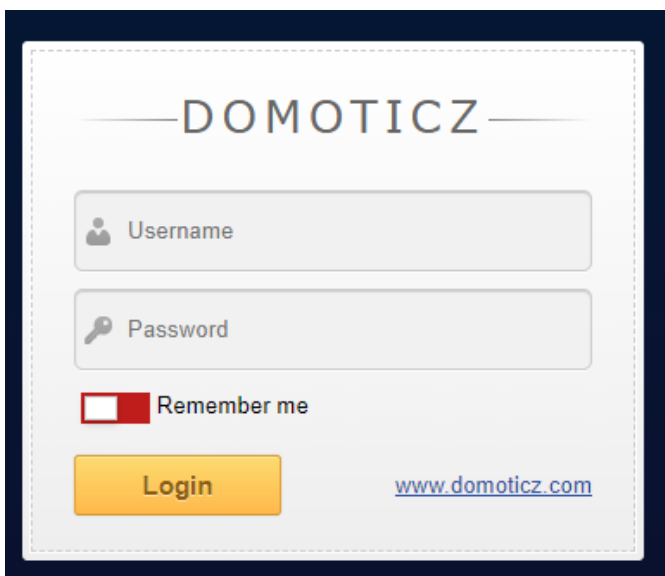
W następnej kolejności warto przejść do ustawień Setup -> Settings i nadać hasło logowania do panelu i ustawić lokalizację (przydatną później przy konfiguracji). W tym celu należy wypełnić formularz „web protection” oraz „location”



System Setup	
User Interface:	
Language:	English
Theme:	default
Dashboard:	Normal
	<input checked="" type="checkbox"/> Allow Widget Re-Order
Mobile:	Mobile
Location:	
Name:	Domoticz
Latitude:	52.40823
Longitude:	16.933364 To find your location click Here
Website Protection:	
Username:	karol
Password:	*****
Authentication:	Login Page
Security Panel:	
Password:	*****
Delay:	30 (seconds, 0=no delay)

Rysunek 5.3-8 Zakończenie instalacji Źródło: Opracowanie własne.

Po zastosowaniu ustawień Domoticz powinien przekierować do panelu logowania:



DOMOTICZ

☐ Remember me

[www.domoticz.com](#)

Login

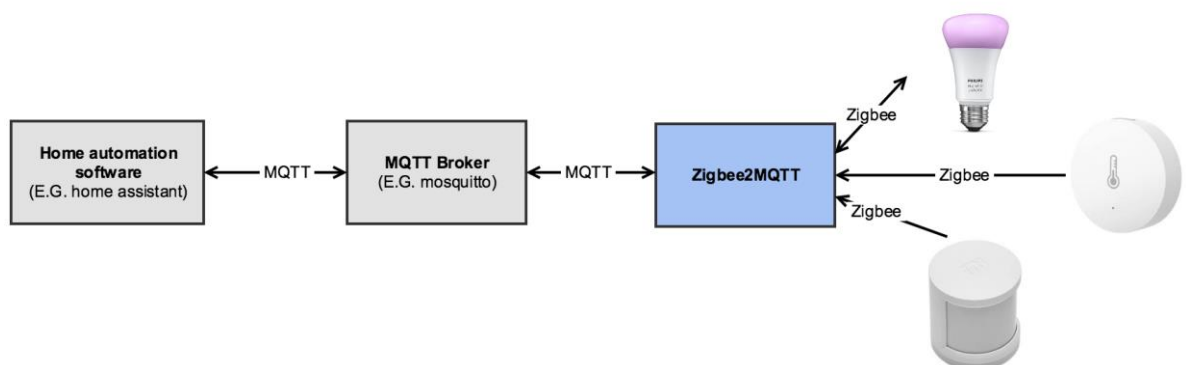
Rysunek 5.3-9 Ekran logowania w Domoticz. Źródło: Opracowanie własne.

5.4. Dostępne pluginy

Pluginy to dodatki dostępne jako komponenty dodatkowe które rozszerzają funkcjonalność środowiska Domoticz o nowe możliwości. W przypadku instalacji pluginów istnieją dwie metody ich instalacji ręczna: Klonowanie repozytorium pod wskazaną ścieżką sposób opisany w podrozdziale: 5.4.1 zigbee2mqtt-plugin lub za pomocą dodatku PP-manager, który ułatwia proces i automatycznie klonuje odpowiednie repozytorium do ścieżki docelowej. Warto jednak pamiętać, że w każdym przypadku w pierwszej kolejności trzeba zapoznać się z dokumentacją pluginu i zainstalować potrzebne komponenty do jego uruchomienia. PP-manager nie sprawdzi się w każdym przypadku, ponieważ nie potrafi doinstalować wszystkich zależności wymaganych przez repozytorium.

5.5. zigbee2mqtt-plugin

Aby podłączyć urządzenia pracujące z protokołem ZigBee należy posiadać urządzenie pełniące funkcje koordynatora i bramy pośredniczącej w komunikacji pomiędzy urządzeniem końcowym a serwerem. Najczęściej spotykany scenariusz to dedykowana brama integrująca wyłącznie urządzenie jednego producenta. W przypadku biblioteki Zigbee2MQTT [31] mamy możliwość użycia uniwersalnego koordynatora łączącego się ze wszystkim urządzeniami kompatybilnymi wykorzystując protokół MQTT¹⁶ i działając jako broker. Takie podejście daje możliwość przyłączania urządzeń różnych producentów koncentrując je w jednym miejscu np. w środowisku automatyki Domoticz:



Rysunek 5.5-1 Schemat działania biblioteki ZigbeeMQTT Źródło: <https://github.com/koenkk/zigbee2mqtt>

¹⁶ MQTT (ang. Message Queuing Telemetry Transport) - Lekki protokół sieciowy typu publikuj-subskrybuj, który przesyła wiadomości między urządzeniami.

Lista kompatybilności urządzeń jest na bieżąco aktualizowana i zawiera informacje o zakresie funkcjonalności każdego z nich [32]. Aby biblioteka działała poprawnie należy posiadać Adapter Zigbee np. firmy Texas Instruments CC253[33] który należy podłączyć i skonfigurować w środowisku Raspberry.

Do poprawnego uruchomienia biblioteki ZigbeeMQTT potrzebny jest:

- Node.js - aby go zainstalować w przypadku Raspberry Pi OS należy użyć polecenia

```
root@raspberrypi:/home/pi# apt-get install -y nodejs git make g++ gcc
Czytanie list pakietów... Gotowe
Budowanie drzewa zależności
```

Rysunek 5.5-2 Instalacja node.js w systemie Raspberry Pi OS. Źródło. Opracowanie własne.

- Narzędzie do zarządzania pakietami npm w środowisku Node.js

```
root@raspberrypi:/home/pi# apt install npm
Czytanie list pakietów... Gotowe
Budowanie drzewa zależności
```

Rysunek 5.5-3 Instalacja npm w systemie Raspberry Pi OS. Źródło. Opracowanie własne.

Do poprawnego działania biblioteki potrzebna jest wersja node.js 10.x,12.x,14.x i npm w wersji 6.x lub 7.x aby sprawdzić poprawność wersji można użyć poleceń w przypadku zainstalowania starszych wersji potrzebna będzie aktualizacja np. korzystając z menadżera pakietów i przełącznika --upgrade

```
pi@raspberrypi:~$ sudo node --version && sudo npm --version
v14.17.2
7.19.1
```

Rysunek 5.5-4 Sprawdzanie wersji pakietów node.js i npm w systemie Raspberry Pi OS. Źródło. Opracowanie własne.

Po spełnieniu wymagań wstępnych można przejść do pobrania biblioteki z platformy github:

```
root@raspberrypi:/home/pi# sudo git clone https://github.com/Koenkk/zigbee2mqtt.git /opt/zigbee2mqtt
Cloning into '/opt/zigbee2mqtt'...
remote: Enumerating objects: 17426, done.
```

Rysunek 5.5-5 Pobieranie biblioteki zigbee2mqtt Raspberry Pi OS. Źródło. Opracowanie własne.

Oraz zmienić jego uprawnienia zmieniając właściciela na użytkownika pi:

```
root@raspberrypi:/home/pi# sudo chown -R pi:pi /opt/zigbee2mqtt
root@raspberrypi:/home/pi#
```

Rysunek 5.5-6 Zmiana uprawnień biblioteki. Źródło. Opracowanie własne.

Jako użytkownik Pi w kolejnym kroku należy zainstalować zależności wymagane do poprawnego działania biblioteki

```
pi@raspberrypi:/opt/zigbee2mqtt $ sudo npm ci
```

Rysunek 5.5-7 Instalacja zależności biblioteki. Źródło. Opracowanie własne.

Żeby zautomatyzować proces warto stworzyć nową usługę, w ramach której automatycznie będzie uruchamiany npm

W tym celu należy utworzyć nowy plik w ścieżce /etc/systemd/system/

Z rozszerzeniem .service np. „zigbee2mqtt.service”

Po podłączeniu Adaptera USB można sprawdzić jego obecność w systemie za pomocą komendy:

```
root@raspberrypi:/home/pi# ls -l /dev/serial/by-id
razem 0
lrwxrwxrwx 1 root root 13 lip  2 20:02 usb-Texas_Instruments_TI_CC2531_USB_CDC___0X00124B0014DA4D3A-if00 -> ../../ttyACM0
```

Rysunek 5.5-8 Listowanie urządzeń USB w systemie Raspberry PI OS. Źródło. Opracowanie własne

(opcjonalnie) Po instalacji zależności warto otworzyć plik konfiguracyjny w celu dokonania potrzebnych zmian np. gdy urządzenie zmapowało się pod inną ścieżką lub gdy jest potrzeba wdrożenia mechanizmu autoryzacji nowych urządzeń (np. za pomocą loginu i hasła):

```
GNU nano 3.2 /opt/zigbee2mqtt/data/configuration.yaml

# Home Assistant integration (MQTT discovery)
homeassistant: false

# allow new devices to join
permit_join: true

# MQTT settings
mqtt:
  # MQTT base topic for zigbee2mqtt MQTT messages
  base_topic: zigbee2mqtt
  # MQTT server URL
  server: 'mqtt://localhost'
  # MQTT server authentication, uncomment if required:
  # user: my_user
  # password: my_password

# Serial settings
serial:
  # Location of CC2531 USB sniffer
  port: /dev/ttyACM0
```

Rysunek 5.5-9 Konfiguracja biblioteki zigbee2mqtt. Źródło. Opracowanie własne.

Po uruchomieniu usługi warto sprawdzić czy działa ona poprawnie:

```
pi@raspberrypi:/opt/zigbee2mqtt $ sudo systemctl status zigbee2mqtt
● zigbee2mqtt.service - zigbee2mqtt
   Loaded: loaded (/etc/systemd/system/zigbee2mqtt.service; disabled; vendor preset: enabled)
   Active: active (running) since Fri 2021-07-02 21:52:58 CEST; 6s ago
```

Rysunek 5.5-10 Sprawdzanie statusu usługi w Raspberry PI OS. Źródło. Opracowanie własne.

Ostatnim krokiem jest instalacja brokera MQTT w postaci pakietu mosquitto:

```
pi@raspberrypi:/opt/zigbee2mqtt $ sudo apt install mosquitto
Czytanie list pakietów... Gotowe
```

Rysunek 5.5-11 Instalacja pakietu mosquitto w systemie Raspberry PI OS. Źródło. Opracowanie własne.

Aby sprawdzić czy urządzenia są wykrywane przez koordynator zigbee można sprawdzić status usług i jego logi

```
pi@raspberrypi:/opt/zigbee2mqtt $ sudo journalctl -u zigbee2mqtt.service -f
```

Rysunek 5.5-12 Wyświetlenie dziennika biblioteki zigbee2mqtt. Źródło. Opracowanie własne.

Jeśli w dzienniku pojawiają się wpisy i wartości odczytów z urządzeń po uruchomieniu czujnika pracującego w protokole ZigBee oznacza to, że koordynator pracuje poprawnie i potrafi sparować się z innymi urządzeniami.

Aby zintegrować kontroler Zigbee z Domoticz potrzebny jest wtyczka dostępna na GitHub w repozytorium github.com/stas-demydiuk/domoticz-zigbee2mqtt-plugin. Aby plugin działał poprawnie na serwerze musi znajdować się interpreter Python3 (który domyślnie zazwyczaj jest zainstalowany). Aby sprawdzić czy w systemie jest już zainstalowany interpreter python3 można użyć polecenia `python3 --version`

W celu bezpośredniego pobrania do repozytorium do folderu z pluginami można użyć polecenia „git clone”. Domyślna ścieżka dla pluginów w Przypadku Domoticz to: `<miejsce_instalacji_domoticz>/plugins` w tym wypadku: `/home/rpi/domoticz/plugins`

```
pi@raspberrypi:/ $ cd /home/pi/domoticz/plugins/
pi@raspberrypi:~/domoticz/plugins $ git clone https://github.com/stas-demydiuk/domoticz-zigbee2mqtt-plugin.git zigbee2mqtt
Cloning into 'zigbee2mqtt'...
remote: Enumerating objects: 3175, done.
remote: Counting objects: 100% (297/297), done.
remote: Compressing objects: 100% (180/180), done.
remote: Total 3175 (delta 164), reused 199 (delta 105), pack-reused 2878
Receiving objects: 100% (3175/3175), 1.16 MiB | 1.76 MiB/s, done.
Resolving deltas: 100% (2221/2221), done.
```

Rysunek 5.5-13 Pobranie pluginu domoticz--zigbee2mqtt. Źródło. Opracowanie własne.

Aby plugin był widoczny z poziomu panelu Domoticz należy zrestartować jego usługę poleceniem: „`sudo service domoticz restart`”

Gdy usługa uruchomi się ponownie w panelu Domoticz można już dodać nowe urządzenie typu Zigbee2MQTT (opcjonalnie) skonfigurować dane dostępowe zgodnie z plikiem, który został wcześniej utworzony tj. `/opt/zigbee2mqtt/data/configuration.yaml`

Enabled: ☒

Name:

Type:

Log Level: ☒ Info ☒ Status ☒ Error

Data Timeout:
 Specifying a Data Timeout will restart the hardware device if no data is received for the specified time.
 Do not enable this option for devices that do not receive data!

Zigbee2MQTT Plugin

Plugin to add support for [zigbee2mqtt](#) project

Features

- Allows to manage and control zigbee devices
- Allows to manage and control zigbee groups
- Custom UI page to improve user experience
- Zigbee network map visualization

Blacklist

Plugin allows to skip processing of some devices by adding them to the blacklist. Blacklisted devices will not be created as logical devices in Domoticz.

Blacklist could contain several entries divided by semi-colon. Each blacklist entry should be valid [Python regexp](#). For example:

- Single item of a device - use full Device ID (ieee addr + alias) to block it i.e. `0x60a423fffe9f3c22_signal`
- All items of a device - use Regexp to block all logical devices (aliases) of a specified zigbee device (ieee address) i.e. `0x60a423fffe9f3c22_*`
- Specific item of all devices - use Regexp to block all devices (ieee address) with a specific alias i.e. `*_signal`

MQTT Server address:

Port:

MQTT Username (optional):

MQTT Password (optional):

MQTT Client ID (optional):

Zigbee2Mqtt Topic:

Devices Blacklist:

Track Link Quality:

Use Battery Devices:

Debug:

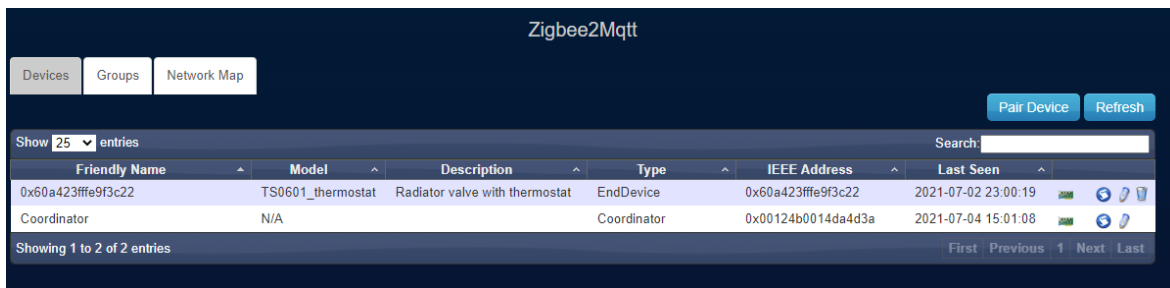
Rysunek 5.5-14 Konfiguracja pluginu domoticz--zigbee2mqtt. Źródło. Opracowanie własne.

Po dodaniu urządzenia przechodząc do zakładki sprzęt, gdzie dostępne są wszystkie czujniki które podłączone urządzenia Zigbee udostępniają.

	Idx ^	Hardware ^	ID ^	Unit ^	Name ^	Type ^	SubType ^	
<input type="checkbox"/>	2	Zigbee2MQTT	0x60a423fffe9f3c22_signal	1	Zigbee2MQTT - 0x60a423fffe9f3c22 (Link Quality)	General	Custom Sensor	97
<input type="checkbox"/>	3	Zigbee2MQTT	0x60a423fffe9f3c22_cell	2	Zigbee2MQTT - 0x60a423fffe9f3c22 (Battery Voltage)	General	Voltage	0 V
<input type="checkbox"/>	4	Zigbee2MQTT	0x60a423fffe9f3c22_btperc	3	Zigbee2MQTT - 0x60a423fffe9f3c22 (Battery)	General	Percentage	0%
<input type="checkbox"/>	5	Zigbee2MQTT	0x60a423fffe9f3c22_mode	4	Zigbee2MQTT - 0x60a423fffe9f3c22 (Mode)	Light/Switch	Selector Switch	On
<input type="checkbox"/>	6	Zigbee2MQTT	0x60a423fffe9f3c22_preset	5	Zigbee2MQTT - 0x60a423fffe9f3c22 (Preset)	Light/Switch	Selector Switch	On
<input type="checkbox"/>	7	Zigbee2MQTT	0x60a423fffe9f3c22_week	6	Zigbee2MQTT - 0x60a423fffe9f3c22 (Week Format)	Light/Switch	Selector Switch	Off
<input type="checkbox"/>	8	Zigbee2MQTT	0x60a423fffe9f3c22_spoint	7	Zigbee2MQTT - 0x60a423fffe9f3c22 (Setpoint)	Thermostat	SetPoint	0.0
<input type="checkbox"/>	9	Zigbee2MQTT	0x60a423fffe9f3c22_sp_eco	8	Zigbee2MQTT - 0x60a423fffe9f3c22 (Eco Setpoint)	Thermostat	SetPoint	15.0
<input type="checkbox"/>	10	Zigbee2MQTT	0x60a423fffe9f3c22_sp_cmf	9	Zigbee2MQTT - 0x60a423fffe9f3c22 (Comfort Setpoint)	Thermostat	SetPoint	20.0
<input type="checkbox"/>	11	Zigbee2MQTT	0x60a423fffe9f3c22_temp	10	Zigbee2MQTT - 0x60a423fffe9f3c22 (Temperature)	Temp	LaCrosse TX3	23.5 C
<input type="checkbox"/>	12	Zigbee2MQTT	0x60a423fffe9f3c22_level	11	Zigbee2MQTT - 0x60a423fffe9f3c22 (Valve position)	Light/Switch	Switch	Off
<input type="checkbox"/>	13	Zigbee2MQTT	0x60a423fffe9f3c22_wnd	12	Zigbee2MQTT - 0x60a423fffe9f3c22 (Window Detection)	Light/Switch	Switch	Off
<input type="checkbox"/>	14	Zigbee2MQTT	0x60a423fffe9f3c22_child	13	Zigbee2MQTT - 0x60a423fffe9f3c22 (Child Lock)	Light/Switch	Switch	Off

Rysunek 5.5-15 Lista czujników udostępnianych przez termostat Tuya Źródło. Opracowanie własne.

Podgląd stanu pluginu dostępny jest w zakładce „custom” pojawi się panel zarządzający usługą Custom > Zigbee2MQTT:



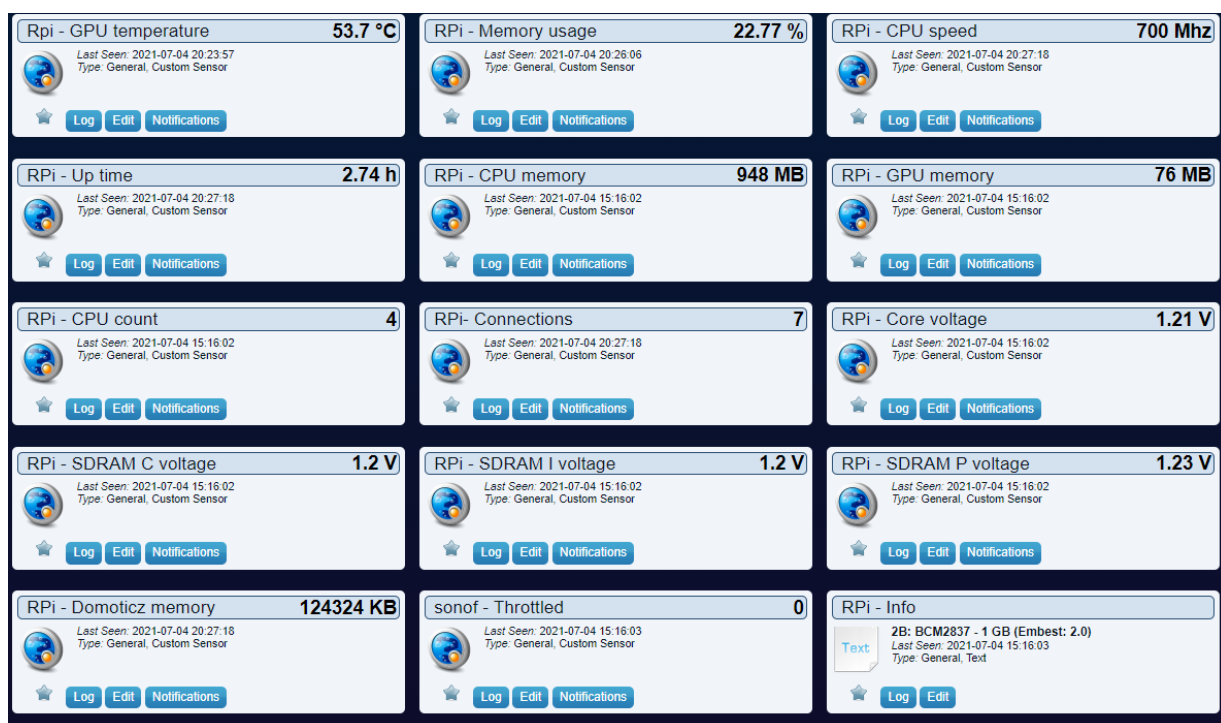
Rysunek 5.5-16 Interfejs pluginu domoticz--zigbee2mqtt w systemie domoticz. Źródło. Opracowanie własne.

W kolejnych krokach można wykorzystać mechanizmy automatyzacji do zarządzania urządzeniem ZigBee

5.6. Pluginy do monitorowania urządzeń i sieci:

Domoticz-PiMonitor-Plugin

Zasługującym na uwagę dodatkiem do Domoticz jest plugin PiMonitor który umożliwia monitorowanie parametrów Raspberry PI z poziomu Domoticz oraz pozwala na reagowanie i powiadamianie po przekroczeniu progów [35]. Korzystając z PP-manager można dodać plugin PiMonitor. Po instalacji pluginu jest możliwość stworzenia instancji urządzenia skojarzonym z tym dodatkiem, aby to zrobić w zakładce „Hardware” trzeba dodać urządzenie Typu PiMonitor. Po dodaniu urządzenia w zakładce Setup > Device powinny pojawić się czujniki które przedstawiają aktualne wartości parametrów RPI: Każdy z wpisów traktowany jest jako osobna encja, którą można spersonalizować, włączyć dla niej powiadomienia lub obserwować jej zmiany na przestrzeni czasu i użyć ich jako wyzwalaczy w skryptach. Domyślnie wszystkie encje trafiają do zakładki Utilities



Rysunek 5.6-1 Czujniki dostępne w ramach pluginu PiMonitor Źródło. Opracowanie własne.

Mikrotik RouterOS plugin

Kolejną ciekawą pozycją do monitorowania sieci jest plugin dedykowany dla systemu RouterOS. Najczęściej system ten spotyka się na urządzeniach firmy Mikrotik, jednak można zainstalować sam system na dowolnym sprzęcie. [36] Plugin łączy się przez API które udostępnia system. Standardowo w przypadku instalacji rozszerzeń w przypadku systemu Domoticz są dwie metody ręczna lub za pomocą PP-Manager. Z poziomu routera w pierwszej kolejności trzeba sprawdzić czy obsługa API jest włączona oraz stworzyć użytkownika z odpowiednimi uprawnieniami. Warto też sprawdzić czy reguły firewall pozwalają na przesyłanie danych przez API. Po Instalacji pluginu standardową procedurą jest dodanie urządzenia z typem zgodnym dla pluginu, czyli „Mikrotik RouterOS” Należy podać login i hasło użytkownika mającego uprawnienia do odczytywania API, port, który będzie monitorowany (domyślnie WAN to ether1) oraz interwał odświeżania. Nie ma przeszkód, aby dodać wszystkie pozostałe porty (fizyczne lub wirtualne) jako osobne urządzenia jednak na potrzeby pokazu zostanie dodanie wyłącznie jeden interfejs „ether1”.

Enabled: ☒

Name:

Type:

Log Level: ☒ Info ☒ Status ☒ Error

Data Timeout:
 Specifying a Data Timeout will restart the hardware device if no data is received for the specified time.
Do not enable this option for devices that do not receive data!

Wiki URL: <https://github.com/mrin/domoticz-routeros-plugin>

IP address:

API Port:

API Username:

API Password:

Update interval (sec):

Bandwidth Interface:

Status Interface:

Debug: ☒ True

Rysunek 5.6-2 Konfiguracja pluginu Mikrotik Router OS Źródło. Opracowanie własne.

W przypadku rozwiązywania problemów przydatna jest funkcja debug która z logach zapisuje komunikaty związane z działaniem pluginu

```
(mt3) Calling message handler 'onHeartbeat'.
(mt3) Pushing 'WriteDirective' on to queue
) Processing 'WriteDirective' message
(mt3) Sending 113 bytes of data
(mt3) 1a 2f 69 6e 74 65 72 66 61 63 65 2f 6d 6f 6e 69 74 6f 72 2d ./interface/monitor-
(mt3) 74 72 61 66 66 69 63 11 3d 69 6e 74 65 72 66 61 63 65 3d 65 traffic.=interface=
(mt3) 74 68 65 72 31 09 3d 6f 6e 63 65 3d 79 65 73 30 3d 2e 70 72 ther1.=once=yes0=.pr
(mt3) 6f 70 6c 69 73 74 3d 72 78 2d 62 69 74 73 2d 70 65 72 2d 73 oplist=rx-bits-per-s
(mt3) 65 63 6f 6e 64 2c 74 78 2d 62 69 74 73 2d 70 65 72 2d 73 65 econd,tx-bits-per-se
(mt3) 63 6f 6e 64 07 2e 74 61 67 3d 62 77 00 .. .. .. cond..tag=bw.
(mt3) Pushing 'onHeartbeatCallback' on to queue
) Processing 'onHeartbeatCallback' message
```

Rysunek 5.6-3 Logi pluginu Mikrotik Router OS w trybie „debug” Źródło. Opracowanie własne.

Po poprawnym dodaniu urządzeń pojawią się one w zakładce „utility” wyświetlając aktualne wartości. W Zakładce przełączniki pojawi się natomiast przełącznik, który pozwala na zarządzanie stanem interfejsu

Mikrotik - Bandwidth UP
0.01 Mbit/s
Last Seen: 2021-07-04 20:27:23
Type: General, Custom Sensor
Log Edit Notifications

Mikrotik - Bandwidth Down
0.01 Mbit/s
Last Seen: 2021-07-04 20:46:28
Type: General, Custom Sensor
Log Edit Notifications

Rysunek 5.6-4 Dostępne encje w wykorzystaniem pluginu Mikrotik Router OS Źródło. Opracowanie własne.

Podobnie jak w przypadku innych encji możliwe jest wyświetlanie historii zużycia i wykonywanie skryptów w zależności o aktualnych wartości

SNMPReader

Uniwersalnym narzędziem do badania aktywności urządzeń połączonych do sieci jest protokół SNMP który umożliwia odczyt (oraz zmianę) wartości udostępnionych przez urządzenia danych zakodowanych w lokalizacjach MIB¹⁷ w ramach pojedynczych zmiennych nazwanych OID¹⁸. Aktualnie istnieje kilka wersji SNMP 1, 2(2c) i 3 późniejsze wersje zawierają lepsze mechanizmy uwierzytelniania jednak co do zasady każda wersja SNMP może korzystać z podstawowych parametrów które są obsługiwane przez plugin SNMP Reader [37]. Aby SNMP Reader działał poprawnie należy zainstalować pakiet w systemie, na którym pracuje Domoticz „snmpd” który zadziała w tle jako usługa, oraz dodać plugin SNMP np. poprzez PP-Manager. Przy dodaniu urządzeń do wyboru powinien być dostępny typ SNMP Value Reader.

Podstawowa konfiguracja odbywa się przez dostarczenie informacji takich jak:

- adres IP serwera SNMP,
- OID (unikalnego identyfikatora wskazującego na konkretny parametr urządzenia np. 1.3.6.1.2.1.25.3.3.1.2.7 – zużycie procesora w procentach w urządzeniach mikrotik uśrednione dla 1 minuty. Zużycie procesora w systemie Windows badane przez funkcje agenta SNMP 1.3.6.1.2.1.25.3.3.1.2.7
- „community” (tzw. Grupa przynależąca urządzenia),
- interwał odświeżania.

¹⁷ MIB (ang. Management Information Base) – Baza danych agenta SNMP przechowująca wartości OID

¹⁸ OID (ang. Object Identifier) - Pojedyncza zmienna w bazie MIB, przechowująca wartość aktualną badanego parametru np. temperature

Enabled: ☒

Name: Mikrotik CPU AVG

Type: SNMP Value Reader

Log Level: ☒ Info ☒ Status ☒ Error

Data Timeout: Disabled
Specifying a Data Timeout will restart the hardware device if no data is received for the specified time.
Do not enable this option for devices that do not receive data!

Wiki URL: [m](#)

Product URL: <https://www.domoticz.com/forum/viewtopic.php?f=65>

Server IP: 192.168.8.194

OID: 1.3.6.1.4.1.2021.11.10.0

Community: public

Check Interval(seconds): 5

Domoticz TypeName: Custom

Debug: True

Rysunek 5.6-5 Konfiguracja plugin SNMP Reader w Domoticz Źródło. Opracowanie własne.

Podobnie jak w przypadku poprzednich pluginów opcja „debug” może być przydatna w przypadku rozwiązywania problemów

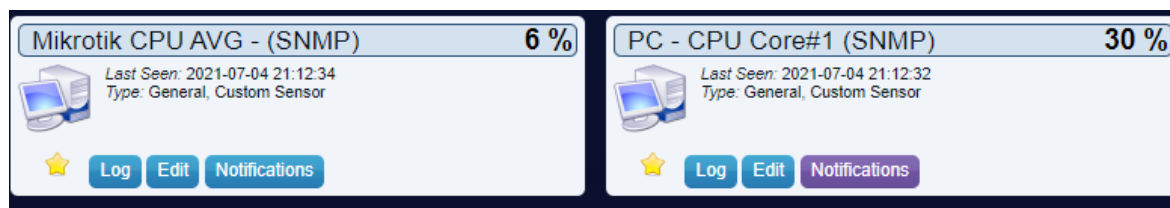
```

2021-07-04 21:06:31.744 Error: Mikrotik CPU AVG: (Mikrotik CPU AVG) No SNMP response received before timeout
2021-07-04 21:06:35.130 Error: mt3: (mt3) Mikrotik connection error. Status [113] [No route to host]
2021-07-04 21:06:42.608 Error: Mikrotik CPU AVG: (Mikrotik CPU AVG) No SNMP response received before timeout
2021-07-04 21:06:47.671 Error: mt3: (mt3) Mikrotik connection error. Status [113] [No route to host]
2021-07-04 21:06:54.160 Status: Mikrotik CPU AVG: (Mikrotik CPU AVG) Stop directive received.
2021-07-04 21:06:56.018 Status: Mikrotik CPU AVG: (Mikrotik CPU AVG) Exiting work loop.
2021-07-04 21:06:56.062 Status: Mikrotik CPU AVG: (Mikrotik CPU AVG) Stopping threads.
2021-07-04 21:06:56.062 Status: Mikrotik CPU AVG: (Mikrotik CPU AVG) Stopped.
2021-07-04 21:06:56.062 Status: Mikrotik CPU AVG: (Mikrotik CPU AVG) Entering work loop.
2021-07-04 21:06:56.062 Status: Mikrotik CPU AVG: (Mikrotik CPU AVG) Started.
2021-07-04 21:07:01.678 Status: Mikrotik CPU AVG: (Mikrotik CPU AVG) Initialized version 1.1.0, author 'ycachome'
2021-07-04 21:08:43.063 Mikrotik CPU AVG: (Mikrotik CPU AVG) SNMP Value (192.168.8.194/public/1.3.6.1.4.1.2021.11.10.0) retrieved:7
2021-07-04 21:08:46.826 (Mikrotik - Bandwidth Down) Updating device from 1:'0.01' to have values 1:'0.03'.
2021-07-04 21:08:48.049 Mikrotik CPU AVG: (Mikrotik CPU AVG) SNMP Value (192.168.8.194/public/1.3.6.1.4.1.2021.11.10.0) retrieved:7
2021-07-04 21:08:51.877 (Mikrotik - Bandwidth Down) Updating device from 1:'0.03' to have values 1:'0.01'.

```

Rysunek 5.6-6 Logi pluginu SNMP Reader w trybie debug Źródło. Opracowanie własne.

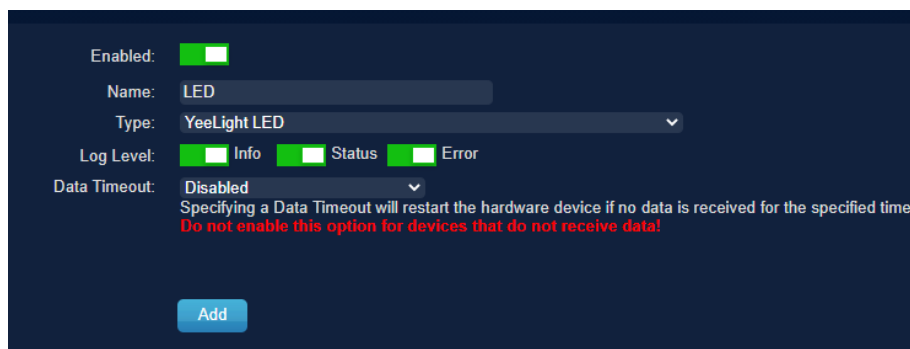
Po poprawnej konfiguracji w zakładce „utility” jest możliwość monitorowania stanu urządzeń co kilka sekund w zależności od ustawionego interwału



Rysunek 5.6-7 Encje dodane przez plugin SNMP Reader w Domoticz Źródło. Opracowanie własne.

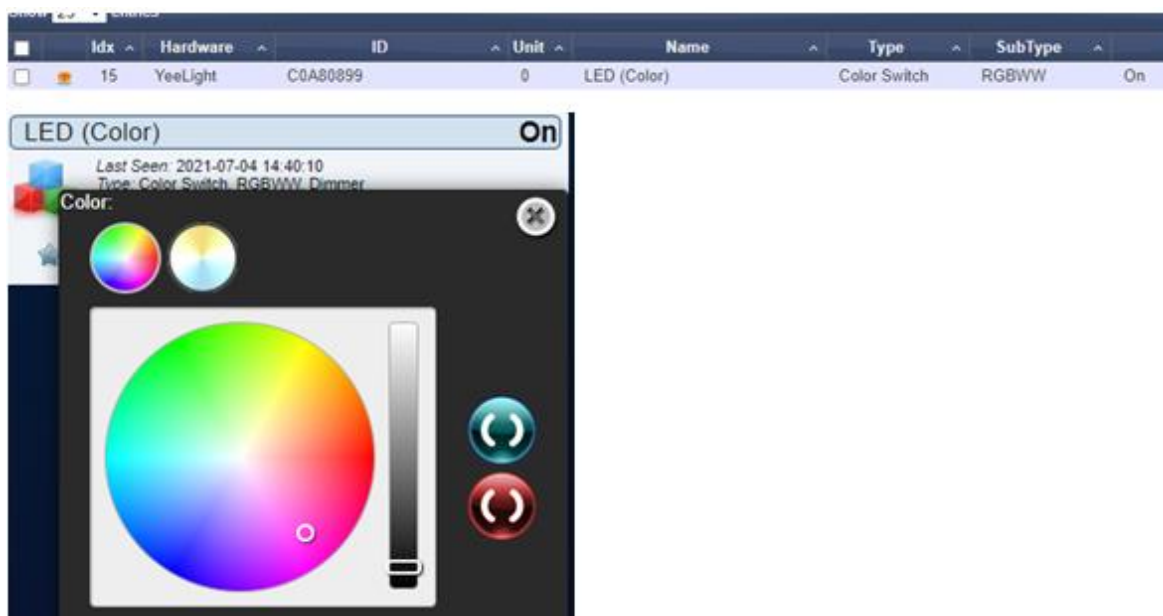
5.7. Dołączanie i konfiguracja kolejny urządzeń

Dołączanie kompatybilnych urządzeń odbywa się według ustalonego schematu. W zakładce Setup > Hardware należy dodać urządzenie z poziomu dostępnego formularza. Np. aby dodać Żarówkę LED Yeelight Smart Bulb 1S konfiguracja będzie wyglądać następująco:



Rysunek 5.7-1 Dodanie nowego urządzenia w Domoticz. Opracowanie własne. Źródło: Opracowanie własne.

Jeżeli żarówka jest już podłączona do lokalnej sieci WiFi domoticz powinien ją wykryć automatycznie, rozpoznać jej typ i dodać na liście Setup > Devices oraz w zakładce „Switches”

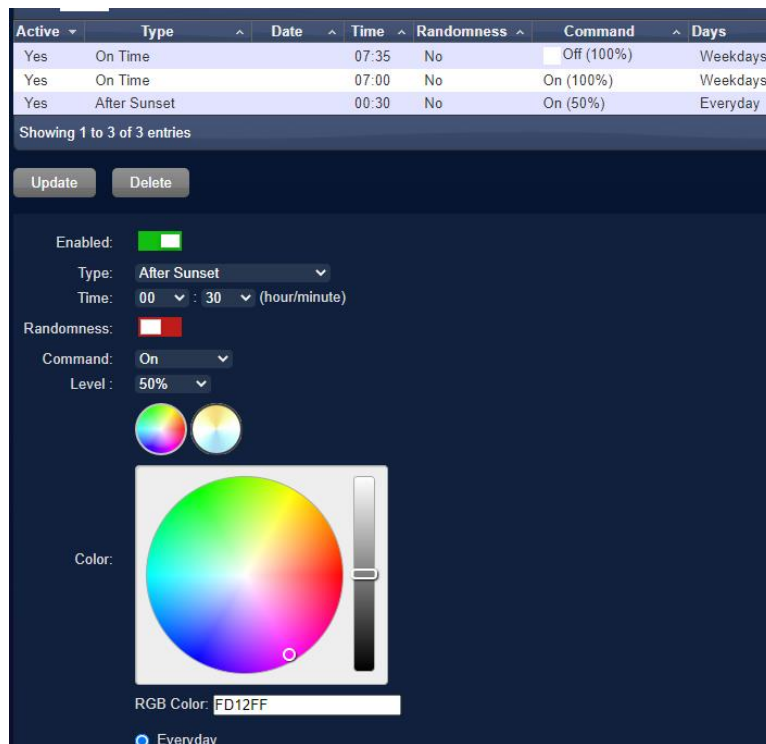


Rysunek 5.7-2 Widok nowo dodanego urządzenia w systemie domoticz. Źródło: Opracowanie własne.

Do możliwości konfiguracji pozostają jeszcze sceny urządzenia, gdzie można ustawić harmonogram zachowania urządzenia:

W poniższym przykładzie zostały ustawione 3 reguły:

- światło włącza się w dni robocze o 7:00 z mocą 100%,
- światło wyłącza się w dni robocze o 7:35,
- światło włącza się 30 minut po zachodzie słońca z mocą 50% i wskazanym kolorze.

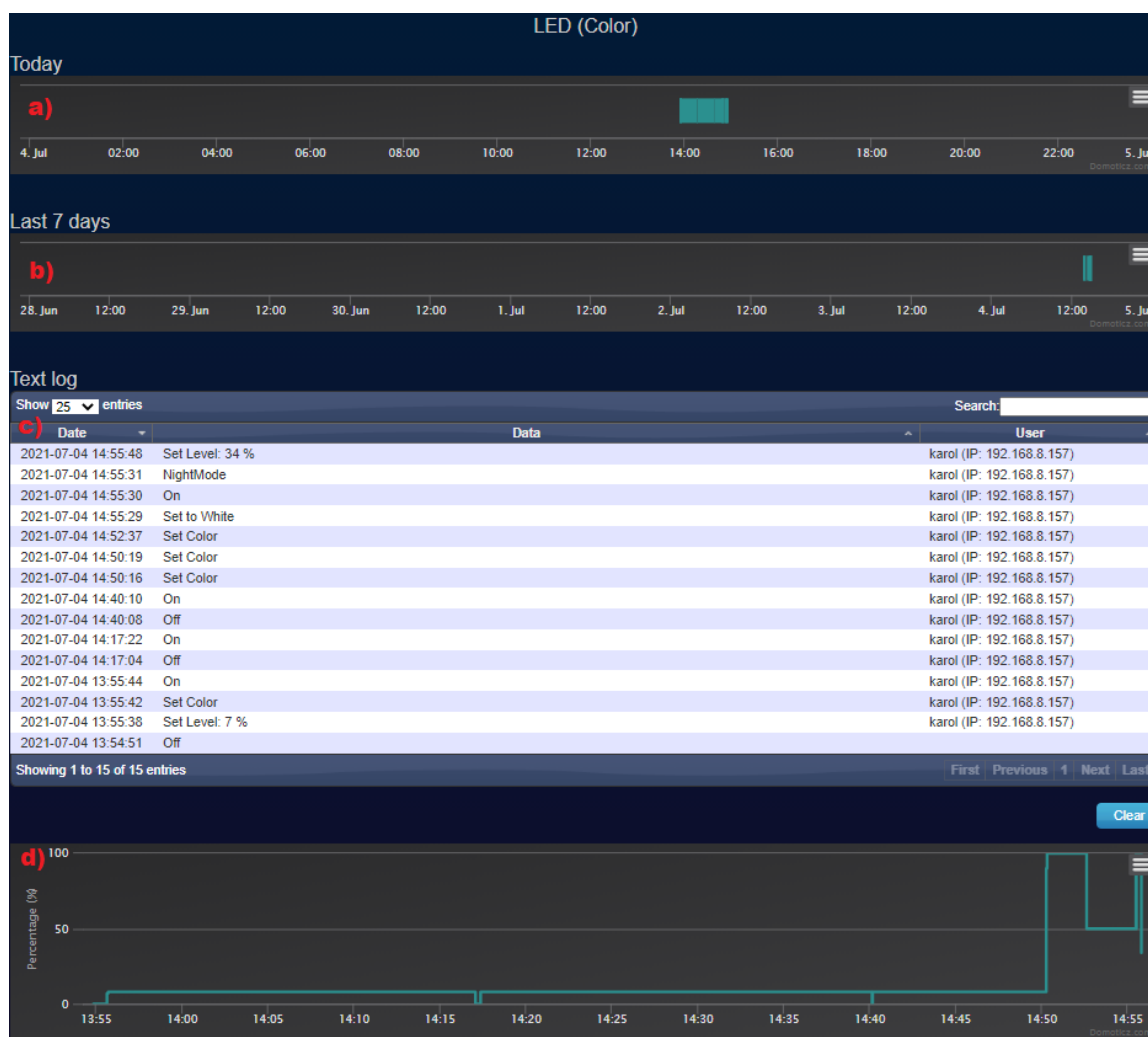


Rysunek 5.7-3 Konfiguracja scen urządzenia. Źródło: Opracowanie własne.

Aktywność urządzenia można obserwować przechodząc do jego logów np. z poziomu „Switches” > <nazwa urządzenia> > Log:

Logi urządzeń podzielone są na sekcje:

- aktywność dzisiaj,
- aktywność przez ostatnie 7 dni,
- tekstowy dziennik zmian stanu,
- wykres zmian wartości (możliwy do pobrania w formacie obrazka):



Rysunek 5.7-4 Logi urządzenia dostępne w Domoticz. Źródło: Opracowanie własne.

5.8. Prezentacja funkcjonalności

Dostęp zdalny

Dostęp zdalny w przypadku lokalnych sieci domowych może być realizowany na wiele sposobów, jednak zazwyczaj wiąże się to udostępnianiem zasobów do sieci Internet. Aby było to możliwe łącze internetowe z którego korzystamy musi posiadać zewnętrzny publiczny adres IP (najlepiej stały) i odpowiednia konfigurację na routerze brzegowym. Można odpłatnie zlecić uruchomienie usługi stałego IP w ramach subskrypcji operatorowi, jednak nie wszyscy operatorzy w Polsce taką usługę oferują. Alternatywą jest korzystanie z bram VPN które mogą pełnić rolę pośrednika pomiędzy usługami znajdującymi się w sieci lokalnej a Internetem. Domoticz oferuje dedykowaną usługę pośredniczącą w udostępnianiu lokalnego serwera Domoticz do Internetu o nazwie „MyDomoticz”. Dodanie własnej instancji Domoticz w ramach serwisu wymaga konta premium które można uzyskać po dokonaniu jednorazowej opłaty w ramach dotacji autora. Minimalna kwota to

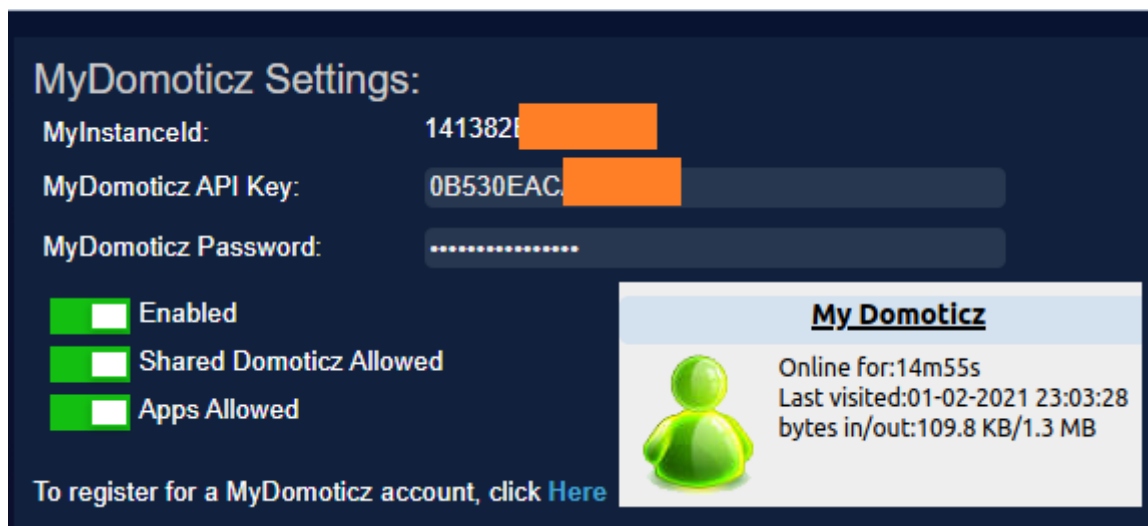
1\$. Po aktywacji konta premium Konfiguracja dostępu zdalnego z użyciem MyDomoticz składa się z kilku prostych etapów:

- włączenie dostępu zdalnego z panelu zarządzającego Domoticz,
- utworzenie konta w serwisie MyDomoticz w celu pozyskania ID użytkownika,
- dodanie instalacji lokalnego Domoticz do MyDomoticz,
- (opcjonalnie) dodanie domyślnego loginu i hasła do panelu,
- (opcjonalnie) ustawienie dwuskładnikowego uwierzytelniania,
- (opcjonalnie) udostępnienie innym dostępu innym użytkownikom.

Po wykonaniu powyższych czynności można podłączyć się do lokalnej instancji Domoticz z dowolnego miejsca na świecie: za pomocą adresu:

http://my.domoticz.com/my/<id_instancji_domoticz>

Zielona ikona po zalogowaniu się do serwisu oznacza, że usługa działa poprawnie:



Rysunek 5.8-1 Konfiguracja i Status połączenia zdalnego. Źródło: Opracowanie własne.

Powiadomienia Push (iOS / Android)

Domoticz udostępnia kilka form wysyłania powiadomień na urządzenia mobilne min. za pomocą aplikacji serwisów zewnętrznych które wykorzystują własne serwery w chmurze. Oznacza to, że mimo iż pomimo braku udostępnionych usług Domoticz w sieci Internet, jest możliwość otrzymywania powiadomień z systemu lokalnego z każdego miejsca na świecie za pośrednictwem usług zewnętrznych. Na potrzeby prezentacji zostanie wykorzystany serwis pushbullet który udostępnia możliwość sparowania usługi powiadomień za pomocą klucz API. Aby stworzyć unikalny klucz dla serwisu i zacząć korzystać z usługi należy stworzyć konto w serwisie pushbullet. [38] W ramach

darmowego konta jest do dyspozycji 500 wywołań API tworzących powiadomienia miesięcznie [39]. Po utworzeniu konta należy wygenerować unikalny token dostępowy w zakładce Ustawienia > Konto:

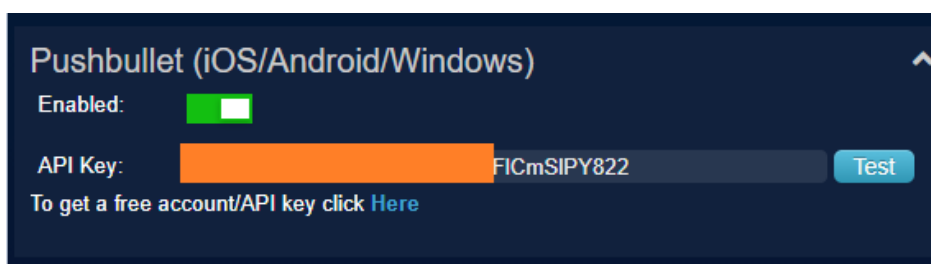
Access Tokens

Using an access token grants full access to your account. Don't share this lightly. You need the access token in order to use the [API](#).

Create Access Token

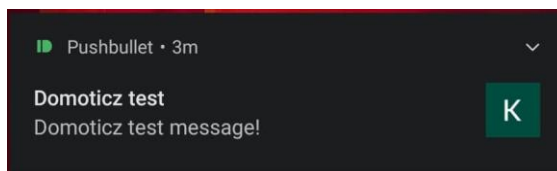
Rysunek 5.8-2 Tworzenie tokenu dostępowego w serwisie Pushbullet Źródło: Opracowanie własne.

Po utworzeniu tokena należy zapisać go w ustawieniach domoticz w zakładce Setup > Settings > Notifications i włączyć usługę:



Rysunek 5.8-3 Konfiguracja usługi Pushbullet w Domoticz. Źródło: Opracowanie własne.

Klikając przycisk test możemy przetestować działanie powiadomień na urządzeniach z zalogowaną wersją kliencką jeżeli wszystko udało się poprawnie to na urządzenie mobilne powinien przyjść komunikat:



Rysunek 5.8-4 Test usługi Pushbullet w Domoticz. Źródło: Opracowanie własne.

Dodanie powiadomień skojarzonych z akcją urządzeń odbywa się np. przez konfigurację w zakładce „switches” gdzie dla każdej dostępnej akcji urządzenia można wygenerować powiadomienie o zdefiniowanej wcześniej treści np. stworzenie powiadomienia o uruchomieniu urządzenia „LED” można skonfigurować wybierając:

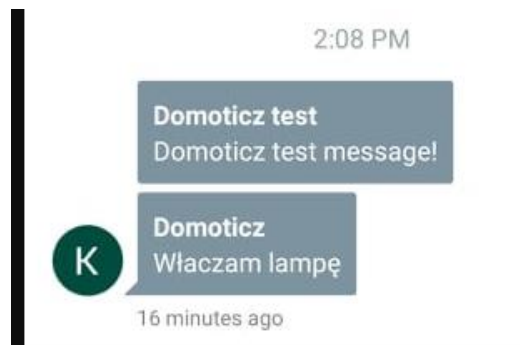
- „type” : Switch ON,
- „custom message” : „ Włączam lampę”,
- „active systems:” : pushbullet > ON,

Efekt działania i wywołania powiadomienia można zauważyć w logach Setup > Log

```
Status: User: karol (IP: 192.168.8.157) initiated a switch command (15/LED (Color)/On)
Status: Notification: Włączam lampę
Notification sent (pushbullet) => Success
```

Rysunek 5.8-5 Logi związane z usługą Pushbullet w Domoticz. Źródło: Opracowanie własne.

Aplikacja pushbullet na urządzeniach mobilnych przechowuje listę powiadomień co może stanowić dziennik zdarzeń który może pomóc w przypadku potrzeby analizy minonych zdarzeń:



Rysunek 5.8-6 Historia powiadomień w serwisie Pushbullet. Źródło: Opracowanie własne.

Powiadomienia Mail

W podobny sposób jak przypadku powiadomień push jest możliwość skonfigurowania skrzynki pocztowej na które będą trafiać powiadomienia, alerty o błędach, lub zrzuty z podglądu kamer. Możliwość dodania skrzynki pocztowej znajduje się w opcjach systemu: Setup > Settings > Mail.

Aby skonfigurować serwis wysyłający maile wystarczy podać dane skrzynki pocztowej nadawczej czyli:

- adres mail,
- serwer poczty email,
- port,
- login i hasło.

Rysunek 5.8-7 Konfiguracja powiadomień E-mail w Domoticz. Źródło: Opracowanie własne.

Po zapisaniu ustawień można przetestować działanie i sprawdzić efekt w logach:

```
Status: Notification: Domoticz test
Notification sent (email) => Success
Status: User: karol (IP: 192.168.8.157) initiated a switch command (15/LED (Color)/On)
Status: Notification: Włączam lampę
Notification sent (email) => Success
Notification sent (pushbullet) => Success
```

Rysunek 5.8-8 Logi powiadomień E-mail w Domoticz. Źródło: Opracowanie własne.

Oraz sprawdzić skrzynkę odbiorczą odbiorcy:

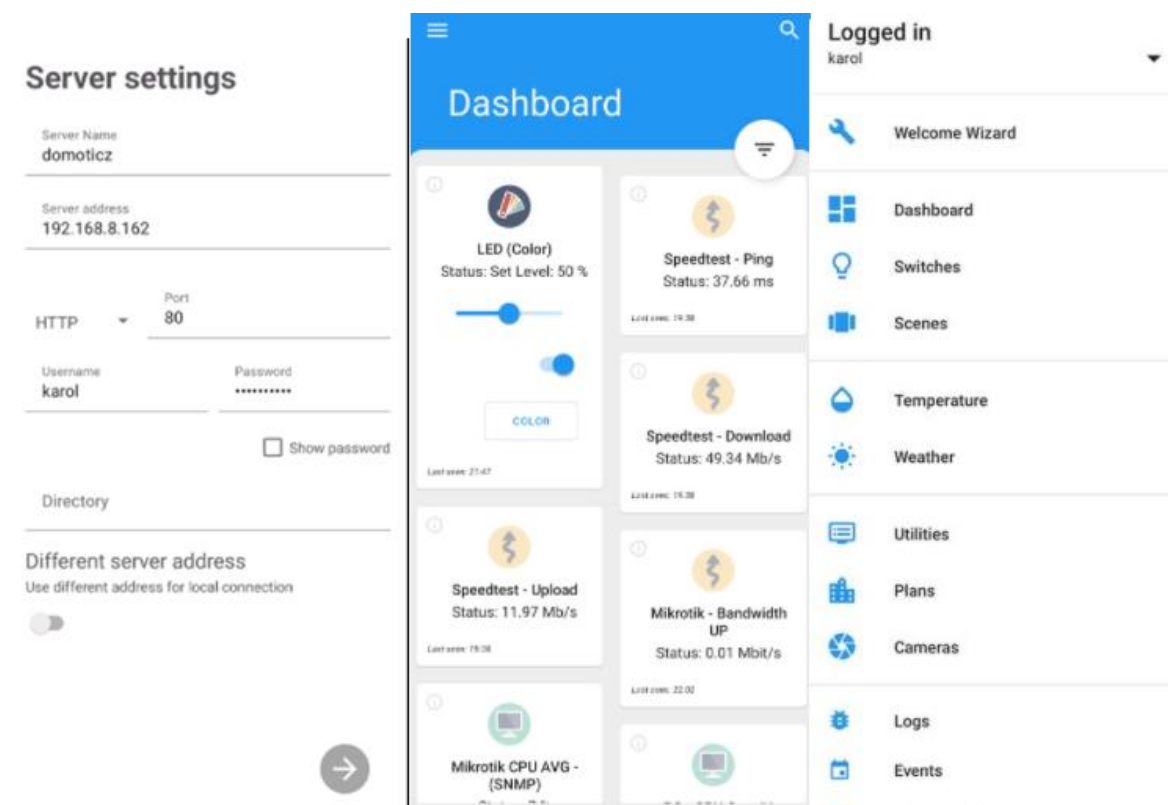


Rysunek 5.8-9 Test powiadomień E-mail w Domoticz. Źródło: Opracowanie własne.

Aplikacja mobilna:

W sklepie „Play Store” dostępna jest oficjalna aplikacja kliencka dla systemów Android Domoticz Lite które oferuje możliwość sterowania za pośrednictwem urządzeń mobilnych bezpośrednio w aplikacji. Aktualna wersja aplikacji to: 0.2.3. Konfiguracja odbywa się w celu autoryzacji urządzenia mobilnego do korzystania z serwera. W podstawowym wariantcie potrzebne dane to:

- adres serwera,
- usługa i port: (http lub HTTPS),
- nazwa użytkownika i hasło.



Rysunek 5.8-10 Konfiguracja i wygląd aplikacji Domoticz Lite na systemy Android. Źródło: Opracowanie własne.

Aplikacja pełni rolę klienta i w większości funkcjonalności działa w trybie tylko do odczytu. Dostępny jest podgląd na żywo do kamer jak i podgląd statystyk gromadzonych przez urządzenia. Wygląd panelu można dostosować, rozmieszczając ikony według uznania. Aplikacja ma podział na kategorie zbliżone do wersji webowej. W płatnej wersji dostępne są dodatkowe funkcjonalności opisane w rozdziale 4. Z poziomu serwera można zarządzać sparowanymi aplikacjami lub przypisywać im unikalną nazwę:



Rysunek 5.8-11 Zarządzanie dostępem przez aplikację w Domoticz. Źródło: Opracowanie własne.

5.9. Przykłady Automatyzacji:

Powiadamianie o braku aktywności urządzeń

Zakładając że urządzenia podczas pracy zawsze wykorzystują min. 1% procesora i nigdy nie zwracają wartości 0 można zastosować automatyzację która będzie powiadamiać użytkownika gdy zużycie będzie równe 0. Do automatyzacji można wykorzystać wcześniej

dodane dwie pozycje z pluginu SNMP Reader Aby dodać powiadomienie Push i mailowe wystarczy dodać w encji urządzenia regułę powiadomień.

W przykładzie przedstawiono jak z wykorzystaniem encji „PC-CPU Core#1 (SNMP)” Utworzono regułę która będzie powiadamiać po spełnieniu warunku:

- typ: „usage”;
- warunek: wartość równa 0,
- wiadomość: „Agent SNMP nie odpowiada, brak internetu lub komputer nie odpowiada”.

Name: PC - CPU Core#1 (SNMP)

Show 25 entries

Type	When	Active Systems	Custom Message	Priority	Ignore Interval	Recovery
Usage	Equal 0	browser,email;pushbullet	Agent SNMP nie odpowiada, brak internetu lub komputer nie odpowiada	Normal	No	No

Showing 1 to 1 of 1 entries 1 row selected

Update Delete Clear

Type: Usage

When: Equal

Value: 0

Priority: Normal

Ignore Interval: ☐ (If enabled, it will bypass the Notification interval specified in the Settings page)

Recovery Notification: ☐ (If enabled, a notification will be send when given condition has been recovered)

Custom Message: Agent SNMP nie odpowiada, brak internetu lub komputer nie (Optional)

☒ browser

☐ clickatell

☒ email

☐ fcm

☐ http

☐ kodi

Active Systems: ☐ lms

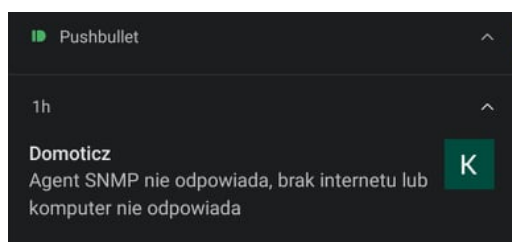
☐ prowl

☐ pushalot

☒ pushbullet

Rysunek 5.9-1 Automatyzacja wykrywania braku aktywności urządzeń. Źródło: Opracowanie własne.

Efekt po spełnieniu warunku powinien być następujący:



Rysunek 5.9-2 Test automatyzacji wykrywania braku aktywności urządzeń. Źródło: Opracowanie własne.

Identyczną regułę ze zmienioną wiadomością można zastosować do encji Mikrotik CPU AVG - (SNMP) aby wykrywać brak aktywności routera (pod warunkiem że dostęp do sieci będzie możliwy bez udziału tego routera).

Sterowanie termostatem na podstawie stanów innych urządzeń

W tym scenariuszu czujnik otwarcia okna gdy kontakt między punktami styku zostanie przerwany zmieni się stan urządzenia „Window sensor” który spowoduje. Wysłanie komendy wyłączenia grzania do termostatu gdy okno zostanie otwarte. Dodatkowo przy zmianie stanu zostanie wysłane powiadomienie push a samo urządzenie będzie reagować ze wskazanym opóźnieniem 30sek. Po wykryciu otwarcia i 5minut po wykryciu zamknięcia. Do zaplanowania i konfiguracji automatyzacji może posłużyć edytor skryptów LUA, dostępny w zakładce Setup > More Options > Events. W ramach skryptu potrzebne będą następujące elementy:

- zagnieżdzone bloki logiczne Jeżeli X1 wykonaj Y oraz jeśli X2 wykonaj Y2
- stany urządzeń (Windows Sensor oraz Termostat)
- akcję ustawienia wartości z wyborem poziomu lub stanu
- akcję wysłania powiadomień

Korzystając z języka skryptowego LUA automatyzację można zrealizować za pomocą poniższego kodu:

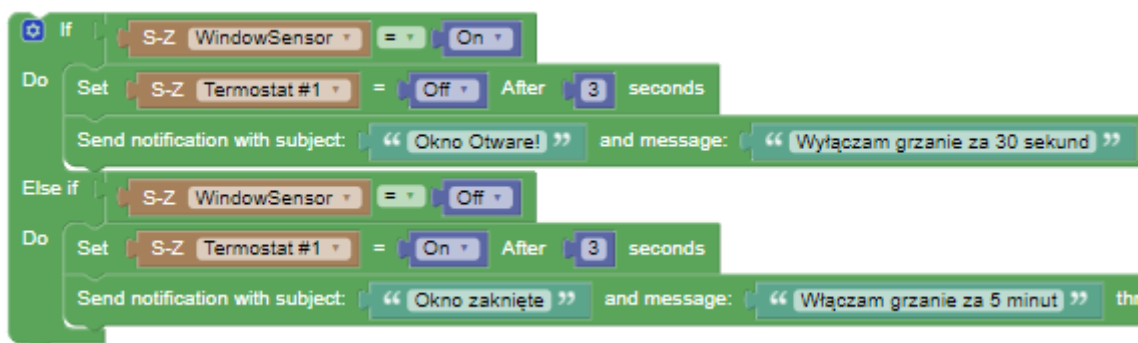
```

1  commandArray = {}
2
3  if otherdevices['WindowSensor'] == 'On'
4      commandArray['Termostat #1'] = 'Off AFTER 30'
5      commandArray['SendNotification']='Okno Otwarte!#Wyłączam grzanie za 30 sekund#0'
6      commandArray['SendEmail']='Okno Otwarte!#Wyłączam grzanie za 30 sekund#domoticz@kkrasuski.com'
7  elseif otherdevices['WindowSensor'] == 'off'
8      commandArray['Termostat #1'] = 'On AFTER 300'
9      commandArray['SendNotification']='Okno zamknięte!#Włączam grzanie za 5minut#0'
10     commandArray['SendEmail']='Okno zamknięte!#Włączam grzanie za 300 sekund#domoticz@kkrasuski.com'
11 end
12
13 return commandArray

```

Rysunek 5.9-3 Skrypt automatyzacji w języku LUA. Źródło: Opracowanie własne.

Tą samą akcję można zobrazować za pomocą kreatora blokowego blocky:



Rysunek 5.9-4 Schemat automatyzacji w blocky. Źródło: Opracowanie własne.

Do uruchomienia skryptu wystarczy zapisanie go i przełączenie przełącznika u góry edytora.

Wywołania skryptu można podejrzeć w logach urządzenia (czas reakcji został zmniejszony do 3 sekund na potrzeby prezentacji wydarzeń)



Date	Data	User
2021-07-04 22:58:22	On	EventSystem/Wyłącz grzanie gdy okno jest otwarte_2
2021-07-04 22:58:15	Off	EventSystem/Wyłącz grzanie gdy okno jest otwarte_1
2021-07-04 22:58:10	On	karol (IP: 192.168.8.150)
2021-07-04 22:54:15	Off	EventSystem/Wyłącz grzanie gdy okno jest otwarte_1
2021-07-04 22:54:13	On	karol (IP: 192.168.8.150)
2021-07-04 22:32:11	Off	karol (IP: 192.168.8.150)

Showing 1 to 6 of 6 entries

First Previous 1 Next Last

Rysunek 5.9-5 Test automatyzacji termostatu Źródło: Opracowanie własne.

Wysyłanie zdjęć z kamery po zmianie statusu urządzenia

Kolejny scenariusz przewiduje sytuacje w której po zmianie stanu czujnika drzwi w godzinach nocnych 23:00 – 5:00 urządzenie typu kamera wykonuje zrzut ekranu, który wysyła na wcześniej zdefiniowany adres mailowy. Kamera IP w ramach tego zadania to smartfon z zainstalowaną aplikacją IP Webcam z uruchomionym serwerem WWW. W tym celu również przydatne będzie narzędzie blocky w którym można zdefiniować potrzebną do tego celu automatyzację.

Aby dodać kamerę należy przejść do zakładki Setup > More options > Camera

Wybrać pozycję Add camera i skonfigurować ustawienia urządzenia:

- nazwę urządzenia,
- protokołów (http lub https) oraz ich porty,
- adres IP Kamry,
- poświadczenia,
- ścieżkę do podglądu.

Rysunek 5.9-6 Konfiguracja kamery w Domoticz. Źródło: Opracowanie własne.

Skrypt automatyzacji który wyśle Email ze zdjęciem i powiadomieniem w przypadku otwarcia drzwi wygląda następująco:

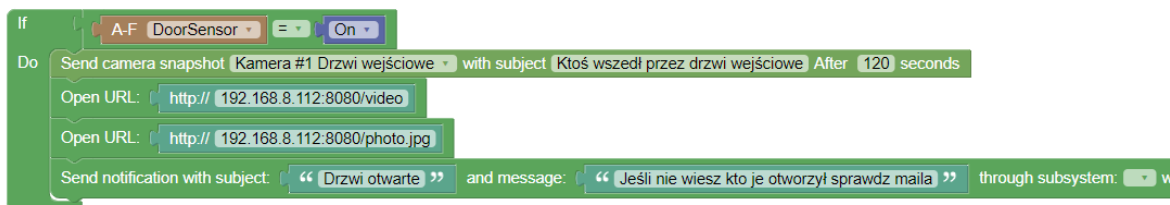
```

1 commandArray = {}
2
3 if otherdevices['DoorSensor'] == 'on'
4   commandArray['SendCamera:1'] = 'Ktoś wszedł przez drzwi wejściowe AFTER 1'
5   commandArray['openURL'] = '192.168.8.112:8080/photo.jpg'
6   commandArray['SendNotification'] = 'Drzwi otwarte!#Jeśli nie wiesz kto je otworzył sprawdź maila#0'
7   commandArray['SendEmail'] = 'Okno otwarte!#Wyłączam grzanie za 30 sekund#domoticz@kkrasuski.com'
8 end
9
10 return commandArray

```

Rysunek 5.9-7 Schemat automatyzacji w LUA wysyłający maile z zrzutem ekranu kamery. Źródło: Opracowanie własne.

Tą samą akcje można wykonać za pomocą bloków z użyciem blocky



Rysunek 5.9-8 Schemat automatyzacji w blocky wysyłający maile. Źródło: Opracowanie własne.

Pożądany efekt po zmianie stanu czujnika drzwi na otwarty (on):



Rysunek 5.9-9 Test wysyłania maili przez automatyzację. Źródło: Opracowanie własne.

5.10. Możliwości rozwoju projektu

Rozpatrując aspekty rozwoju projektu można wziąć pod uwagę następujące elementy:

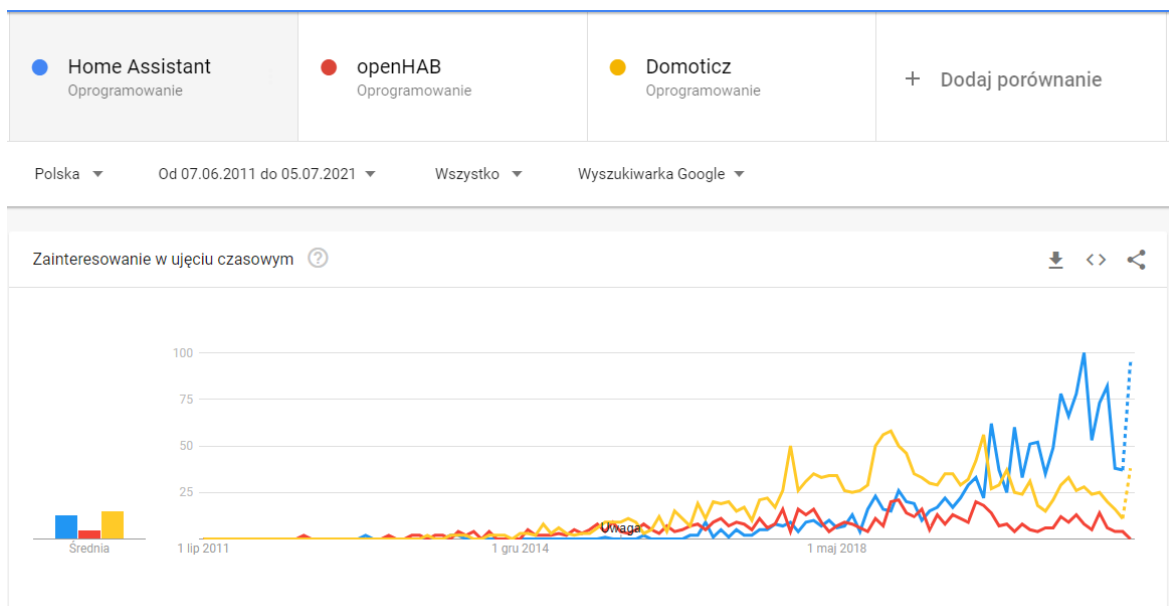
- instalacja dodatkowych pluginów zwiększających liczbę dostępnych serwisów do integracji,
- instalacja dodatkowych pluginów zwiększających wachlarz dostępnych urządzeń,
- skonfigurowanie bardziej zaawansowane skryptów automatyzujących np. całkowicie zautomatyzować prace termostatu przez cały tydzień oraz zdefiniować dodatkowe wyjątki,
- komercjalizacja projektu, zaplanowanie przypadków użycia, scenariusza wdrożenia u klienta końcowego oraz przygotowanie gotowego zestawu czujników i urządzeń z gotowym obrazem zawierającym konfigurację,
- wykorzystanie Raspberry PI i jego system operacyjny do uruchomienia dodatkowych usług na przykład:
 - Docker z aplikacją Portainer (lub podobną) która pełniłaby funkcję nadzorca nad dodatkowymi serwisami [40],
 - Kodi który mógłby pełnić rolę centrum multimedialnym [41],
 - PiHole Dedykowany lokalny serwer DNS blokujący reklamy [42].

Rozdział 6

Zakończenie

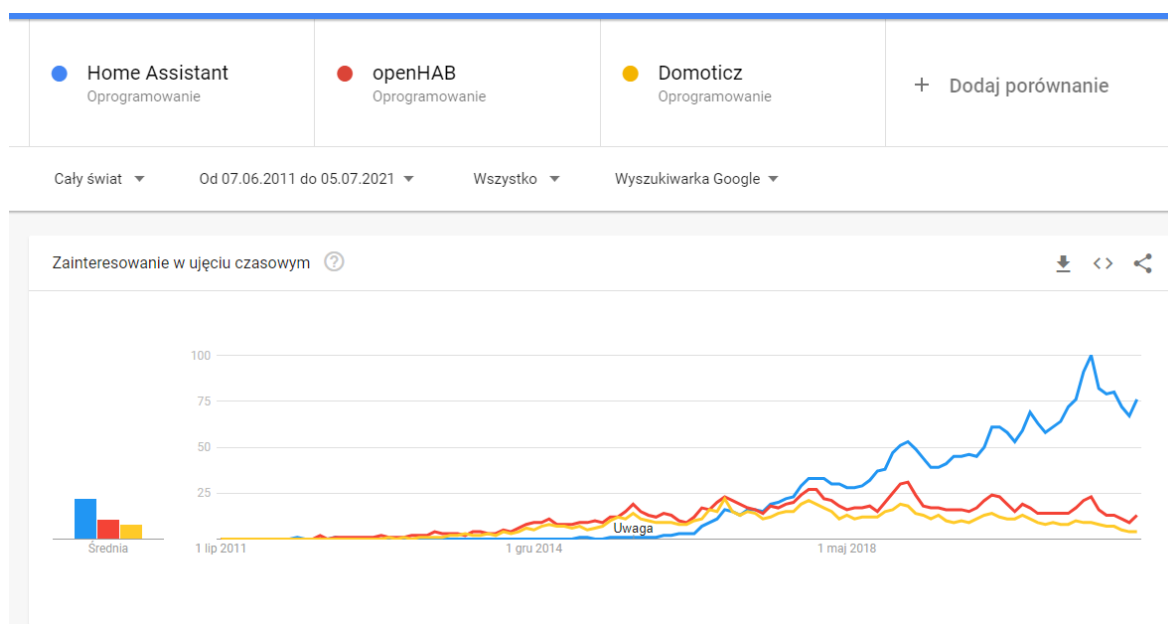
Realizowany w niniejszej pracy temat dotyczący systemu automatyki domowej to szerokie pojęcie składające się z wielu mniejszych komponentów. Dowodem na to jest mnogość przedstawionych w części teoretycznej rozwiązań pozwalających na automatyzację w mniejszym lub większym stopniu codziennych rutyn. Zgodnie z tym co zostało przedstawione w części teoretycznej istnieje spora konkurencja na rynku komercyjnych rozwiązań automatyki domowej, jednak do rozpoczęcia korzystania z dobrodziejstw płynących z takiego systemu nierzadko trzeba na początek zainwestować sporą ilość pieniędzy. W ramach hobbystycznego rozwoju i poszerzenia wiedzy z zakresu IoT warto pochylić się nad platformą Raspberry Pi która za niewielką cenę może stanowić fundament pod wielozadaniowy system automatyki domowej, dodatkowo nieograniczony produktami wyłącznie jednego producenta. Dowodem na to stwierdzenie jest część praktyczna w której to w formie prezentacji na przykładzie platformy Domoticz zostały przedstawione najważniejsze aspekty związane z mechanizmami automatyzacji i konfiguracji urządzeń z rodziny Internetu Rzeczy. Oczywiście możliwości jest dużo więcej i nie wszystkie mogłyby być w ramach jednego rozdziału przedstawione, niemniej jednak na podstawie przytoczonych przykładów wniosek jaki należy wyciągnąć to fakt, że korzystając z otwartego oprogramowania istnieje możliwość rozbudowy funkcjonalności bez względu na ograniczenia nałożone przez producentów gotowych rozwiązań. Warto też mieć na uwadze, że w przypadku otwartego oprogramowania można natknąć się na niedoskonałości lub braki w dokumentacji, jednak moim zdaniem rekompensuje to bogata społeczność chętna do dzielenia się wiedzą i chęcią pomocy. Część usług oferowanych w ramach darmowych platform jest z ekonomicznych względów płatna, ponieważ za tymi usługami stoją ludzie inwestujący w swoje rozwiązania czas i pieniądze, którzy są odpowiedzialni za utrzymanie infrastruktury i dbanie o nieprzerwane działanie produktu. Właśnie z tego względu myślę, że warto docenić prace takich ludzi, ponieważ zastrzyk finansowy pozwala na dalsze rozwijanie produktu i motywacje do działania, aby ulepszyć swój produkt. Kolejnym z ważnych wniosków pochodzący z pracy nad tym projektem to fakt, iż dobre produkty wymagają czasu i sporego nakładu pracy dlatego też nie można oczekiwać, że projekty które zostały uruchomione w niedalekiej przeszłości będą perfekcyjne. W przypadku oprogramowania open source ważnym bodźcem do zmian jest

informacja zwrotna od użytkowników dzięki której autor wie na czym powinien się skupić i gdzie można znaleźć szerszą grupę użytkowników, jeżeli nie zadba o ten aspekt to z biegiem czasu może się okazać, że użytkownicy przejdą do konkurencji, która lepiej dostosowuje się do rynku i proponowanych zmian. Niestety jako przykład można podać projekty Domoticz lub/i OpenHAB które od pewnego czasu pomimo dużej popularności tracą kolejnych użytkowników ze względu na brak wsparcia popularnych rozwiązań i nierozwiązane problemy. Poniższy wykres stworzony za pomocą silnika Google Trends pokazuje wskaźniki wyszukiwania fraz wg. Wyszukiwarki Google na przestrzenie ostatnich 10lat dla ruchu internetowego skojarzonego z Polską. Wskaźnika na poziomie 100 jednostek w porównaniu do 50 jednostek oznacza dwukrotnie mniejszą liczbę skojarzonych wyszukiwanych fraz



*Rysunek 6.1 Trendy wyszukiwania w Polsce wybranych systemów automatyki w wyszukiwarce Google.
Źródło trends.google.pl*

Dane dla całego świata pozują ten sam trend jeszcze wyraźniej gdzie od 2018 roku Home Assistant obejmuje sporą przewagę nad konkurencją



Rysunek 6.1 Trendy wyszukiwania na świecie wybranych systemów automatyki w wyszukiwarce Google.
Źródło trends.google.pl

Odwrotną zależność można zauważyć przy oprogramowaniu Home Assistant które pomimo najmniejszego stażu oferuje w tym momencie najwięcej integracji i cieszy się największą popularnością czym przyciąga do siebie entuzjastów i inwestorów. Podchodząc do wyboru platformy w przyszłości dobrą strategią będzie wybranie najpopularniejszej platformy, ponieważ im większe grono użytkowników tym większe prawdopodobieństwo, że problem, z którym będzie trzeba sobie poradzić wystąpił już w przeszłości i jest dla niego rozwiązanie. Rozwiązywanie wcześniej nieopisanych problemów i próba ich naprawy często bywa czasochłonna i bardziej opłacalna może się okazać zmiana platformy. Jeżeli z kolei wymagamy od oprogramowania konkretnej funkcjonalności to lepiej wybrać inne rozwiązanie, które je najlepiej wspiera jednak należy mieć wtedy na uwadze, że w pozostałych kwestiach trzeba będzie wybierać kompromisy. Niemniej jednak wszystkie z przedstawionych systemów Automatyki domowej mają swoje zalety i jeżeli czas na to pozwoli, to warto choć na moment przyjrzeć się każdemu z nich, aby podjąć świadomą decyzję.

Bibliografia

Akty prawne

1. Ustawa z dnia 10 maja 2018 r. *o ochronie danych osobowych* Dz.U. z 2018 r. poz. 1000
2. Rozporządzenie Parlamentu Europejskiego i Rady (UE) 2016/679 z dnia 27 kwietnia 2016 r. *w sprawie ochrony osób fizycznych w związku z przetwarzaniem danych osobowych i w sprawie swobodnego przepływu takich danych oraz uchylenia dyrektywy 95/46/WE (ogólne rozporządzenie o ochronie danych)* Dz. U. UE L 119 4 maja 2016

Artykuły i dokumenty elektroniczne

1. Definicja i podstawowe informacje. Źródło: <https://cloud.google.com/solutions/iot-overview> –o IoT opracowane przez firmę Google [dostęp 19.12.2020]
2. Znaczenie IoT w sporcie. Źródło:
<https://www2.deloitte.com/content/dam/Deloitte/us/Documents/consumer-business/us-cb-internet-of-things-sports.pdf> IoT w sporcie wg. Firmy Deloitte Digital [dostęp 19.12.2020]
3. Artykuł o rosnącym rynku asystentów głosowych. Źródło:
<https://www.forbes.com/sites/danielnewman/2018/08/22/voice-interface-technology-the-future-of-business/?sh=16fdc96a316a> [dostęp 19.12.2020]
4. Models in Industrial IoT Wi-Fi 6 and Private LTE/5G Technology and Business Models in Industrial. Źródło: IoT Cisco-IoT-5G-WP_V1-20191016 [dostęp 19.12.2020]
5. Cisco Annual Internet Report (2018–2023). Źródło: White Paper C11-741490-01 03/20 [dostęp 21.12.2020]
6. Internet of Things (IoT) communication protocols: Review. Źródło:
10.1109/ICITECH.2017.8079928 [dostęp 21.12.2020]
7. Opis standardów sieci w IoT. Źródło: <https://przemysl-40.pl/index.php/2018/07/30/lorawan-i-sigfox-dwa-standardy-sieci-iot/> opis protokołów lorawan i sigfox [dostęp 26.03.2021]
8. Opis technologii 5G. Źródło: <https://www.gov.pl/web/5g/technologie-5g> [dostęp 26.03.2021]
9. Opis LTE-Cat1. Źródło: <https://www.siretta.com/2019/04/3g-vs-cat-1/> [dostęp 25.12.2020]

10. Aspekty bezpieczeństwa w IoT. Źródło: <https://www.intellectsoft.net/blog/biggest-iot-security-issues/> [dostęp 29.12.2020]
11. Najlepsze praktyki bezpieczeństwa według platformy AWS. Źródło: https://pages.awscloud.com/rs/112-TZM-766/images/IoT_Security_Best_Practices_Guide_design_v3.1.pdf [dostęp 29.12.2020]
12. Aktualna lista podatności w różnych systemach. Źródło: <https://www.exploit-db.com/>
13. Katalog lekkich algorytmów szyfrujących wg. NIST. Źródło: <https://csrc.nist.gov/Projects/lightweight-cryptography> [dostęp 03.01.2021]
14. Informacje o google home Źródło: <https://developers.google.com/assistant/smarthome/overview>
15. Artykuł dotyczący wsparcia protokołu Zigbee w Alexa. Źródło: <https://developer.amazon.com/en-US/docs/alexa/smarthome/zigbee-support.html> [dostęp 03.01.2021]
16. FAQ producenta Wink. Źródło: <https://www.wink.com/help/faq/#about> [dostęp 03.01.2021]
17. Artykuł dotyczący zarządzanie urządzeniami smarthings lokalnie. Źródło: <https://homealarmreport.com/smart-home/smart-devices-work-locally/> [dostęp 03.01.2021]
18. Informacje nt. Fibaro. Źródło: <http://archive.is/6KT08#selection-618.0-618.3> [dostęp 03.01.2021]
19. Artykuł nt. sprzedaży firmy Fibaro. Źródło: <https://www.telepolis.pl/wiadomosci/wydarzenia/fibaro-zmienia-wlasciciela> [dostęp 03.01.2021]
20. Porównanie Home assistant vs. OpenHAB. Źródło: <https://smarthome.university/your-smart-home-platform-home-assistant-vs-openhab/> [dostęp 05.02.2021]
21. Kompatybilność z HA. Źródło: <https://www.home-assistant.io/integrations/#all> [dostęp 05.02.2021]
22. Platforma Home assistant cloud. Źródło: <https://www.nabucasa.com/> [dostęp 05.02.2021]
23. Lista systemów operacyjnych Raspberry. Źródło: <https://www.raspberrypi.org/software/operating-systems/> [dostęp 05.02.2021]
24. HACS – „sklep” z rozszerzeniami tworzonymi przez użytkowników do Home Assistant Źródło: <https://hacs.xyz/> [dostęp 05.02.2021]
25. Instalacja Open HAB. Źródło: <https://www.openhab.org/docs/installation/> [dostęp 15.03.2021]

- 26.** Dodawanie urządzeń w OpenHAB. Źródło:
https://www.openhab.org/docs/tutorial/things_advanced.html [dostęp 15.03.2021]
- 27.** GadgetFreakz – Sklep partnerski platformy Domoticz udostępniający listę urządzeń kompatybilnych. Źródło: <https://gadget-freakz.com/product-category/gadgetz/domoticz-compatible-products/> [dostęp 15.03.2021]
- 28.** Portal GadgetFreakz z poradnikami i nowościami o Domoticz. Źródło: <https://gadget-freakz.com/tutorials/> [dostęp 15.03.2021]
- 29.** Domoticz Wiki. Źródło: https://www.domoticz.com/wiki/Main_Page [dostęp 19.03.2021]
- 30.** Domoticz forum. Źródło: <https://www.domoticz.com/forum/> [dostęp 19.03.2021]
- 31.** Kod źródłowy biblioteki zigbee2mqtt. Źródło: <https://github.com/stas-demydiuk/domoticz-zigbee2mqtt-plugin> [dostęp 19.03.2021]
- 32.** Lista kompatybilności z biblioteki zigbee2mqtt. Źródło:
https://www.zigbee2mqtt.io/information/supported_devices.html [dostęp 19.03.2021]
- 33.** Koordynator ZigBee CC2531 Źródło: <https://www.ti.com/product/CC2531> [dostęp 19.03.2021]
- 34.** Pi Imager Źródło: <https://www.raspberrypi.org/software/> [dostęp 24.05.2021]
- 35.** Repozytorium pluginu PiMonitor Źródło: <https://github.com/Xorfor/Domoticz-PiMonitor-Plugin> [dostęp 24.05.2021]
- 36.** Repozytorium pluginu Mikrotik RouterOS Źródło: <https://github.com/mrin/domoticz-routeros-plugin> [dostęp 24.05.2021]
- 37.** Repozytorium pluginu SNMP Reader Źródło:
<https://github.com/ycahome/SNMPReader> [dostęp 24.05.2021]
- 38.** Rejestracja w serwisie Pushbullet Źródło:
<https://www.pushbullet.com/signin?next=%2F> [dostęp 29.06.2021]
- 39.** Limit powiadomień push w Pushbullet Źródło:
<https://docs.pushbullet.com/#ratelimiting> [dostęp 29.06.2021]
- 40.** Instalacja Docker na Raspberry. Źródło: <https://www.docker.com/blog/happy-pi-day-docker-raspberry-pi/> [dostęp 29.06.2021]
- 41.** Instalacja centrum multimediiów Kodi na Raspberry. Źródło:
https://kodi.wiki/view/HOW-TO:Install_Kodi_on_Raspberry_Pi [dostęp 29.06.2021]
- 42.** Instalacja lokalnego serwera DNS na Raspberry. Źródło:
<https://www.smarthomebeginner.com/pi-hole-setup-guide> [dostęp 29.06.2021]

Spis tabel

Tabela 2.4-1: Podział LPWAN – Źródło: Opracowanie własne.	17
Tabela 2.6-1 Podstawowe parametry Google NestHub	27
Tabela 2.6-2 Podział urządzeń amazon Echo Źródło: Opracowanie własne.	28

Spis rysunków (ilustracji)

Rysunek 2.1-1 Podział elementów systemu IoT.	7
Rysunek 2.2-1 Podział elementów systemu IoT.	9
Rysunek 2.3-1 Przewidywania liczby użytkowników asystentów głosowych wg. eMarketer.	10
Rysunek 2.3-2 Udział urządzeń z asystentem głosowym w rynku w 2018 roku.	11
Rysunek 2.3-3 Rewolucja przemysłowa 4.0	12
Rysunek 2.3-4 Prognoza podłączonych urządzeń do Internetu: Coroczny raport cisco nt. Internetu 2020	14
Rysunek 2.3-5 Wzrost podłączonych urządzeń komunikujących się w trybie M2M - Podział na kategorie	14
Rysunek 2.9-1 schemat architektury Google Home.	25
Rysunek 2.9-2 Możliwości kontrolowania urządzeń zewnętrznych w Google Home... ..	25
Rysunek 2.9-3 Google Nest Hub	26
Rysunek 2.9-4 Porównanie wersji SmartThings hub.....	31
Rysunek 2.9-5 Prognoza udziałów w rynku USA z podziałem na inteligentne głośniki wg. Loup Ventures.	33
Rysunek 2.9-6 Rysunek 2-9-6 Schemat połączeń w implanicie fibaro:.....	34
Rysunek 2.9-7 Specyfikacja połączeń bezprzewodowych w Fibaro Home center 3	35
Rysunek 2.10-1 Interfejs Home Assistant Web.	36
Rysunek 2.10-2 Architektura Home Assistant	37
Rysunek 2.10-3 Porównanie typów instalacji i ich funkcjonalności.	38
Rysunek 2.10-4 Specyfikacja HA Blue	39
Rysunek 2.10-5 Przykładowy widok OpenHAB	40
Rysunek 2.10-6 Widok aplikacji mobilnej OpenHAB.....	41
Rysunek 3.1-1 Porównanie płytek RPi 3 i 4.	44
Rysunek 4.2-1 Dostępne obrazy Domoticz.....	51
Rysunek 5.3-1 Status usług SSH	54
Rysunek 5.3-2 Instalacja aktualizacji	54
Rysunek 5.3-3 Wybór dostępnych serwisów dla Domoticz.....	54
Rysunek 5.3-4 Konfiguracja portów dla serwisów Domoticz	55
Rysunek 5.3-5 Wybór miejsca instalacji Domoticz	55
Rysunek 5.3-6 Zakończenie instalacji	55
Rysunek 5.3-7 Strona powitalna Domoticz.	55
Rysunek 5.3-8 Zakończenie instalacji.....	56
Rysunek 5.3-9 Ekran logowania w Domoticz.	56
Rysunek 5.5-1 Schemat działania biblioteki ZigbeeMQTT https://github.com/koenkk/zigbee2mqtt	57
Rysunek 5.5-2 Instalacja node.js w systemie RaspberryPi OS.....	58
Rysunek 5.5-3 Instalacja npm w systemie RaspberryPi OS.	58
Rysunek 5.5-4 Sprawdzanie wersji pakietów node.js i npm w systemie RaspberryPi OS.	58

Rysunek 5.5-5 Pobieranie biblioteki zigbee2mqtt RaspberryPi OS.	58
Rysunek 5.5-6 Zmiana uprawnień biblioteki..	58
Rysunek 5.5-7 Instalacja zależności biblioteki..	59
Rysunek 5.5-8 Listowanie urządzeń USB w systemie Raspberry PI OS.	59
Rysunek 5.5-9 Konfiguracja biblioteki zigbee2mqtt.	59
Rysunek 5.5-10 Sprawdzanie statusu usługi w Raspberry PI OS.	60
Rysunek 5.5-11 Instalacja pakietu mosquitto w systemie Raspberry PI OS.	60
Rysunek 5.5-12 Wyświetlenie dziennika biblioteki zigbee2mqtt.	60
Rysunek 5.5-13 Pobranie pluginu domoticz--zigbee2mqtt.	60
Rysunek 5.5-14 Konfiguracja pluginu domoticz--zigbee2mqtt.	61
Rysunek 5.5-15 Lista czujników udostępnianych przez termostat Tuya	61
Rysunek 5.5-16 Interfejs pluginu domoticz--zigbee2mqtt w systemie domoticz..	62
Rysunek 5.6-1 Czujniki dostępne w ramach pluginu PiMonitor	63
Rysunek 5.6-2 Konfiguracja pluginu Mikrotik Router OS	64
Rysunek 5.6-3 Logi pluginu Mikrotik Router OS w trybie „debug”	64
Rysunek 5.6-4 Dostępne encje w wykorzystaniem pluginu Mikrotik Router OS	64
Rysunek 5.6-5 Konfiguracja plugin SNMP Reader w Domoticz	66
Rysunek 5.6-6 Logi pluginu SNMP Reader w w trybie debug	66
Rysunek 5.6-7 Encje dodane przez plugin SNMP Reader w Domoticz	66
Rysunek 5.7-1 Dodanie nowego urządzenia w Domoticz.	67
Rysunek 5.7-2 Widok nowo dodanego urządzenia w systemie domoticz.	67
Rysunek 5.7-3 Konfiguracja scen urządzenia.	68
Rysunek 5.7-4 Logi urządzenia dostępne w Domoticz.	69
Rysunek 5.8-1 Konfiguracja i Status połączenia zdalnego..	70
Rysunek 5.8-2 Tworzenie tokenu dostępowego w serwisie Pushbullet	71
Rysunek 5.8-3 Konfiguracja usługi Pushbullet w Domoticz.	71
Rysunek 5.8-4 Test usługi Pushbullet w Domoticz.	71
Rysunek 5.8-5 Logi związane z usługą Pushbullet w Domoticz.	72
Rysunek 5.8-6 Historia powiadomień w serwisie Pushbullet.....	72
Rysunek 5.8-7 Konfiguracja powiadomień E-mail w Domoticz. własne.....	73
Rysunek 5.8-8 Logi powiadomień E-mail w Domoticz.	73
Rysunek 5.8-9 Test powiadomień E-mail w Domoticz..	73
Rysunek 5.8-10 Konfiguracja i wygląd aplikacji Domoticz Lite na systemy Android..	74
Rysunek 5.8-11 Zarządzanie dostępem przez aplikację w Domoticz.....	74
Rysunek 5.9-1 Automatyzacja wykrywania braku aktywności urządzeń.....	75
Rysunek 5.9-2 Test automatyzacji wykrywania braku aktywności urządzeń.....	75
Rysunek 5.9-3 Skrypt automatyzacji w języku LUA..	76
Rysunek 5.9-4 Schemat automatyzacji w blocky.....	76
Rysunek 5.9-5 Test automatyzacji termostatu.	77
Rysunek 5.9-6 Konfiguracja kamery w Domoticz.....	78
Rysunek 5.9-7 Schemat automatyzacji w LUA wysyłający maile z zrzutem ekranu kamery.	78
Rysunek 5.9-8 Schemat automatyzacji w blocky wysyłający maile.	78
Rysunek 5.9-9 Test automatyzacji wysyłającej maile.....	79
Rysunek 6.1 Trendy wyszukiwania na świecie wybranych systemów automatyki w wyszukiwarce Google.....	82
Rysunek 6.2 Trendy wyszukiwania w Polsce wybranych systemów automatyki w wyszukiwarce Google.....	82