# Wstęp teoretyczny

Rozdział ten zawiera informacje o koncepcie IoT, wykorzystaniu urządzeń IoT, systemie Android Things, a także informacje o decyzji wyboru tematu pracy.

## Wstęp

Technologie IoT od wielu lat zdobywają popularność na rynku. Koncepcja IoT opiera się połączeniu fizycznych obiektów codziennego użycia do Internetu. Pozwala to na zdalne kontrolowanie przedmiotów oraz gromadzenie danych w sposób umożliwiający ich prostą analizę. Pozwala to przyśpieszyć i ulepszyć codzienne czynności i procesy. Inteligentne urządzenia mają również swoją niszę w zastosowaniach domowych (np. inteligentne włączniki świateł, czajniki), ale także i przemysłowych. Urządzenia IoT są stosowane w systemach bezpieczeństwa, transporcie, energetyce, służbie zdrowia, produkcji, kontroli jakości itd. Jaki jest powód wdrażania tam tych technologii? Umożliwiają one automatyzację, synchronizację, a także wdrażanie procesów typu DevOps w procesach produkcyjnych.

W przypadku zastosowań domowych mogą być to kamery IP, gniazdka monitorujące zużycie prądu, inteligentne lodówki, czajniki, wagi. Tak naprawdę zastosowań jest bardzo dużo, ponieważ zestawy prototypowe są sprzedawane w przystępnych cenach, często gotowych do wykonaniu prototypowania swojego pierwszego urządzenia klasy IoT. Do swojej dyspozycji domowi konstruktorzy mają do dyspozycji gotowe mikrokomputery, często wyposażone w złącze GPIO (general purpouse input-output), które pozwala na wyprowadzenie połączeń do innych układów. Przykładami takich urządzeń jest mikrokomputer Arduino, BeagleBone czy Rasberry Pi. Układy, które możemy podłączyć do takich układów to termometry, diody, czujniki (np. zbliżeniowe), wyświetlacze, kamery, regulatory napięcia, przyciski, serwomechanizmy. Sterowanie nimi odbywa się przy użyciu znanych i dobrze opisanych interfejsów, takich jak np. I2C.

W roku 2018 firma Google również postanowiła wkroczyć w ten rynek. Nie zaoferowała ona jednak swojego autorskiego układu, lecz oficjalnie udostępniała system Android Things w wersji 1.0. System Android Things jest kompatybilny z mikrokomputerami takimi jak Rasberry Pi 3 Model B oraz NPX i.MX7D. Układy te są oficjalnie wspierane przez kalifornijskiego giganta. System ten został zaprojektowany, aby wraz z Things Support Library dostarczyć wsparcia do integracji urządzeń dotąd niewspieranych na urządzeniach mobilnych. Pozwala on na bardziej przyjazny dostęp do urządzeń peryferyjnych i sterowników niż urządzenia mobilne, jednocześnie zapewniając dobrodziejstwa systemu Android, takie jak wysoko-poziomowe API.

Jak zostało wspomniane wyżej urządzenia IoT mogą mieć wiele zastosowań domowych. Zgodnie z założeniem urządzenia tej klasy powinny wspomagać użytkownika w jego działaniach, poprzez automatyzację pewnych czynności jak i gromadzenie danych umożliwiających analizę. Jednym z problemów, które urządzenie klasy IoT może rozwiązać jest domowa uprawa roślin, które wymagają szczególnych warunków, takich jak utrzymanie temperatury, wilgotności, cykli światła. Nie chodzi tu tylko o regulację pewnych parametrów, ale także o regularne zbieranie danych w taki sposób, aby użytkownik mógł odpowiednio reagować na zmiany zachodzące w środowisku rośliny.

Mając na uwadze tak szybki rozwój branży IoT w tej pracy postanowiłem zaprojektować system wspomagania uprawy roślin. Będzie on miał na celu wspomóc domowych ogrodników.

## Platforma Android Things

Platforma Android Things została oficjalnie ogłoszona w roku 2015, pod kodową nazwą Brillo, na konferencji Google I/O. Jak klasyczny Android oparty jest on na jądrze systemu Linux. Przez kilka kolejnych lat developerze trudzili się, aby zaprojektować system, który będzie używał mało energii, działający na różnych platformach MCU. Od samego początku wspiera Low Energy Bluetooth oraz WiFi.

System pozwala na projektowanie aplikacji przy użyciu istniejących bibliotek i narzędzi dla systemu Android. Wśród nowego API odnajdziemy komponenty takie jak niskopozoiomowe API do obsługi I/O, a także sterowniki dla najpopularniejszych urządzeń, takich jak sensory, wyświetlacze, termometry itp. Nie jest żadnym problemem połączenie z Google Services. System ma wbudowane narzędzia do obsługi złącz GPIO, I2C, SPI, UART a także PWM. Opis złącz znajduje się poniżej:

* GPIO – złącze ogólnego przeznaczenia. Może przyjąć stan logiczny niski oraz wysoki.
* I2C – szeregowa, dwukierunkowa magistrala. Wysyła dane za pomocą linii sygnałowych SDA (linia danych) oraz SCL (linia sygnału taktującego).
* SPI – Interfejs szeregowy urządzeń peryferyjnych. Służy do dwukierunkowej transmisji danych. Interfejs jest trójprzewodowy. Transfer danych odbywa się w układzie master-slave.
* UART – służy do przekazywania i odbierania informacji przez port szeregowy. Wspiera on buforowanie danych w przypadku szybkiej transmisji.
* PWM – złącze służące do modulacji sygnału cyfrowego.

To, jakie złącza znajdziemy na mikrokontrolerze zależy od niego.

Udostępniona jest również usługa Android Things Console, która pozwala na zdalne aktualizacje aplikacji. Wystarczy, że urządzenie jest podpięte do internetu.

Jak zostało wspomniane wyżej platforma obecnie obsługuje dwa układy – Rasberry Pi 3 Model B, a także NPX i.MX7D. Sam system Android Things opiera się na popularnym systemie Android dla urządzeń mobilnych. Tutaj jednak, wraz z Android SDK zostało dostarczone dodatkowe API w postaci Things Support Library, pozwalające na integrację w zasadzie każdego urządzenia działającego na w/w złączach.

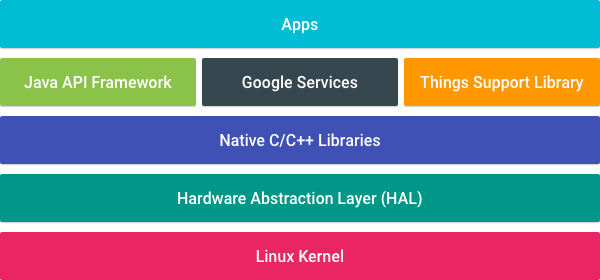
Mówiąc o systemie Android Things nie sposób jest nie porównać go do jego „starszego brata”, systemu Android. W porównaniu do niego, Android Things pozwala na łatwiejszy dostęp do urządzeń peryferyjnych, nie posiada wielu aplikacji wbudowanych w system, pozwala na instalację tylko jednej aplikacji, która zostanie uruchomiona automatycznie po starcie aplikacji (jeżeli jest obecna). Samo Android SDK jest również ograniczone – nie posiada części UI do którego przyzwyczaił nas Android (np. status bar, navigation buttons). Innymi, najważniejszymi kompontentami są:

* android.hardware.telephony (usługa odpowiedzialna za wykonywanie połączeń GSM)
* android.hardware.nfc (usługa odpowiedzialna za komunikację z modułem NFC)
* android.hardware.fingerprint (usługa odpowiedzialna za autoryzację odciskiem palca)
* android.hardware.wifi.aware (usługa odpowiedzialna za optymalizację połączeń WiFi)
* android.software.backup (usługa odpowiedzialna za automatyczny backup)

Nie są to wszystkie usunięte komponenty z Android API. Warto jednak wspomnieć, że część z tych komponentów można zastąpić zewnętrznymi bibliotekami i modułami.

O czym warto wspomnieć, z racji, że urządzenia IoT często nie posiadają interfejsu użytkownika tzw. Runtime Permissions, wprowadzone w Androidzie Marshmallow zostały wyłączone, i są nadawane podczas instalacji aplikacji, lub poprzez konsole adb. System wspiera natywny kod, jednak wymaga tego, aby natywne biblioteki były trzymane w pliku APK.

Aby lepiej zrozumieć jak system jest zbudowany warto spojrzeć na graficzne przedstawienie warstw systemu.

<https://developer.android.com/things/images/platform-architecture.png>

## Opis projektowanego systemu

Jak zostało wcześniej wspomniane, celem tej pracy jest zaprojektowanie systemu wspomagania uprawy roślin. Pomoc ta miałaby sprowadzać się do:

* Zdalnego kontrolowania światła (włączone/wyłączone), w tym możliwość ustawienia godzin, w których światło ma być w włączone/wyłączone.
* Monitorowania temperatury pomieszczenia
* Monitorowania wilgotności gleby i powietrze
* Regulacji temperatury (poprzez włączanie/wyłączanie wentylatorów)
* Możliwość zrobienia zdjęcia wewnątrz pomieszczenia.

W tym celu zostaną zaprojektowane:

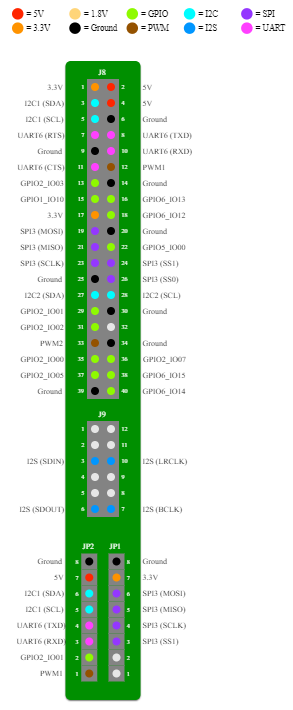
* Prototyp urządzenia IoT (hardware)
* Aplikacja kontrolujące prototyp klasy IoT działająca pod kontrolą systemu Android Things
* Aplikacja kliencka na system Android pozwalająca na zdalne monitorowanie parametrów środowiska, a także zdalną kontrolę nad aplikacją.

Jako mikrokomputer zostanie użyty komputer NPX i.MX7D. Układ ten posiada dwurdzeniowy procesor ARM Cortex A7, 512MB pamięci RAM, wbudowane 4GB pamięci eMMC, wbudowany ekran LCD8000-43T VL050-80128NM-CO1, a także kamerę O5640 MIPI CSI. Interfejsy, które udostępnia ten układ to I2S (Audio), UART, I2C, SPI, PWM, GPIO (czyli wszystkie wspierane przez system Android Things). Interfejsy bezprzewodowej komunikacji to 10/100/1000 Ethernet, Wi-Fi 802.11ac (2.4/5.0Ghz), a także Bluetooth 4.1. Na płytce znajdziemy również gniazda USB – USB 2.0 Host, a także USB 2.0 OTG.



<https://developer.android.com/things/images/nxp-pico7-module.png>

Prezentacja listy złączy na płytce:



<https://developer.android.com/things/hardware/imx7d.html>

Powyższa konfiguracja urządzenia powinna pozwolić na zaprojektowanie urządzenia klasy IoT spełniającego wszystkie w/w wymagania. Mikrokomputer ten działać będzie na systemie Android Things w wersji 1.0. Zdalna kontrola odbywać będzie się z aplikacji na urządzenie Android, w wersji co najmniej 8.0. Aplikacje działające na systemie Android zostaną napisane w języku Kotlin. Do przechowywania danych w chmurze zostanie użyty Firebase, udostępniający zarówno usługę przechowywania statycznych zdjęć, jak i Real Time Database.

Opis, a także projekt całego układu znajduje się w dalszej części pracy.

# Projektowanie i implementacja

Ten rozdział zawiera informacje na temat procesu projektowania układu, począwszy od instalacji systemu na mikrokomputerze, skończywszy na implementacji aplikacji klienckiej.

## Projekt urządzenia

Biorąc pod uwagę wymagania wyspecyfikowane powyżej, podjęto decyzję jakie podzespoły zostaną użyte.

|  |  |  |
| --- | --- | --- |
| Wymaganie | Zastosowany układ | Opis |
| Zdalna kontrola światła w tym możliwość ustalenia godzin, w których światło ma być włączone/wyłączone | Moduł 2 przekaźników TONGLING JQC-3FF-S-Z | Przekaźnik jest urządzeniem, które pozwala na wywołanie pewnej ustalonej nagłej zmiany stanu. Zmiana stanu jest skokowa.  Zastosowanie przekaźników pozwoli na sterowanie stanem oświetlenia. Możliwość planowania zostanie zapewniona na warstwie softwareowej. |
| Monitorowanie temperatury pomieszczenia | Czujnik temperatury LM35-DZ (\*) | Analogowy czujnik temperatury pozwalający mierzyć temperatury z zakresu -55 °C do 150 °C z dokładnością do 0.5 °C. |
| Monitorowanie wilgotności gleby i powietrza | TODO(pmaterna) |  |
| Regulacja temperatury | - | Regulacja temperatury będzie odbywać się poprzez ustawienie sterownika wybranego wentylatora komputerowego w stan niski bądź wysoki |
| Możliwość zrobienia zdjęcia wnętrza pomieszczenia | - | Zastosowany układ posiada wbudowaną kamerę. To ona zostanie wykorzystana w tym celu. |

(\*) – Układ jest układem z wyjściem analogowym. Niestety, zastosowana płytka nie posiada wejść analogowych. Aby odczytać wartość zwracaną przez układ zostanie użyty konwerter analogowo cyfrowy – ADAFRUIT ADS1015. Układ ten pozwala na podłączenie do 4 urządzeń analogowych. Komunikuje się z hostującym układem przez interfejs I2C.

## Przygotowanie układu – instalacja systemu

Pierwszym krokiem instalacji systemu Android Things na mikrokomputerze jest pobranie Android Things Setup Utility z Android Things Console. Wymagało to posiadania konta Google oraz zaakceptowania licencji.

Wstęp teoretyczny:

<https://www.computerworld.pl/news/IIoT-jak-zmieni-swiat-i-dlaczego-jest-to-wazne,409830.html>

<https://developer.android.com/things/get-started>

<https://en.wikipedia.org/wiki/Android_Things>

Android things:

<https://developer.android.com/things/get-started/platform-differences>

<https://developer.android.com/things/get-started>

<https://developer.android.com/things/hardware/>