2023-05-15

Piotr Wyrwas

Wydział matematuki i informatyki, uniwersytet Łódzki

Projekt zaliczeniowy IoT

# Spis treści

[Spis treści 1](#_Toc135038801)

[Wstęp 2](#_Toc135038802)

[Cel 2](#_Toc135038803)

[Oprogramowanie klienta 2](#_Toc135038804)

[Logika Biznesowa 2](#_Toc135038805)

[Klient 2](#_Toc135038806)

[1.1 Klasa Virtual Device 2](#_Toc135038807)

[1.2 Pobieranie danych 2](#_Toc135038808)

[1.3 Wysyłanie danych 2](#_Toc135038809)

[1.4 Inicjalizacja 2](#_Toc135038810)

[1.5 Device Twin 2](#_Toc135038811)

[1.6 Direct Method 3](#_Toc135038812)

[Logika biznesowa po stronie Azure. 3](#_Toc135038813)

[1.7 Stream Analytics 3](#_Toc135038814)

[1.8 Logic Apps 3](#_Toc135038815)

# Wstęp

## Cel

Celem niżej opisanego projektu jest zapewnienie komunikacji między urządzeniami w fabryce a chmurą Azure oraz implementacja, zgodnej z podanymi wymaganiami, logiki biznesowej.

Obraz zawierający diagram, Plan, tekst, Rysunek techniczny

Opis wygenerowany automatycznie

## Oprogramowanie klienta

Oprogramowanie zostało napisane w języku C#. Zapewnia ona cykliczne wysyłanie wiadomości D2C oraz aktualizację informacji zawartych w Device Twin

## Logika Biznesowa

Cała logika, zgodnie z wymaganiami, jest zaimplementowana po stronie chmury azure. W celu jej implementacji zostały wykorzystane Logic Apps oraz Stream Analytics

# Klient

## Klasa Virtual Device

Klient do wysyłania, aktualizowania oraz odczytywania DirectMethod korzysta z metod klasy VirtualDevice. W głównym programie klienta jest zaimplementowane tylko łączenie się z odpowiednim urządzeniem w IoT Hubie. Za połączenie z odpowiadającym mu urządzeniem na serwerze fabryki odpowiadają również metody klasy VirtualDevice.

## Pobieranie danych

Klient podczas tworzenia wiadomości D2C łączy się z serwerem fabryki i na bieżąco pobiera aktualne dane, następnie przygotowuje je do wysłania.

## Wysyłanie danych

Przygotowane dane zostają wysłane do IoT Huba jako dane telemetryczne.

Proces ten powtarza się co 10 sekund (wartość ustawiona dla testów).

## Inicjalizacja

Inicjalizacja nagłówków potrzebnych do wykonywania Direct Method oraz wykrycia zmian po stronie Device Twin odbywa się przed rozpoczęciem cyklicznego wysyłania widomości.

## Device Twin

Device Twin przechowuje aktualne dane o błędach występujących na urządzeniu oraz aktualnej szybkości produkcji.

## Direct Method

Direct Method są to wcześniej przygotowane metody które można wywołać z poziomu IoT Huba. W założeniu są one wykorzystywane przez Logikę Biznesową do wdrożenia podjętych decyzji.

## Schemat

Obraz zawierający tekst, diagram, zrzut ekranu, Prostokąt

Opis wygenerowany automatycznie

# Logika biznesowa po stronie Azure.

## Stream Analytics

Stream Analytics odpowiada za przeanalizowanie odpowiednich danych telemetrycznych oraz umieszczenie ich w odpowiednim miejscu, takim jak kontener na dane lub kolejka. Aktualnie dane które są zbierane to:

1. Dane procentowe o ilości poprawnie wyprodukowanych przedmiotów, są one zbierane co 5 minut i zapisywane do kontenera o nazwie „KPIs”
2. Informacje o wystąpieniu w przeciągu mniej niż jednej minuty trzech błędów na raz. Dane te zostają zapisane w kontenerze „ErrorRaport”
3. Dane o średniej temperaturze z ostatnich pięciu minut. Są one zapisywane co minutę do pojemnika „tempAVG”

## Logic Apps

Logic Apps odpowiada za podjęcie właściwych działań w zależności od wcześniej zdefiniowanego przypadku. Na chwilę obecna zostały zdefiniowane trzy takie sytuację:

1. W przypadku gdy w kontenerze „ErrorRaport” znajdzie się nowy wpis od razu jest wywoływana DirectMethod’a odpowiadająca za całkowite zatrzymanie produkcji na danym urządzeniu (EmergencyStop)
2. W przypadku kiedy procentowa wartość poprawnie wykonanych dóbr spadnie poniżej 90% zostaje zmniejszone tempo produkcji o 10 punktów.
3. Jeśli jakikolwiek error zostanie wykryty natychmiastowo zostaje wysłany email na podany wcześniej adres.

