

Dokumentacja projektu

Contents

1.	Instrukcja uruchomienia aplikacji	1
2.	Funkcjonalności	2
•	Komunikacja D2C.....	2
•	Direct methods.....	3
•	Device Twin	6
•	Data Calculations	7
•	Buisness Logic – NIEZAIMPLEMENTOWANA	8

1. Instrukcja uruchomienia aplikacji

Na platformie Azure należy utworzyć:

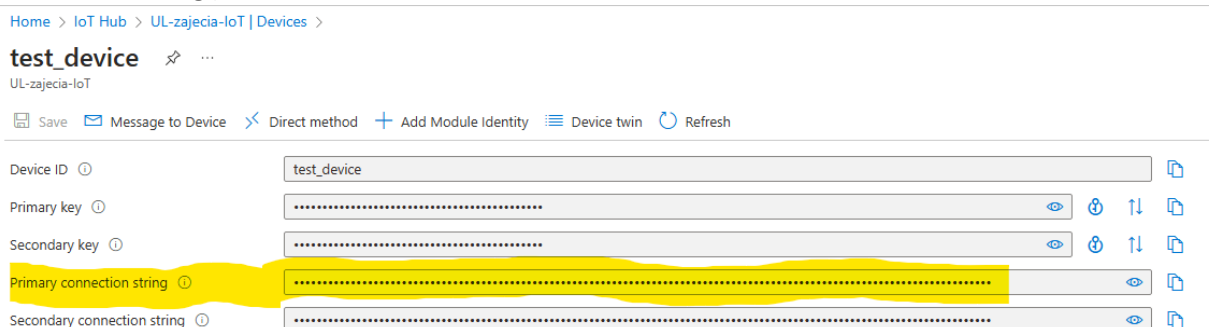
- IoT Hub wraz z docelową ilością urządzeń
- Storage account wraz z 4 kontenerami Blob (np. „telemetry”, „errors”, „kpi”i „temperature”)
- Stream Analytics job z 1 inputem (nasz IoT Hub) oraz 4 outputami (nasze Blob kontenery)

Następnie należy pobrać projekt z GitHub i uruchomić plik „IoTHub.Console”
(IoTHub\IoTHub.Console\bin\Debug\net6.0\ IoTHub.Console.exe).

Jedna instancja aplikacji odpowiada jednemu urządzeniu!

W oknie konsoli użytkownik zostanie poproszony o podanie kolejno:

- Azure Device Connection String (IoT Hub -> Devices -> [nazwa urządzenia] -> Primary connection string)



- OPC UA Client connection string (np. opc.tcp://localhost:4840/)

- Nazwa urządzenia w OPC UA (np. „Device 1”, „Device 2”)

W przypadku podania złej wartości lub innych problemów zostanie wyświetlony komunikat i będzie możliwość ponownego wpisania

```
Connection string to Azure IoT:
WrongConnectionString
Connection failed. Please check connection string to Azure IoT or resolve other problems
Connection string to Azure IoT:
```

W przypadku, gdy wszystkie wartości będą poprawne – wyświetli się następujący komunikat:

```
Connection string to Azure IoT:
WrongConnectionString
Connection failed. Please check connection string to Azure IoT or resolve other problems

Connection string to Azure IoT:
HostName=UL-zajecia-IoT.azure-devices.net;DeviceId=test_device;SharedAccessKey=/yDZTwwCjiKGayFrVZA95jIDZ0zSSISV3AIoTI7xe
jA=

Connection string to OPC UA:
opc.tcp://localhost:4840/

Device name in OPC UA:
Device 1

Connection successfull!
```

Wszystkie połączenia nawiązywane są w projekcie IoTHub.Console, gdzie tworzony jest obiekt klasy „Virtual_device” i istnieją do momentu zatrzymania procesu.

Dane z urządzenia odczytywane są co 10 sekund.

2. Funkcjonalności

- Komunikacja D2C

Telemetria wysyłana jest co 10 sekund i jej przykładowa zawartość wygląda następująco:

Urządzenie włączone:

```
"body": {
  "device": "Device 1",
  "temperature": 78.13413409903372,
  "production_status": 1,
  "worker_id": "130e1800-e3b2-4673-9b0d-c1e6d6fbd350",
  "good_count": 33,
  "bad_count": 1
}
```

Urządzenie wyłączone:

```
"body": {
  "device": "Device 1",
  "temperature": 24.281148949975474,
  "production_status": 0,
  "worker_id": "",
  "good_count": 0,
  "bad_count": 0
}
```

Event wysyłany tylko kiedy na urządzeniu wystąpi błąd (kiedy błędy zostaną wyczyszczone – nie zostanie wysłana wiadomość). Przykładowa zawartość:

```
"body": {
  "error_device": "Device 1",
  "device_errors": "[Sensor Failure] [Power Failure] ",
  "error_count": 2
},
```

Wszystkie wiadomości są w formacie JSON.

- [Direct methods](#)

Emergency stop – zatrzymuje urządzenie i aktywuje flagę błędu „Emergency Stop”.

Można ją wywołać na platformie Azure lub za pomocą programu „Azure IoT Explorer” (na którym pokażę przykładowe wywołanie)

The screenshot shows the Azure IoT Explorer interface. The breadcrumb navigation at the top reads: Home > UL-zajecia-IoT > Devices > test_device > Direct method. On the left sidebar, the 'Direct method' option is selected. The main panel is titled 'Invoke method' and contains the following elements:

- Direct method**: A heading with a subtext: "You can send direct methods to a device. Direct methods have a name, payload, and configurable connection and method timeouts."
- Method name ***: A text input field containing the value "EmergencyStop".
- Payload ⓘ**: A large, empty text area for entering the JSON payload.

Przed:

Industrial Device Simulator

New Device

Remove Selected

Device 1

☒ Production Status

Start

Stop

Production Rate: 40

+

-

Workorder ID: 130e1800-e3b2-4673-9b0d-c1e6d6fb

Temperature: 70.81425078112933

Good Count: 802

Bad Count: 75

☐ Emergency Stop

☐ Power Failure

☐ Sensor Failure

☐ Unknown

Po:

Industrial Device Simulator

New Device

Remove Selected

Device 1

☐ Production Status

Start

Stop

Production Rate: 40

+

-

Workorder ID: 130e1800-e3b2-4673-9b0d-c1e6d6fb

Temperature: 24.3918452399056

Good Count: 820

Bad Count: 79

☒ Emergency Stop

☐ Power Failure

☐ Sensor Failure

☐ Unknown

Reset error status – ustawia wszystkie flagi błędów na 0.

Home > UL-zajecia-IoT > Devices > test_device > Direct method

≡

Device identity

Device twin

Telemetry

< Direct method

Cloud-to-device message

Module identities

IoT Plug and Play components

< Invoke method

Direct method

You can send direct methods to a device. Direct methods have a name, payload, and configurable connection and method timeouts.

Method name *

ResetErrorStatus

Payload ⓘ

Przed:

Industrial Device Simulator

New Device Remove Selected

Device 1

☐ Production Status

Start Stop

Production Rate: 40

+

-

Workorder ID: 130e1800-e3b2-4673-9b0d-c1e6d6ft Temperature: 25

Good Count: 820 Bad Count: 79

☒ Emergency Stop

☒ Power Failure

☐ Sensor Failure

☐ Unknown

Po:

Industrial Device Simulator

New Device Remove Selected

Device 1

☐ Production Status Start Stop

Production Rate: 40 + -

Workorder ID: 130e1800-e3b2-4673-9b0d-c1e6d6f0 Temperature: 24.1572598609059

Good Count: 820 Bad Count: 79

☐ Emergency Stop

☐ Power Failure

☐ Sensor Failure

☐ Unknown

- Device Twin

Desired properties – własności pożądane składają się tylko z „ProductionRate” (procentowo), którego zmiana będzie zaaplikowana na urządzeniu. Zmienić je można np. w „Azure IoT Explorer”.

Przykładowa struktura:

```
"properties": {
  "desired": {
    "ProductionRate": 100,
    "$metadata": {
      "$lastUpdated": "2024-05-09T21:22:18.4989618Z",
      "$lastUpdatedVersion": 13,
      "ProductionRate": {
        "$lastUpdated": "2024-05-09T21:22:18.4989618Z",
        "$lastUpdatedVersion": 13
      }
    }
  },
  "reported": {
    "ProductionRate": 40,
    "deviceErrors": []
  }
}
```

Reported properties – własności zgłoszone odzwierciedlają obecny stan urządzenia (ostatnio sczytane własności). Składają się z „ProductionRate” (procentowo) oraz „deviceErrors” (nazwy errorów lub

„none”). Przykładowa struktura:

```
"reported": {
  "DateTimeLastDesiredPropertyChangeReceived": "2024-05-09T23:22:21.0844353+02:00",
  "ProductionRate": 70,
  "deviceErrors": "[Unknown Error] [Sensor Failure] ",
  "$metadata": {
    "$lastUpdated": "2024-05-19T21:26:27.6444419Z",
    "DateTimeLastDesiredPropertyChangeReceived": {
      "$lastUpdated": "2024-05-09T21:22:21.4930877Z"
    },
    "ProductionRate": {
      "$lastUpdated": "2024-05-19T21:26:27.6444419Z"
    },
    "deviceErrors": {
      "$lastUpdated": "2024-05-19T21:24:54.5895745Z"
    }
  }
},
```

- Data Calculations

Kalkulacje danych zostały zaimplementowane w Azure Stream Analytics Jobs. Aby uruchomić serwis należy wypełnić „Query” (na GitHub plik „ASA-QUERY.txt”) na platformie Azure i nacisnąć „Start job”.

Wszystkie dane wykorzystywane w kalkulacjach pochodzą z D2C message wysyłanych do chmury.

Kalkulacje danych to:

- KPIs - oblicza zsumowany procent dobrej produkcji z ostatnich 5 minut. Wyniki są zapisywane co 5 minut i zapisywane w kontenerze „kpi”. Przykładowa zawartość kontenera:

```
1 [{"Device": "Device 1", "prod_kpi": 91.65876777251185}]
2 [{"Device": "Device 1", "prod_kpi": 91.64705882352942}]
3 [{"Device": "Device 1", "prod_kpi": 91.12534309240623}]
```

- Temperature – oblicza średnią, maksymalną i minimalną temperaturę urządzenia z ostatnich 5 minut. Wyniki są zapisywane co jedną minutę w kontenerze „temperature”. Przykładowa zawartość kontenera:

```
1 [{"Device": "Device 1", "tempAvg": 24.95923777351174, "tempMin": 24.00551991992175, "tempMax": 25.99651586055412}]
2 [{"Device": "Device 1", "tempAvg": 24.956925960517303, "tempMin": 24.00551991992175, "tempMax": 25.99651586055412}]
3 [{"Device": "Device 1", "tempAvg": 27.39071562580029, "tempMin": 24.00551991992175, "tempMax": 70.3319519927853}]
4 [{"Device": "Device 1", "tempAvg": 41.71456026573164, "tempMin": 24.059216447168087, "tempMax": 72.84539486279303}]
5 [{"Device": "Device 1", "tempAvg": 46.28924151801923, "tempMin": 24.059216447168087, "tempMax": 75.69435571577439}]
6 [{"Device": "Device 1", "tempAvg": 60.88766625863201, "tempMin": 24.093765363685552, "tempMax": 75.69435571577439}]
7 [{"Device": "Device 1", "tempAvg": 68.52998694226538, "tempMin": 61.94102730474544, "tempMax": 75.97448256968583}]
8 [{"Device": "Device 1", "tempAvg": 68.2996178484383, "tempMin": 62.12091597078274, "tempMax": 75.97448256968583}]
9 [{"Device": "Device 1", "tempAvg": 68.13189943384603, "tempMin": 60.884578040231816, "tempMax": 75.97448256968583}]
10 [{"Device": "Device 1", "tempAvg": 67.42720213163066, "tempMin": 60.884578040231816, "tempMax": 75.97448256968583}]
```

- Device errors – w momencie kiedy w ciągu minuty pojawi się nie urządzeniu więcej niż 3 błędy, zapisuje w kontenerze „errors” liczbę błędów napotkanych przez urządzenie.

Przykładowa zawartość kontenera:

```
1 [{"Device": "Device 1", "errorCount": 5.0}]
2 [{"Device": "Device 1", "errorCount": 4.0}]
3 [{"Device": "Device 1", "errorCount": 5.0}]
4 [{"Device": "Device 1", "errorCount": 8.0}]
5 [{"Device": "Device 1", "errorCount": 5.0}]
6 [{"Device": "Device 1", "errorCount": 6.0}]
7 [{"Device": "Device 1", "errorCount": 4.0}]
8 [{"Device": "Device 1", "errorCount": 7.0}]
9 [{"Device": "Device 1", "errorCount": 4.0}]
```

- Buisness Logic – NIEZAIMPLEMENTOWANA

Logiki biznesowe opierają się na wynikach „Data Calculations” i odbieranych wiadomościach D2C.

Powinny działać następująco:

- Jeśli w kontenerze „errors” pojawi się nowy wpis (urządzenie napotka powyżej 3 błędów w ciągu minuty) – zostanie wywołana direct method „EmergencyStop”.
- Jeśli w kontenerze „kpi” pojawi się wartość „prod_kpi” mniejsza niż 90 – desired property „ProductionRate” w device twin zostanie zmniejszona o 10.
- Jeśli zostanie wysłana wiadomość event informująca o błędzie – zostanie wysłana wiadomość email na podany adres mailowy