

Dokumentacja IoT

1. Repozytorium

https://github.com/Bartflo/Iot_Project_2k24

2. Połączenie z OPC UA

Aby połączyć się z serwerem OPC UA potrzebujemy dwóch plików:

- Plik aplikacji z folderu „bin -> Release -> net6.0” o nazwie „IoT.Project.Console.exe”
- Plik konfiguracyjny „config.json”, który musi znajdować się w tym samym folderze

Przykładowe uzupełnienie pliku „config.json”

```
{
  "deviceConnectionString": "HostName=IoT2023-Bart.azure-devices.net;DeviceId=Device;SharedAccessKey=ORM2xabUxuxFnnK0NxpVU6wKpqsE7PSY1NyZtwvL/E20=",
  "OPCserverURL": "opc.tcp://localhost:4840/",
  "serviceBusConnectionString": "Endpoint=sb://iot2024-bk.servicebus.windows.net/;SharedAccessKeyName=RootManageSharedAccessKey;SharedAccessKey=b5NquQ/ycR6HP11c4MhXRvvbVchUeqBhw+ASbITONoA=",
  "registryManagerConnectionString": "HostName=IoT2023-Bart.azure-devices.net;SharedAccessKeyName=iothubowner;SharedAccessKey=UEbBomU44sPB05qghWqgeobB5YIA8CgyEETyisM60=",
  "emergencyStopQueue": "device-errors-bus",
  "decreaseProductionRateQueue": "production-kpi-bus",
  "azureDeviceName": "Device"
}
```

deviceConnectionString – klucz do łączenia z się z naszym urządzeniem na platformie Azure

OPCserverURL – adres do łączenia się z serwerem OPC UA

serviceBusConnectionString – klucz do łączenia się z usługą „Service Bus” na platformie Azure

registryManagerConnectionString – klucz do łączenia się z „IoT Hub” na platformie Azure

emergencyStopQueue – nazwa kolejki stworzona w „Service Bus” do wywoływania metody „EmergencyStop”

decreaseProductionQueue – nazwa kolejki stworzona w „Service Bus” do wywoływania metody „DecreaseProductionRate”

azureDeviceName – nazwa urządzenia z którym będziemy się łączyć (wykorzystywane do obniżania produkcji)

Po uruchomieniu aplikacji dostaniemy informację czy plik konfiguracyjny załadował się poprawnie czy też brakuje nam jakichś danych oraz czy połączenie z usługami się udało.

```
Started
Configuration file read successfully
Connection to Azure success
Connection to OPC UA success
```

A tutaj przykładowa wiadomość o braku jednego z „connection string”

```
Started
Error: 'deviceConnectionString' not found in the configuration file.
```

Po poprawnym uruchomieniu aplikacji i załadowaniu całego pliku konfiguracyjnego aplikacja jest gotowa do działania.

3. Wiadomości D2C

Wiadomości Device To Cloud wysyłane są co 10 sekund – dane które są przekazywane

- DeviceName – nazwa fizycznego urządzenia
- ProductionStatus – zawiera aktualny stan produkcji(1 – włączona, 0 – wyłączona)
- WorkorderID – zawiera unikalne ID pracującego urządzenia, jeśli urządzenia aktualnie nie pracuje jest wypełnione samymi zerami.
- GoodCount – zawiera ilość prawidłowo wykonanych produktów
- BadCount – zawiera ilość niepoprawnie wykonanych produktów
- Temperature – temperatura urządzenia w danym momencie
- DeviceErrors – zawiera numer błędu, który aktualnie jest na urządzeniu(znajduje się w wiadomości tylko wtedy jak błąd się zmieni, np. uszkodzenie czujnika temperatury)

Przykładowa wiadomość bez błędu na urządzeniu

Mon Oct 07 2024 09:18:01 GMT+0200 (czas środkowoeuropejski letni):

```
{
  "body": {
    "DeviceName": "Device 1",
    "ProductionStatus": 1,
    "WorkorderId": "a3ec8549-1e3e-4276-932c-9683e622f80d",
    "GoodCount": 9241,
    "BadCount": 936,
    "Temperature": 127.60520904281192
  },
  "enqueuedTime": "Mon Oct 07 2024 09:18:01 GMT+0200 (czas
środkowoeuropejski letni)"
}
```

Natomiast, gdy maszynie zaraportuje błąd tak wygląda wiadomość

Mon Oct 07 2024 09:38:02 GMT+0200 (czas środkowoeuropejski letni):

```
{
  "body": {
    "DeviceName": "Device 1",
    "ProductionStatus": 1,
    "WorkorderId": "a3ec8549-1e3e-4276-932c-9683e622f80d",
    "Temperature": 82.56142721706901,
    "GoodCount": 14720,
    "BadCount": 1542,
    "DeviceErrors": 2
  },
  "enqueuedTime": "Mon Oct 07 2024 09:38:02 GMT+0200 (czas
środkowoeuropejski letni)"
}
```

4. Device Twin

Device Twin przechowuje nasz aktualny stan maszyny oraz porządkany stan maszyny (np. zmiana procentu produkcji)

- 1) Desired Device Twin – jeśli chcemy zarządzać procentem produkcji na danym urządzeniu, dzięki tej metodzie jest to wykonalne
 - Wstawianie danych odbywa się za pomocą formatu JSON, jeśli chcemy umieścić porządkany procent produkcji to w strukturze „desired” musimy dodać nowe pole w formacie „**deviceName_production_rate**” : **liczba**
 - deviceName musi zostać zastąpione nazwą urządzenia, które chcemy modyfikować oraz liczba musi być w zakresie od 0 do 100(% produkcji urządzenia) np. „Device1_production_rate”: 90
- 2) Reported Device Twin – każde urządzenie, które jest podpięte pod nasz serwer ma zapisany swój aktualny stan

- Format wiadomości błędu na urządzeniu jest w postaci „**deviceName_error_code**” : „**liczba**”

Gdzie „deviceName” oraz „liczba” są zastępowane aktualnymi danymi z urządzenia

Np. „Device1_error_code”:”0”

Kody błędów:

- Brak – 0 = 0000
- Emergency Stop = 1 = 0001
- Power Failure = 2 = 0010
- Sensor Failure = 4 = 0100
- Unknown = 8 = 1000

Przy czym te kody mogą się sumować np. Połączenie „Power Failure” z „Sensor Failure” oraz „Unknown” da nam liczbę błędu = **14**.

- Format wiadomości aktualnej produkcji na urządzeniu „**deviceName_production_rate**”:”**liczba**”

Gdzie „deviceName” oraz „liczba” są zastępowane aktualnymi danymi z urządzenia

Np. „Device1_production_rate”:”100”

Oznacza to, że aktualnie Device1 pracuje na 100%, liczba ta jest z przedziału od 0 do 100%.

Przykładowa zawartość

```
{
  "deviceId": "Device",
  "etag": "AAAAAAAAAB8=",
  "deviceEtag": "MTE4NjQyODA3",
  "status": "enabled",
  "statusUpdateTime": "0001-01-01T00:00:00Z",
  "connectionState": "Connected",
  "lastActivityTime": "2024-10-07T08:37:39.9531214Z",
  "cloudToDeviceMessageCount": 0,
  "authenticationType": "sas",
  "x509Thumbprint": {
    "primaryThumbprint": null,
    "secondaryThumbprint": null
  },
  "modelId": "",
  "version": 519,
  "properties": {
    "desired": {
      "restartCount": "false",
      "Device1_production_rate": 100,
      "Device2_production_rate": 100,
      "Device4_production_rate": 70,
      "Device3_production_rate": 100,
      "$metadata": {
        "$lastUpdated": "2024-10-07T07:05:12.0122396Z",
        "$lastUpdatedVersion": 31,
        "restartCount": {
          "$lastUpdated": "2024-10-07T07:05:12.0122396Z",
          "$lastUpdatedVersion": 31
        },
        "Device1_production_rate": {
          "$lastUpdated": "2024-10-07T07:05:12.0122396Z",
          "$lastUpdatedVersion": 31
        },
        "Device2_production_rate": {
          "$lastUpdated": "2024-10-07T07:05:12.0122396Z",
          "$lastUpdatedVersion": 31
        },
        "Device4_production_rate": {
          "$lastUpdated": "2024-10-07T07:05:12.0122396Z",
          "$lastUpdatedVersion": 31
        },
        "Device3_production_rate": {
          "$lastUpdated": "2024-10-07T07:05:12.0122396Z",
          "$lastUpdatedVersion": 31
        }
      },
      "$version": 31
    },
    "reported": {
      "Device1_error_code": "0",
      "Device1_production_rate": "100",
      "Device2_error_code": "14",
      "Device2_production_rate": "100",
      "Device3_error_code": "0",
      "Device3_production_rate": "100",
      "Device4_error_code": "0",
      "Device4_production_rate": "70",
      "Device5_error_code": "0",
      "Device5_production_rate": "50",
    }
  }
}
```

5. Direct Methods

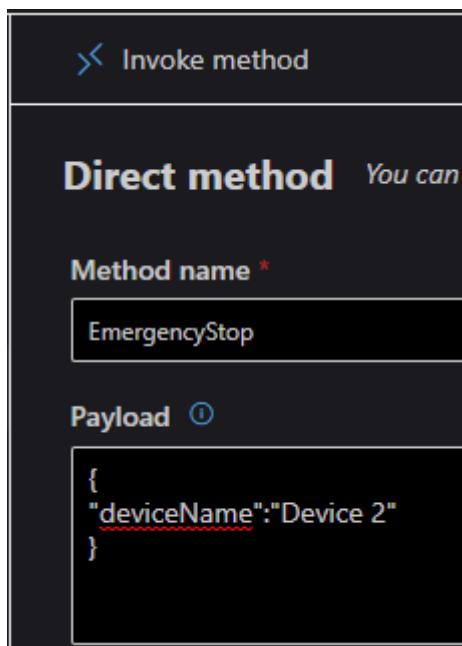
Direct Methods są zaimplementowane w kodzie aplikacji, które można potem wywoływać bezpośrednio z Azure

- 1) EmergencyStop – awaryjne zatrzymanie produkcji urządzenia, wywoływane w następujący sposób

„Method name” – tutaj wpisuje się nazwę metody, w tym wypadku „EmergencyStop”

„Payload” – tutaj przekazywane są dane w formacie JSON, które urządzenie ma zostać zatrzymane. Np.

```
{  
  „deviceName”: „Device 2”  
}
```



The screenshot shows the 'Invoke method' interface in the Azure portal. At the top, there is a header with a blue arrow icon and the text 'Invoke method'. Below this, the section is titled 'Direct method' with a subtitle 'You can'. There are two main input fields: 'Method name' and 'Payload'. The 'Method name' field contains the text 'EmergencyStop'. The 'Payload' field contains a JSON object:

```
{  
  "deviceName": "Device 2"  
}
```

. The 'Payload' field has a help icon (a circle with an 'i') next to it.

Stan urządzenia przed i po wykonaniu „EmergencyStop”

The screenshot shows the 'Industrial Device Simulator' window. On the left, a list of devices includes 'Device 2' which is highlighted. The main panel on the right displays the configuration for the selected device. The 'Production Status' checkbox is unchecked, with 'Start' and 'Stop' buttons to its right. The 'Production Rate' is set to 100, with '+' and '-' buttons. The 'Workorder ID' is '96d9475b-2b37-4d08-ab43-bae777b' and the 'Temperature' is 25. The 'Good Count' is 87 and the 'Bad Count' is 9. Under the 'Emergency Stop' section, the 'Unknown' checkbox is checked, while 'Emergency Stop', 'Power Failure', and 'Sensor Failure' are unchecked.

This screenshot shows the same 'Industrial Device Simulator' window after the 'Emergency Stop' has been triggered. The 'Emergency Stop' checkbox is now checked, and the 'Unknown' checkbox is unchecked. The other parameters, including 'Production Status', 'Production Rate', 'Workorder ID', 'Temperature', 'Good Count', and 'Bad Count', remain unchanged from the previous state.

Oraz dostajemy informację od naszej aplikacji o wywołaniu danej metody

```
{
  "DeviceName": "Device 2",
  "ProductionStatus": 0,
  "WorkorderId": "96d9475b-2b37-4d08-ab43-bae777b98f59",
  "GoodCount": 87,
  "BadCount": 9,
  "Temperature": 25.801352213031535
}

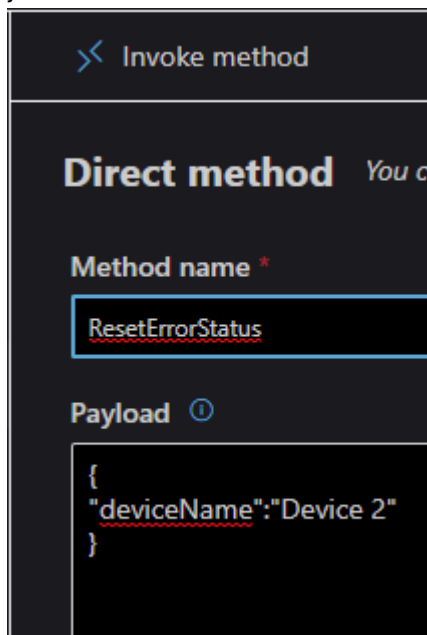
Emergency stop executed for: Device 2
```

2) ResetErrorStatus – kasuje błędy na urządzeniu

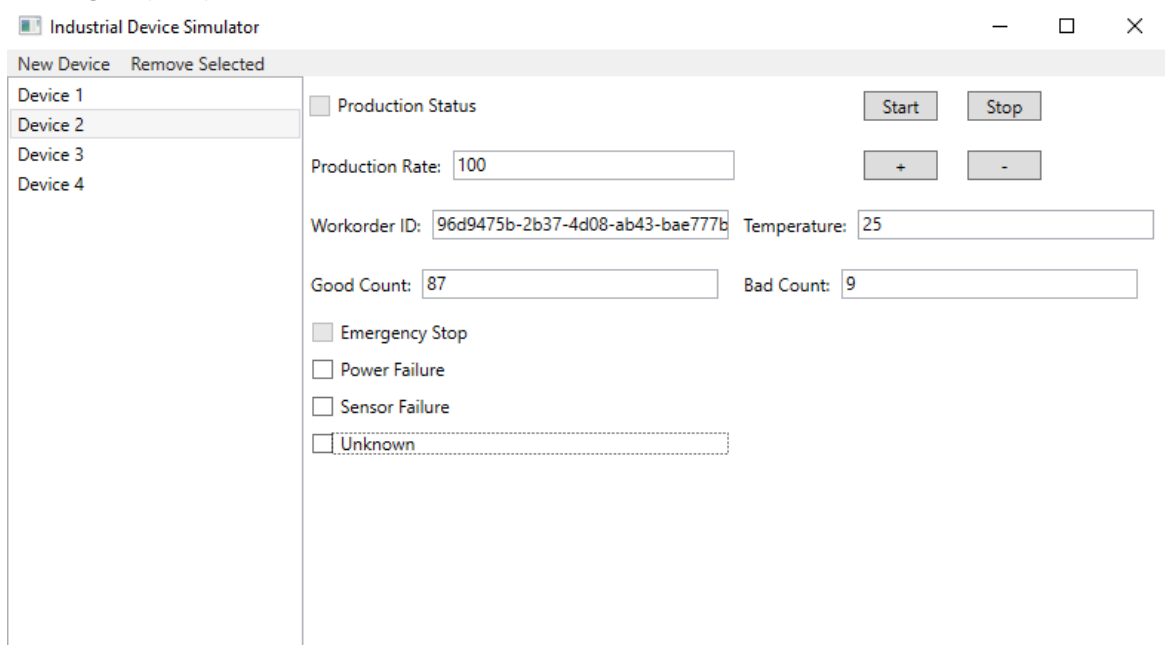
„Method name” – tutaj wpisuje się nazwę metody, w tym wypadku „ResetErrorStatus”

„Payload” – tutaj przekazywane są dane w formacie JSON, które urządzenie ma mieć skasowane błędy. Np.

```
{  
  „deviceName”: „Device 2”  
}
```



Stan aplikacji po wywołaniu „ResetErrorStatus” po uprzednim zatrzymaniu go poprzez „EmergencyStop”



Oraz informacja w aplikacji o wywołaniu metody

Reset error status executed for: Device 2

6. Data Calculations

Wszystkie zapytania dotyczące tego podpunktu znajdują się w osobnym pliku „Query.txt” – zapytania zostały umieszczone w usłudze „Azure Stream Analytics”. Dane wejściowe pobierane są z naszego urządzenia i przekazywane do OPC, które następnie trafia do IoT Huba i końcowo do „Azure Stream Analytics”, dane wyjściowe umieszczane są w usłudze „Storage Account” i każde z zapytań trafia do swojego kontenera.

Name
<input type="checkbox"/> \$logs
<input type="checkbox"/> device-errors
<input type="checkbox"/> production-kpi
<input type="checkbox"/> temp-asa
<input type="checkbox"/> temperature

1) Production KPIs

% dobrej produkcji w całkowitej objętości, pogrupowane według urządzenia w ciągu 5 minutowego okna – przechowywane w kontenerze „production-kpi” w Storage Account

Przykładowe dane wyjściowe:

```
{"DeviceName": "Device 1", "SnapshotDateTime": "2024-10-06T18:05:00.000000Z", "KPI": 90.41}
{"DeviceName": "Device 2", "SnapshotDateTime": "2024-10-06T18:05:00.000000Z", "KPI": 90.13}
{"DeviceName": "Device 3", "SnapshotDateTime": "2024-10-06T18:05:00.000000Z", "KPI": 91.14}
{"DeviceName": "Device 4", "SnapshotDateTime": "2024-10-06T18:05:00.000000Z", "KPI": 79.7}
{"DeviceName": "Device 1", "SnapshotDateTime": "2024-10-06T18:10:00.000000Z", "KPI": 90.69}
{"DeviceName": "Device 2", "SnapshotDateTime": "2024-10-06T18:10:00.000000Z", "KPI": 88.69}
{"DeviceName": "Device 3", "SnapshotDateTime": "2024-10-06T18:10:00.000000Z", "KPI": 90.51}
{"DeviceName": "Device 4", "SnapshotDateTime": "2024-10-06T18:10:00.000000Z", "KPI": 72.18}
```

2) Temperature

co 1 minutę wysyła średnią, minimalną i maksymalną temperaturę z ostatnich 5 minut (pogrupowane według urządzenia) - przechowywane w kontenerze „temperature” w Storage Account

Przykładowe dane wyjściowe:

```
{"DeviceName": "Device 1", "SnapshotDateTime": "2024-10-06T18:01:00.000000Z", "MinTemperature": 71.2375609325803, "MaxTemperature": 95.92494193793422, "AvgTemperature": 79.5913202982458}
{"DeviceName": "Device 2", "SnapshotDateTime": "2024-10-06T18:01:00.000000Z", "MinTemperature": 86.04569024347815, "MaxTemperature": 134.21082541167357, "AvgTemperature": 109.5306158715992}
{"DeviceName": "Device 3", "SnapshotDateTime": "2024-10-06T18:01:00.000000Z", "MinTemperature": 85.86734755139634, "MaxTemperature": 128.24539149175834, "AvgTemperature": 101.82880212713283}
{"DeviceName": "Device 4", "SnapshotDateTime": "2024-10-06T18:01:00.000000Z", "MinTemperature": 93.0, "MaxTemperature": 747.0, "AvgTemperature": 340.93395690119735}
{"DeviceName": "Device 1", "SnapshotDateTime": "2024-10-06T18:02:00.000000Z", "MinTemperature": 66.85967824208036, "MaxTemperature": 118.39321808127173, "AvgTemperature": 84.76091148268841}
{"DeviceName": "Device 2", "SnapshotDateTime": "2024-10-06T18:02:00.000000Z", "MinTemperature": 60.075174099639845, "MaxTemperature": 134.21082541167357, "AvgTemperature": 92.58608206103357}
{"DeviceName": "Device 3", "SnapshotDateTime": "2024-10-06T18:02:00.000000Z", "MinTemperature": 60.19928735590487, "MaxTemperature": 137.26656818413397, "AvgTemperature": 97.42602989514164}
{"DeviceName": "Device 4", "SnapshotDateTime": "2024-10-06T18:02:00.000000Z", "MinTemperature": -812.0, "MaxTemperature": 747.0, "AvgTemperature": 145.34197845059867}
```


3) Device errors

W sytuacji jeśli na urządzeniu pojawi się więcej, niż 3 błędy w ciągu 1 minuty – przechowywane w „device-errors” w Storage Account

Przykładowe dane wyjściowe:

```
{ "DeviceName": "Device 2", "SnapshotDateTime": "2024-10-06T10:45:05.558000Z", "ErrorCount": 4 }  
{ "DeviceName": "Device 2", "SnapshotDateTime": "2024-10-06T10:45:20.028000Z", "ErrorCount": 5 }  
{ "DeviceName": "Device 2", "SnapshotDateTime": "2024-10-06T10:45:22.132000Z", "ErrorCount": 4 }
```

7. Business logic

Zapytania wykorzystane do tego punktu znajdują się w osobnym pliku „Query.txt”. Do obsługi wszystkiego zostały wykorzystane usługi „Azure Stream Analytics”. Dane wejściowe pobierane są z naszego urządzenia i przekazywane do OPC, które następnie trafia do IoT Huba i końcowo do „Azure Stream Analytics”, które przekazuje dane do kolejek „Service Bus”

Name ↑↓	Status ↑↓
device-errors-bus	Active
production-kpi-bus	Active

- 1) Jeśli w urządzeniu wystąpią więcej niż 3 błędy w czasie krótszym niż 1 minuta
- Natychmiast wykonaj „EmergencyStop” na urządzeniu

Stan urządzenia przed:

Industrial Device Simulator

New Device Remove Selected

Device 1
Device 2
Device 3
Device 4

☒ Production Status Start Stop

Production Rate: 100 + -

Workorder ID: 2ba48597-d49a-4c2f-844e-8ccf2d82 Temperature: -506

Good Count: 1383 Bad Count: 303

☐ Emergency Stop

☒ Power Failure

☒ Sensor Failure

☒ Unknown

Stan urządzenia po wykonaniu:

Industrial Device Simulator

New Device Remove Selected

Device 1
Device 2
Device 3
Device 4

☒ Production Status Start Stop

Production Rate: 90 + -

Workorder ID: 2ba48597-d49a-4c2f-844e-8ccf2d82 Temperature: 620

Good Count: 1730 Bad Count: 488

☐ Emergency Stop

☒ Power Failure

☒ Sensor Failure

☒ Unknown

Oraz informacja w aplikacji

Production Rate decreased for: Device 2 by 10%

- 3) Jeśli urządzenia napotka błąd
 - Wyślij e-mail na wcześniej zdefiniowany adres

Brak implementacji.