

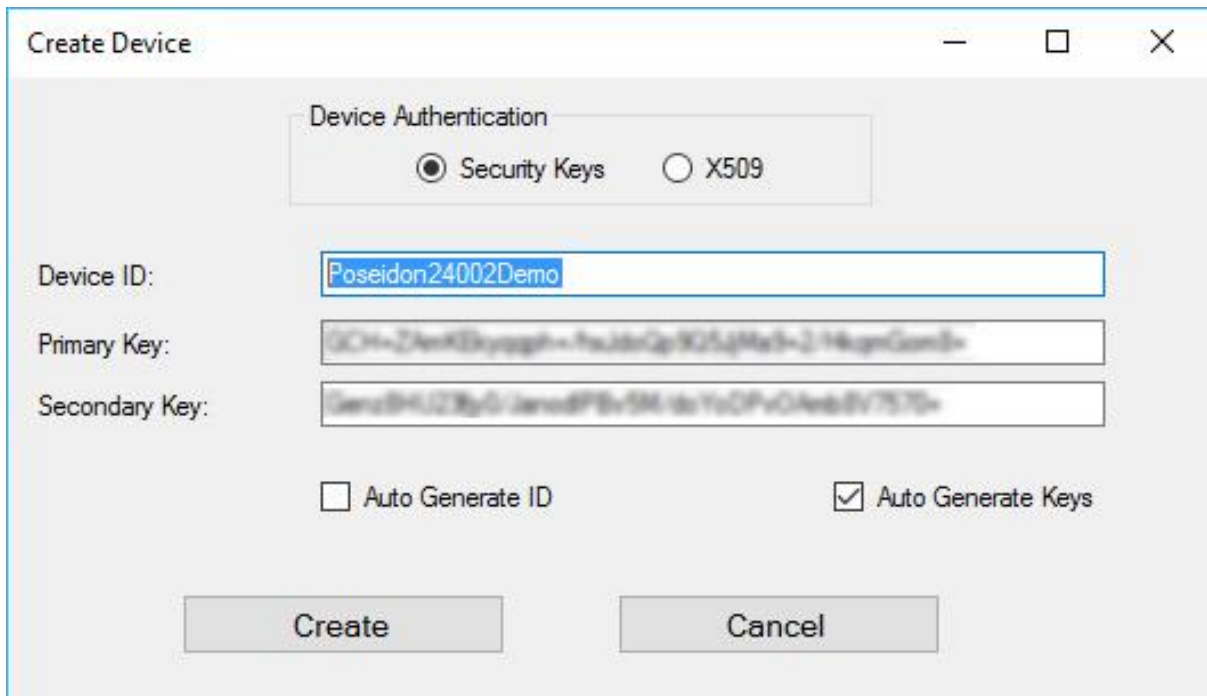
Use of Poseidon2 and Damocles2 with MS Azure

For work with Microsoft Azure MQTT protocol is used with Poseidon2 and Damocles2 units. It is necessary to have FW 3.2.6 and higher

For settings and activity verification we use *Device Explorer Twin* software. It can be downloaded on <https://github.com/Azure/azure-iot-sdks/releases/download/2016-11-17/SetupDeviceExplorer.msi> When open application we enter string into window IoT Hub Connection String, which was received when account MS Azure was created and we generate SAS key for required time. In our case 365 days:

The screenshot shows the 'Device Explorer Twin' application window. It has a menu bar with 'Configuration', 'Management', 'Data', 'Messages To Device', and 'Call Method on Device'. The 'Configuration' tab is active. Under 'Connection Information', there is a text area for the 'IoT Hub Connection String' containing a long alphanumeric string. Below it is a 'Protocol Gateway HostName:' field. An 'Update' button is present. The 'Shared Access Signature' section has fields for 'Key Name' (iothubowner), 'Key Value' (a long alphanumeric string), and 'Target' (hwg.azure.devices.net). A 'TTL (Days)' dropdown is set to 365. A 'Generate SAS' button is highlighted. At the bottom, there is a text area for the generated SAS token.

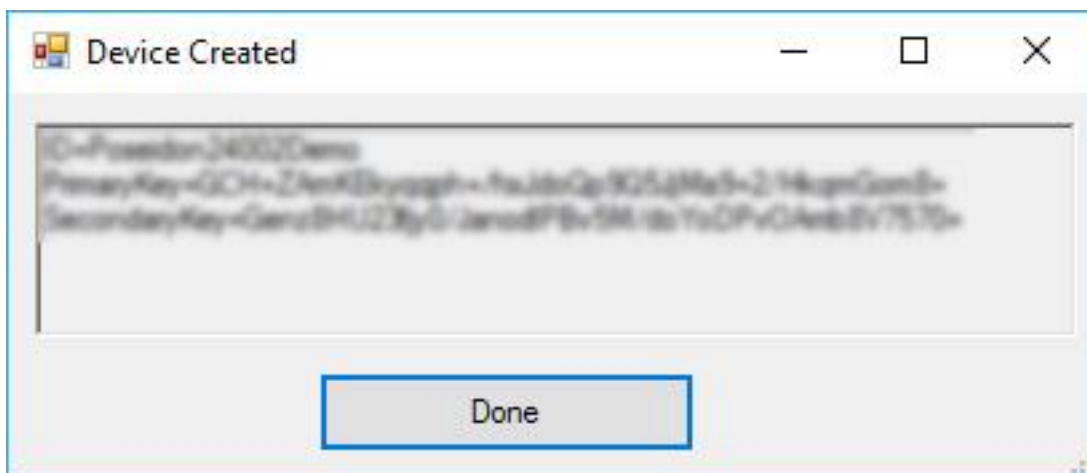
Now please switch on Management tab and create new product on *Device IP* with name e.g. Poseidon24002Demo



The 'Create Device' dialog box contains the following elements:

- Device Authentication:** A group box containing two radio buttons: 'Security Keys' (selected) and 'X509'.
- Device ID:** A text input field containing 'Poseidon24002Demo'.
- Primary Key:** A text input field containing a long alphanumeric string.
- Secondary Key:** A text input field containing a long alphanumeric string.
- Auto Generate ID:** An unchecked checkbox.
- Auto Generate Keys:** A checked checkbox.
- Buttons:** 'Create' and 'Cancel' buttons at the bottom.

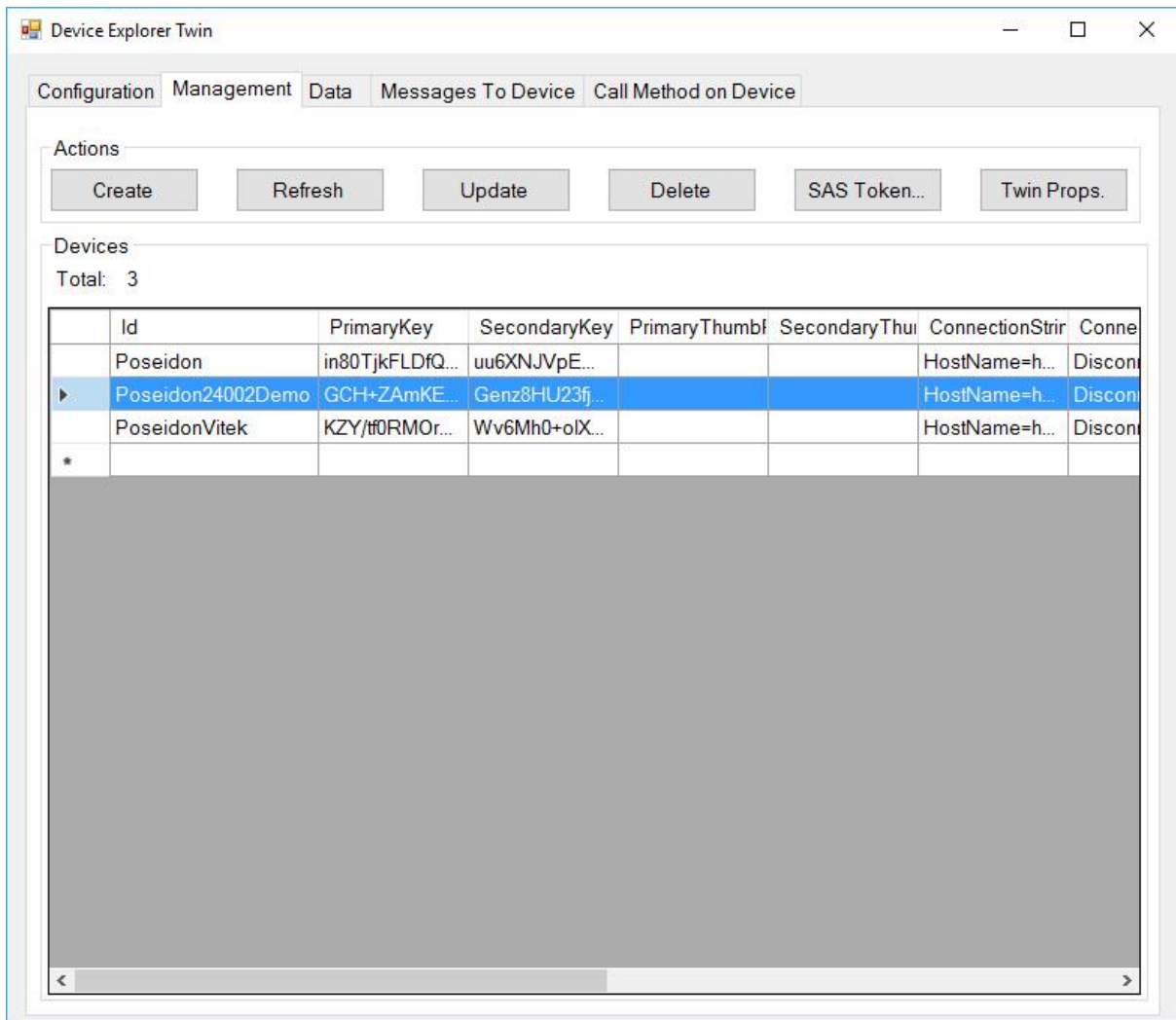
Creation is confirmed by this window:



The 'Device Created' confirmation dialog box displays the following information:

- Title:** Device Created
- Text:** ID=Poseidon_24002Demo
PrimaryKey=<GCH~ZAnK8kypph~FuJdrQp9Q5jMa9~214qgnGom8>
SecondaryKey=<Genc8H1J28y0JancdPBvSM id YuOPv0Amd8V757D>
- Button:** 'Done' button at the bottom.

Now you can see the device in the list of devices



We will select new device and by pressing SAS Token password can be generated:

Here we notice *DeviceID* column which contains right device and field *TTL (Days)* should have suitable length, the length indicate the length of validity:

SASTokenForm

DeviceID Poseidon24002Demo

DeviceKeys 5CH+Z4nK3kyagH+7uJdrGp9025jMe3+214q+Gmb+

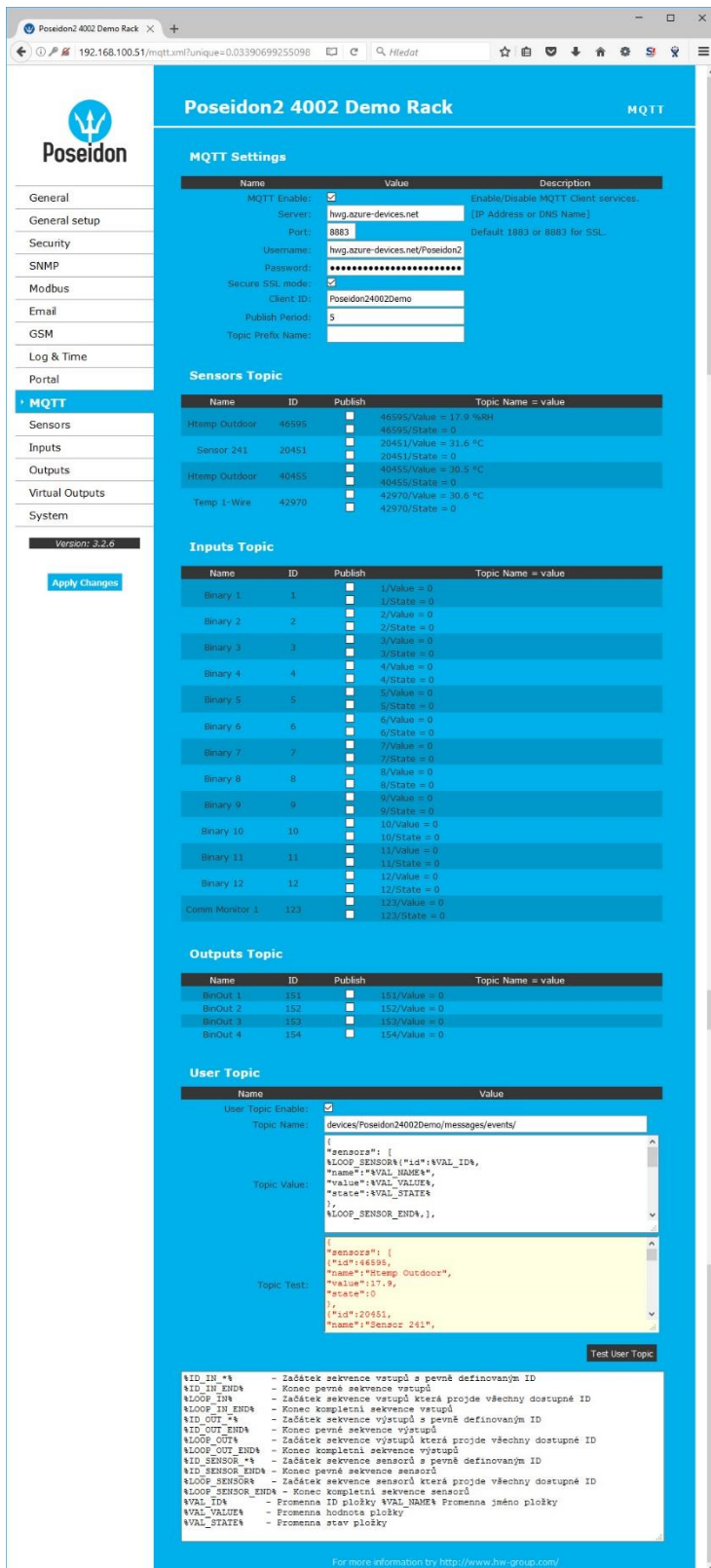
TTL (Days) 365

```
HostName=hwg.acure  
DeviceID=Poseidon24002Demo SharedAccessSignature sr  
hwg.acure-devices.net/1?deviceid=Poseidon24002Demo&sig=5  
CH+Z4nK3kyagH+7uJdrGp9025jMe3+214q+Gmb+  
17E6e35q09W2agp+L2B+L2B+K7pOCrYHQGp+K0UQwF207L3D4e+  
519737833
```

Generate Done

We copy password from Token. It contains everything what starts with string *SharedAccessSignature sr*

Now we set Poseidon unit:



MQTT Settings

| Name | Value | Description |
|--------------------|-------------------------------------|--------------------------------------|
| MQTT Enable: | <input checked="" type="checkbox"/> | Enable/Disable MQTT Client services. |
| Server: | hwg.azure-devices.net | [IP Address or DNS Name] |
| Port: | 8883 | Default 1883 or 8883 for SSL. |
| Username: | hwg.azure-devices.net/Poseidon2 | |
| Password: | ***** | |
| Secure SSL mode: | <input checked="" type="checkbox"/> | |
| Client ID: | Poseidon24002Demo | |
| Publish Period: | 5 | |
| Topic Prefix Name: | | |

Sensors Topic

| Name | ID | Publish | Topic Name = value |
|---------------|-------|--------------------------|-----------------------|
| Htemp Outdoor | 46595 | <input type="checkbox"/> | 46595/Value = 17.9 °C |
| Sensor 241 | 20451 | <input type="checkbox"/> | 20451/Value = 31.6 °C |
| Htemp Outdoor | 40455 | <input type="checkbox"/> | 40455/Value = 30.5 °C |
| Temp 1-Wire | 42970 | <input type="checkbox"/> | 42970/Value = 30.6 °C |

Inputs Topic

| Name | ID | Publish | Topic Name = value |
|----------------|-----|--------------------------|--------------------|
| Binary 1 | 1 | <input type="checkbox"/> | 1/Value = 0 |
| Binary 2 | 2 | <input type="checkbox"/> | 2/Value = 0 |
| Binary 3 | 3 | <input type="checkbox"/> | 3/Value = 0 |
| Binary 4 | 4 | <input type="checkbox"/> | 4/Value = 0 |
| Binary 5 | 5 | <input type="checkbox"/> | 5/Value = 0 |
| Binary 6 | 6 | <input type="checkbox"/> | 6/Value = 0 |
| Binary 7 | 7 | <input type="checkbox"/> | 7/Value = 0 |
| Binary 8 | 8 | <input type="checkbox"/> | 8/Value = 0 |
| Binary 9 | 9 | <input type="checkbox"/> | 9/Value = 0 |
| Binary 10 | 10 | <input type="checkbox"/> | 10/Value = 0 |
| Binary 11 | 11 | <input type="checkbox"/> | 11/Value = 0 |
| Binary 12 | 12 | <input type="checkbox"/> | 12/Value = 0 |
| Comm Monitor 1 | 123 | <input type="checkbox"/> | 123/Value = 0 |

Outputs Topic

| Name | ID | Publish | Topic Name = value |
|---------|-----|--------------------------|--------------------|
| BrOut 1 | 151 | <input type="checkbox"/> | 151/Value = 0 |
| BrOut 2 | 152 | <input type="checkbox"/> | 152/Value = 0 |
| BrOut 3 | 153 | <input type="checkbox"/> | 153/Value = 0 |
| BrOut 4 | 154 | <input type="checkbox"/> | 154/Value = 0 |

User Topic

User Topic Enable:

Topic Name: devices/Poseidon24002Demo/messages/events/

Topic Value:

```
{
  "sensors": [
    {
      "id": "46595",
      "name": "Htemp Outdoor",
      "value": 17.9,
      "state": 0
    },
    {
      "id": "20451",
      "name": "Sensor 241",
      "value": 31.6,
      "state": 0
    }
  ]
}
```

Topic Test:

```
{
  "sensors": [
    {
      "id": "46595",
      "name": "Htemp Outdoor",
      "value": 17.9,
      "state": 0
    },
    {
      "id": "20451",
      "name": "Sensor 241",
      "value": 31.6,
      "state": 0
    }
  ]
}
```

Test User Topic

Legend:

- #ID_IN_* - Začátek sekvence vstupů s pevně definovaným ID
- #ID_IN_END# - Konec pevné sekvence vstupů
- #LOOP_IN - Začátek sekvence vstupů která projde všechny dostupné ID
- #LOOP_IN_END# - Konec kompletní sekvence vstupů
- #ID_OUT_* - Začátek sekvence výstupů s pevně definovaným ID
- #ID_OUT_END# - Konec pevné sekvence výstupů
- #LOOP_OUT - Začátek sekvence výstupů která projde všechny dostupné ID
- #LOOP_OUT_END# - Konec kompletní sekvence výstupů
- #ID_SENSOR_* - Začátek sekvence senzorů s pevně definovaným ID
- #ID_SENSOR_END# - Konec pevné sekvence senzorů
- #LOOP_SENSOR - Začátek sekvence senzorů která projde všechny dostupné ID
- #LOOP_SENSOR_END# - Konec kompletní sekvence senzorů
- #VAL_ID# - Proměnná ID položky #VAL_NAME# Proměnná jméno položky
- #VAL_VALUE# - Proměnná hodnota položky
- #VAL_STATE# - Proměnná stav položky

On MQTT page we enable MQTT protocol support. Enter server address which was assigned to us when creating MS Azure account, port add 8883 and at the same time we check *Secure SSL*

Mode. Into field *Password* we enter password generated by previous step. *Username* is always a compound of server address and *DeviceID*, for example `hwg.azure-devices.net/Poseidon24002Demo` and *Client ID* is in this case on same level as *DeviceID*.

At the same time, you still need to allow user defined *Topic* and as *Topic* name set: `devices/YOUR_DEVICE_ID/messages/events/`, for example `devices/Poseidon24002Demo/messages/events/`:

Into *Topic Value* we can fill definated content using macros:

// XML template

```
<?xml version="1.0" encoding="utf-8" ?>
<BinaryInSet>%LOOP_IN%<Entry>
  <ID>%VAL_ID%</ID>
  <Name>%VAL_NAME%</Name>
  <Value>%VAL_VALUE%</Value>
  <State>%VAL_STATE%</State>
</Entry>%LOOP_IN_END%</BinaryInSet>

<BinaryOutputSet>%LOOP_OUT%<Entry>
  <ID>%VAL_ID%</ID>
  <Name>%VAL_NAME%</Name>
  <Value>%VAL_VALUE%</Value>
  <State>%VAL_STATE%</State>
</Entry>%LOOP_OUT_END%</BinaryOutputSet>

<SensorSet>%LOOP_SENSOR%<Entry>
  <ID>%VAL_ID%</ID>
  <Name>%VAL_NAME%</Name>
  <Value>%VAL_VALUE%</Value>
  <State>%VAL_STATE%</State>
</Entry>%LOOP_SENSOR_END%</SensorSet>

<BinaryOutputSet>%ID_OUT_3%<Entry>
  <ID>%VAL_ID%</ID>
  <Name>%VAL_NAME%</Name>
  <Value>%VAL_VALUE%</Value>
  <State>%VAL_STATE%</State>
</Entry>%ID_OUT_END%</BinaryOutputSet>
```

// Json template

```
{
  "sensors": [
    %LOOP_SENSOR%{"id":%VAL_ID%,
    "name": "%VAL_NAME%",
    "value": %VAL_VALUE%,
```

```
    "state":%VAL_STATE%
  },
  %LOOP_SENSOR_END%, ],
  "inputs": [
    %LOOP_IN%{"id":%VAL_ID%,
    "name": "%VAL_NAME%",
    "value":%VAL_VALUE%,
    "state":%VAL_STATE%
  },
  %LOOP_IN_END%, ],
  "outputs": [
    %LOOP_OUT%{"id":%VAL_ID%,
    "name": "%VAL_NAME%",
    "value":%VAL_VALUE%,
  },
  %LOOP_OUT_END%, ]
}
```

