



## **TwinCAT 3 IoT Communicator**

**Konfiguracja i uruchomienie TwinCAT 3 IoT Communicator  
od strony sterownika i urządzenia mobilnego**

Wersja dokumentacji 2.2

Aktualizacja: 05.01.2024

Kontakt: *support@beckhoff.pl*

Beckhoff Automation Sp. z o. o.

## Spis treści

1	Wstęp.....	5
1.1	Wykorzystana konfiguracja .....	5
1.2	Opis technologii .....	5
2	Przygotowanie sterownika oraz programu PLC.....	6
2.1	Uruchomienie brokera MQTT.....	6
2.2	Przygotowanie sterownika .....	6
2.3	Przygotowanie struktury danych .....	7
2.4	Przygotowanie funkcji do komunikacji .....	7
2.4.1	Komunikacja bez podawania użytkownika i hasła .....	8
2.4.2	Komunikacja z podawaniem nazwy użytkownika i hasła .....	8
2.5	Przygotowanie struktury programu.....	9
3	Aplikacja mobilna.....	11
3.1	Uruchomienie .....	11
3.1.1	Uruchomienie bez nazwy użytkownika i hasła .....	11
3.1.2	Uruchomienie chronione nazwą użytkownika i hasłem .....	12
3.2	Dodatkowe funkcje.....	12
3.2.1	Nadpisywanie wartości.....	12
3.2.2	Toggle booleans.....	12
3.2.3	Live graph.....	13
4	Informacje dodatkowe .....	14

© Beckhoff Automation Sp. z o.o.

*Wszystkie obrazy są chronione prawem autorskim. Wykorzystywanie i przekazywanie osobom trzecim jest niedozwolone.*

*Beckhoff®, TwinCAT®, EtherCAT®, Safety over EtherCAT®, TwinSAFE®, XFC® i XTS® są zastrzeżonymi znakami towarowymi i licencjonowanymi przez Beckhoff Automation GmbH. Inne oznaczenia użyte w niniejszej prezentacji mogą być znakami towarowymi, których użycie przez osoby trzecie do własnych celów może naruszać prawa właścicieli.*

*Informacje przedstawione w tej prezentacji zawierają jedynie ogólne opisy lub cechy wydajności, które w przypadku rzeczywistego zastosowania nie zawsze mają zastosowanie zgodnie z opisem lub które mogą ulec zmianie w wyniku dalszego rozwoju produktów. Obowiązek przedstawienia odpowiednich cech istnieje tylko wtedy, gdy zostanie to wyraźnie uzgodnione w warunkach umowy.*

Uwaga! Poniższy dokument zawiera przykładowe zastosowanie produktu oraz zbiór zaleceń i dobrych praktyk. Służy on wyłącznie celom szkoleniowym i wymaga szeregu dalszych modyfikacji przed zastosowaniem w rzeczywistej aplikacji. Autor dokumentu nie ponosi żadnej odpowiedzialności za niewłaściwe wykorzystanie produktu. Dany dokument w żadnym stopniu nie zastępuje dokumentacji technicznej dostępnej online na stronie [infosys.beckhoff.com](http://infosys.beckhoff.com).

## 1 Wstęp

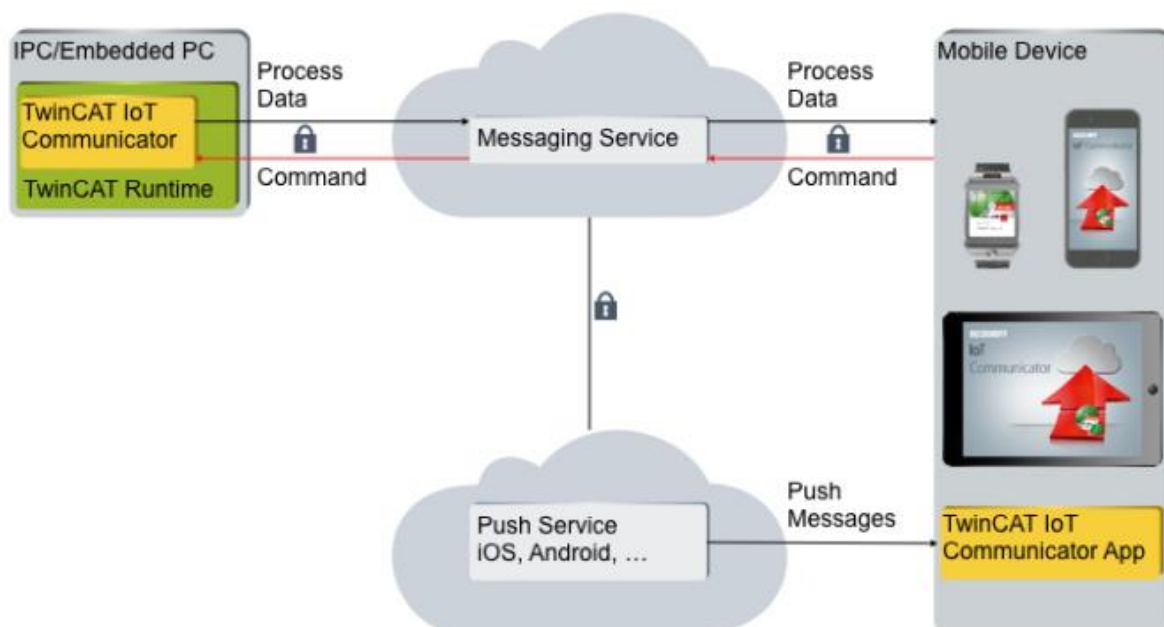
### 1.1 Wykorzystana konfiguracja

W celu opracowania instrukcji posłużono się następującą konfiguracją sprzętową:

- Sterownik C6015-0010 z systemem operacyjnym Windows 10 oraz TwinCAT 3.1.4024.53
- Telefon komórkowy z dostępem do Internetu oraz pobraną aplikacją TwinCAT IoT
- Komputer z uruchomionym brokerem MQTT

### 1.2 Opis technologii

Biblioteka Tc3\_IotCommunicator pozwala na wymianę danych pomiędzy programem PLC a brokerem poprzez protokół komunikacyjny MQTT. Rozwiązanie takie pozwala np. na zdalne odczytywanie zmiennych statusowych procesu (subscribe mode) lub zadawanie parametrów (publish mode) bez konieczności fizycznego przebywania w pobliżu sterownika czy komputera, z którego sterownik był programowany. W przykładzie przedstawiona zostanie komunikacja dla dwóch pokoi, z czego jeden będzie zabezpieczony nazwą użytkownika i hasłem, a drugi nie.



## 2 Przygotowanie sterownika oraz programu PLC

### 2.1 Uruchomienie brokera MQTT

W tej instrukcji instalacja brokera MQTT zostanie przedstawiona w sposób skrócony. W celu otrzymania dokładnej instrukcji prosimy o kontakt poprzez skrzynkę mailową [support@beckhoff.pl](mailto:support@beckhoff.pl).

W pierwszej kolejności należy zainstalować broker MQTT <https://mosquitto.org/download/> (w naszym przypadku będzie to Eclipse Mosquitto), a następnie odblokować port firewall 1883 na urządzeniu serwera oraz klienta.

Dla wersji 2.0.0 oraz nowszych, wymagana jest zmiana domyślnej konfiguracji, aby zezwolić na dostęp innych urządzeń do brokera. W tym celu należy edytować plik konfiguracyjny, znajdujący się w folderze instalacyjnym, o nazwie **mosquitto.conf**. Potrzebne parametry:

- **listener 1883 0.0.0.0** – deklarujemy port oraz adres po których będzie przebiegała komunikacja (0.0.0.0 oznacza dostęp z dowolnego adresu)
- **allow\_anonymous true** – zezwalamy na połączenia z innych urządzeń

```
232 #  
233 # listener port-number [ip address/host name/unix socket path]  
234 listener 1883 0.0.0.0
```

```
529 # Defaults to false, unless there are no listeners defined in the configuration  
530 # file, in which case it is set to true, but connections are only allowed from  
531 # the local machine.  
532 allow_anonymous true
```

Aby załadować nową konfigurację, należy zapisać edytowany plik oraz w linii poleceń cmd uruchomić broker:  
**mosquitto -v -c mosquitto.conf**

```
C:\Users\Administrator>cd C:\Program Files\mosquitto  
C:\Program Files\mosquitto>\mosquitto -v -c mosquitto.conf
```

Należy pamiętać, aby broker był cały czas uruchomiony podczas komunikacji!

Pełna dokumentacja pliku konfiguracyjnego dostępna na stronie [mosquitto.org/man/mosquitto-conf-5](https://mosquitto.org/man/mosquitto-conf-5).

### 2.2 Przygotowanie sterownika

W pierwszej kolejności należy połączyć się ze sterownikiem, a następnie odblokować w firewall port 1883 dla połączeń przychodzących i wychodzących. Dodatkowa instalacja biblioteki nie jest konieczna, ponieważ biblioteka Tc3\_lotCommunicator jest domyślnie zainstalowana w wersji TwinCAT 3.1.4022 oraz wyższych. W przypadku uruchamiania brokera MQTT na urządzeniu posiadającym adres IP przydzielany przy pomocy serwera DHCP należy się upewnić, czy urządzenie mobilne, sterownik i urządzenie na którym uruchomiony jest broker znajdują się w tej samej podsięci.

## 2.3 Przygotowanie struktury danych

Należy uruchomić środowisko TwinCAT XAE, a następnie dołączyć do programu PLC biblioteki Tc3\_IotCommunicator oraz Tc3\_Module. W następnej kolejności należy utworzyć strukturę danych, która przesyłana będzie przy pomocy protokołu MQTT, np. jak poniżej:

```
TYPE ST_ProcessData :  
STRUCT  
  {attribute 'iot.DisplayName' := 'Kitchen Lights'}  
  bLamp1 : BOOL;  
  
  {attribute 'iot.DisplayName' := 'Living Room Lights'}  
  bLamp2 : BOOL;  
  
  {attribute 'iot.DisplayName' := 'Outside Temperature'}  
  {attribute 'iot.ReadOnly' := 'false'}  
  {attribute 'iot.Unit' := 'Celsius'}  
  {attribute 'iot.MinValue' := '5'}  
  {attribute 'iot.MaxValue' := '30'}  
  nTemp : REAL;  
END_STRUCT  
END_TYPE
```

Atrybut `iot.DisplayName` odpowiada za wyświetlanie nazwy zmiennej w urządzeniu klienta, atrybut `ReadOnly` pozwala na ustawienie braku możliwości zmiany wartości zmiennej, atrybut `Unit` określa jednostkę w jakiej wyświetlana jest wartość, a atrybut `MinValue` i `MaxValue` pozwala określić zakres wyświetlania zmiennej w urządzeniu klienta.

## 2.4 Przygotowanie funkcji do komunikacji

W celu zapewnienia komunikacji przy pomocy protokołu MQTT używa się bloku funkcyjnego FB\_IotCommunicator dostępnego w bibliotece Tc3\_IotCommunicator. Blok ten posiada następujące wejścia:

- `sHostName` – adres IP lub Hostname brokera MQTT
- `nPort` – port wykorzystywany do komunikacji (w naszym przypadku port 1883)
- `sClientID` – opcjonalne wejście dla ustawienia unikalnej nazwy klienta
- `sMainTopic` – nazwa głównego tematu
- `sDeviceName` – nazwa pokoju
- `sUser` – nazwa użytkownika (gdy skonfigurowane w brokerze)
- `sPassword` – hasło (gdy skonfigurowane w brokerze)
- `stTLS` – struktura dla komunikacji zabezpieczanej przy pomocy TLS
- `bRetain` – zmienna dla ustalenia, czy broker ma przechowywać poprzednie wiadomości
- `eQoS` – zmienna dla „Quality of Service”

Wyjścia:

- `bError` – gdy wystąpi błąd
- `hrErrorCode` – kod błędu
- `eConnectionState` – stan komunikacji między klientem i brokerem
- `bConnected` – TRUE jeśli jest poprawna komunikacja między klientem i brokerem
- `fbCommand` – wyjście do ewaluacji otrzymanych danych („komend”)

Oraz metody:

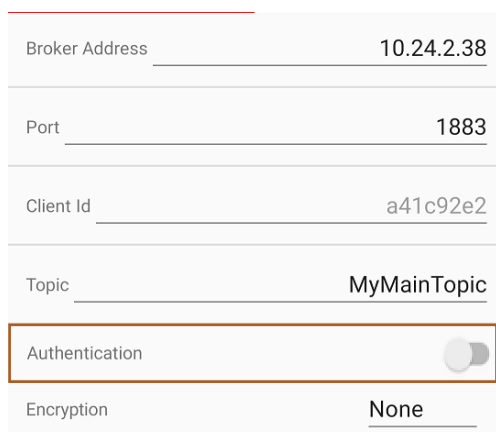
- Execute – wywoływana cyklicznie dla utrzymania komunikacji
- SendData – metoda do wysłania danych do brokera
- SendMessage – metoda do wysłania wiadomości do brokera

#### 2.4.1 Komunikacja bez podawania użytkownika i hasła

W celu zapewnienia podstawowej komunikacji (bez zabezpieczenia przy pomocy podawania nazwy użytkownika i hasła), konfiguracja wygląda jak na obrazku poniżej (deklarujemy tylko 4 zmienne wejściowe):

```
fblot      : FB_lotCommunicator := (  
                                sHostName      := '10.24.2.38',  
                                nPort          := 1883,  
                                sMainTopic     := 'MyMainTopic',  
                                sDeviceName    := 'Room One');
```

W aplikacji mobilnej w zakładce Settings odznaczamy opcję „Authentication”:



Należy pamiętać, że broker MQTT również powinien być uruchomiony bez opcji zakładającej nazwę użytkownika i hasło dla komunikacji.

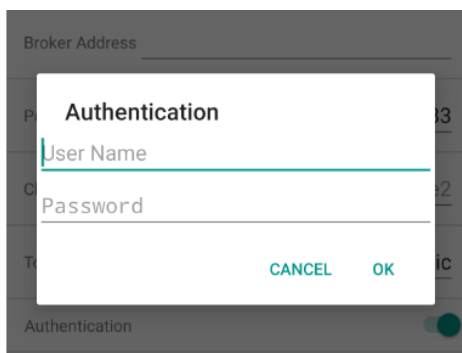
#### 2.4.2 Komunikacja z podawaniem nazwy użytkownika i hasła

W celu zapewnienia komunikacji bezpieczniejszej (zabezpieczonej nazwą użytkownika i hasłem) poza zmiennymi zadeklarowanymi w przykładzie powyżej należy zadeklarować również w bloku zmienne sUser oraz sPassword, jak na przykładzie poniżej (muszą być one zgodne z użytkownikiem i hasłem skonfigurowanymi dla brokera):

```
fblot2     : FB_lotCommunicator := (  
                                sHostName      := '10.24.2.38',  
                                nPort          := 1883,  
                                sMainTopic     := 'MyMainTopic',  
                                sDeviceName    := 'Room One',  
                                sUser          := 'UserA',  
                                sPassword     := 'PassA');
```

W aplikacji mobilnej należy zaznaczyć opcję „Authentication”, a następnie wpisać nazwę użytkownika i hasło w oknie, które się pojawi:





Należy pamiętać, że broker MQTT również powinien być uruchomiony z opcją zakładającą nazwę użytkownika i hasło dla komunikacji.

## 2.5 Przygotowanie struktury programu

W celu zapewnienia cyklicznej komunikacji klienta z brokerem należy co cykl wywoływać metodę Execute dla funkcji FB\_lotCommunicator:

```
fbloT.Execute(TRUE); //keep communication alive
fbloT2.Execute(TRUE); //keep communication alive
```

Przesyłanie danych realizowane będzie co 500 ms przy pomocy timera TON, w przypadku gdy wyjście bConnected bloku będzie w stanie TRUE (komunikacja będzie poprawna):

```
stData          : ST_ProcessData;
stData2         : ST_ProcessData2;

timer(IN := NOT timer.Q, PT := T#500MS); //cyclic message sending

IF bSendMessage THEN //if TRUE
    bSendMessage := FALSE; //set to FALSE
    fbloT.SendMessage(sMessage); //send message
END_IF

IF fbloT.bConnected AND timer.Q THEN //if 500 ms passed and communication ok
    fbloT.SendData(ADR(stData), SIZEOF(stData)); //send data
END_IF

IF fbloT2.bConnected AND timer.Q THEN //if 500 ms passed and communication ok
    fbloT2.SendData(ADR(stData2), SIZEOF(stData2)); //send data
END_IF
```

Powyższy fragment kodu służy do odczytu informacji z programu PLC np. za pomocą urządzenia mobilnego. Aby możliwe było nadpisywanie wartości zmiennych z poziomu urządzenia klienta należy zaimplementować fragment kodu jak poniżej:

```
IF fbloT.fbCommand.bAvailable THEN //if command sending available
    IF fbloT.fbCommand.sVarName = 'bLamp1' THEN //if Lamp 1
        fbloT.fbCommand.GetValue(ADR(stData.bLamp1), SIZEOF(stData.bLamp1), E_lotCommunicatorDatatype.type_BOOL); //set new value
    ELSEIF fbloT.fbCommand.sVarName = 'bLamp2' THEN //if Lamp 2
        fbloT.fbCommand.GetValue(ADR(stData.bLamp2), SIZEOF(stData.bLamp2), E_lotCommunicatorDatatype.type_BOOL); //set new value
    ELSEIF fbloT.fbCommand.sVarName = 'nTemp' THEN //if Temp
        fbloT.fbCommand.GetValue(ADR(stData.nTemp), SIZEOF(stData.nTemp), E_lotCommunicatorDatatype.type_REAL); //set new value
    END_IF
```

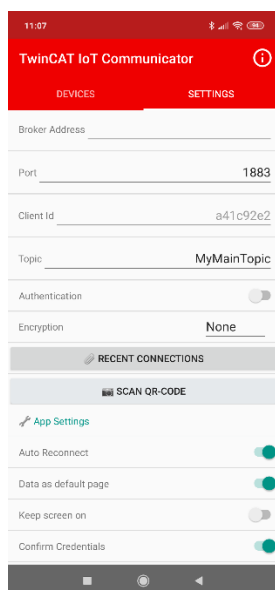
```
fbIoT.fbCommand.Remove();  
END_IF
```

```
//discard command
```

Kod ten pozwala na nadpisanie wartości zmiennej w przypadku gdy dostępna jest możliwość wysłania nowej komendy do brokera (fbIoT.fbCommand.bAvailable).

### 3 Aplikacja mobilna

W celu odczytu i zapisywania danych w programie PLC z poziomu telefonu komórkowego należy pobrać i zainstalować aplikację TwinCAT IoT (dostępna w sklepie Google Play oraz AppStore). Następnie w aplikacji należy przejść do zakładki Settings, gdzie dokonujemy konfiguracji urządzenia klienta. W polu Broker Address wpisujemy adres IP lub Hostname brokera, w polu Port wpisujemy port komunikacyjny, w polu Client ID ID klienta (o ile zadeklarowane w programie PLC), w polu Topic nazwę tematu, a opcję Authentication zaznaczamy w zależności od rodzaju komunikacji (opisane wcześniej). Pole Encryption pozwala na wybranie kodowania, o ile zostało zaimplementowane w brokerze.

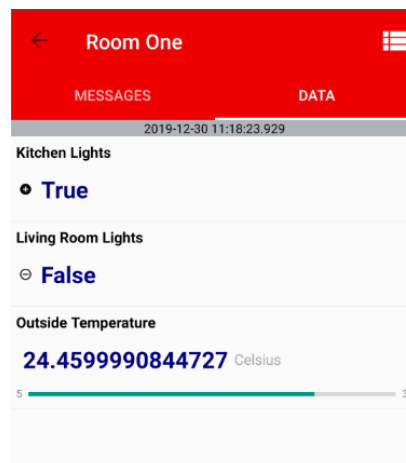
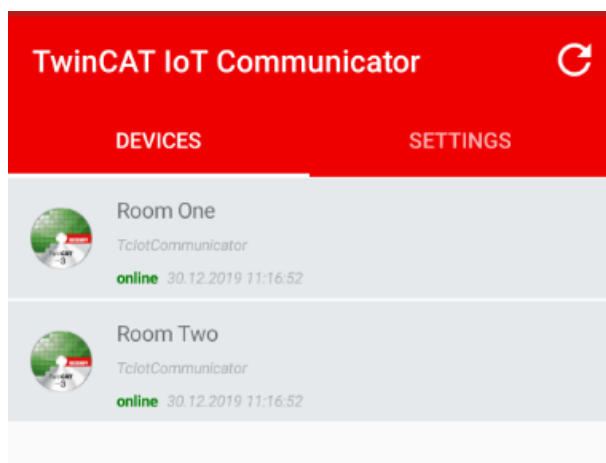


#### 3.1 Uruchomienie

Po uruchomieniu programu PLC na sterowniku oraz aplikacji mobilnej otrzymujemy możliwość odczytywania i zmiany wartości parametrów programu PLC przy pomocy aplikacji mobilnej, jak przedstawiono to na poniższym obrazku:

##### 3.1.1 Uruchomienie bez nazwy użytkownika i hasła

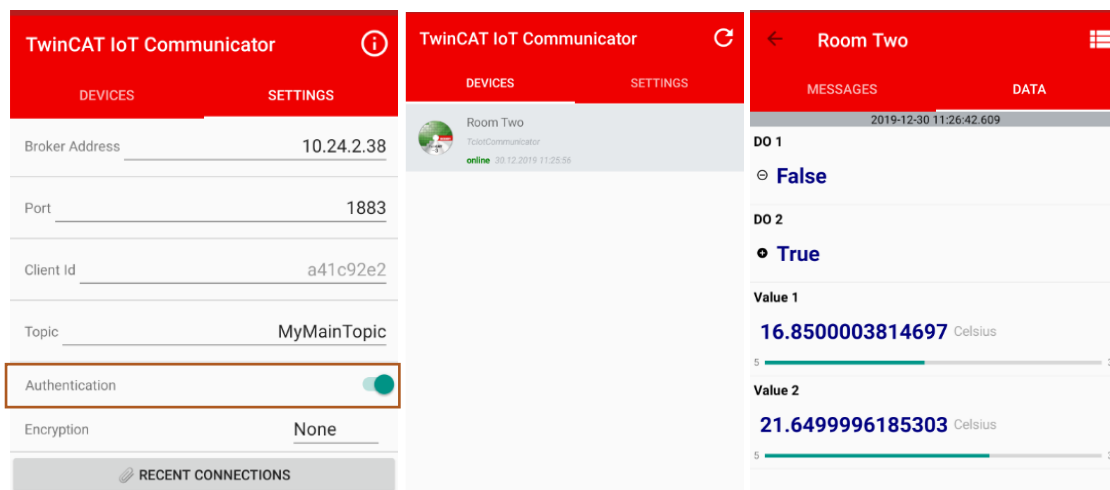
W pierwszej kolejności uruchomiono komunikację niechronioną nazwą użytkownika i hasłem. Widok działającej aplikacji przedstawiono na obrazkach poniżej:



Jak można zauważyć, oba pokoje są w trybie online, mimo że drugi z nich został zadeklarowany jako chroniony nazwą użytkownika i hasłem. Spowodowane jest to uruchomieniem brokera bez opcji logowania użytkowników.

### 3.1.2 Uruchomienie chronione nazwą użytkownika i hasłem

Następnie uruchomiono komunikację wymagającą zalogowania się przy pomocy nazwy użytkownika i hasła. Konfiguracja i widok działającej aplikacji przedstawiono na obrazkach poniżej:

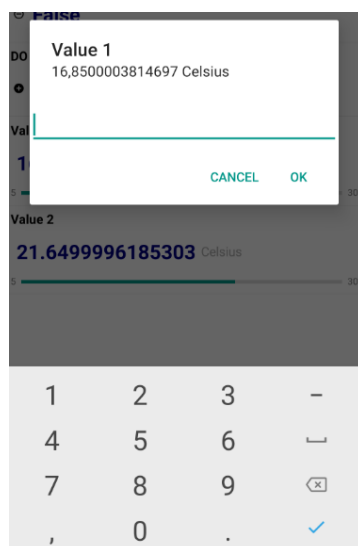


Jak można zauważyć, w zakładce Devices widoczny jest tylko Room 2. Spowodowane jest to uruchomieniem brokera z opcją logowania użytkowników, a niezaimplementowaniem tej opcji przy deklaracji bloku funkcyjnego do komunikacji dla pokoju 1.

## 3.2 Dodatkowe funkcje

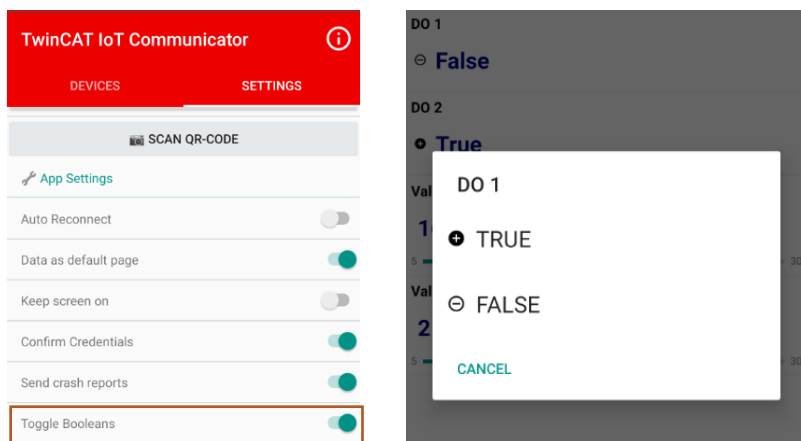
### 3.2.1 Nadpisywanie wartości

Opcja nadpisywania wartości dostępna jest po kliknięciu na daną wartość, a następnie pojawi się możliwość wpisania żądanej wartości.



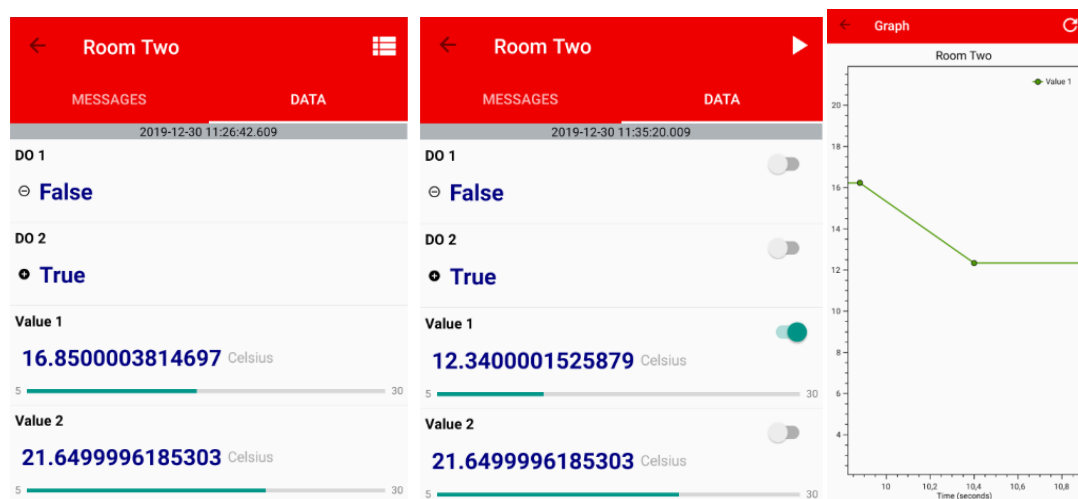
### 3.2.2 Toggle booleans

Opcja „Toggle booleans” pozwala na zmianę wartości zmiennej bool na przeciwną od razu po kliknięciu na nią. W przypadku odznaczenia opcji pojawia się okno wyboru wartości zmiennej.



### 3.2.3 Live graph

Opcja „Live graph” pozwala na monitorowanie wartości zmiennych na wykresie. Aby ją uruchomić, należy kliknąć ikonę w prawym górnym rogu ekranu, a następnie wybrać zmienne które chcemy monitorować i w tym samym miejscu uruchomić rysowanie wykresu.



## 4 Informacje dodatkowe

Dodatkowe informacje na temat samej biblioteki Tc3\_IotCommunicator dostępne są pod linkiem [https://infosys.beckhoff.com/english.php?content=../content/1033/tf6730\\_tc3\\_iot\\_communicator/index.html](https://infosys.beckhoff.com/english.php?content=../content/1033/tf6730_tc3_iot_communicator/index.html)

Dodatkowe informacje na temat działania protokołu MQTT dostępne są pod linkiem [https://infosys.beckhoff.com/english.php?content=../content/1033/tf6701\\_tc3\\_iot\\_communication\\_mqtt/27021601282764171.html](https://infosys.beckhoff.com/english.php?content=../content/1033/tf6701_tc3_iot_communication_mqtt/27021601282764171.html)