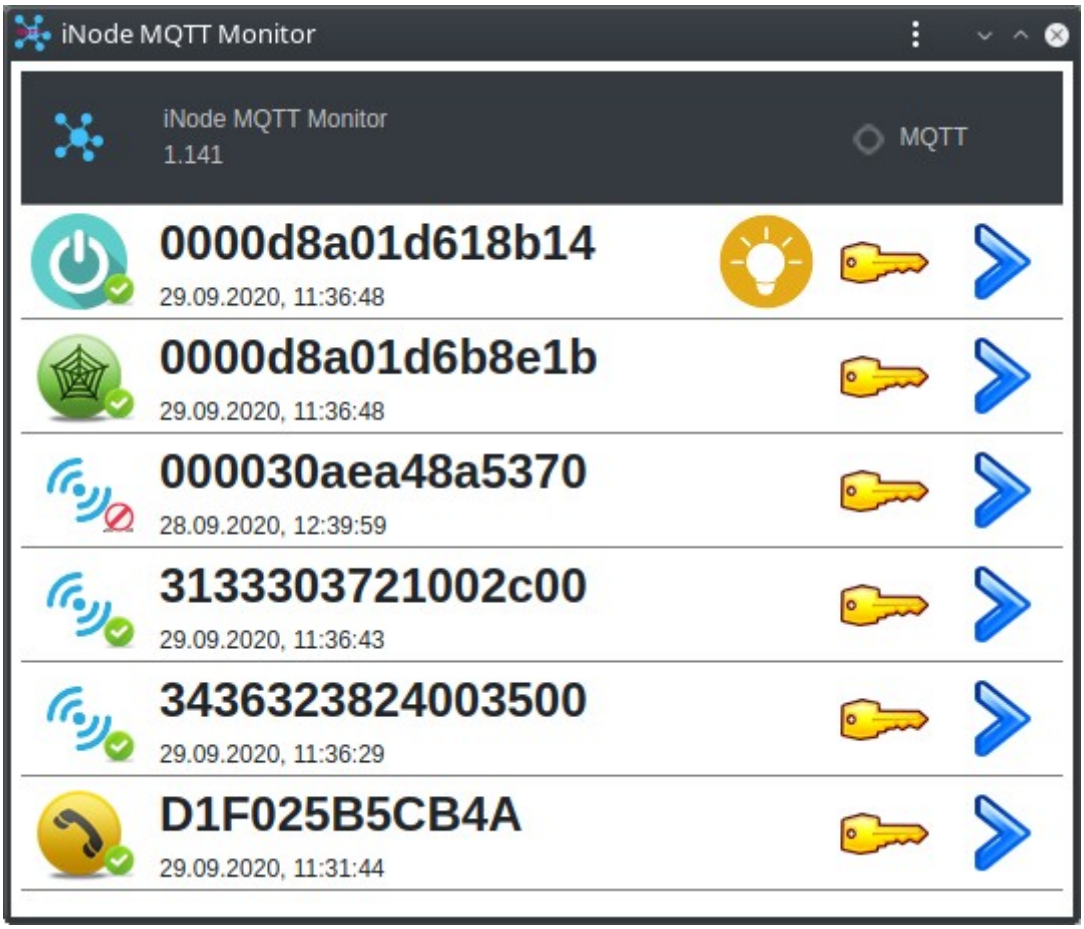
iNode MQTT Monitor

instrukcja użytkownika

1. Wstęp

Aplikacja **iNode MQTT Monitor** umożliwia przetestowanie komunikacji pomiędzy urządzeniem, a serwerem MQTT lub HTTP/POST. Obsługuje również adaptery USB BT lub LPWAN oraz wspomaga technologię Web Bluetooth. Umożliwia zdalne sterowanie urządzeniami np. wyjściem w **iNode MCU Relay**. Aplikacja **iNode MQTT Monitor** jest dedykowana dla przeglądarki Google Chrome i działa na systemach operacyjnych Android, Windows, Linux itp. Po wczytaniu się aplikacji można ją zainstalować w celu późniejszego łatwiejszego uruchamiania. Na głównym ekranie pojawi się wtedy ikona aplikacji.



Po uruchomieniu się aplikacji pokazuje ona urządzenia, które przesyłają dane na serwer MQTT iot.inode.pl. Jest to bezpłatny testowy serwer MQTT dla użytkowników produktów **iNode**.

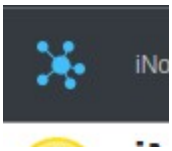
Ważne !

Firma ELSAT s.c. nie gwarantuje, że serwer MQTT iot.inode.pl będzie dostępny w przyszłości oraz na jakich warunkach. Użytkownik musi być świadom, że dane wysyłane na ten serwer mogą być odbierane przez innych.

Dla zachowania prywatności, należy zapewnić, aby wysłane na ten serwer

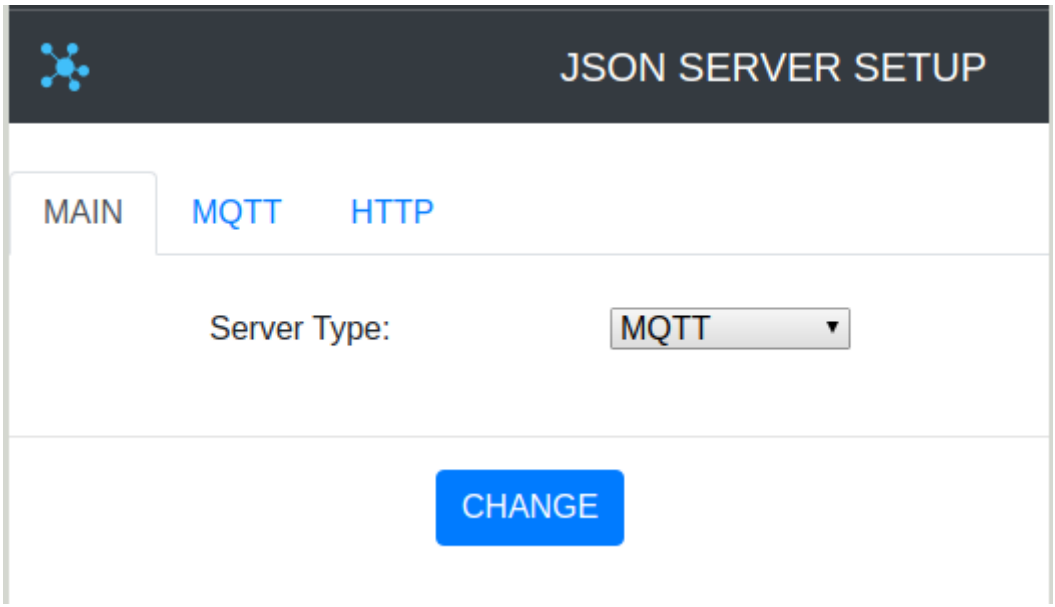
dane były szyfrowane – jest to opcja domyślna w **iNode LAN Central**, **iNode MCU Relay** lub **iNode GSM MQTT LPWAN**. Domyślne hasło do ich szyfrowania jest w każdym urządzeniu inne i tworzone losowo. Dane na serwerze nie są w żaden sposób archiwizowane, są jednak publicznie dostępne co wynika ze specyfiki działania serwera MQTT jeżeli dostęp do niego nie jest ograniczony za pomocą nazwy użytkownika i hasła. Firma ELSAT s.c. w żaden sposób nie odpowiada za treść tych danych i w żaden sposób w nie nie ingeruje – moderuje.

Konfiguracja aplikacji **iNode MQTT Monitor** odbywa się po kliknięciu na obrazku w lewym górnym rogu okna aplikacji:

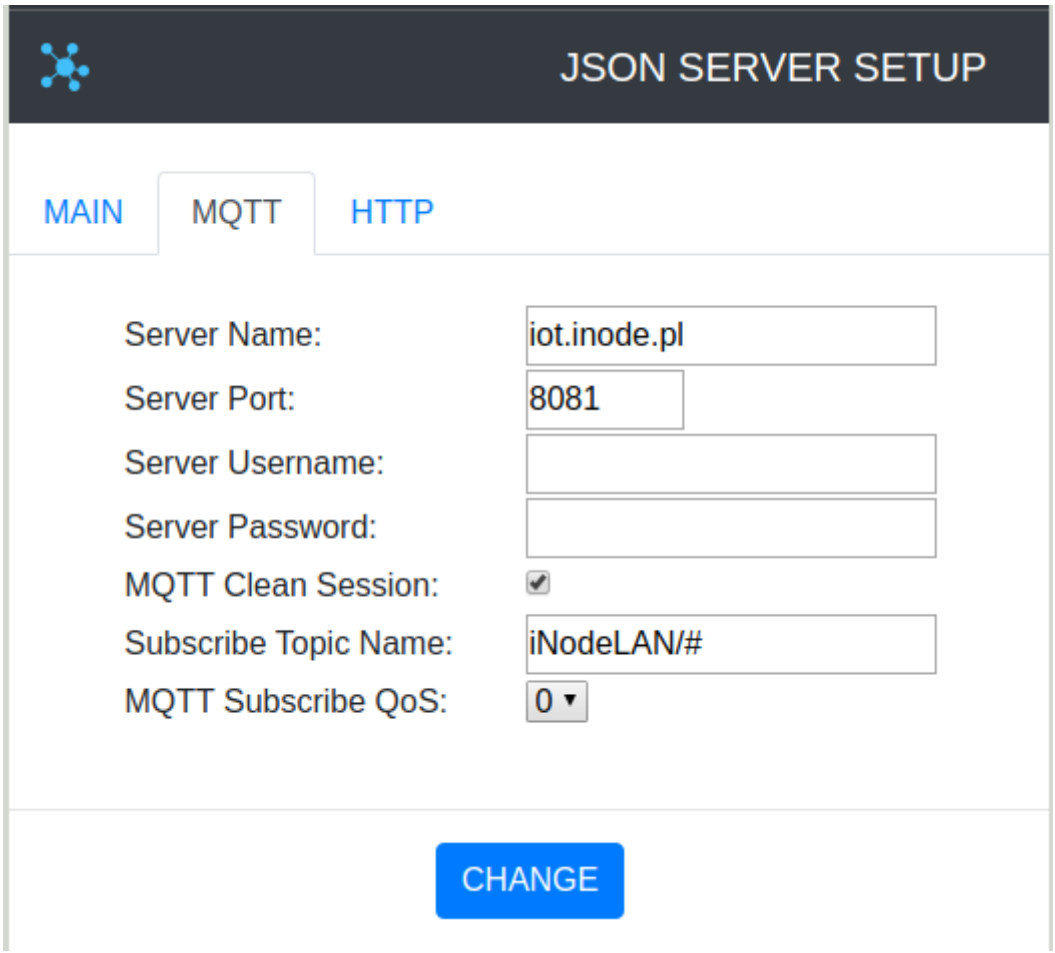


Pojawi się wtedy następujące okno aplikacji – **JSON SERVER SETUP**:

Zakładka **MAIN** umożliwia wybranie rodzaju serwera z którym aplikacja ma współpracować. Może to być HTTP, MQTT, USB lub BLUETOOTH. Ta ostatnia opcja jest dostępna od wersji Google Chrome 79 jednak jak na razie działa tylko pod systemem Android OS, a pod Linux OS i Windows OS w ograniczonym zakresie. Może być konieczne włączenie w **chrome://flags/#enable-experimental-web-platform-features** dla USB lub BLUETOOTH.



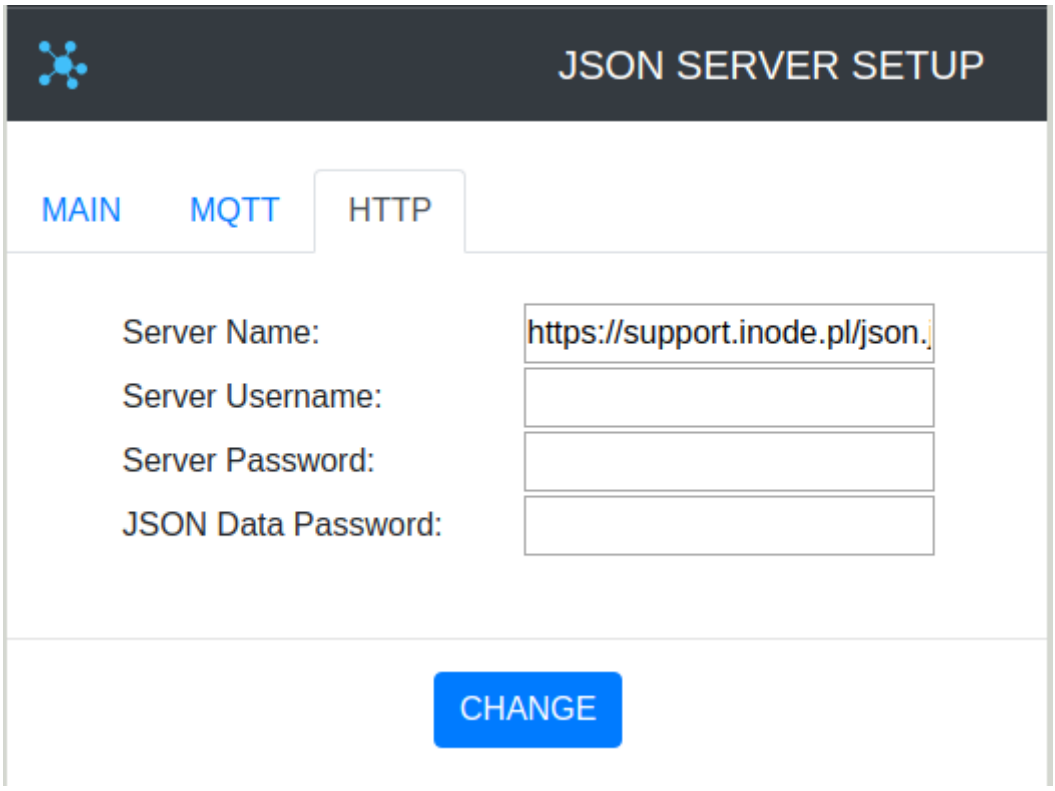
Zakładka **MQTT** umożliwia podanie parametrów serwera MQTT.



- ***Server Name*** – nazwa serwera
- ***Server Port*** – port pod którym dostępna jest usługa WebSocket serwera MQTT
- ***Server Username*** – nazwa użytkownika jeśli dostęp do serwera MQTT jest ograniczony
- ***Server Password*** – hasło dostępu do serwera MQTT
- ***MQTT Clean Session*** - gdy flaga ***MQTT Clean Session*** jest ustawiona, klient nie chce trwałej sesji. Jeśli klient rozłącza się z jakiegokolwiek powodu, wszystkie informacje i komunikaty w kolejce z poprzedniej trwałej sesji zostają utracone.
- ***Subscribe Topic Name*** – musi to być taka sama wartość jak w ustawieniach **iNode LAN Central** w polu **Page/Topic Name** lub jej fragment.
- ***MQTT Subscribe QoS:***
  - ***QoS 0*** - klient nie otrzyma od serwera żadnego potwierdzenia. Podobnie wiadomość dostarczona klientowi z serwera nie musi być potwierdzona. Jest to najszybszy sposób publikowania i odbierania wiadomości, ale także ten, w którym najprawdopodobniej nastąpi utrata wiadomości.

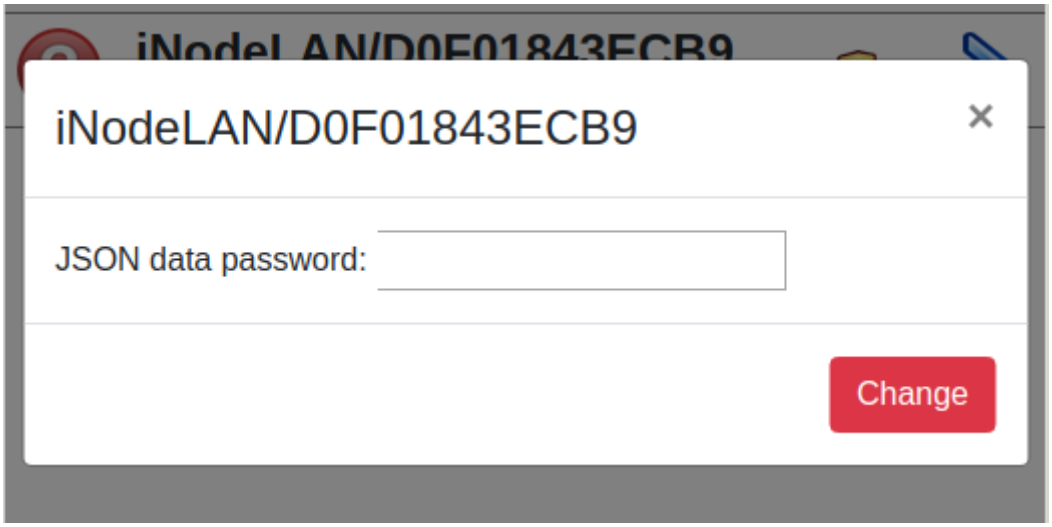
- ***QoS 1*** - klient otrzyma wiadomość potwierdzającą z serwera po jej opublikowaniu. Jeśli oczekiwane potwierdzenie nie zostanie odebrane w określonym czasie, klient musi ponowić wiadomość. Wiadomość otrzymana przez klienta również musi zostać potwierdzona na czas, w przeciwnym razie serwer ponownie dostarczy wiadomość.

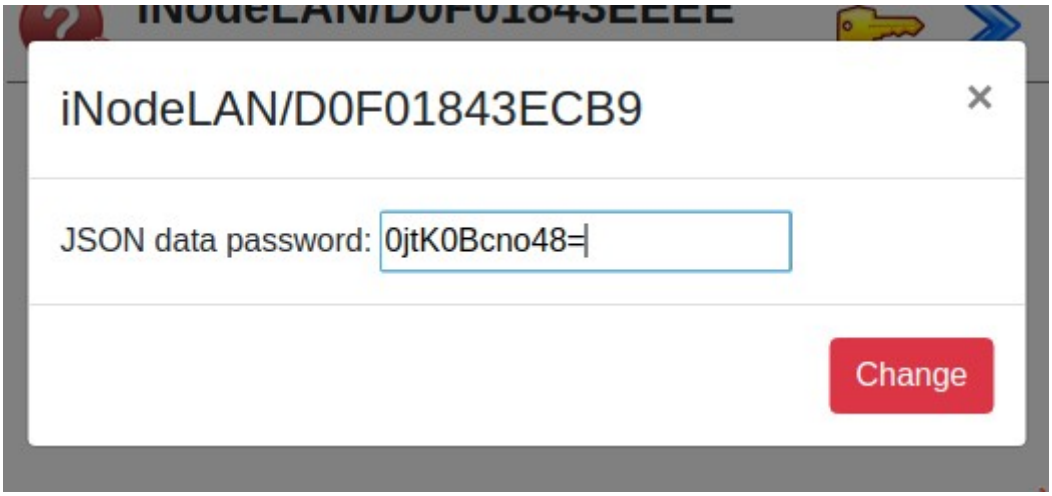
Zakładka **HTTP** umożliwia podanie parametrów serwera HTTP.



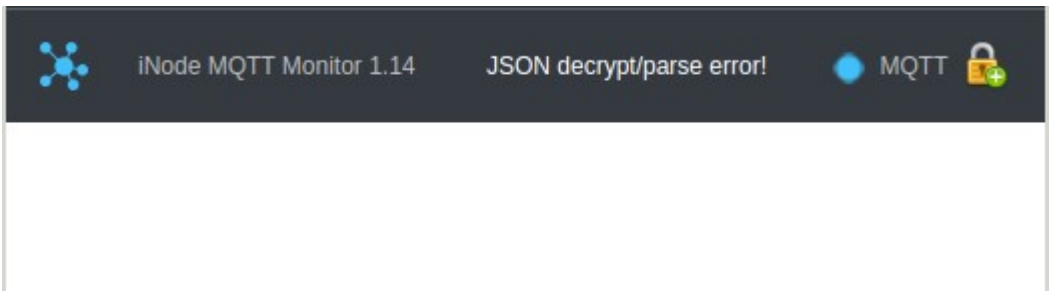
- ***Server Name*** – pełen url zawierający nazwę serwera oraz ścieżkę do pliku z danym JSON
- ***Server Port*** – port pod którym dostępna jest usługa serwera HTTP
- ***Server Username*** – nazwa użytkownika jeśli dostęp do serwera HTTP jest ograniczony. Rodzaj autoryzacji Basic.
- ***Server Password*** – hasło dostępu do serwera HTTP

Jeżeli **iNode LAN Central** przesyła dane JSON zaszyfrowane, to w aplikacji należy, po wybraniu obrazka z kluczykiem, podać hasło do ich odkodowania i nacisnąć przycisk **CHANGE**. Musi to być to samo hasło co w ustawieniach **SETUP** w polu ***JSON data password***.

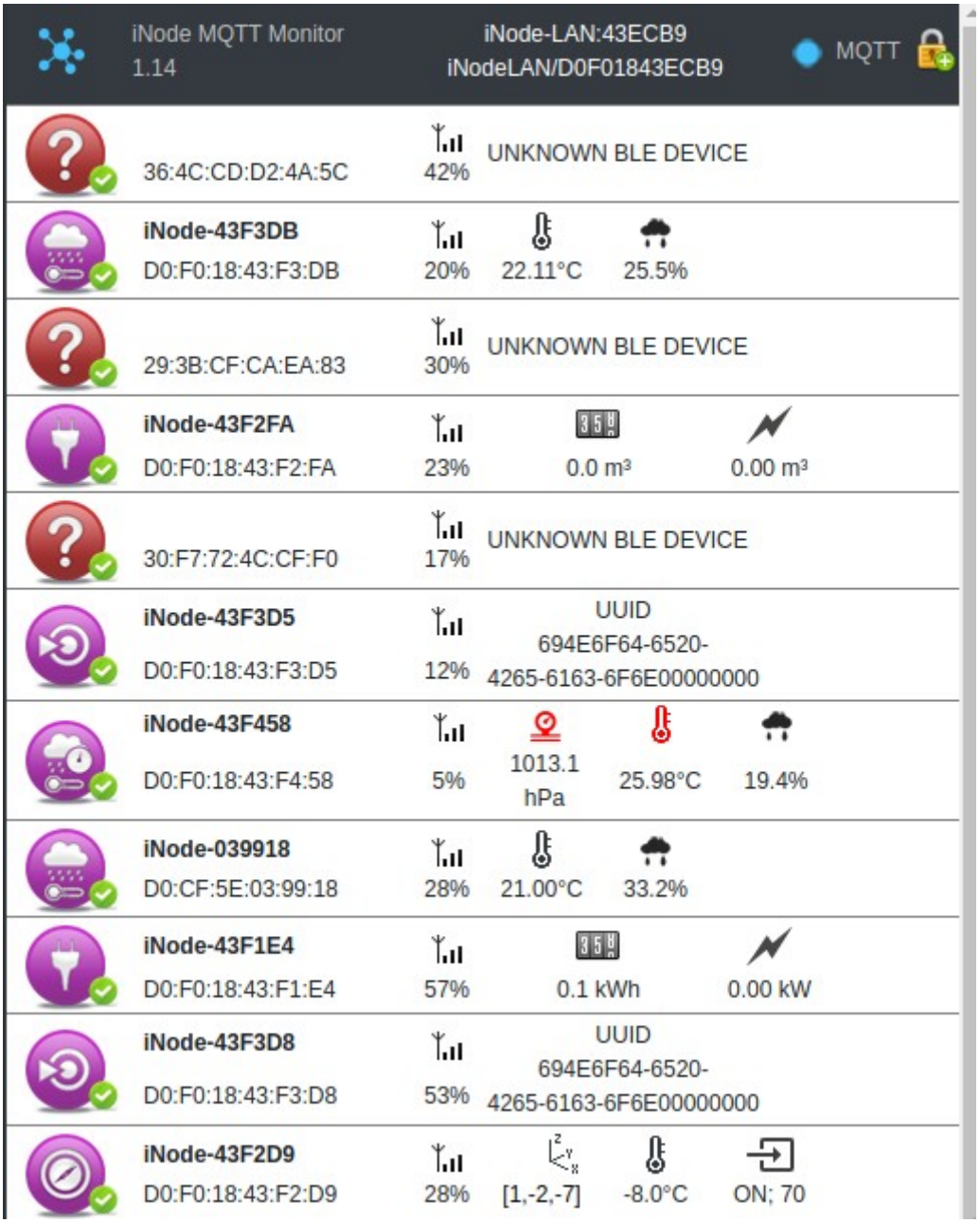




Jeżeli hasło jest nieustawione lub złe to po wybraniu niebieskiej strzałki w prawo pojawi się informacja o błędzie - ***JSON decrypt / parse error***. To, że dane są zaszyfrowane pokazuje obrazek kłódki w prawym górnym rogu ekranu.



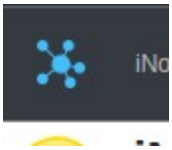
Jeżeli hasło zostało podane prawidłowo to aplikacja wyświetli informację o urządzeniach BLE, które są wysyłane przez dany ***iNode LAN Central***. Najeżdżając myszką lub dotykając palcem poszczególnych elementów (smartfon) będą wyświetlane dodatkowe informacje o danym urządzeniu BLE.



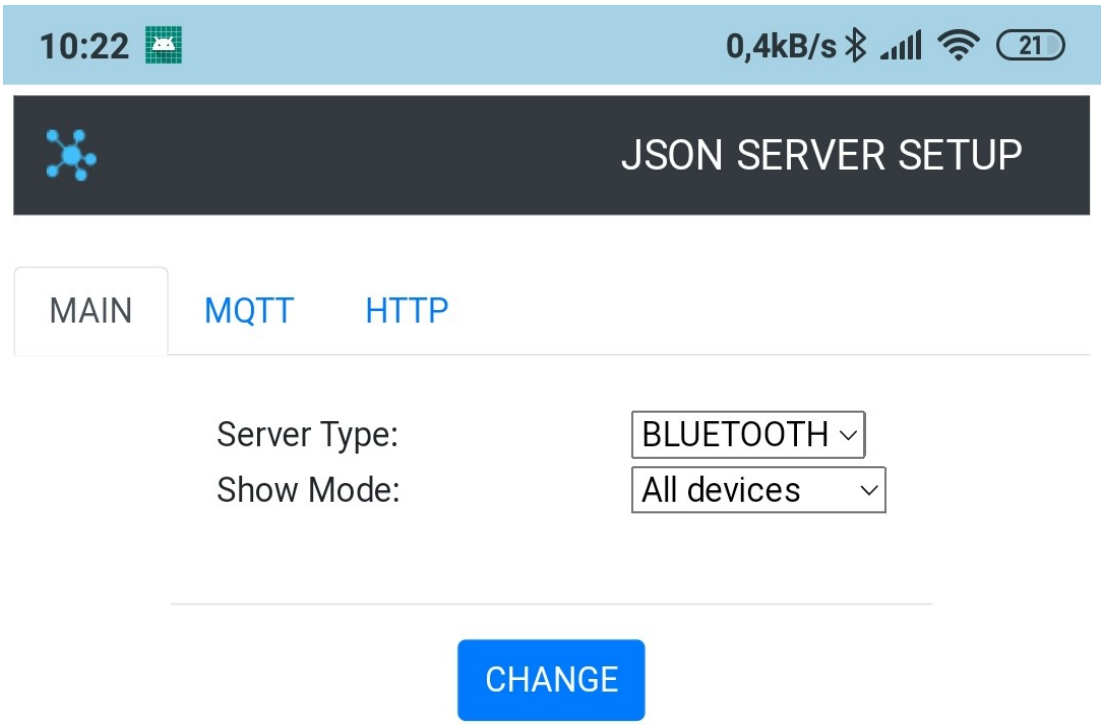
2. Android OS

Ponieważ aplikacja **iNode MQTT Monitor** wykorzystuje technologię Web Bluetooth to jest możliwe jej używanie na telefonie z Android OS. Konieczne jest jednak do tego wykonanie kilku czynności:

- Przede wszystkim w telefonie musi być włączona obsługa Bluetooth.
- Kolejnym krokiem jest uruchomienie przeglądarki Chrome i wpisanie w pasku adresu: **chrome://flags/#enable-experimental-web-platform-features** a następnie włączenie tej funkcji i zrestartowanie przeglądarki.
- Teraz możemy już uruchomić aplikację **iNode MQTT Monitor**.
- Konfiguracja aplikacji **iNode MQTT Monitor** odbywa się po kliknięciu na obrazku w lewym górnym rogu ekranu:



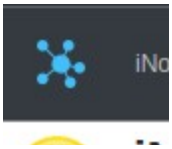
Pojawi się wtedy następujący ekran aplikacji – **JSON SERVER SETUP**:



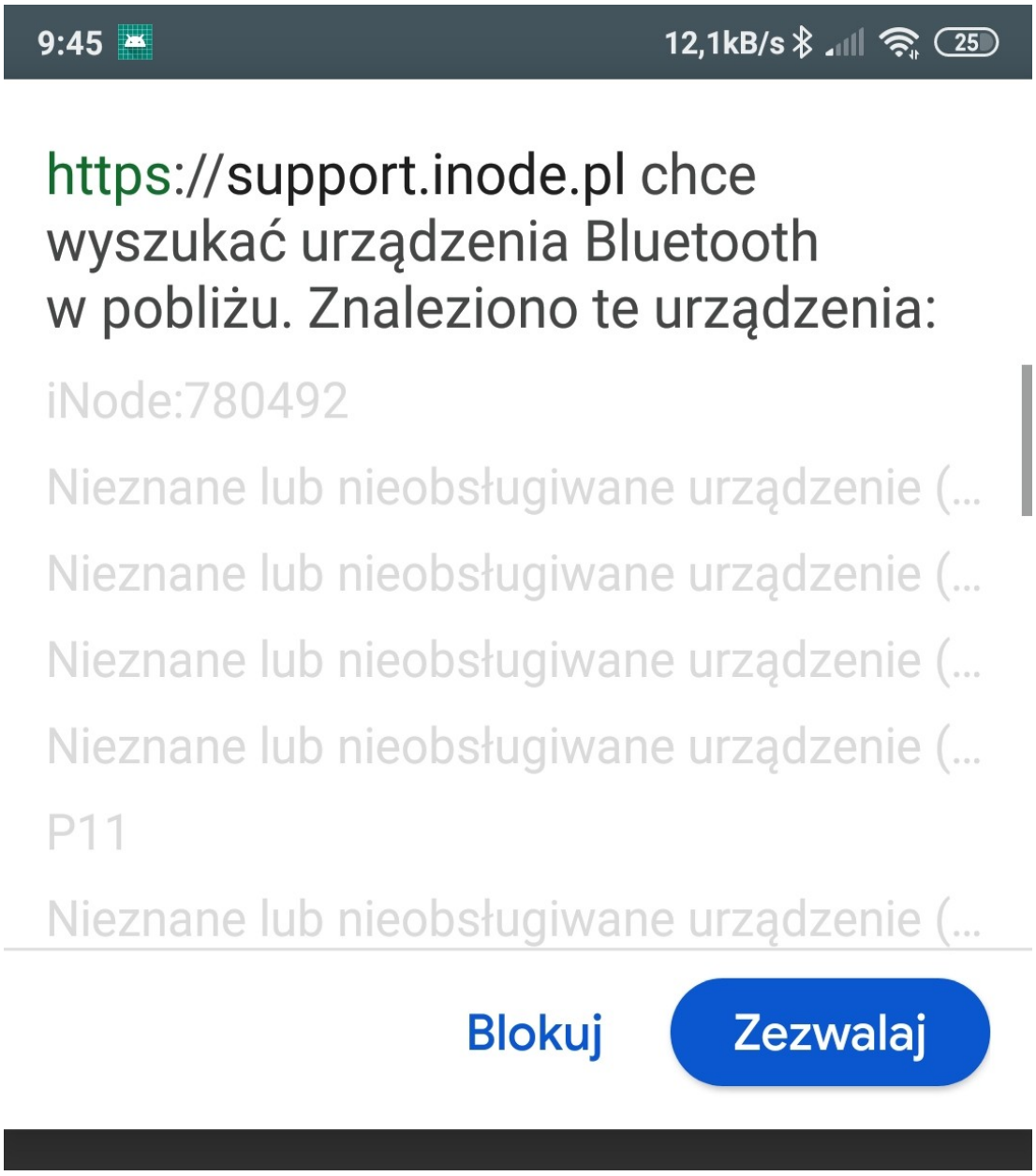
Zakładka **MAIN** umożliwia wybranie rodzaju serwera z którym aplikacja ma współpracować. W tym przypadku musi to być **BLUETOOTH**. Ta opcja jest dostępna od wersji Google Chrome 79 jednak

jak na razie działa tylko pod systemem Android. W celu zapamiętania zmienionych ustawień należy wcisnąć przycisk **CHANGE**.

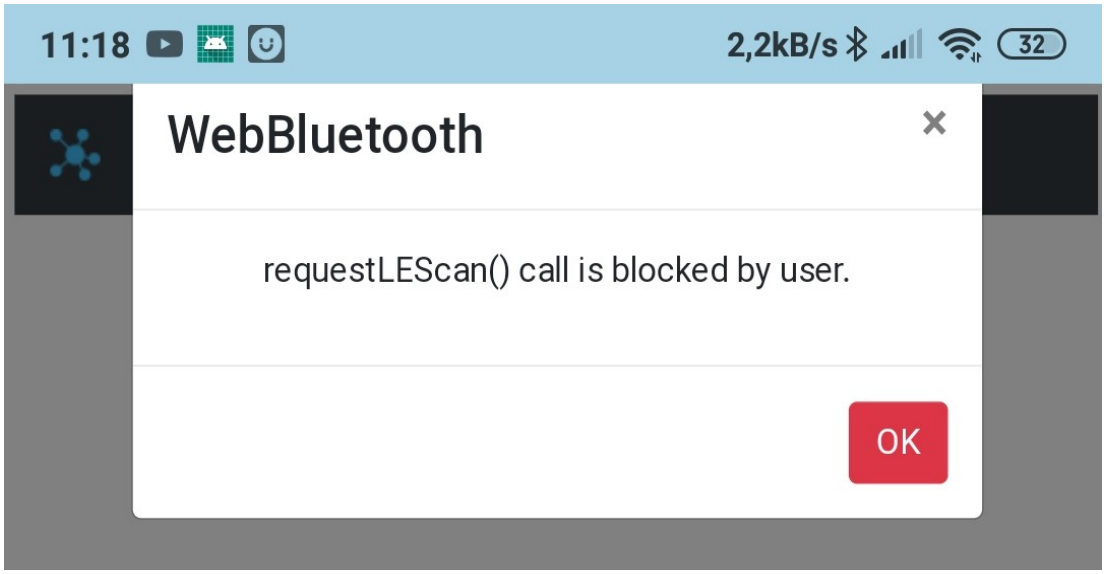
Po zakończeniu konfiguracji należy ponownie kliknąć na obrazku:



- Jeżeli wszystko jest skonfigurowane prawidłowo to pojawi się wtedy ekran systemowy z następującym komunikatem, aby użytkownik zezwolił na skanowanie:

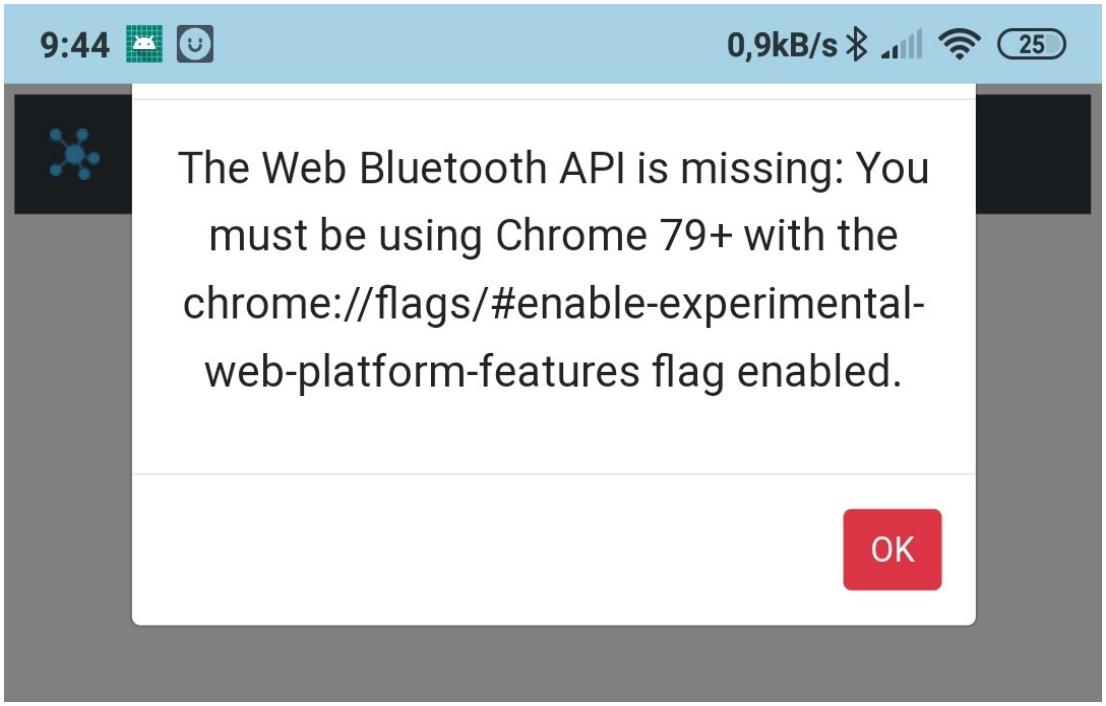


Należy wtedy wcisnąć przycisk **Zezwalaj**. Wciśnięcie przycisku **Blokuj** spowoduje wyłączenie dostępu do **Web Bluetooth** na stałe.

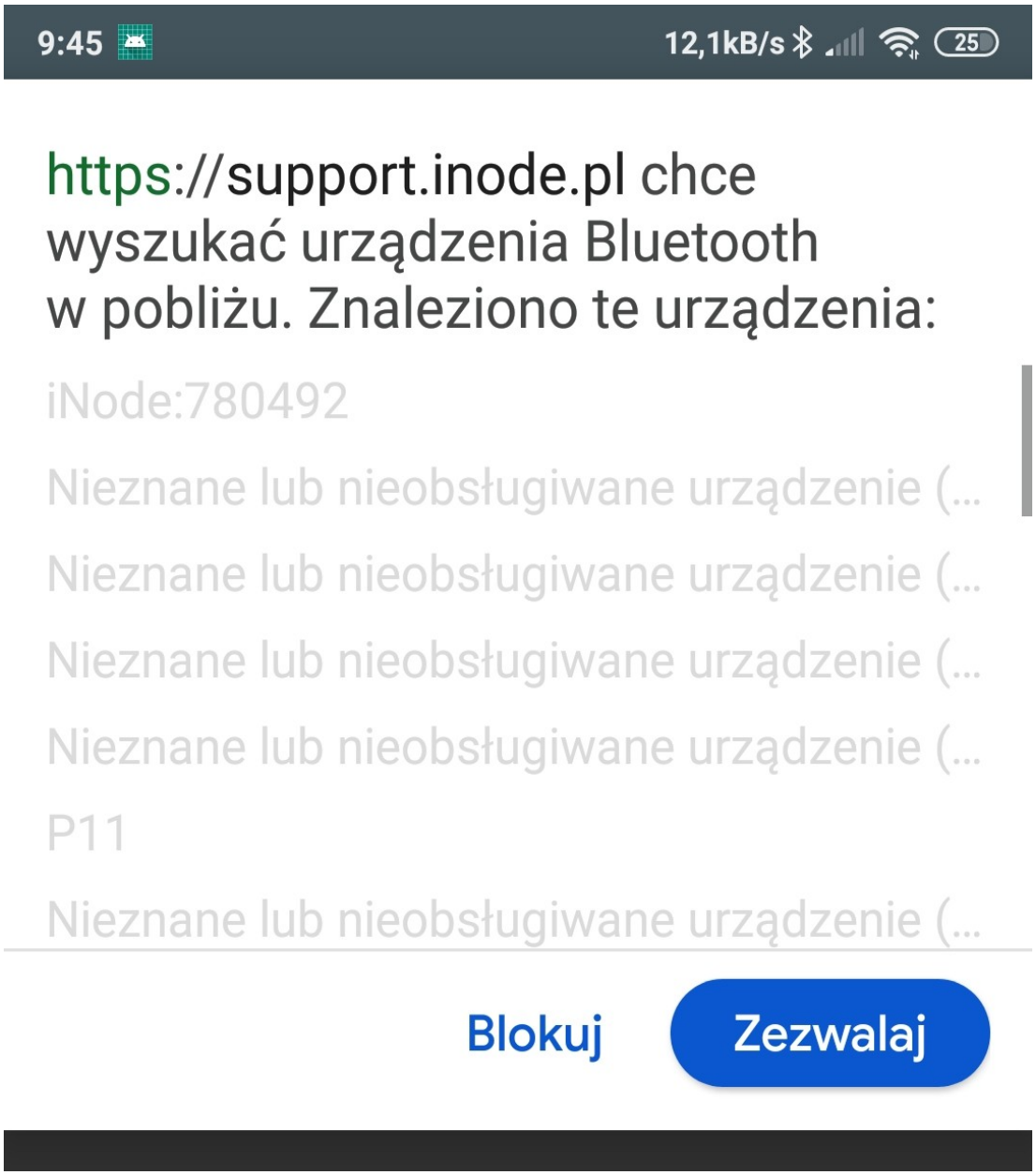


Dostęp do **Web Bluetooth** można wtedy uaktywnić tylko przez wybranie z menu Chrome → Ustawienia → Ustawienia witryn → Skanowanie Bluetooth i dopuszczenie tej funkcjonalności dla witryny https://support.inode.pl.

Pomimo włączenia prawidłowego dostępu do **Web Bluetooth**, przy uruchamianiu aplikacji może pojawiać się komunikat:

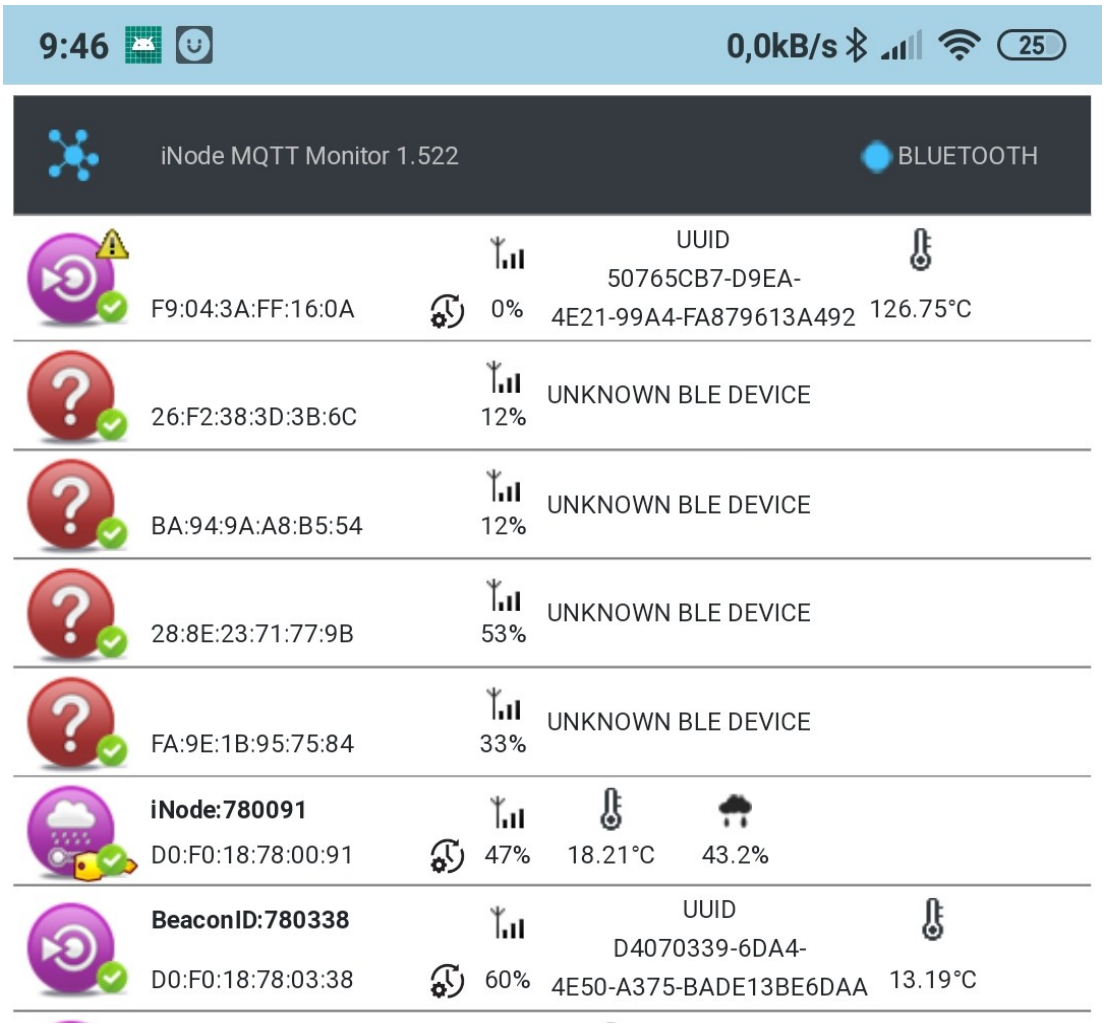


Po potwierdzeniu komunikatu przez wciśnięcie **OK** pojawi się systemowe okienko zezwolenia na skanowanie.



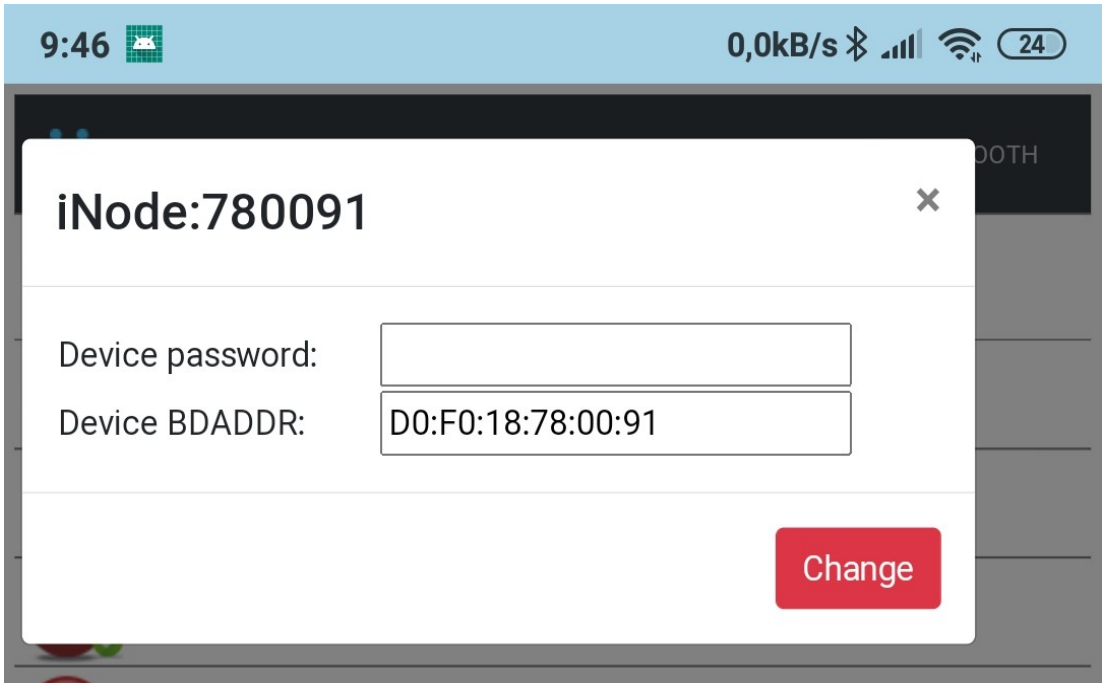
2.1. iNode MQTT Monitor i Web Bluetooth

Tak wygląda ekran aplikacji, jeżeli w zasięgu są jakieś urządzenia BLE:

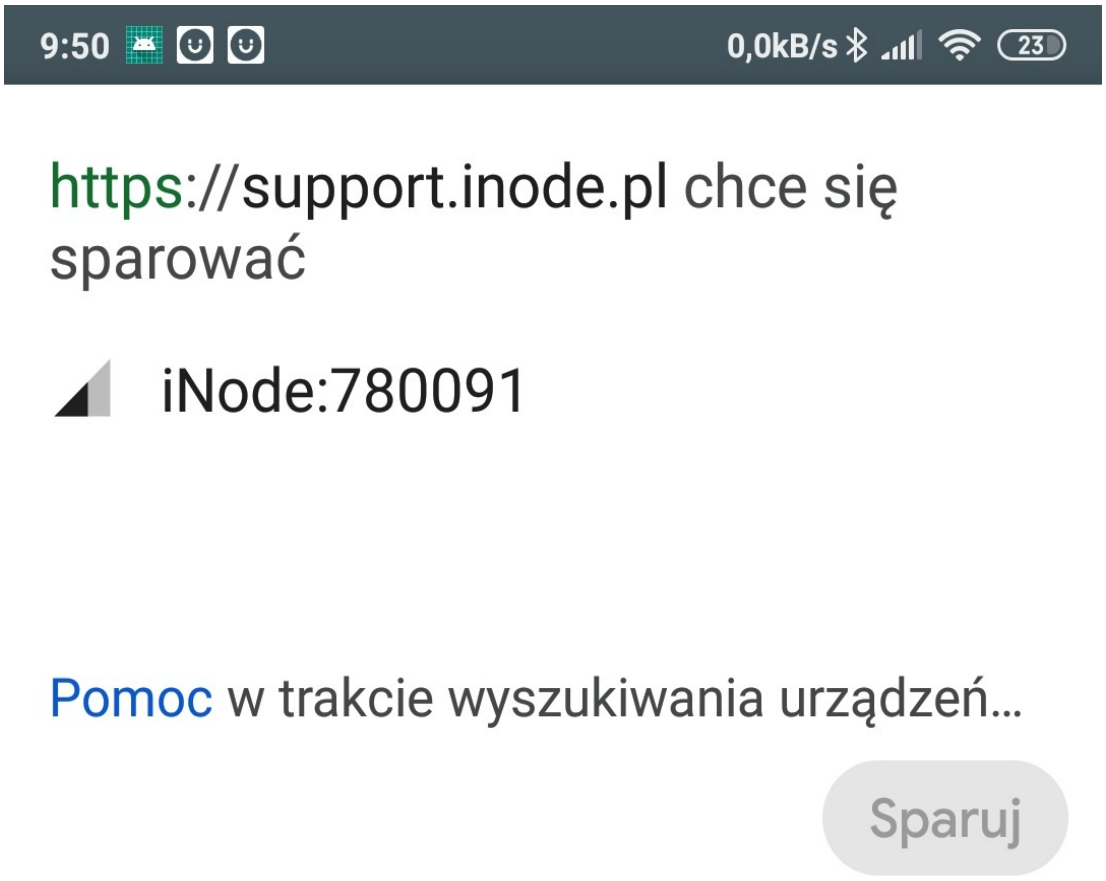


Urządzenia (czujniki BLE), dla których konfiguracja przez **Web Bluetooth** jest możliwa mają wyświetloną ikonę:

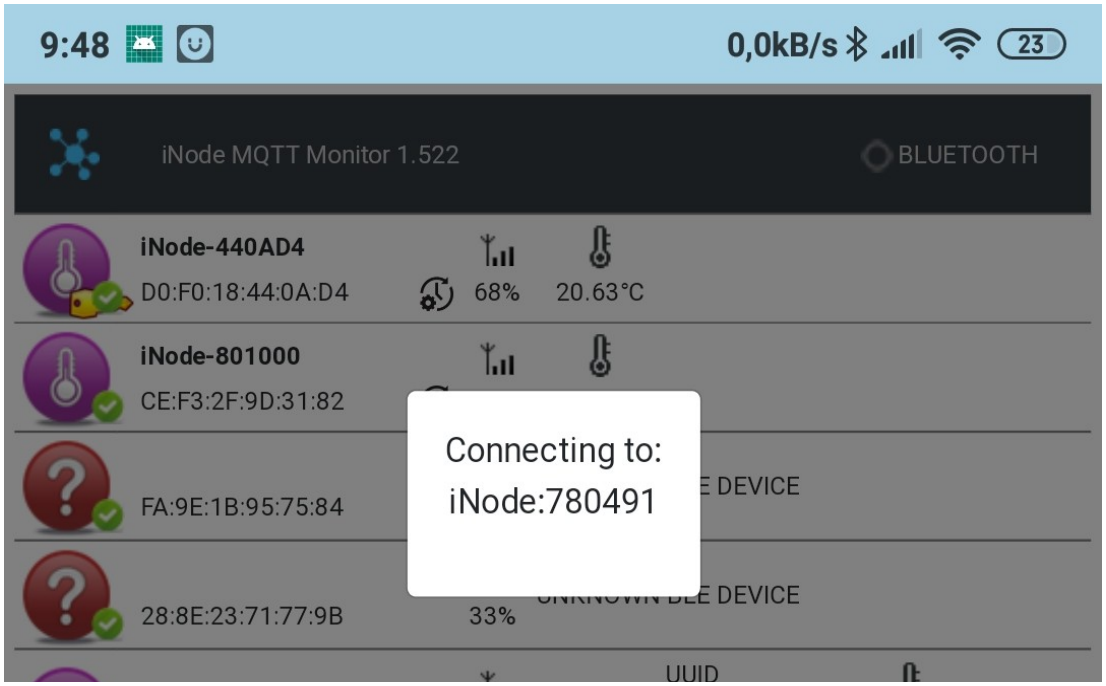
Po jej kliknięciu pojawi się najpierw okienko z prośbą o podanie hasła. Nie ma znaczenia, czy hasło jest potrzebne do nawiązania połączenia, czy też nie. Aplikacja wymaga, aby hasło było wprowadzone. W przypadku, gdy nie jest ono potrzebne należy tylko wcisnąć przycisk **Change** co spowoduje zamknięcie okienka.



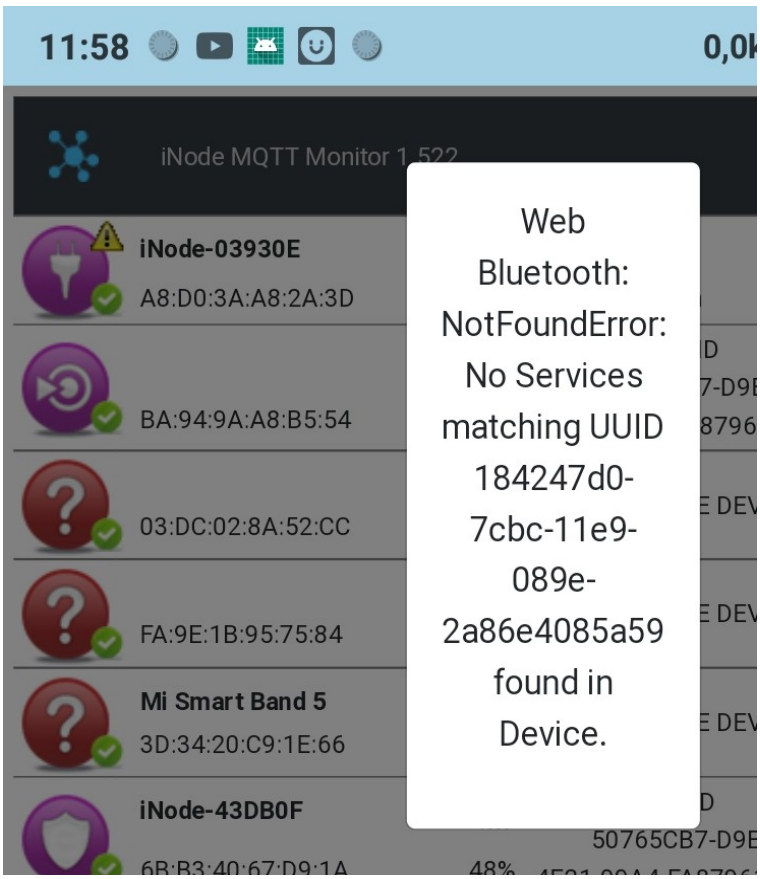
Po ponownym kliknięciu ikony pojawi się okienko systemowe zezwolenia na połączenie z określonym urządzeniem BLE:



Po zaznaczeniu urządzenia i wciśnięciu przycisku **Sparuj** pojawi się komunikat o próbie nawiązania połączenia z nim:

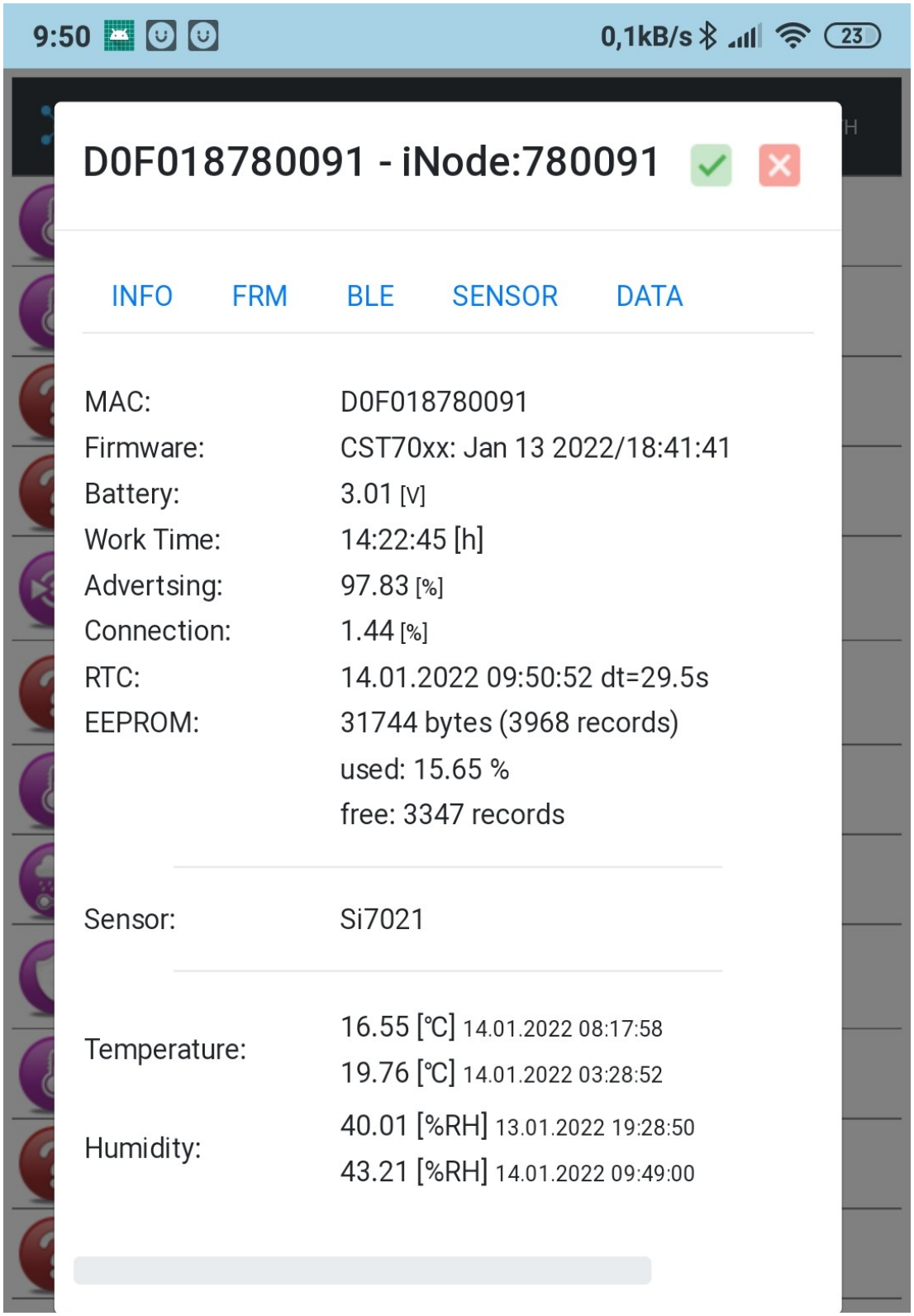


Zezwolenie na połączenie z wybranym urządzeniem BLE jest zapamiętywane przez Android OS do wyłączenia telefonu lub interfejsu Bluetooth. Ponieważ tylko na podstawie wyników skanowania aplikacja nie jest w stanie stuprocentowo określić, czy może z danym urządzeniem się połączyć możliwa jest sytuacja, że dany czujnik nie jest obsługiwany przez nią. Dotyczy to przede wszystkim wersji BT 4.0 czujników. Pojawi się wtedy komunikat o błędzie:



2.2. Konfigurowanie czujników BLE BT 5.1

Po nawiązaniu połączenia z wybranym z listy czujnikiem BLE aplikacja wyświetli okienko, które umożliwi zmianę parametrów czujnika:



2.2.1. INFO

Na pierwszej zakładce **INFO** są wyświetlone informacje o wersji firmware, napięciu baterii oraz statystyczne o urządzeniu:

- **MAC** – adres MAC urządzenia;
- **Firmware** – wersja i data utworzenia firmware w urządzeniu;
- **Battery** – napięcie baterii;
- **Work Time** – czas pracy od ostatniego resetu;
- **Advertising** – ile procent czasu pracy zajmowało rozgłaszanie;
- **Connection** – ile procent czasu pracy urządzenie było połączone;
- **RTC** – czas z zegarka RTC oraz dt – różnica w stosunku do czasu w telefonie; po włożeniu baterii w urządzeniu ustawia się czas utworzenia firmware;

Kolejne informacje zależą od rodzaju czujnika BLE. W przypadku **iNode CS T** lub **HT** jest to dodatkowo:

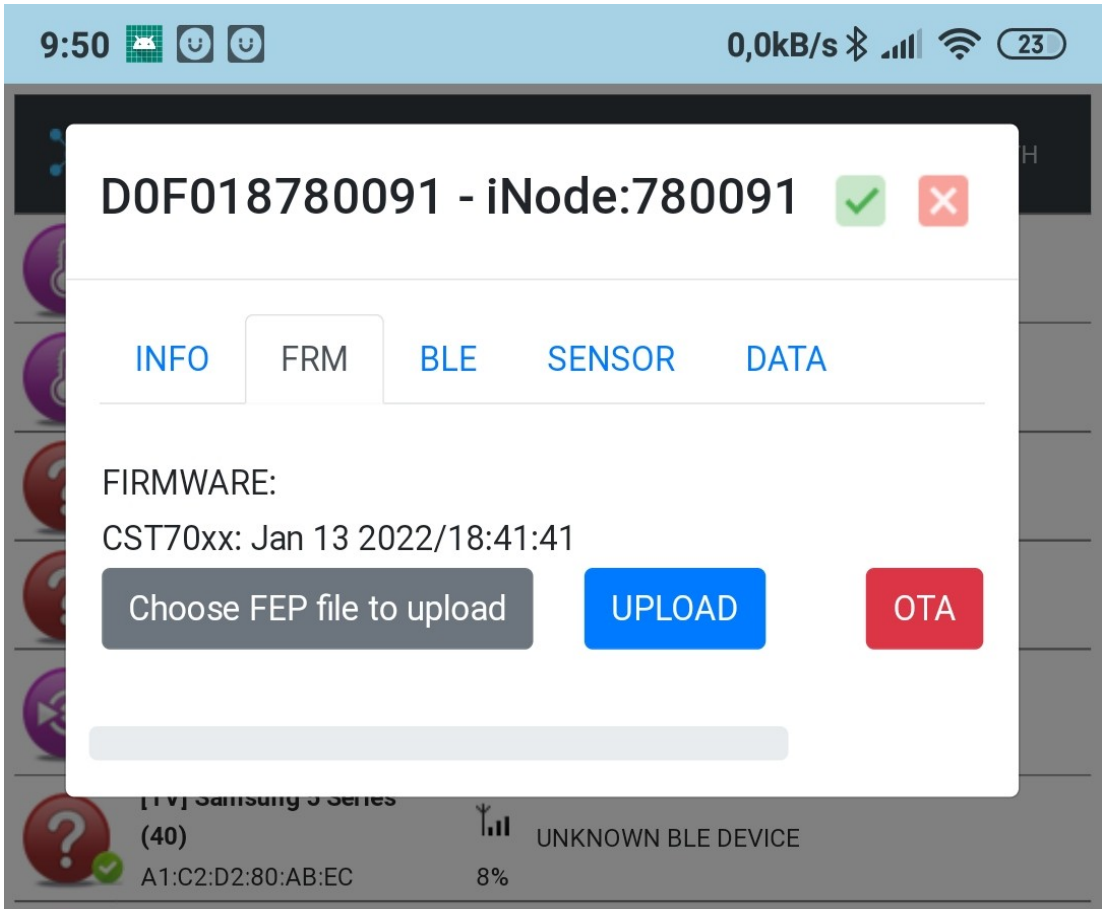
- **EEPROM** – jaka wielkość pamięci EEPROM jest przeznaczona na rejestrację danych, ile z niej jest użyte, a ile wolne;
- **Sensor** – rodzaj sensora użytego w czujniku;
- **Temperature** – zarejestrowana temperatura: najniższa, najwyższa i w trakcie ostatniego alarmu (jeżeli był);
- **Humidity** - zarejestrowana wilgotność: najniższa, najwyższa i w trakcie ostatniego alarmu (jeżeli był);

Gdy połączenie z czujnikiem jest utrzymywane, świadczy o tym ikonka wyświetlona na górnej belce okienka. Po jej naciśnięciu zmodyfikowane parametry są przesyłane do czujnika.

Jeżeli nie wykonujemy żadnych czynności, które wymagają wymiany danych z czujnikiem to połączenie zostanie automatycznie rozłączone po około 1 minucie. Nie będzie wtedy możliwe przesłanie do czujnika zmodyfikowanych parametrów. Należy wtedy zamknąć okienko i ponownie połączyć się z czujnikiem.

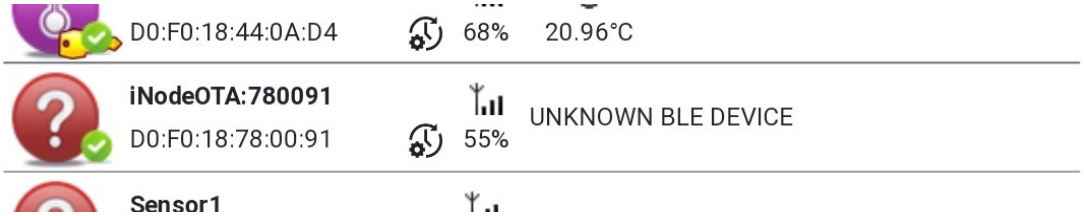
2.2.2. FRM

Kolejna zakładka to **FRM**:



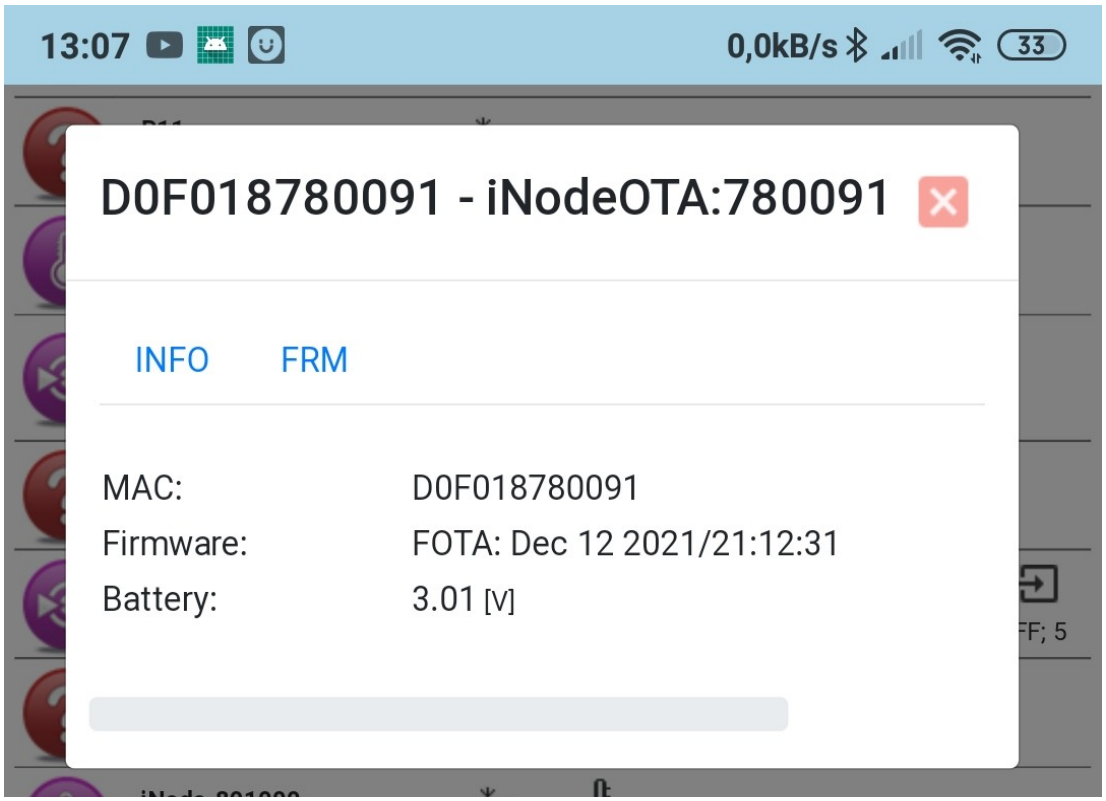
Podaje ona wersję firmware znajdującą się w urządzeniu. Po wciśnięciu przycisku **Choose FEP file to upload** można wybrać plik z firmware, który chcemy przesłać do urządzenia. Po jego wczytaniu i sprawdzeniu przez aplikację można wysłać go do urządzenia wciskając przycisk **UPLOAD**. Sam transfer danych zajmuje kilka sekund i jeżeli wszystko przebiegło prawidłowo to połączenie jest rozłączane i urządzenie uruchamia się już z nowym firmware.

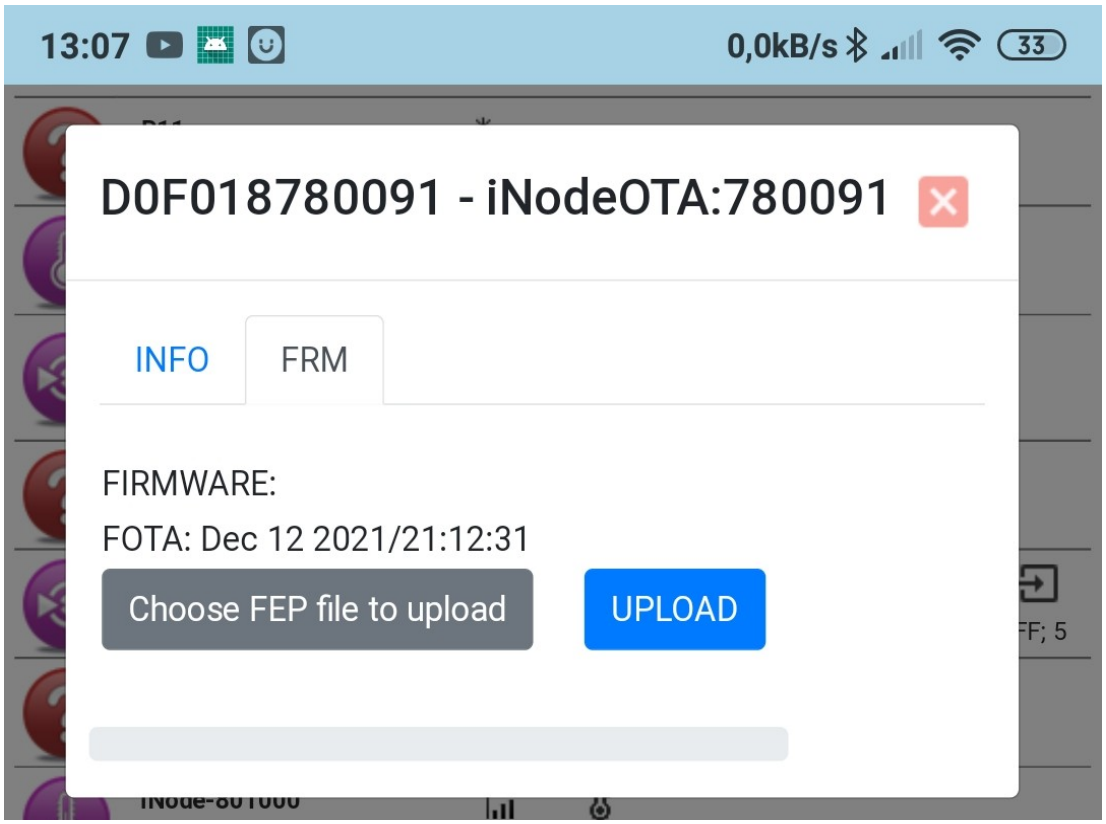
Przycisk **OTA**, jeżeli jest wyświetlony, umożliwia wprowadzenie urządzenia w tryb wymiany firmware. Zacznie się ono rozgłaszać jako iNodeOTA:xxxxxx. Aby to zauważyć należy ponownie uruchomić aplikację **iNode MQTT Monitor**.



Aby się z nim połączyć należy ponownie podać hasło, gdyż jest ono powiązane z nazwą urządzenia.

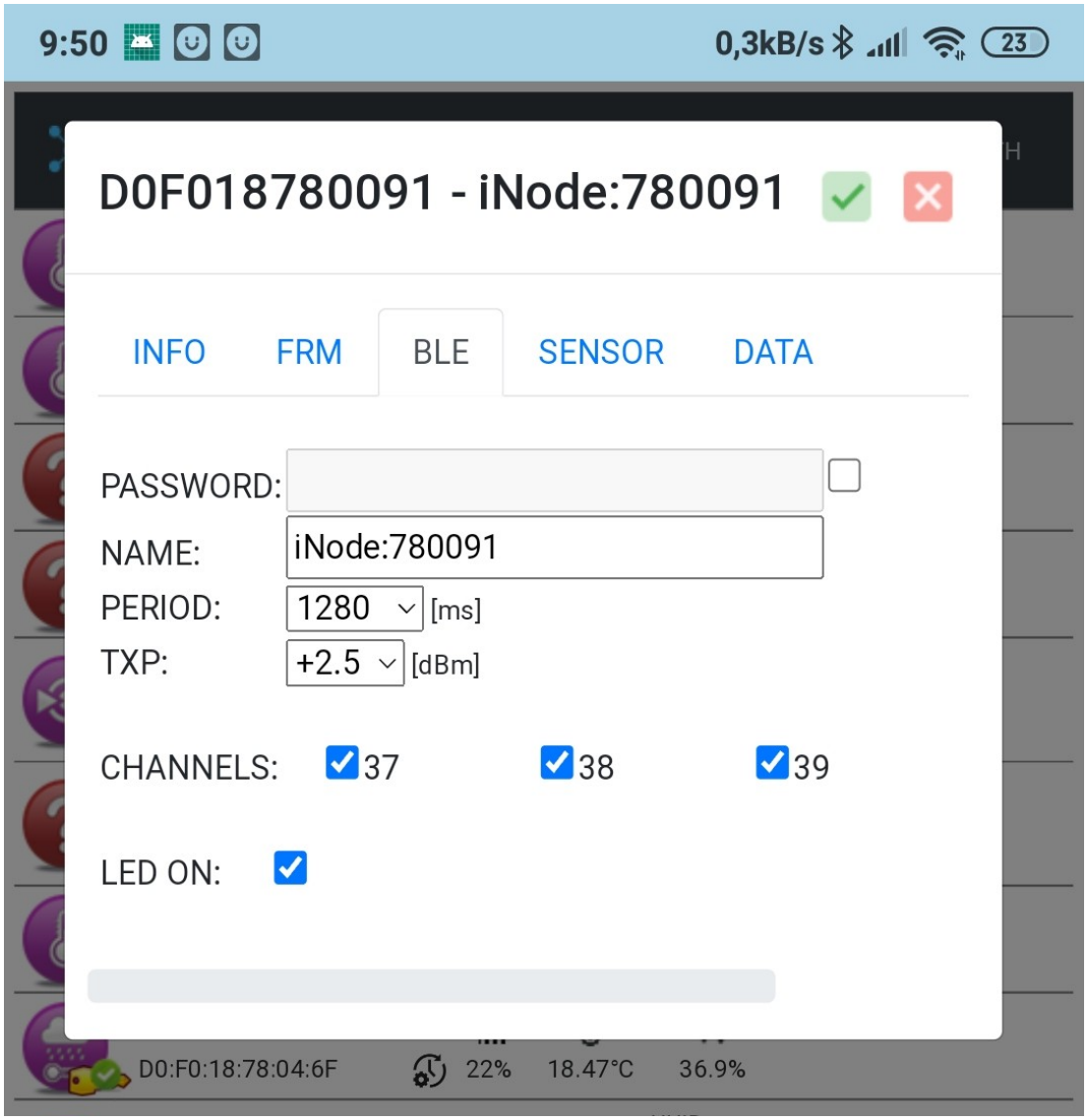
Okienko z dostępnymi funkcjami ogranicza się do zakładek **INFO** i **FRM**:





2.2.3. BLE

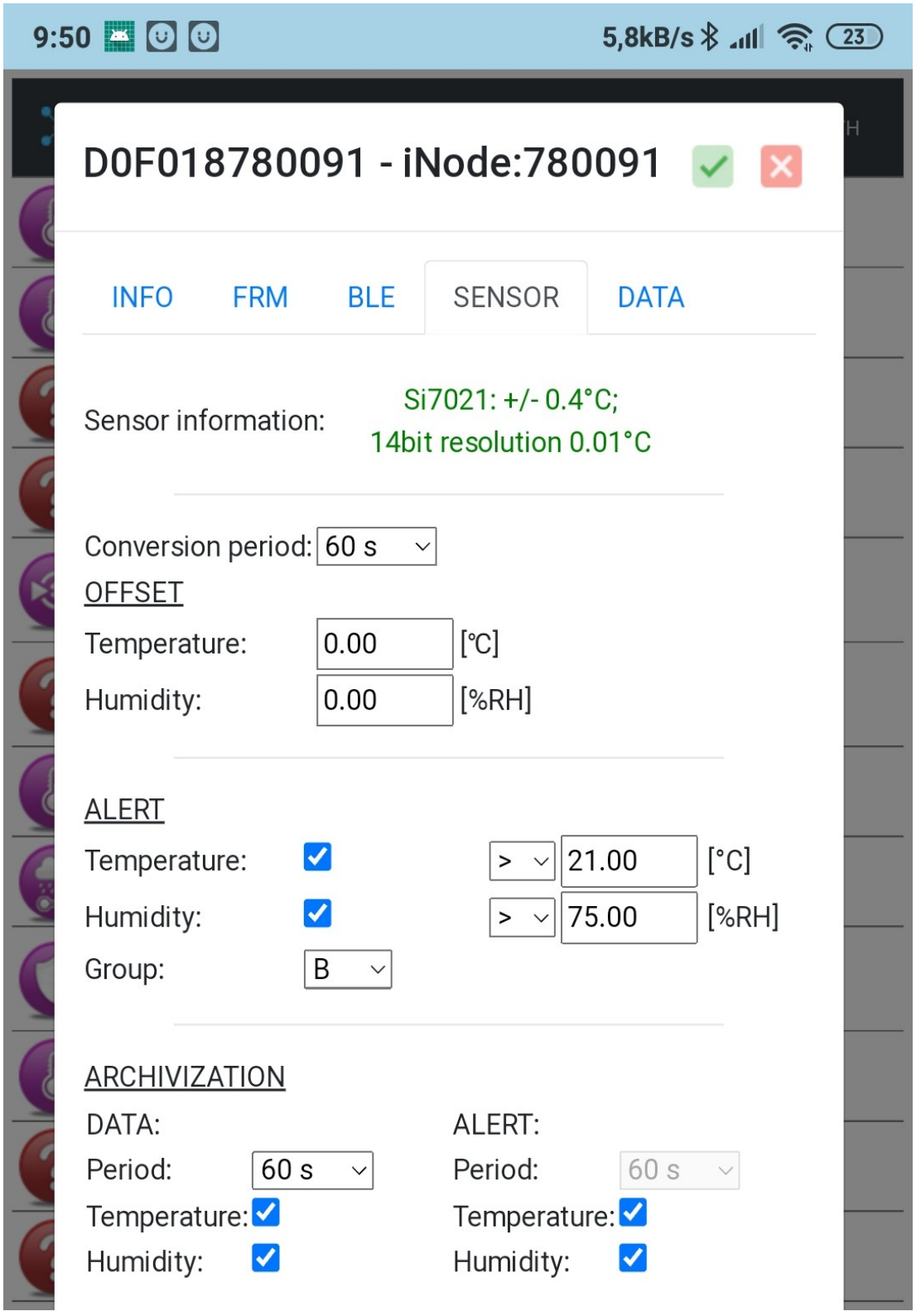
Zakładka **BLE** umożliwia zmianę parametrów związanych z rozgłaszaniem się czujnika:



- **PASSWORD** – po zaznaczeniu umożliwia wprowadzenia hasła dostępu do urządzenia; UWAGA! Nie jest możliwe skasowanie wprowadzonego hasła przez np. wyjęcie baterii;
- **NAME** – nazwa urządzenia; aby zmiana była widoczna należy uruhomić ponownie aplikację;
- **PERIOD** – okres rozgłaszania się;
- **TXP** – moc z jaką urządzenie nadaje;
- **CHANNELS** – kanały na których urządzenie nadaje; przynajmniej jeden musi być włączony; im jest ich mniej tym trudniej połączyć się z urządzeniem;
- **LED ON** – po zaznaczeniu włącza diody LED w czujniku; ma to wpływ na zużycie energii;

2.2.4. SENSOR

Zakładka **SENSOR** umożliwia zmianę parametrów związanych z wbudowanym czujnikiem środowiskowym:



- **Sensor information** – informacje o czujniku środowiskowym znajdującym się w urządzeniu;
- **Conversion period** – okres czasu z jakim są wykonywane przez sensor pomiary; ma wpływ na zużycie energii przez czujnik;

- **OFFSET**
  - **Temperature** – korekcja wartości temperatury;
  - **Humidity** – korekcja wartości wilgotności;

- **ALERT**
  - **Temperature** – włączenie dla temperatury alarmów od przekroczenia ustalonego warunku i progu;
  - **Humidity** – włączenie dla wilgotności alarmów od przekroczenia ustalonego warunku i progu;
  - **Group** – grupa logiczna do jakiej alarm jest wysyłany;

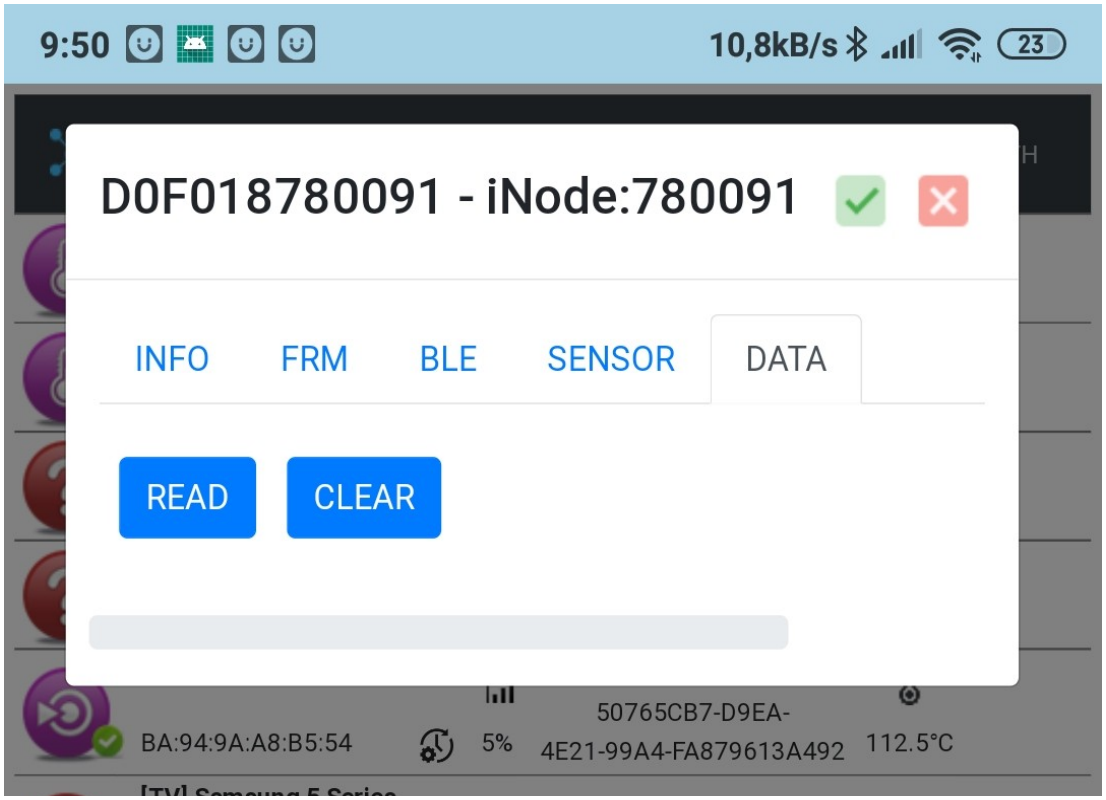
- **ARCHIVIZATION**
  - **DATA**
    - **Period** – okres archiwizacji danych (zapisywania do pamięci EEPROM); nie może być krótszy niż okres wykonywania pomiarów – **Conversion period**;
    - **Temperature** – włączenie archiwizacji pomiarów temperatury;
    - **Humidity** – włączenie archiwizacji pomiarów wilgotności;

  - **ALERT**
    - **Period** - okres archiwizacji alarmów;
    - **Temperature** - włączenie dla temperatury archiwizacji alarmów od przekroczenia ustalonego warunku i progu;
    - **Humidity** - włączenie dla wilgotności archiwizacji alarmów od przekroczenia ustalonego warunku i progu;

Dane archiwizowane w pamięci EEPROM są zapisywane w buforze cyklicznym. To znaczy, że jeżeli pamięć zostanie zapełniona danymi to przy zapisie nowych danych najstarsze z nich są kasowane.

2.2.5. DATA

Zakładka **DATA** umożliwia odczytanie z urządzenia danych zapisanych w pamięci EEPROM:



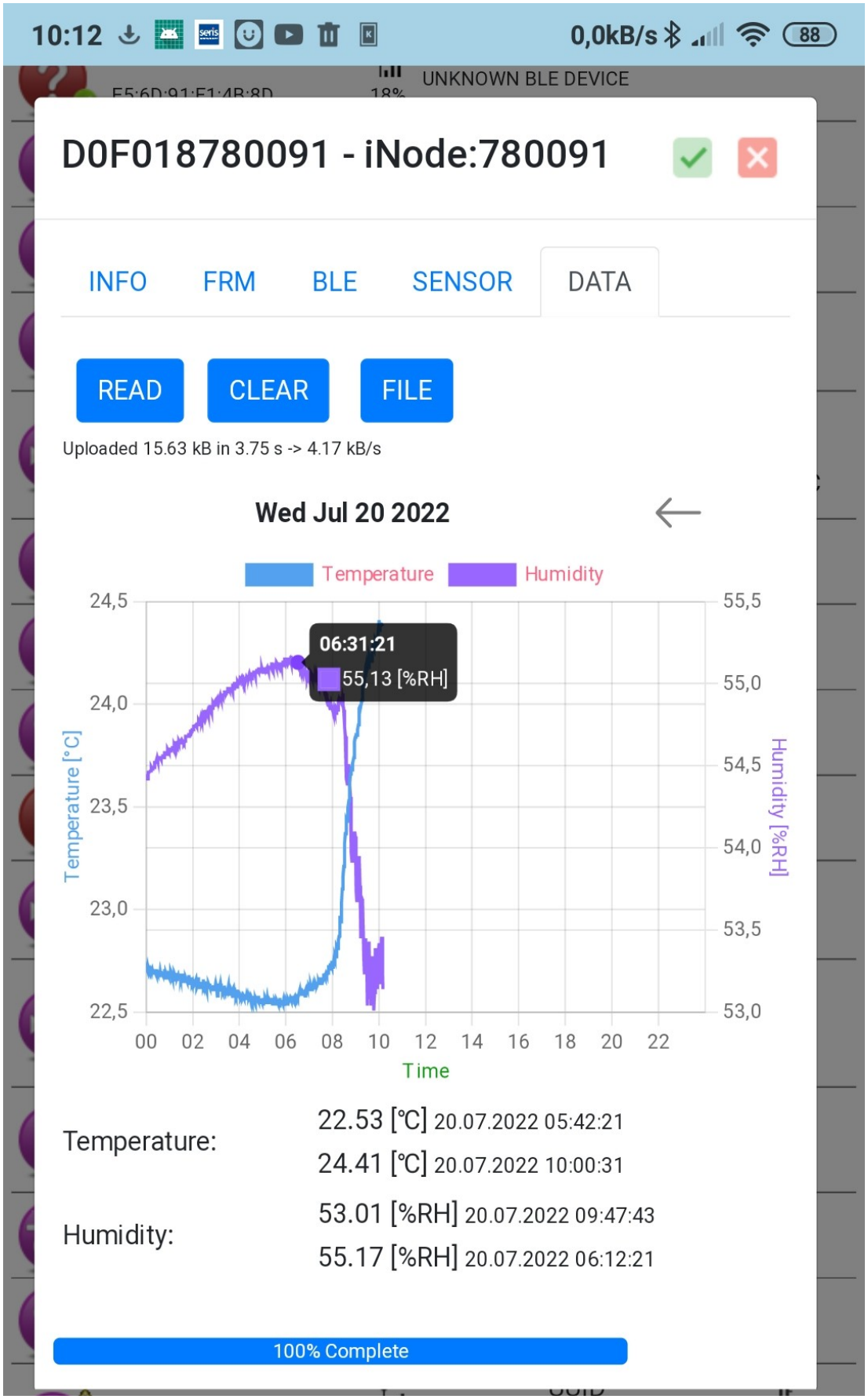
Odczytanie danych odbywa się po naciśnięciu przycisku **READ**. Jeżeli transfer danych odbył się prawidłowo to pojawi się przycisk **FILE**, który umożliwia pobranie z aplikacji odczytanych z czujnika danych. Dane są zapisywane w pliku ZIP zarówno w postaci binarnej – do dalszego ich przetworzenia oraz już przetworzonej, jako pliki typu CSV.

Przycisk **CLEAR** służy do skasowania pamięci danych w czujniku.

Jeżeli dane zostały odczytane prawidłowo to pojawi się wykres z ich wizualizacją za dany dzień. Strzałka w lewo i w prawo umożliwia zmianę dnia za jaki wyświetlane są dane na wykresie. Dane na wykresie mogą być powiększane i przesuwane.

Pod wykresem wyświetlane są informacje (zależne od typu czujnika):

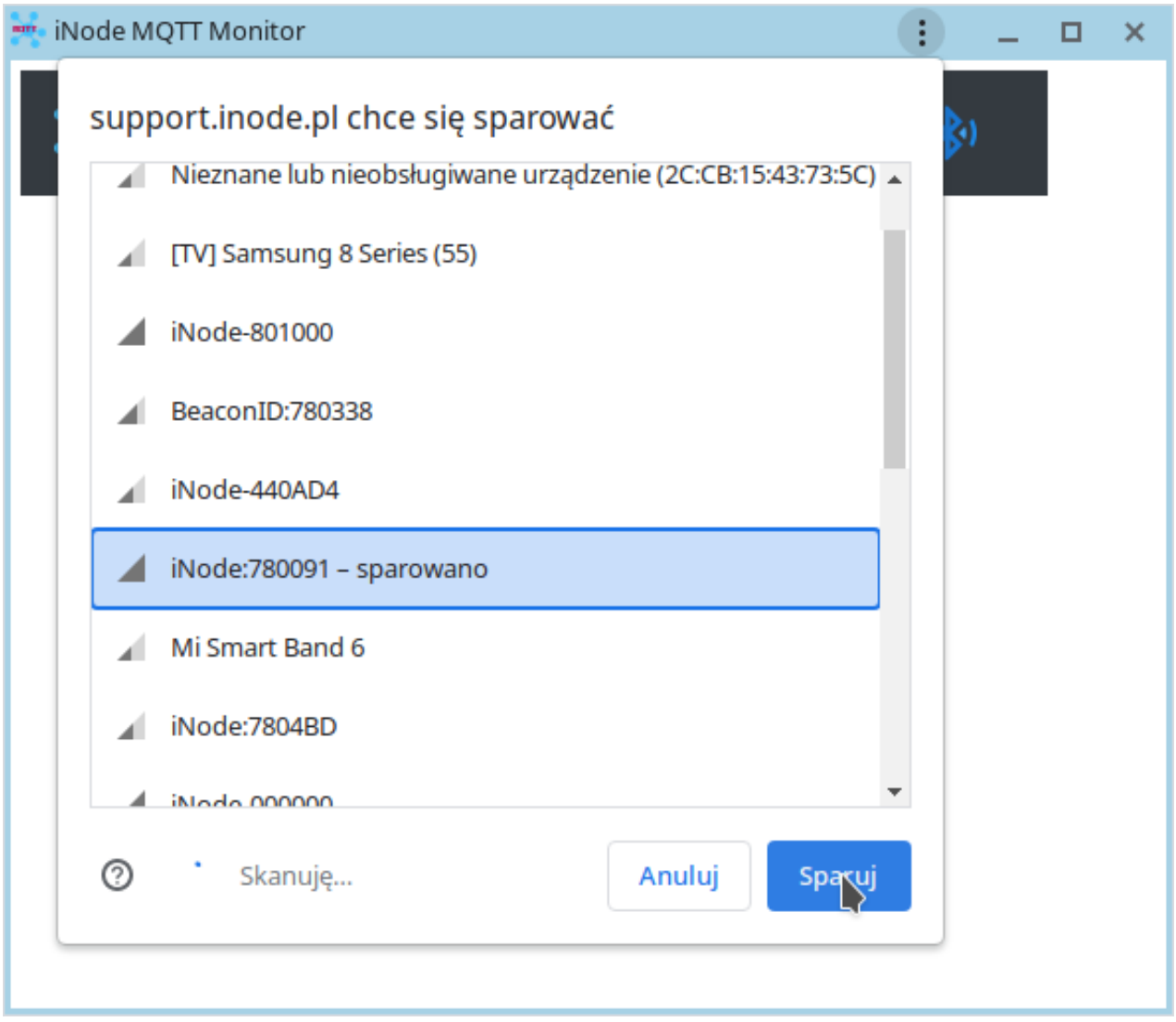
- **Temperature** – zarejestrowana temperatura: najniższa, najwyższa w danym dniu;
- **Humidity** - zarejestrowana wilgotność: najniższa, najwyższa w danym dniu;


10:12
0,0kB/s
88
UNKNOWN BLE DEVICE
D0F018780091 - iNode:780091
INFO
FRM
BLE
SENSOR
DATA
READ
CLEAR
FILE
Uploaded 15.63 kB in 3.75 s -> 4.17 kB/s
Wed Jul 20 2022
Temperature
Humidity
06:31:21
55,13 [%RH]
Temperature [°C]
Humidity [%RH]
Time
Temperature:
22.53 [℃] 20.07.2022 05:42:21
24.41 [℃] 20.07.2022 10:00:31
Humidity:
53.01 [%RH] 20.07.2022 09:47:43
55.17 [%RH] 20.07.2022 06:12:21
100% Complete

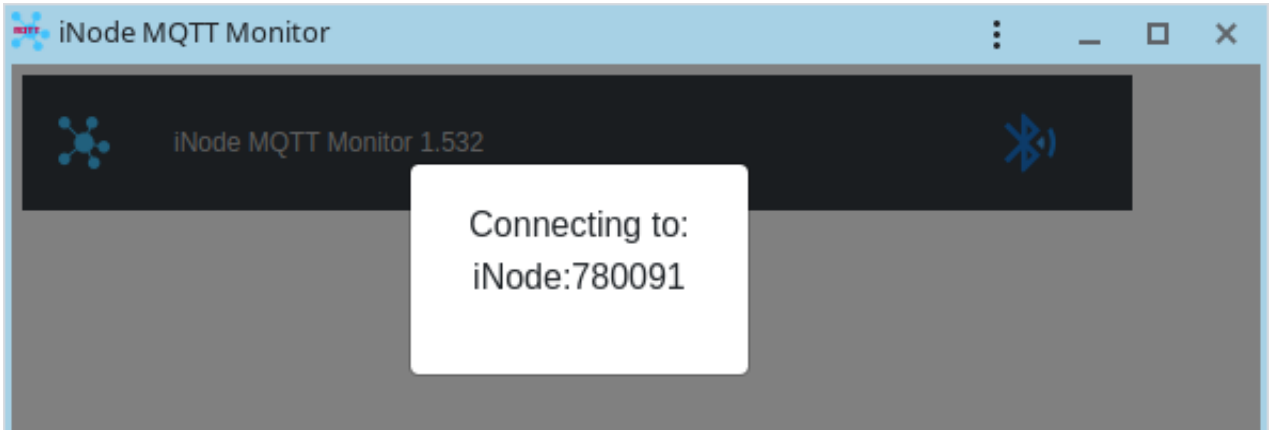
3. Linux OS lub Windows OS

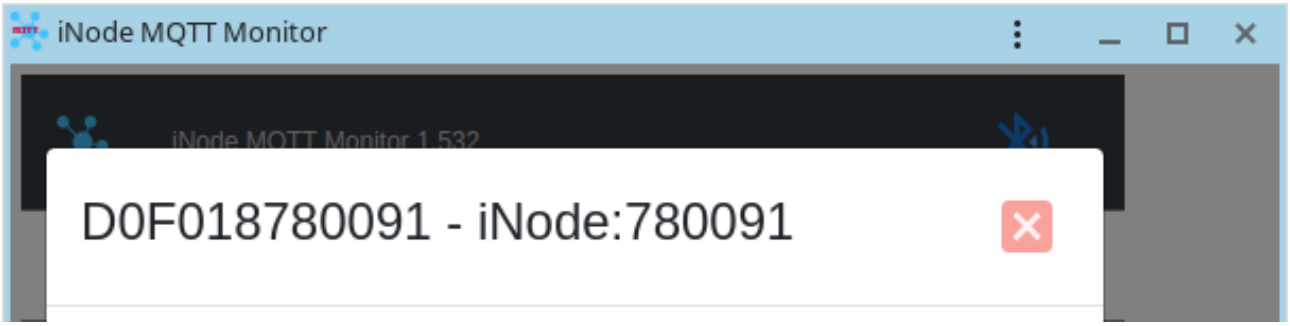
Ponieważ aplikacja **iNode MQTT Monitor** wykorzystuje technologię Web Bluetooth to jest możliwe jej używanie pod systemem Linux lub Windows, tak, jak na telefonie z Android OS. Różnica jest w tym, że w przeglądarce Chrome uruchomionej w systemie Linux OS lub Windows OS nie działa skanowanie. Można jednak połączyć się z urządzeniem Bluetooth wybierając je w okienku systemowym.

W tym celu należy kliknąć na obrazku  w prawym górnym rogu okienka aplikacji. Pojawi się wtedy systemowe okienko skanowania:

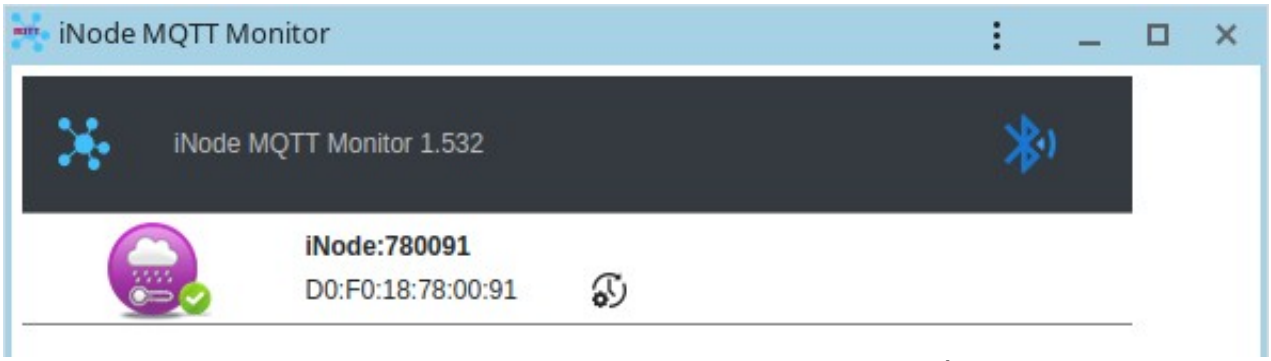


Po wybraniu urządzenia i wciśnięciu przycisku ***Sparuj*** aplikacja połączy się z nim.

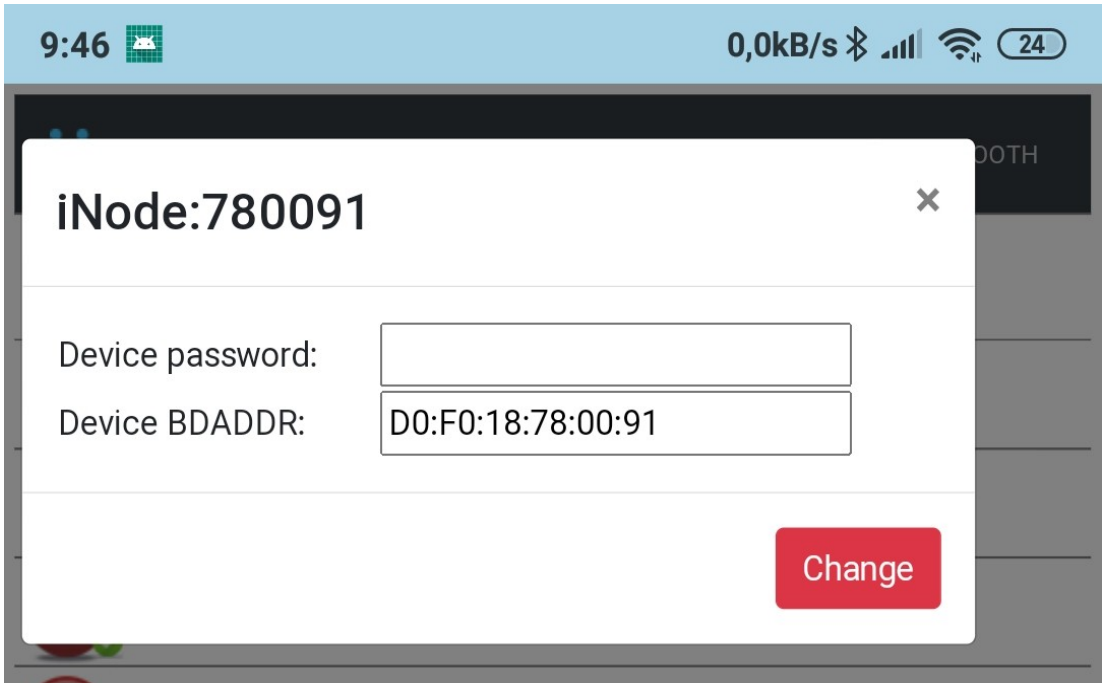




Po zamknięciu okienka urządzenie zostanie dopisane do listy szybkiego dostępu. Można się będzie z nim wtedy połączyć po wybraniu ikony



Przy ponownym połączeniu się z urządzeniem może być konieczne podanie hasła. Nie ma znaczenia, czy hasło jest potrzebne do nawiązania połączenia, czy też nie. Aplikacja wymaga, aby hasło było wprowadzone. W przypadku, gdy nie jest ono potrzebne należy tylko wcisnąć przycisk **Change** co spowoduje zamknięcie okienka.


9:46
0,0kB/s
24
iNode:780091
Device password:
Device BDADDR:
D0:F0:18:78:00:91
Change

4. Format danych JSON

4.1. Odszyfrowane dane JSON:

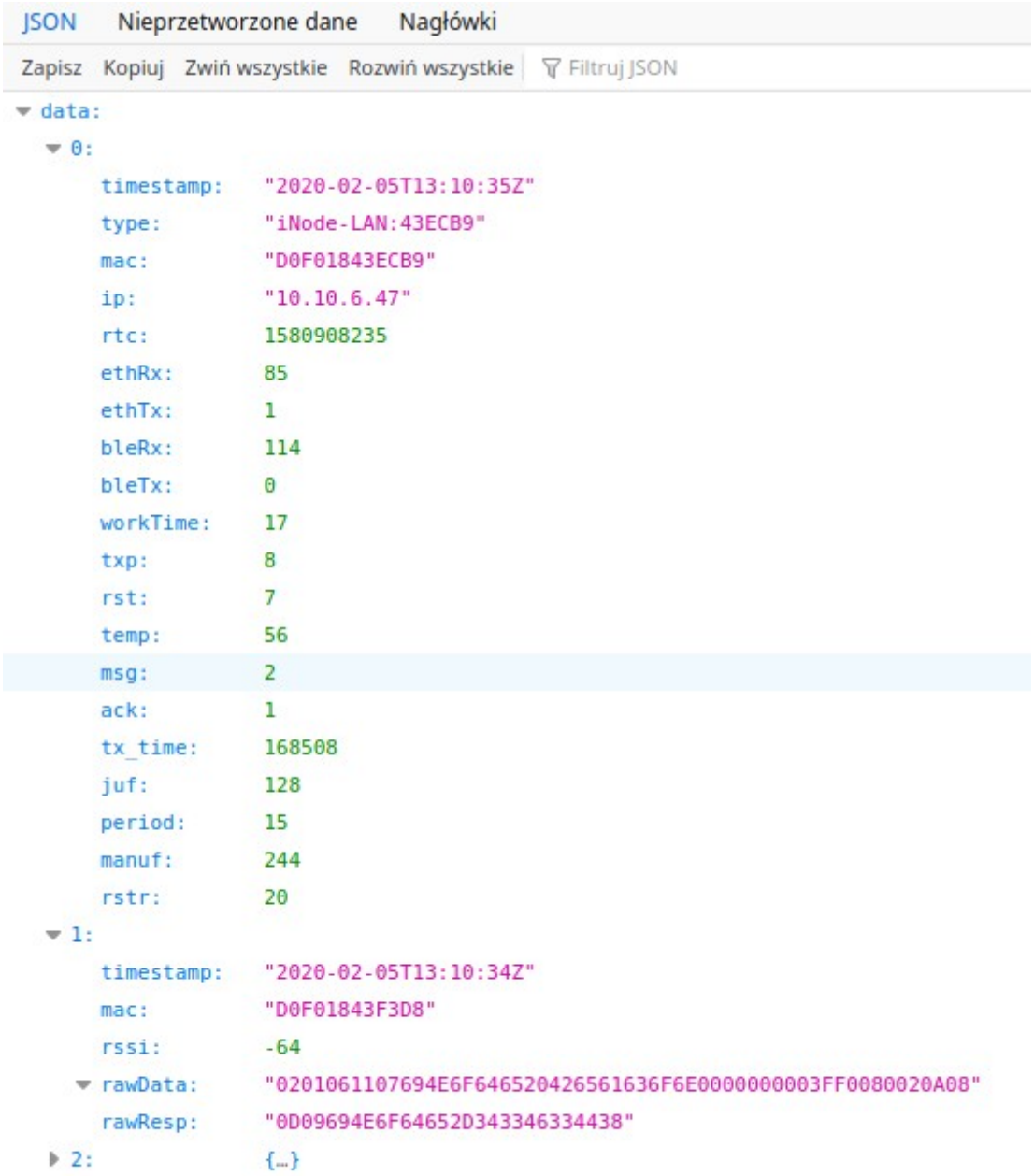
Na pierwszej pozycji przesyłanych danych JSON - tablicy ***data*** znajdują się informacje na temat **iNode LAN Central**.

- timestamp – znacznik czasu
- type – nazwa
- mac – adres mac
- rtc – czas urządzenia w sekundach
- ethRx – liczba ramek odebranych przez LAN
- ethTx – liczba ramek wysłana przez LAN
- bleRx- liczba ramek odebrana przez BLE
- bleTx - liczba ramek wysłanych przez BLE (tylko jeśli jest włączone skanowanie aktywne)
- workTime – czas pracy urządzenia w sekundach
- txp – ustawiona moc nadawania
- rst – liczba resetów urządzeniach
- temp – temperatura urządzenia w stopniach Celsjusza
- msg – liczba wysyłek danych JSON
- ack – liczba potwierdzonych wysyłek danych
- tx_time – czas wysyłki danych JSON w mikrosekundach
- juf – informacje o zabezpieczeniach
- period – okres wysyłania danych JSON
- manuf – kod typu urządzeniach
- rstr – przyczyna ostatniego resetu

{"data": [{"timestamp": "2020-02-05T13:10:35Z","type": "iNode-LAN:43ECB9","mac": "D0F01843ECB9","ip": "10.10.6.47","rtc": 1580908235,"ethRx": 85,"ethTx": 1,"bleRx": 114,"bleTx": 0,"workTime": 17,"txp": 8,"rst": 7,"temp": 56,"msg": 2,"ack": 1,"tx_time": 168508,"juf": 128,"period": 15,"manuf": 244,"rstr": 20},{"timestamp": "2020-02-05T13:10:34Z","mac": "D0F01843F3D8","rssi": -64,"rawData": "0201061107694E6F646520426561636F6E0000000003FF0080020A08","rawResp": "0D09694E6F64652D343346334438"},{"timestamp": "2020-02-05T13:10:35Z","mac": "35FD6890709F","rssi": -64,"rawData": "1EFF06000109200240FFC4543224734A4054BEABBFCF2DE413BBB209EC8750","rawResp": ""},{"timestamp": "2020-02-05T13:10:35Z","mac": "1DB6F43813AF","rssi": -66,"rawData": "1EFF060001092002468330 7B82213C7F54DD5BBB25CE60CE90ED7FC8E15B97","rawResp": ""},{"timestamp": "2020-02-05T13:10:34Z","mac": "FE2BD474F9EC","rssi": -57,"rawData": "031941030201060303E7FE09FFF8F8ECF974D42BFE0409503131","rawResp": ""},{"timestamp": "2020-02-05T13:10:35Z","mac": "D0F01843DB0F","rssi": -57,"rawData": "02010603FFC088020AFE0D09694E6F64652D343344423046","rawResp": ""},{"timestamp": "2020-02-05T13:10:34Z","mac": "D0F01843F3DA","rssi": -79,"rawData": "02010619FFC09B01B000000000931800000F00CCF6B00428FEE4821F30","rawResp": "0D09694E6F64652D343346334441020AFE"},{"timestamp": "2020-02-05T13:10:33Z","mac": "D0F01843F15C","rssi": -77,"rawData": "0201060EFFA0820000C31300006440B100A0020A08","rawResp": "0D09694E6F64652D343346313543"},{"timestamp": "2020-02-05T13:10:33Z","mac": "D8A01D6B8E1A","rssi": -79,"rawData": "0F0843656E7472616C2D364238453141","rawResp": ""},{"timestamp": "2020-02-05T13:10:35Z","mac": "30F7724CCFF0","rssi": -83,"rawData": "1AFF4C00021550765CB7D9EA4E2199A4FA879613A49270D8B708CE","rawResp": ""},{"timestamp": "2020-02-05T13:10:34Z","mac": "D0F01843F3DB","rssi": -84,"rawData": "02010619FF909B01C000000000471836100F0090F6D67324F4862721D3","rawResp": "0D09694E6F64652D343346334442020AFE"},{"timestamp": "2020-02-05T13:10:34Z","mac": "D0F01843F1E4","rssi": -52,"rawData": "0201060EFF908200004D000000E803B20080020AFE","rawResp": "0D09694E6F64652D343346314534"},{"timestamp": "2020-02-05T13:10:34Z","mac": "D0CF5E03930E","rssi": -66,"rawData": "0201060EFFA08200009F000000E803A00000020A14","rawResp":

"0D09694E6F64652D303339333045"},{"timestamp": "2020-02-05T13:10:34Z","mac": "D0F01843E2C5","rssi": -74,"rawData": "02010619FF129D01B00004643F3519D812390034FE30FD8292C68368FC","rawResp": "0D09694E6F64652D343345324335020A06"},{"timestamp": "2020-02-05T13:10:34Z","mac": "D0F01843F2FA","rssi": -72,"rawData": "0201060EFFA08200001000000E843B00000020AFE","rawResp": "0D09694E6F64652D343346324641"},{"timestamp": "2020-02-05T13:10:34Z","mac": "D0F01843F3D5","rssi": -74,"rawData": "0201061107694E6F646520426561636F6E0000000003FF0080020AFE","rawResp": "0D09694E6F64652D343346334435"},{"timestamp": "2020-02-05T13:10:34Z","mac": "D0F01843F2D9","rssi": -70,"rawData": "02010619FF1A9500C00000B807F81F46001B001ADC26436A4EC8AAD4CF","rawResp": "0D09694E6F64652D343346324439020A08"},{"timestamp": "2020-02-05T13:10:34Z","mac": "D0CF5E039918","rssi": -74,"rawData": "02010619FF009BFF80000000009E1813130000A9FEC617BB7D98BD395D","rawResp": "0D09694E6F64652D303339393138"},{"timestamp": "2020-02-05T13:10:33Z","mac": "D0F01843F3DC","rssi": -79,"rawData": "02010619FF909101B00000077CA08000000F0090F605A530C80FB0D0F4","rawResp": "0D09694E6F64652D343346334443020AFE"},{"timestamp": "2020-02-05T13:10:35Z","mac": "D0F01843F150","rssi": -79,"rawData": "02010619FF149E00C000006D02A10000000900A24846D5481621C47BE9","rawResp": "0D09694E6F64652D343346313530020A08"},{"timestamp": "2020-02-05T13:10:33Z","mac": "D0F01843F458","rssi": -79,"rawData": "02010619FF129D01C000047C3FD818C80E0000163E538DEF07F9AF5D84","rawResp": "0D09694E6F64652D343346343538020AFE"},{"timestamp": "2020-02-05T13:10:34Z","mac": "0CF3EEA355A1","rssi": -74,"rawData": "02010417FF0C0300B8DC2023FEE4110FBE2000145E45041DBF000003030118","rawResp": ""}]}

4.2. Przetworzone dane JSON



4.3. Zaszyfrowane dane JSON:

W przypadku, gdy dane z **iNode LAN Central** są zakodowane na początku pliku JSON jest pole ***key***. Jest to tymczasowy klucz użyty do zaszyfrowania danych JSON. Jest on zaszyfrowany kluczem głównym wpisanym do **iNode LAN Central**. Dane są szyfrowane algorytmem ARCFour.

{"key": "33FE46D546832247CC91A8EA733D56E9","data": [Q}H/k_ {.m Z u Ö-
G z Tw e Zn&v1 H T} qE l m qq O J)};)x m n
+e UC IA7!j* @] + /k'v 0Ee3@^f[+">
y aK B\ 9: l BQ]k A6 hY(} w |SL_h)
j .xOLT S @

T□h□_O
□□Q1□□]}

4.4. Odszyfrowywanie danych JSON

Poniżej przykładowa funkcja funkcja w JavaScript dekodująca zaszyfrowane dane JSON. Plik jsaes.js jest do pobrania spod adresu:
https://support.inode.pl/apps/iNodeMqttMonitor/js/jsaes.js

```
/*
 * RC4 symmetric cipher encryption/decryption
 * @license Public Domain
 * @param string key - secret key for encryption/decryption
 * @param string str - string to be encrypted/decrypted
 * @return string
 */

var rc4_s = [];
var rc4_i;
var rc4_j;

function rc4_init(rc4_key, rc4_key_length)
{
   var j = 0, x;
   for (var i = 0; i < 256; i++) {
      rc4_s[i] = i;
   }
   for (i = 0; i < 256; i++) {
      j = (j + rc4_s[i] + rc4_key[i % rc4_key_length]) % 256;
      x = rc4_s[i];
      rc4_s[i] = rc4_s[j];
      rc4_s[j] = x;
   }

   rc4_i=0;
   rc4_j=0;
}

function rc4_get_xor_byte()
{
   var x;

   rc4_i = (rc4_i + 1) % 256;
   rc4_j = (rc4_j + rc4_s[rc4_i]) % 256;
   x = rc4_s[rc4_i];
   rc4_s[rc4_i] = rc4_s[rc4_j];
   rc4_s[rc4_j] = x;
   return rc4_s[(rc4_s[rc4_i] + rc4_s[rc4_j]) % 256];
}

var JSON_USER_KEY = new Array(16);
var JSON_DECRYPT_KEY = new Array(16);

function searchCommentValue(sstr, key)
{
   var offset_start=sstr.search(key);

   if(offset_start>=0)
   {
```

```
        var rsstr=sstr.slice(offset_start+key.length);
        var offset_end=rsstr.search('"');
        return rsstr.substring(0,offset_end);
    }
    else
    {
        return "";
    }
}

function decodeJSON(json_raw, json_key)
{
    var img_dataView = new DataView(json_raw);
    var ik;
    var img_byte;
    var xor_byte=0;
    var ik_img_offset=0;

    for(var i = 0; i < 16; i++)
    {
        JSON_USER_KEY[i] = 0;
    }

    for(var i = 0; i < json_key.length; i+=2)
    {
        JSON_USER_KEY[15-i] = json_key.charCodeAt(i+1);
        JSON_USER_KEY[14-i] = json_key.charCodeAt(i);
    }

    var JSON_KEY=searchCommentValue(ab2str(json_raw),'{"key": "');

    if(JSON_KEY.length!=0)
    {
        AES_Init();

        var block = new Array(16);
        for(var i = 0; i < 16; i++)
          { block[15-i] = parseInt(JSON_KEY.substr(i*2,2), 16) };

        var key = new Array(16);
        for(var i = 0; i < 16; i++)
            { key[i] = JSON_USER_KEY[i]; }

        AES_ExpandKey(key);
        AES_Decrypt(block, key);

        for(var i = 0; i < 16; i++)
            { JSON_DECRYPT_KEY[15-i] = block[i] };

        AES_Done();

        rc4_init(JSON_DECRYPT_KEY,16);

        var json_data_start=ab2str(json_raw).search(',"data": ')+10;

         for(ik=json_data_start;ik<(img_dataView.byteLength-2);ik++)
        {
```

```
            img_byte=img_dataView.getUint8(ik);

            img_dataView.setUint8(ik,(img_byte^rc4_get_xor_byte())& 0xff);
        }
    }
    return json_raw;
}
```

4.5. Dekodowanie danych BLE:

Sposób kodowania danych w ramce rozgłoszeniowej BLE lub odpowiedzi na zapytanie aktywne. Informacje na temat kodów **AD Type** można znaleźć w Core_V4.0.pdf: Volume 3 Part C, Section 8.i na stronie https://www.bluetooth.org/en-us/specification/assigned-numbers/generic-access-profile

ramka rozgłoszeniowa:
02010619FF12930110000017 00AB18951F485435BE5B809D6F571E40E 8FE0000

020106
02 -> długość pola danych: 2 bajty
0106 -> dane
01 -> 0x01 -> EIR Data Type = 0x01 -> «Flags»
06 -> 0x06 -> EIR Data = 0x06 -> LE General Discoverable Mode (bit 1), BR/EDR Not Supported (bit 2)
19FF12930110000017 00AB18951F485435BE5B809D6F571E40E8
19 -> długość pola danych: 25 bajtów
FF12930110000017 00AB18951F485435BE5B809D6F571E40E8 -> dane(25 bajtów)
FF -> 0xFF -> EIR Data Type = 0xFF «Manufacturer Specific Data»
12930110000017 00AB18951F485435BE5B809D6F571E40E8 ->
1293 -> 0x9312 -> 0x93XX identyfikator iNodeCareSensor #3; 0xXX1X wersja 1; 0xXXX2 od ostatniego odczytu pamięci minęły 24 h;

0110 -> 0x1001 type -> bit 15 do bit 12 -> zarezerwowane, bit 11 do bit 0 -> adres czujnika w grupie
0000 -> 0x0000 flags ->
SENSOR_ALARM_MOVE_ACCELEROMETER=1,
SENSOR_ALARM_LEVEL_ACCELEROMETER=2,
SENSOR_ALARM_LEVEL_TEMPERATURE=4,
SENSOR_ALARM_LEVEL_HUMIDITY=8,
SENSOR_ALARM_CONTACT_CHANGE=16
1700 -> 0x0017 value1
/* motion sensor */
0x8000 czujnik jest w ruchu (bit 15 =1)

bity 14 do 10:
składowa X położenia (wartość 5 bitowa ze znakiem) -> 0x00= 0
bity 9 do 5:
składowa Y położenia (wartość 5 bitowa ze znakiem) -> 0x00= 0
bity 4 do 0:
składowa Z położenia (wartość 5 bitowa ze znakiem) -> 0x17= -9

AB18 -> 0x18AB value2
/* temperature sensor */
Temperature= ((175.72 * Temp_Code)/65536)-46.85 [°C]
Temp_Code = 0x18AB *4 = 0x62AC = 25260
Temperature = 20,879 °C

951F -> 0x1F95 value3
/* humidity sensor */
%RH= ((125*RH_Code)/65536)-6 [%]
RH_Code = 0x1f95 *4 = 0x7e54 = 32340
%RH= 55,68 %

485435BE -> 0x5448BE35 time (znacznik czasu; liczba sekund od 01.01.1970)

5B80 9D6F 571E 40E8 -> cyfrowy podpis AES128 dla powyższych danych

ramka z odpowiedzią na zapytanie aktywne:

0D09694E6F64652D3335363134410 20A020000000000000000000000000

0D09694E6F64652D333536313441

0D -> długość pola danych: 13 bajtów

09694E6F64652D333536313441 -> dane

09 -> 0x09 -> EIR Data Type = 0x09 -> «Complete Local Name»

694E6F64652D333536313441 -> iNode-35614A

020A02

02 -> długość pola danych: 2 bajty

0A02 -> dane

0A-> 0x0A -> EIR Data Type = 0x0A -> «Tx Power Level»

02 -> 0x02 -> Tx Power Level = +2dBm