

iNode MCU USB

instrukcja użytkownika

© 2019 ELSAT®

1. Wstęp

Chcielibyśmy Państwu przedstawić rodzinę urządzeń **iNode** działających w technologii **Bluetooth Low Energy** ®. Pokażemy Państwu, że BLE to nie tylko tagi do znajdowania zagubionych kluczy, czy tagi lokalizacyjne, lecz jeszcze coś więcej.

Nasze urządzenia potrafią to i jeszcze więcej:

- Są to przede wszystkim urządzenia bateryjne.
- Działają bez jej wymiany do 36 miesięcy w zależności od zastosowania i sposobu użycia.
- Mają pamięć do rejestrowania zdarzeń, odczytów pomiarów etc.
- Precyzyjne czujniki temperatury, wilgotności, przyspieszenia czy pola magnetycznego pozwalają na precyzyjne sterowanie automatyką domową czy też opiekę nad ludźmi starszymi.
- Jako urządzenia zdalnego sterowania, mimo małego poboru mocy, mają duży zasięg i cechy niedostępne dla innych konkurencyjnych urządzeń – własne hasło użytkownika, szyfrowanie AES, sterowanie bezpośrednio ze smartfona.

iNode może też pomóc w kontroli przemieszczania się osób czy towarów, zapisując czas pojawienia się i zniknięcia z zasięgu rejestratora (aktywne **RFID**® o dużym zasięgu). Nowe funkcjonalności związane z rozwojem produktu to też nie problem – umożliwia to zdalna wymiana firmware przez OTA a funkcjonalności (skryptów LUA) przez smartfona z **Bluetooth 4.0/4.1** ® i obsługą **Bluetooth Low Energy** ® (**Bluetooth Smart** ®).

iNode MCU USB jest urządzeniem o modyfikowalnej funkcjonalności, gdyż działa pod kontrolą systemu NodeMCU ESP32 – interpretera skryptowego języka programowania [LUA](#) (opartego o Lua 5.1.4). NodeMCU to system oparty o kod open source Lua dla układu [ESP32 SOC](#) firmy Espressif, który używa wykorzystującego pamięć flash modułu systemu plików [SPIFFS](#). iNode MCU USB łączy ze sobą dwie bezprzewodowe technologie: Bluetooth 4.1 i WiFi. Dla bezpieczeństwa użytkownika firmware Node MCU jest podpisany cyfrowo i można go wymienić tylko przez OTA.

Użytkownik może napisać własne skrypty LUA i wpisać je do urządzenia. Służy to tego aplikacja **iNode MCU FFS**. W serwisie pomocy technicznej <https://support.inode.pl> są do dyspozycji użytkownika darmowe skrypty LUA z różnymi funkcjonalnościami np. **iNode Central MCU** – odpowiednika iNode Care GSM.

W wersji WebUSB **iNode MCU USB** jest możliwa bezpośrednia komunikacja ze skryptu LUA z przeglądarką Google CHROME. WebUSB to swego rodzaju mostek pomiędzy urządzeniami USB a przeglądarkami.

<https://developers.google.com/web/updates/2016/03/access-usb-devices-on-the-web>

Znaki towarowe lub zarejestrowane znaki towarowe:

Bluetooth Low Energy ®, **Bluetooth 4.0** ®, **RFID**®, **CSR**®, **Windows**®, **Android**, **Google**, **Microsoft**, **ThingSpeak**, **Raspberry Pi**, **Domoticz**, **BlueZ** , **Linux** , **NodeMCU**, **ESP32**, **Espressif**® są użyte w niniejszej broszurze wyłącznie w celach informacyjnych i należą do ich właścicieli.

2. NodeMCU ESP32

NodeMCU ESP32 jest interpreterem języka [LUA](#) dla układów ESP32 WiFi SOC firmy Espressif. Wykorzystuje Espressif IoT Development Framework i używa systemu plików (również na karcie SD - FAT).

NodeMCU ESP32:

- to prosty w programowaniu bezprzewodowy węzeł i/lub punkt dostępowy (*access point*);
- jest oparty o [Lua](#) 5.1.4 (bez modułów debug i os);
- to asynchroniczny i zdarzeniowy model programowania;
- to ponad 10 wbudowanych modułów np. WiFi, Bluetooth, MQTT, https;
- to uaktualniana dokumentacja na <https://nodemcu.readthedocs.io/en/dev-esp32/>;

Model programowania zastosowany w **NodeMCU** jest podobny do tego w Node.js, tylko wykorzystuje Lua. Jest asynchroniczny i zdarzeniowy. Wiele funkcji wymaga dlatego jako parametru funkcji powrotu (callback). Poniżej pokazujemy przykład programu w **NodeMCU**. Więcej przykładów można znaleźć w katalogu /lua na GitHub.

prosty serwer HTTP

```
srv = net.createServer(net.TCP)
srv:listen(80, function(conn)
    conn:on("receive", function(sck, payload)
        print(payload)
        sck:send("HTTP/1.0 200 OK\r\nContent-Type: text/html\r\n\r\n<h1>
Hello, NodeMCU.</h1>")
    end)
    conn:on("sent", function(sck) sck:close() end)
end)
-- connect to WiFi access point
wifi.mode(wifi.STATION, true)

wifi.sta.on("connected", function() print("connected") end)
wifi.sta.on("got_ip", function(event, info) print("got ip "..info.ip) end)

-- mandatory to start wifi after reset
wifi.start()
wifi.sta.config({ssid="SSID", pwd="password", auto=true}, true)
```

otwarcie dodatkowego UART z funkcjonalnością echa do komunikacji przez WebUSB

```
uart.setup(2, 921600, 8, uart.PARITY_NONE, uart.STOPBITS_2, {tx = 19, rx = 22})
uart.on(2, "data", "\r", function(dat) print("rx:",dat) uart.write(2, dat) end)
uart.start(2)
```

Dokumentację do **NodeMCU** można znaleźć na <https://nodemcu.readthedocs.io/en/dev-esp32/>

Licencja: MIT © zeroday/nodemcu.com

2.1 moduł *bthci*

bthci.rawhci(hcibytes, callback)
bthci.reset(callback)
bthci.adv.enable(onoff, callback)
bthci.adv.setdata(advbytes, callback)
bthci.adv.setparams(paramtable, callback)
bthci.scan.enable(onoff, callback)
bthci.scan.setparams(paramstable, callback)
bthci.scan.on(event, callback)

bthci.notification(charhandle, notbytes) - sends notbytes as BLE notification; charhandle = 0x2c

bthci.gatt.enable(onoff) - enable/disable gatt server

bthci.gatt.on(event, callback) - registers the callback to be passed hci/gatt server event:

- 0 - ERROR
- 1 - CONNECTED
- 2 - DISCONNECTED
- 3 - GATT WRITE REQUEST
- 4 - GATT READ REQUEST
- 5 - CUSTOM DATA
- 6 - FORMAT REQUEST
- 7 - DISCONNECT REQUEST

bthci.gatt.disconnect() - disconnect current BLE connection

bthci.access.enable(onoff) - enable/disable access to custom service

bthci.LED_R - LED red gpio pin

bthci.LED_G - LED G gpio pin

bthci.LED_B - LED B gpio pin

2.2 moduł *fota*

fota.https_ota(server_uri, certfilename) - starts OTA from *server_uri* and with PEM-encoded certificate (or certificates) - *certfilename*

fota.bpaddr() - boot partition address

fota.rpaddr() - running partition address

fota.upaddr() - update partition address

2.3 moduł *node*

node.reset_reason() - returns the boot reason and extended reset info; boot reason: 1 - power on, 3 - software reset; extended reset info: 0x464D54 - reset after file format, 0x434F4F54 - reset after BLE disconnection;

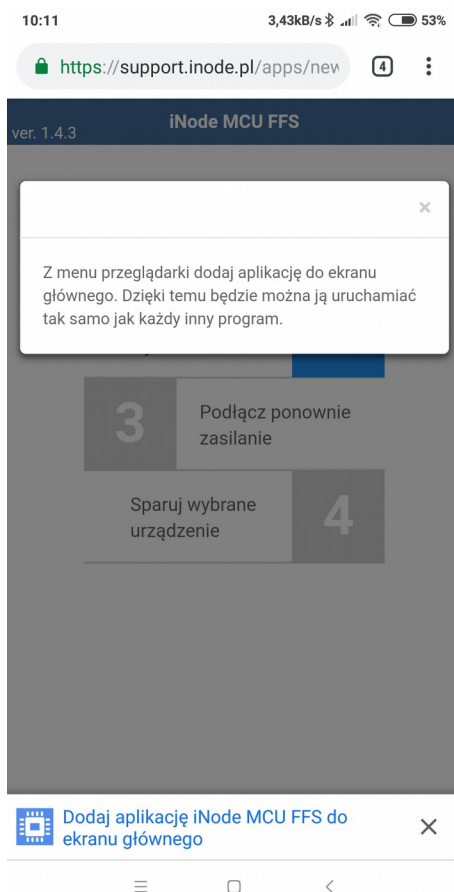
node.restart(rst_reason) - restarts the chip with *rst_reason* value; rst_reason: 0x424C45 - LUA bootloader starts BLE advertising for 10 secs;

3. Zmiana funkcjonalności iNode MCU USB

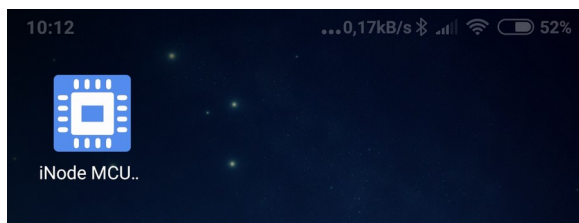
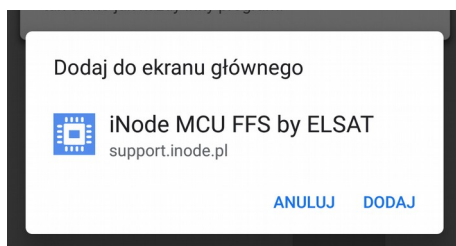
Uruchom w przeglądarce Google CHROME aplikację **iNode MCU FFS**, która umożliwia formatowanie wewnętrznej pamięci flash, wymianę firmware Node MCU przez OTA oraz wgrywanie skryptów LUA do wszystkich urządzeń rodziny **iNode MCU**. Aplikacja działa na systemach: Android OS, Linux, Windows 10;

3.1 Android OS

W przeglądarce CHROME należy wpisać w pasku adresu <https://support.inode.pl/apps/new/iNodeMcuFfs> Aplikacja **iNode MCU FFS** zostanie pobrana z serwera i uruchomiona

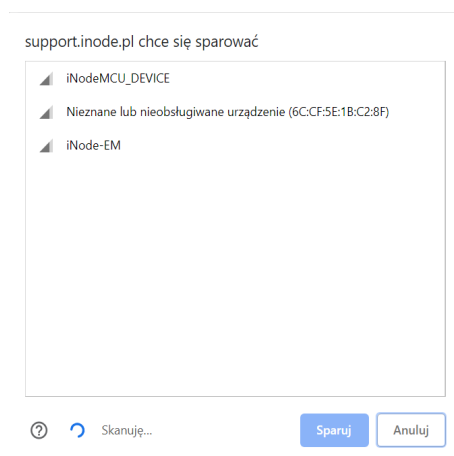


U dołu ekranu pojawi się informacja o możliwości dodania aplikacji **iNode MCU FFS** do ekranu głównego. Należy to zrobić, aby uruchamiać ją w przyszłości tak, jak każdą inną aplikację.



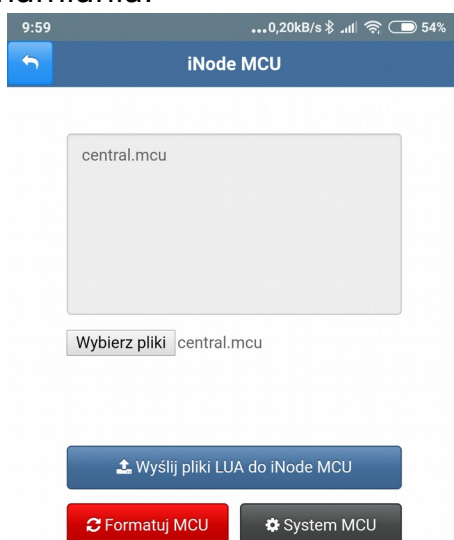
Urządzenie po włączeniu zasilania lub programowym resecie rozgłasza się przez Bluetooth przez 10 sekund co sygnalizowane jest miganiem diody LED na niebiesko. W przypadku, gdy nie ma do niego wgranego żadnego skryptu LUA, który może zostać uruchomiony (brak jest np. pliku *init.lua*) to urządzenie rozgłasza się cały czas. W tym trybie do połączenia się z urządzeniem nie jest potrzebne hasło. Jeżeli natomiast w urządzeniu jest już uruchomiony skrypt LUA z określoną funkcjonalnością to hasło będzie wymagane i aplikacja o nie poprosi. Domyślne hasło do funkcjonalności iNode MCU Central to 1234.

Kliknięcie na polu „2” powoduje pojawienie się systemowego okienka ze znalezionymi urządzeniami BLE. Jeśli takie okienko się nie pojawia to najprawdopodobniej oznacza to wyłączony interfejs BT4.0 (Windows, Linux) lub jego brak.

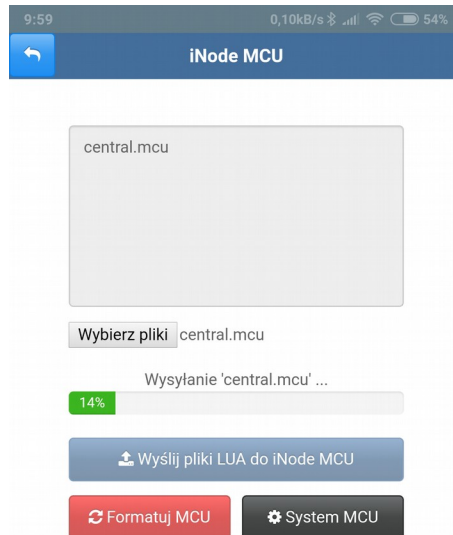


Po połączeniu się z wybranym urządzeniem niebieska dioda LED zaczyna się świecić stale i pojawia się ekran z przyciskami:

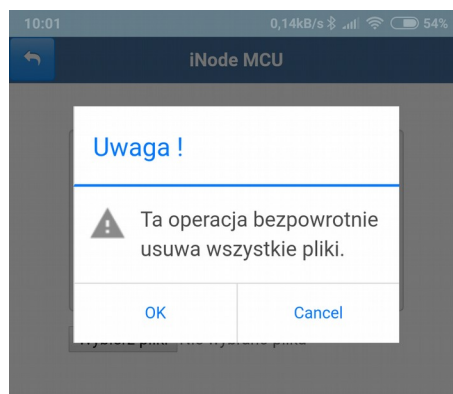
- **Wybierz pliki** - służy do wyboru plików, które użytkownik chce przesłać do urządzenia; Pliki LUA z określoną funkcjonalnością mają rozszerzenie *.mcu*; Przy ich wgrywaniu aplikacja automatycznie tworzy plik *init.lua* służący do ich uruchamiania.



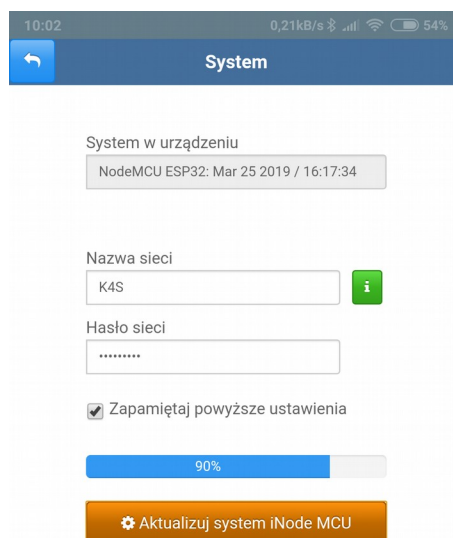
- **Wyślij pliki LUA do iNode MCU** - rozpoczyna transfer plików do urządzenia;



- **Formatuj MCU** - usuwa wszystkie zapisane w pamięci flash pliki; Operacja ta nie dotyczy zbiorów zapisanych na karcie microSD.



- **System MCU** - otwiera okienko umożliwiające aktualizację firmware NodeMCU; Urządzenie samo pobierze odpowiednie pliki z serwera iNode; Do przeprowadzenia tej operacji wymagany jest dostęp do Internetu przez sieć WiFi.



Po wybraniu **Aktualizuj system iNode MCU** zacznie migać czerwona dioda LED. Oznacza to oczekiwanie na nawiązanie połączenia z lokalną siecią WiFi. Zwykle udaje się to zrobić w ciągu kilku sekund. Jeśli dioda nie przestaje migać należy zweryfikować ustawienia dotyczące sieci WiFi.

Po nawiązaniu połączenia rozpocznie się pobieranie plików i czerwona dioda LED zacznie stale świecić. W trakcie pobierania danych dioda LED świeci na przemian na niebiesko i czerwono. Należy odczekać aż dioda zgaśnie. Oznaczać to będzie reset urządzenia. Po ponownym połączeniu się z urządzeniem będzie można sprawdzić, czy firmware Node MCU został wymieniony.

4. Funkcjonalność WebUSB

W wersji [WebUSB iNode MCU USB](#) jest możliwa bezpośrednia komunikacja ze skryptu LUA z przeglądarką Google CHROME. Komunikację przez WebUSB obsługuje układ STM32L082. Do wymiany firmware w tym układzie przeznaczona jest aplikacja [iNode STM32 Utility](#).

Poniżej parametry interfejsu WebUSB konieczne do nawiązania połączenia:

Identyfikator USB producenta i wyrobu:
Vendor ID = 1155, Product ID = 22336

Numer interfejsu przeznaczonego do komunikacji WebUSB:
Interface Number = 2

Identyfikator endpoint przeznaczonego do komunikacji WebUSB:
Endpoint ID = 3

Klasa interfejsu przeznaczonego do komunikacji WebUSB:
Interface class = 255

Vendor-specific requests:

0x30 - open interface

```
this.device_.controlTransferOut({
  'requestType': 'class',
  'recipient': 'interface',
  'request': 0x30,
  'value': 0x01,
  'index': this.interfaceNumber_,LCS))
```

0x33 - reopen interface

```
this.device_.controlTransferOut({
  'requestType': 'class',
  'recipient': 'interface',
  'request': 0x33,
  'value': 0x01,
  'index': this.interfaceNumber_,LCS);
```

```

/*****
/* Line Coding Structure
/*-----*/
/* Offset | Field      | Size | Value | Description
/* 0      | dwDTERate  | 4    | Number | Data terminal rate, in bits per second
/* 4      | bCharFormat | 1    | Number | Stop bits
/*
/*          |             |      |        | 0 - 1 Stop bit
/*          |             |      |        | 1 - 1.5 Stop bits
/*          |             |      |        | 2 - 2 Stop bits
/* 5      | bParityType | 1    | Number | Parity
/*          |             |      |        | 0 - None
/*          |             |      |        | 1 - Odd
/*          |             |      |        | 2 - Even
/*          |             |      |        | 3 - Mark
/*          |             |      |        | 4 - Space
/* 6      | bDataBits  | 1    | Number | Data bits (5, 6, 7, 8 or 16).
*****/
```

0x31 - close interface

```
this.device_.controlTransferOut({
  'requestType': 'class',
  'recipient': 'interface',
  'request': 0x31,
  'value': 0x00,
  'index': this.interfaceNumber_})
```

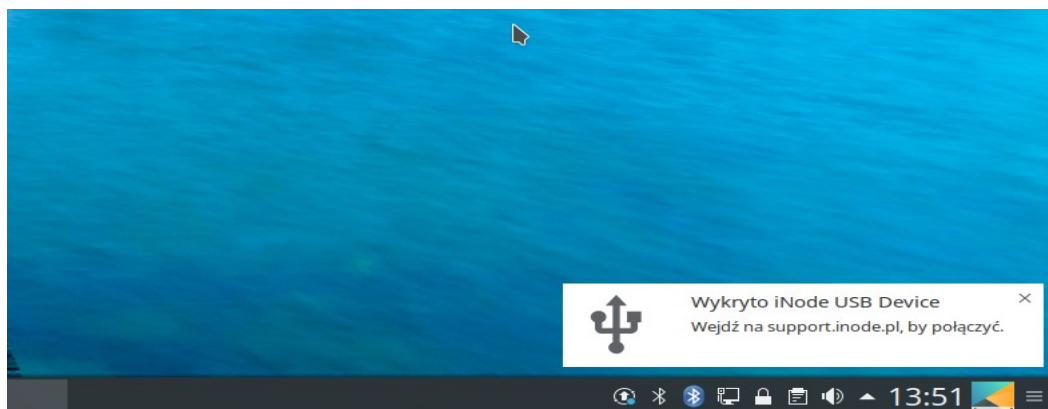
0x32 - read device type

```
this.device_.controlTransferIn({
  'requestType': 'class',
  'recipient': 'interface',
  'request': 0x32,
  'value': 0x01,
  'index': this.interfaceNumber_,1}))
```

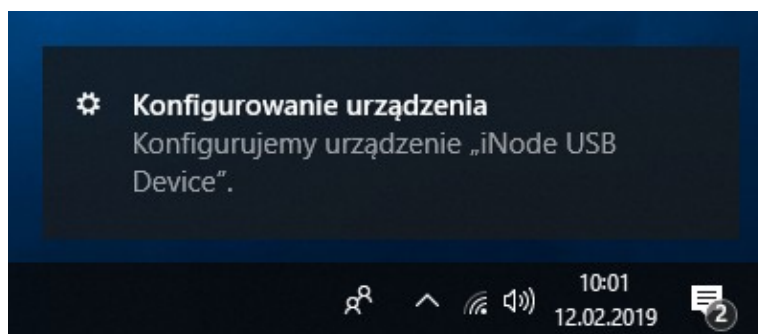
Pod systemem Linux uruchomienie aplikacji wykorzystującej funkcjonalność WebUSB jest trochę bardziej skomplikowane, gdyż na większości systemów opartych o Linux, urządzenia USB są standardowo dołączane z uprawnieniami typu read-only. Żeby zezwolić Chrome na użycie adaptera **iNode MCU USB** należy dodać nową regułę udev. W tym celu, w katalogu `/etc/udev/rules.d/` stworzymy plik o nazwie **50-inode.rules** z następującą zawartością:

```
SUBSYSTEM=="usb", ATTR{idVendor}=="0483", MODE="0664", GROUP="plugdev"
```

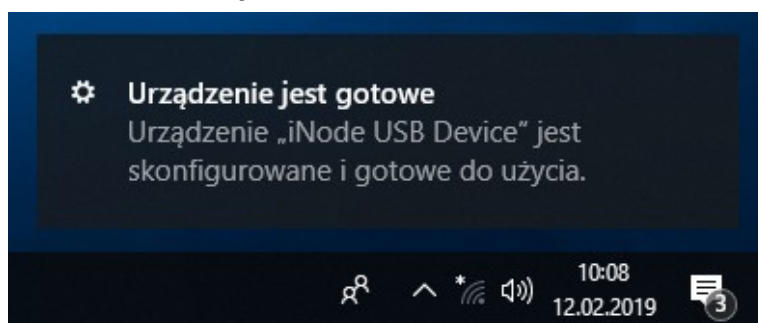
Dodatkowo należy upewnić się, że użytkownik jest członkiem grupy *plugdev*. Następnie należy się wylogować i zalogować ponownie, aby wprowadzone modyfikacje zostały użyte. Wtedy możemy już uruchomić przeglądarkę Chrome i wpisać na pasku adresu <chrome://flags/#enable-experimental-web-platform-features> po to, aby Chrome mogło wykorzystywać technologię WebUSB. Po podłączeniu adaptera **iNode MCU USB** powinien pojawić się następujący komunikat systemowy:



Przy pierwszym podłączeniu adaptera **iNode MCU USB** do komputera z systemem Windows 10 zostanie on rozpoznany jako *iNode USB Device*:

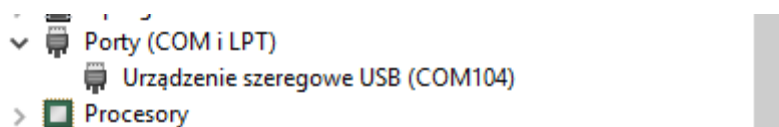


i następnie zainstalowany:

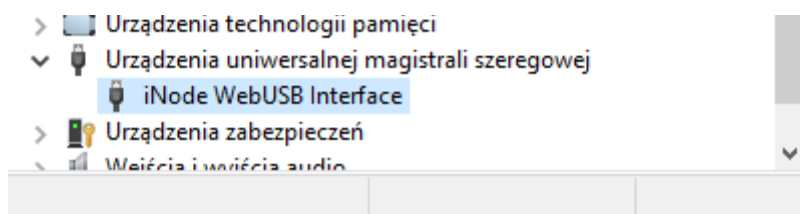


W menedżerze urządzeń adapter, jako urządzenie kompozytowe, jest widziany w kilku kategoriach:

jako port COM:



i jako interfejs WebUSB:



Nie jest wymagane ręczne instalowanie żadnych sterowników do obsługi adaptera!

5. Parametry techniczne

Parametry radiowe:

- RX/TX:
 - Bluetooth/BLE: 2402-2480 MHz
 - WiFi: 2412-2472 MHz / 2422-2462 MHz
- moc wyjściowa (maksymalna):
 - Bluetooth: 6,72 dBm
 - BLE: 8,23 dBm
 - WiFi: 16,81 dBm (802.11b); 17,60 dBm (802.11g); 18,04 dBm (802.11n20); 17,85 dBm (802.11n40);
- modulacja:
 - Bluetooth: GFSK, $\pi/4$ -DQPSK, 8DPSK;
 - BLE: GFSK;
 - WiFi: DSSS, OFDM;
- antena:
 - wewnętrzna PCB typu MIFA, 1,6dBi;

Parametry pracy:

- konfigurowalne z PC przez Bluetooth:
 - nazwa sieci WiFi i hasło do niej;
 - skrypty LUA sterujące pracą urządzenia;
 - wymiana firmware NodeMCU przez OTA (WiFi);

Node MCU:

- wbudowane moduły:
 - ADC, Bit, Bluetooth HCI interface, CRYPTO, Encoder, File, FOTA, GPIO, HTTP, LEDC, MQTT, Net, Node, SD-MMC, Struct, Timer, Uart, WiFi, time;

Zasilanie:

- 5VDC, 50-160 mA w zależności od trybu pracy i wersji;

Obudowa:

- plastikowa;
- wymiary: 55mm x 20mm x 15mm;

Pozostałe:

- sygnalizacja za pomocą diod LED:
 - pracy w trybie konfiguracji;
- złącze standardu USB 2.0;
- pamięć FLASH na skrypty LUA: około 800kB;
- czytnik kart micro SD (opcja):
 - karty o pojemności do 64GB (FAT);
 - obsługa FAT;
- WebUSB (opcja):
 - bezpośrednia komunikacja z przeglądarką Google CHROME;
 - UART podstawowy: 115200 bps 8N1 – komunikacja z Node MCU;
 - UART dodatkowy: 921600 bps 8N2 – komunikacja z aplikacją LUA;
- temperatura pracy: od -20 do 45°C;
- wilgotność: 20 - 80% RHG;
- masa: 15 g;

Oprogramowanie:

- Google CHROME: Android OS, Linux, Windows 10;

Chipset:

- [ESP32](#);

Producent zastrzega sobie prawo do zmiany parametrów urządzenia i oprogramowania oraz wprowadzenia innych rozwiązań konstrukcyjnych.

6. Prawidłowe usuwanie produktu (zużyty sprzęt elektryczny i elektroniczny)



Materiały z opakowania nadają się w 100% do wykorzystania jako surowiec wtórny. Utylizacji opakowania należy dokonać zgodnie z przepisami lokalnymi. Materiały z opakowania należy zabezpieczyć przed dziećmi, gdyż stanowią dla nich źródło zagrożenia. Oznaczenie umieszczone na produkcie lub w odnoszących się do niego tekstach wskazuje, że produktu po upływie okresu użytkowania nie należy usuwać z innymi odpadami pochodzącymi z gospodarstw domowych. Aby uniknąć szkodliwego wpływu na środowisko naturalne i zdrowie ludzi wskutek niekontrolowanego usuwania odpadów, prosimy o

oddzielenie produktu od innego typu odpadów oraz odpowiedzialny recykling w celu promowania ponownego użycia zasobów materialnych jako stałej praktyki.

Właściwa utylizacja urządzenia:



- Zgodnie z dyrektywą WEEE 2012/19/EU symbolem przekreślonego kołowego kontenera na odpady oznacza się wszelkie urządzenia elektryczne i elektroniczne podlegające selektywnej zbiórce.
- Po zakończeniu okresu użytkowania nie wolno usuwać niniejszego produktu razem z normalnymi odpadami komunalnymi, lecz należy go oddać do punktu zbiórki i recyklingu urządzeń elektrycznych i elektronicznych.

Informuje o tym symbol przekreślonego kołowego kontenera na odpady, umieszczony na produkcie lub w instrukcji obsługi lub opakowaniu.

- Zastosowane w urządzeniu tworzywa nadają się do powtórnego użycia zgodnie z ich oznaczeniem. Dzięki powtórnemu użyciu, wykorzystaniu materiałów lub innym formom wykorzystania zużytych urządzeń wnoszą Państwo istotny wkład w ochronę naszego środowiska naturalnego.
- Informacji o właściwym punkcie usuwania zużytych urządzeń elektrycznych i elektronicznych udzieli Państwu administracja gminna lub sprzedawca urządzenia.
- Zużyte, całkowicie rozładowane baterie i akumulatory muszą być wyrzucane do specjalnie oznakowanych pojemników, oddawane do punktów przyjmowania odpadów specjalnych lub sprzedawcom sprzętu elektrycznego.
- Użytkownicy w firmach powinni skontaktować się ze swoim dostawcą i sprawdzić warunki umowy zakupu. Produktu nie należy usuwać razem z innymi odpadami komunalnymi.

Numer Deklaracji: 1/03/2019
Number of declaration of Conformity

Data wystawienia Deklaracji: 01.03.2019 r.
Date of issue of declaration

DEKLARACJA ZGODNOŚCI UE RED
UE RED DECLARATION OF CONFORMITY

Producent / *Manufacturer:*

ELSAT s.c.

(nazwa producenta / producer's name)

ul. Warszawska 32E/1, 05-500 Piaseczno k/Warszawy

(adres producenta / producer's address)

niniejszym deklaruje, że następujący wyrób:

declare, under our responsibility, that the electrical product:

iNode MCU

(nazwa wyrobu / product's name)

USB

(model / model)

spełnia wymagania następujących norm zharmonizowanych:

to which this declaration relates is in conformity with the following harmonized norm:

Radio Spectrum ISM (Article 3.2 of the RED directive):

PN-ETSI EN 300 328 V2.1.1:2016-11

EMC (Article 3.1.b of the RED directive):

PN-ETSI EN 301 489-1 V2.2.0:2017-03

PN-ETSI EN 301 489-17 V3.2.0:2017-03

Safety (Article 3.1.a of the RED directive):

PN-EN 62368-1:2015-03

Health (Article 3.1.a of the RED directive):

PN-EN 62311:2008

RoHS:

PN-EN IEC 63000:2019-01

jest zgodny z postanowieniami następujących dyrektyw Unii Europejskiej:

is compatible with the following European Union directives:

Dyrektywa RED 2014/53/UE

Dyrektywa EMC 2014/30/UE

Dyrektywa LVD 2014/35/UE

Dyrektywa RoHS 2011/65/UE

Procedura oceny zgodności: wewnętrzna kontrola produkcji zgodnie z załącznikiem II RED

Acceptance procedure: internal production control in accordance with Annex II of the RED Directive

01.03.2019 r.

Piaseczno k/Warszawy

(data i miejscowość / date and place)

Paweł Rzepecki



Współwłaściciel

(podpis i stanowisko / signature and function)



ELSAT s.c. ul. Warszawska 32E/1 05-500 Piaseczno k/Warszawy

tel.: 22 716 43 06 <https://iNode.pl/>