

Programátorská príručka

Tento dokument slúži ako programátorská príručka pre nahranie firmvéru na koncové zariadenie typu LILYGO® TTGO 1/2Pcs LoRa32 V1.0 LoRa 868MHz 915MHz [1]. Príručka obsahuje postup vytvorenie koncového zariadenia s použitím SW knižnice LMIC node.

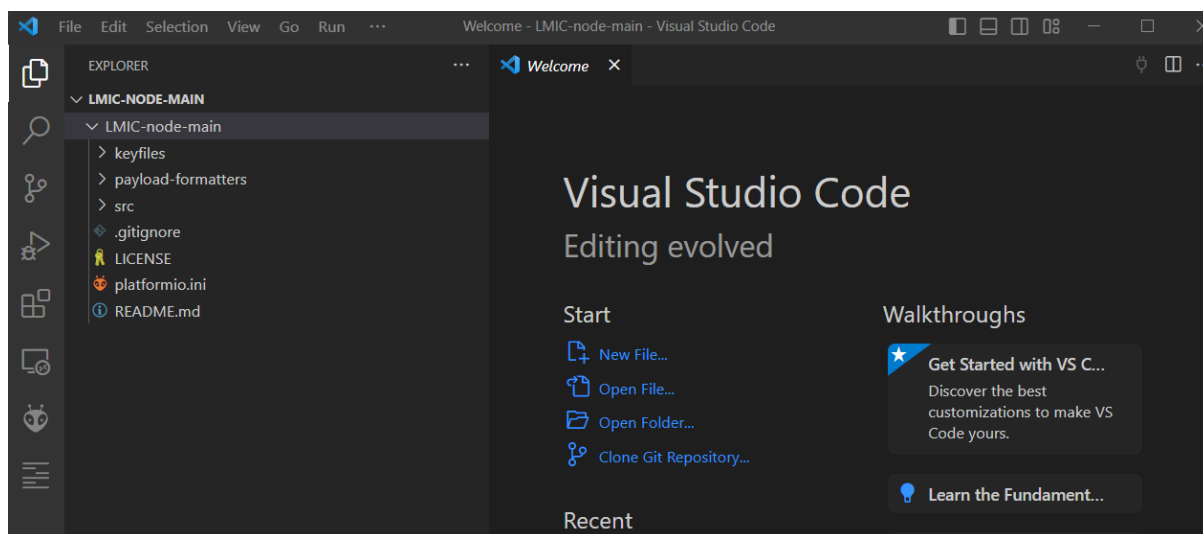
HW koncového zariadenia

Ako už bolo spomenuté v úvode ako HW koncového zariadenia je možné použiť rôzne HW implementácie využívajúce technológiu LoRa na odosielanie dát. My sme sa rozhodli na prototypovanie v rámci nášho projektu zariadenie LILYGO® TTGO vo verzii 1. Toto zariadenie disponuje micro USB konektorom, prostredníctvom ktorého je možné ho pripojiť k PC a nahráť nový firmvér prípadne monitorovať jeho činnosť prostredníctvom sériového monitoru. Zároveň zariadenie disponuje aj vstavaným LCD displejom, ktorý možno využiť na vypisovanie debugovacích informácií o činnosti zariadenia v reálnom čase.

Inštalácia vývojového prostredia

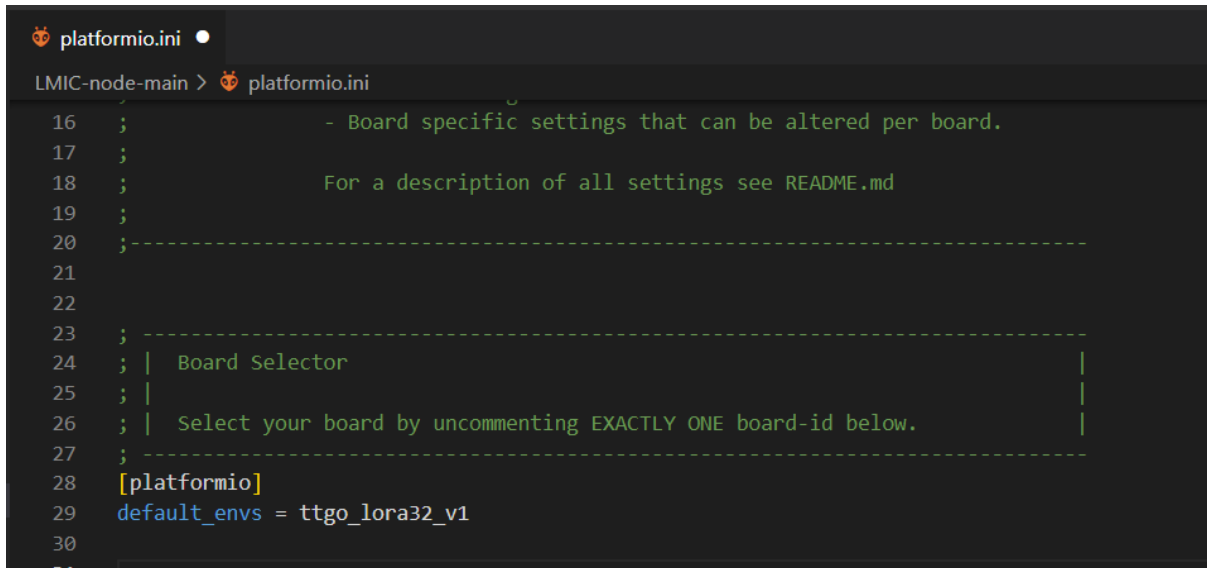
Na úpravy firmvéru [2] koncového zariadenia možno použiť vývojové prostredie Visual Studio Code (VSC) [3], do ktorého ju nutné doinštalovať plugin PlatformIO [4] potrebný pre nahrávanie firmvéru na koncové zariadenie. Tento plugin je možné do-inštalovať vyhľadávači rozšírení nájsť plugin PlatformIO kliknúť inštalovať a po inštalácii je nutné reštartovať VSC. Po opätovnom spustení by mala byť viditeľná ikona pluginu PlatformIO v ľavej lište.

Následne je možné vo VSC po kliknutí na ikonu PlatformIO pluginu vytvoriť nový projekt do ktorého presunieme priečinok so zdrojovým kódom [2]. Po pridaní zdrojových súborov možno upravovať vybrané súbory.



Obrázok č. 1: Pridanie zdrojového kódu knižnice LMIC-node do vytvoreného projektu v rámci pluginu PlatformIO

Následne je nutné špecifikovať v súbore platformio.ini typ použitého HW koncového zariadenia v našom prípade je to ttgo_lora32_v1.

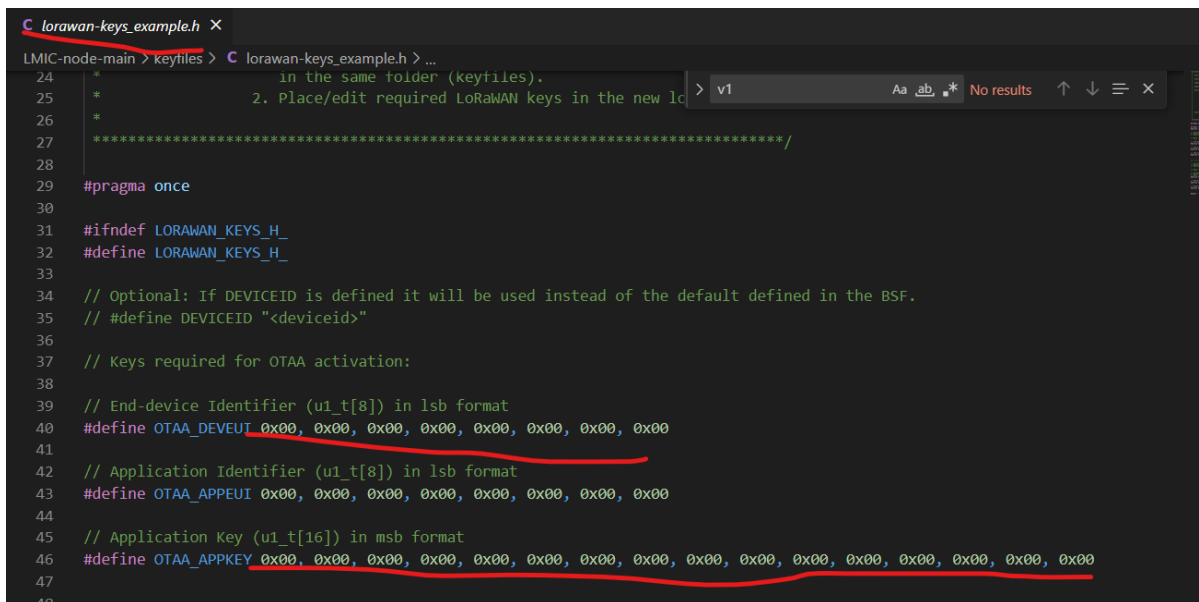


```
platformio.ini
LMIC-node-main > platformio.ini
16 ; - Board specific settings that can be altered per board.
17 ;
18 ; For a description of all settings see README.md
19 ;
20 ;-----
21
22
23 ; -----
24 ; | Board Selector |
25 ; | |
26 ; | Select your board by uncommenting EXACTLY ONE board-id below. |
27 ; |-----|
28 [platformio]
29 default_envs = ttgo_lora32_v1
30
31
```

Obrázok č. 1: Pridanie zdrojového kódu knižnice LMIC-node do vytvoreného projektu v rámci pluginu PlatformIO

V ďalšom kroku je nutné v súbore lorawan-keys_examle.h nastaviť kľúče pre Device a Aplikáciou prostredníctvom ktorých sa jednotlivé sieťové zariadenia autentifikujú voči chirpstacku resp. sieťovému prípadne aplikačnému serveru. Kľúče v súbore lorawan-keys_examle.h sa musia zhodovať s kľúčmi uvedenými v chirpstacku pre vybrané koncové zariadenia. Postup pridania nového koncového zariadenia do chirpstacku resp. nastavenia kľúčov je popísaný v používateľskej príručke chirpstacku.

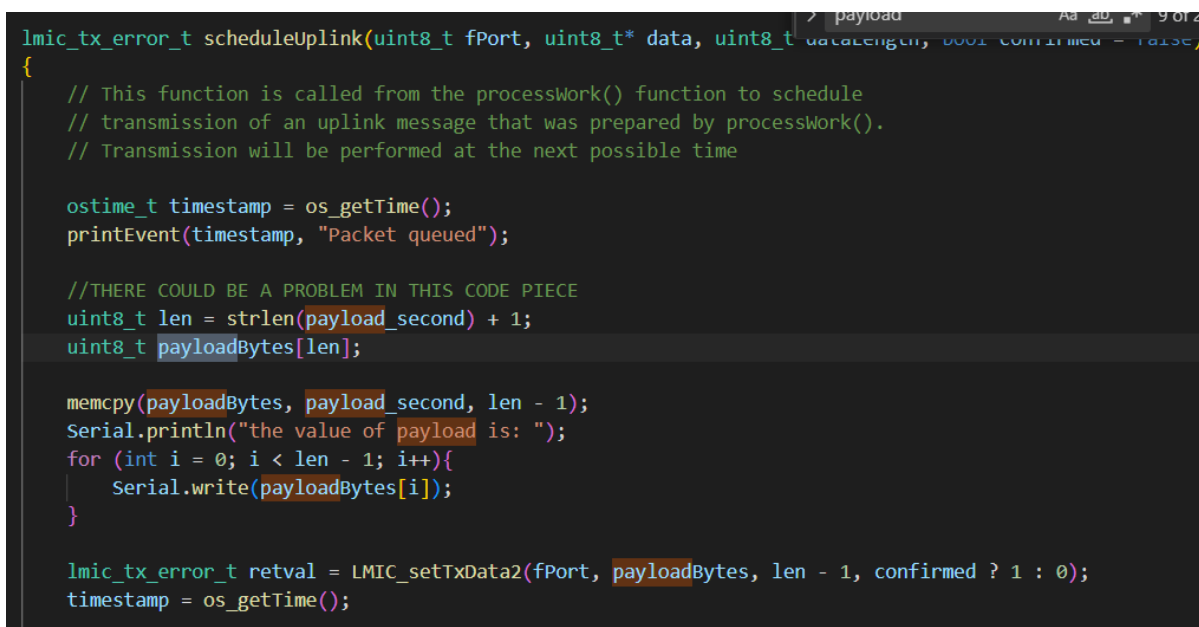
Odporúča sa nastaviť kľúče ktoré boli vytvorené náhodne. Pričom si treba dať pozor na fakt, že v zdrojovom kóde sú kľúče nastavené vo formáte LSB teda Least Significant Bit je zapísaný prvý, zatiaľ čo v chirpstacku sú zapísané v MSB formáte, teda Most Significant Bit je prvý. Rozdiel medzi MSB a LSB je popísaný v tomto článku [5].



```
24 * in the same folder (keyfiles).
25 * 2. Place/edit required LoRaWAN keys in the new folder
26 *
27 *****/
28
29 #pragma once
30
31 #ifndef LORAWAN_KEYS_H_
32 #define LORAWAN_KEYS_H_
33
34 // Optional: If DEVICEID is defined it will be used instead of the default defined in the BSF.
35 // #define DEVICEID "<deviceid>"
36
37 // Keys required for OTAA activation:
38
39 // End-device Identifier (u1_t[8]) in lsb format
40 #define OTAA_DEVEUI 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00
41
42 // Application Identifier (u1_t[8]) in lsb format
43 #define OTAA_APPEUI 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00
44
45 // Application Key (u1_t[16]) in msb format
46 #define OTAA_APPKEY 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00
47
48
```

Obrázok č. 2: Nastavenie aplikačných kľúčov a kľúčov zariadenia vo firmvéri

Následne môžeme upraviť súbor LMIC-node.cpp a nakonfigurovať odosielanie dát zo senzoru resp. nastavenia dátovej časti uplink správ odosielaných z koncového zariadenia v rámci funkcie `scheduleUplink(uint8_t fPort, uint8_t* data, uint8_t dataLength, bool confirmed = false)`



```
lmic_tx_error_t scheduleUplink(uint8_t fPort, uint8_t* data, uint8_t dataLength, bool confirmed = false)
{
    // This function is called from the processWork() function to schedule
    // transmission of an uplink message that was prepared by processWork().
    // Transmission will be performed at the next possible time

    ostime_t timestamp = os_getTime();
    printEvent(timestamp, "Packet queued");

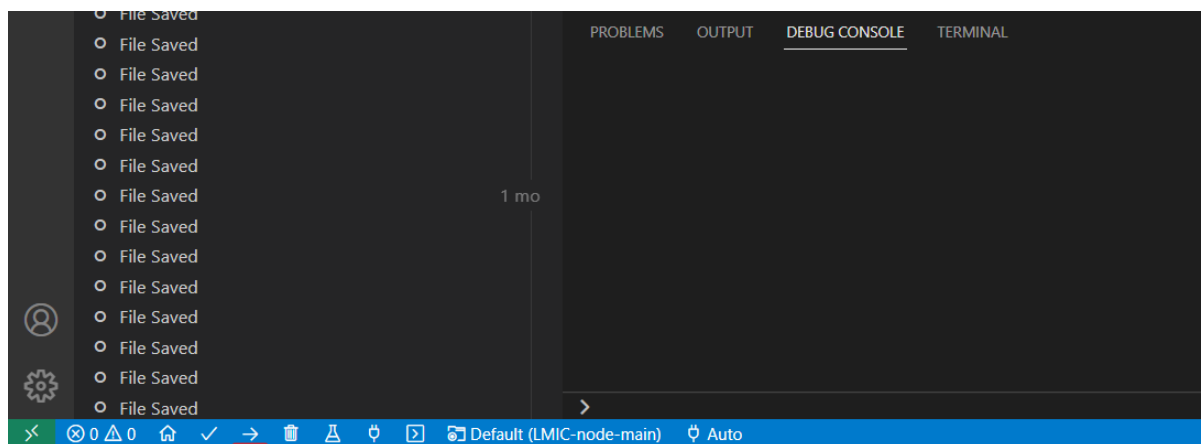
    //THERE COULD BE A PROBLEM IN THIS CODE PIECE
    uint8_t len = strlen(payload_second) + 1;
    uint8_t payloadBytes[len];

    memcpy(payloadBytes, payload_second, len - 1);
    Serial.println("the value of payload is: ");
    for (int i = 0; i < len - 1; i++){
        Serial.write(payloadBytes[i]);
    }

    lmic_tx_error_t retval = LMIC_setTxData2(fPort, payloadBytes, len - 1, confirmed ? 1 : 0);
    timestamp = os_getTime();
}
```

Obrázok č. 3: Konfigurácia dátovej časti odosielanej Uplink správy

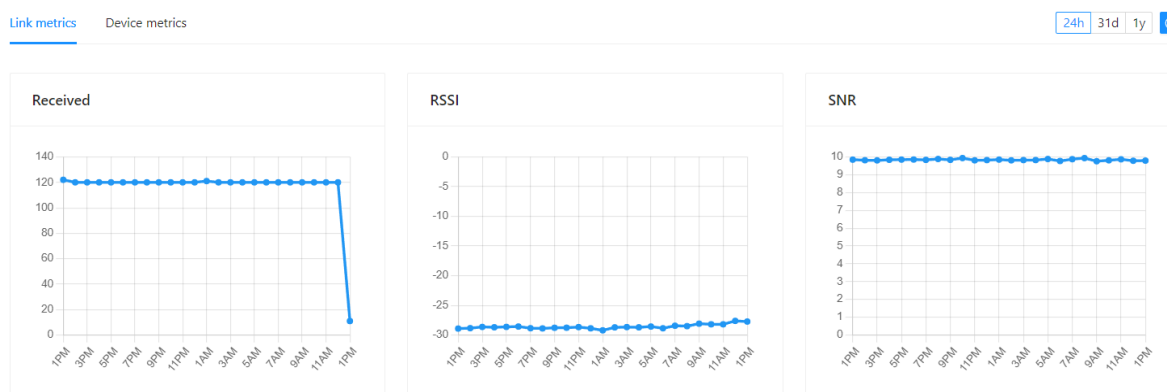
Nakoniec po vykonaní všetkých potrebných úprav v rámci zdrojového kódu je možné nahráť firmvér na koncové zariadenie po jeho pripojení prostredníctvom kábla k micro USB portu na koncovom zariadení kliknutím na tlačidlo Upload v rámci pluginu PlatformIO.



Obrázok č. 4: Postup nahratia upraveného firmvéru do HW koncového zariadenia

Následne možno v terminály sledovať Serial výstup z koncového zariadenia a kontrolovať tak jeho činnosť.

Po úspešnom nahratí zdrojového kódu na zariadenie by malo zariadenie začať proces registrácie do siete a po úspešnej registrácii pristúpiť k odosielaní dátových správ v zmysle nastaveného pracovného intervalu. Základný prednastavený interval v rámci firmvérovej knižnice je 60 sekúnd. Po úspešnej registrácii by malo byť v chirpstacku viditeľná komunikácia z koncového zariadenia.



Obrázok č. 5: Informácie o priebehu komunikácie zo sledovaného koncového zariadenia

Referencie

1. LilyGO. (n.d.). LoRa32 v1.0: LoRa 868MHz/915MHz. Retrieved April 26, 2023, from <https://www.lilygo.cc/products/lora32-v1-0-lora-868mhz-915mhz>
2. lnlp. (n.d.). LMIC-node. GitHub. Retrieved April 26, 2023, from <https://github.com/lnlp/LMIC-node>
3. Microsoft. (n.d.). Visual Studio Code. Retrieved April 26, 2023, from <https://code.visualstudio.com/>
4. PlatformIO. (n.d.). PlatformIO: An open source ecosystem for IoT development. Retrieved April 26, 2023, from <https://platformio.org/>
5. Morningstar Engineering. (2016, December 25). MIDI MSB and LSB. Retrieved April 26, 2023, from <https://www.morningstar.io/post/2016/12/25/midi-msb-and-lsb>