LoRa bütykölés

Baráth Áron
baratharon@caesar.elte.hu

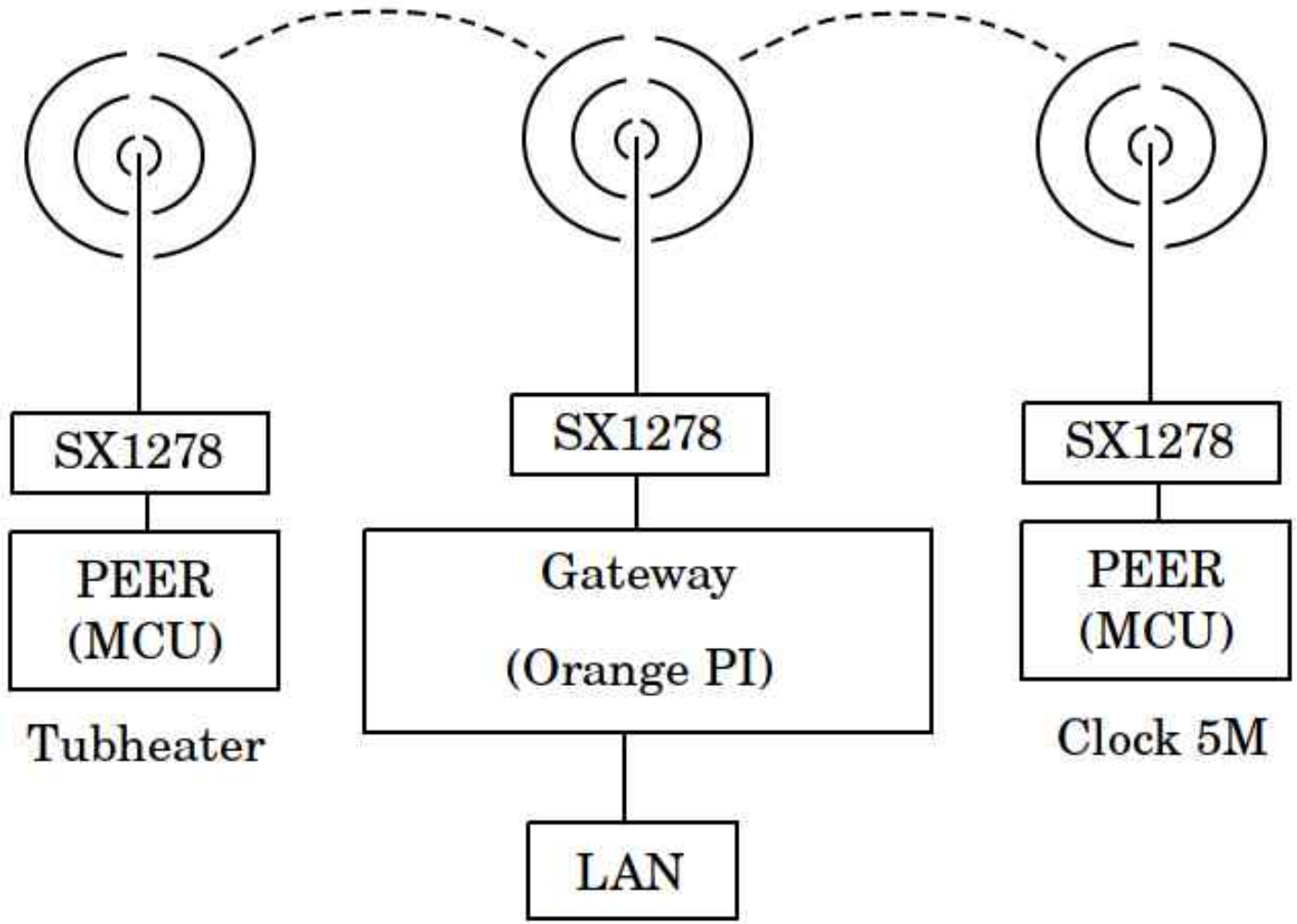
2020. augusztus 15.

Tartalom

Bevezetés

- Szükségem volt egy vezetéknélküli hálózatra mérési és vezérlő adatok továbbítására
- Ezek kis mennyiségű adatok, és ritkán kell közvetíteni őket
- A soros vonal elég lett volna, de új kábel lefektetésére nem volt lehetőség, illetve a kábel túl hosszú lett volna
- Több lehetőséget is megvizsgáltam
- Végül a LoRa-ra esett a választás

A hálózati vázlata



Alternatívák: 3G/4G/NB-IoT

- Magas fogyasztás
  - Főleg 3G/4G esetén
  - Aktív NB-IoT is sok
- Nincs megbízható NB-IoT lefedettség
  - Hónapokon át tartó tesztelés alapján
- SIM kártya
  - „Rejtett” plusz költség
  - „Hetente” változnak a kondíciók

Alternatívák: Bluetooth/BLE

- Elvileg 100-200 métert is képes áthidalni
  - Sajnos a tesztek alapján nem szereti az akadályokat
- Fogyasztás: volt lehetőségem BLE-s MCU-t kipróbálni, és nem bírta 2 hétig
  - 200 mAh
- A megrendelt BLE serial modul nem érkezett meg, így nem tudtam kipróbálni
  - Így utólag nem bánom :-)

WiFi

- Hatókör kétséges
  - Még „jó” eszközzel sem látszódik a hálózat
  - Közelebb nem tudok AP-t telepíteni
- Fogyasztás szintén kétséges
- Az ESP MCU-kal kivitelezhető lenne

Nyers RF

- Ennyire azért nem értek ehhez
- A hatókör valószínűleg nem probléma

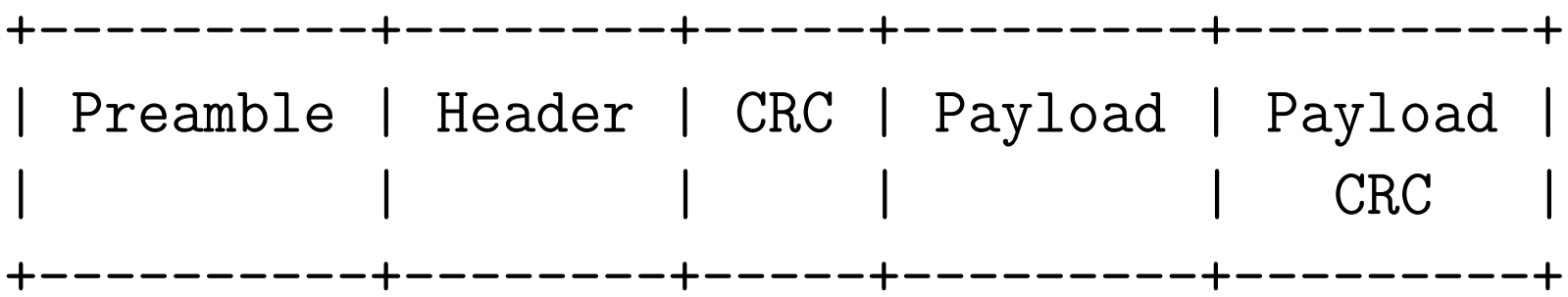
Long Range

- SX1278-as modul (433 MHz)
- Mérések alapján erős jel
  - 100+ méter: több akadály (falak, gépház, épületek)
  - 400+ méter: végül kitakarta egy domb
- Alacsony fogyasztás
  - Adatlap szerint 11-13 mA RX, és 20-28/90 mA TX
  - Egy félig feltöltött 500 mAh-es battery-vel kezdtem, még nem merült le
  - Kiterjedt tesztelés szükséges
- Hátrány: alacsony adatátviteli sebesség
  - De ez nem baj!
- SPI

LoRaWAN

- Globális hálózat
- Számos design probléma
- Külső cég által kezelt hálózat
- Out-of-scope, nekem arra van szükségem, hogy pár száz méterre elküldjek néhány csomagot
- Ezért csak a LoRa PHY-t használjuk
  - LoRaWAN helyett SecPAN

LoRa packet



- Mindig: `Preamble`
  - Default: 12 szimbólum hosszú (max. 65535)
- Explicit mode: `Header` + `CRC`
  - Az explicit módot használom
- Mindig: `Payload` + `Payload CRC`

Modem működési módok

- SLEEP: majdnem kikapcsolt állapot
- STANDBY: az oszcillátor már működik
- *(FSTX)*
- *(FSRX)*
- TX: küldési mód aktív, majd kiküldés után vissza STANDBY-ba
- RXCONTINUOUS: fogadási mód aktív, folyamatos fogadási üzemmód
- RXSINGLE: fogadási mód aktív, egy érvényes csomag fogadása után visszaáll STANDBY-ba
- *(CAD)*

Config

- Frequency: 433 MHz
- Bandwidth: 500 kHz (default: 125 kHz)
- Coding Rate: 4/8 (default: 4/5)
  - Kb: 4 bitet kódol 8 biten
- Spread factor: 1024 (default: 128)
- Sync Word (default: 0x12, LoRaWAN: 0x34)
- Symbol Timeout (default: 100)
- Preamble length: sok mindentől függ (default: 12)
  - Hosszú TX preamble: magas fogyasztás, de könnyebb a szinkronizáció

Interrupt

- Események hatására a modem beállít interrupt bit-eket
  - Van néhány I/O láb, amit limitáltan lehet konfigurálni
  - Tisztán software: poll-ozás
  - Nem mindig éri meg bekötni az összes lábat!
- RxTimeout, RxDone, PayloadCrcError, ValidHeader, TxDone, CadDone, FhssChangeChannel, CadDetected

FIFO

- 256 byte
- RX és TX adatok ide kerülnek
- Speciális SPI regiszter (nincs auto-increment)
- A layout testreszabható, de én kizárólagos módban használom a maximális csomagméret elérése miatt

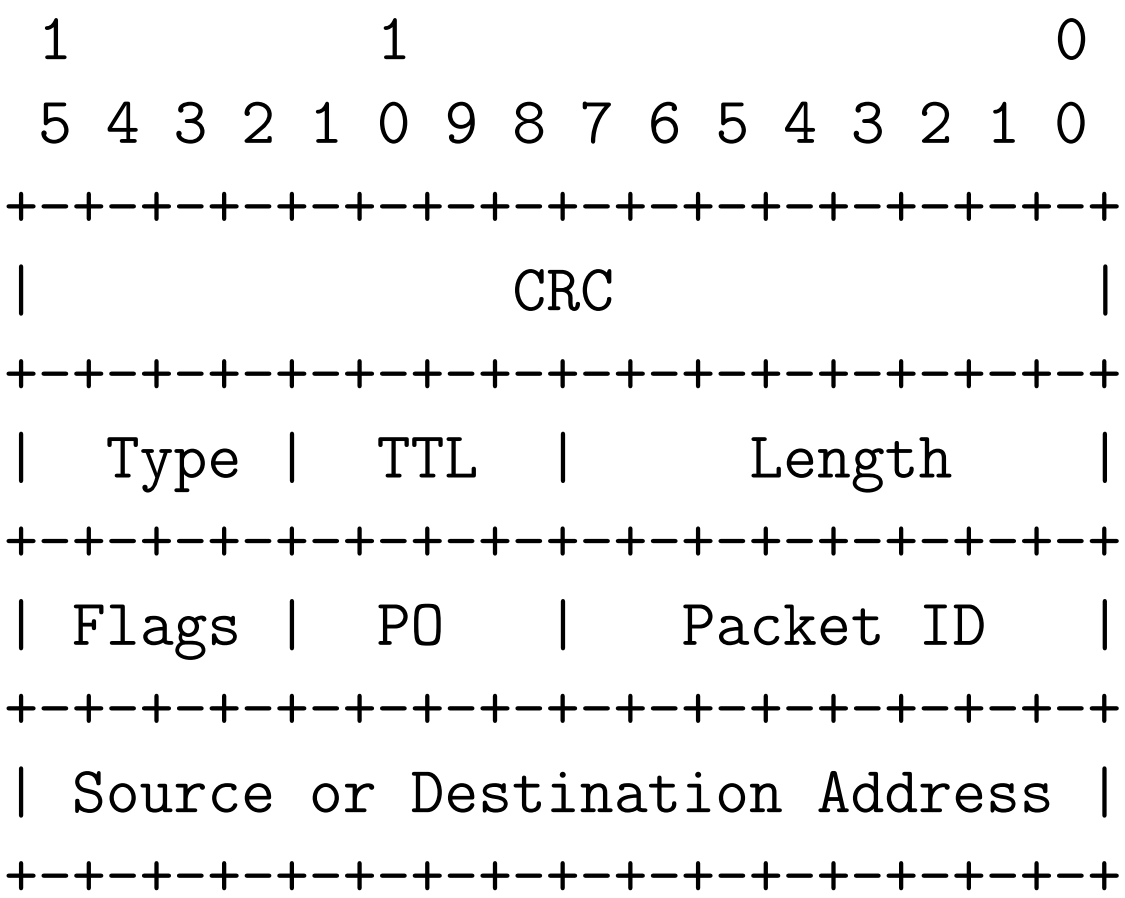
Áttekintés

- Secure Personal Area Network
  - Az eredeti feladathoz mérten biztonságos
- Egyszerűen implementálhatónak szánt protokoll
- Feladata: kellően biztonságos és megbízható adatkapcsolat a végpont és a gateway között
- Titkosított
- Protokoll szinten van újraküldés és ACK
- A végpontok tetszőleges ideig lehetnek tétlenek, akár ki is kapcsolhatnak
  - Célja az energia takarékosság
- Open source
  - A freestd32 projekt keretein belül
  - svn://svn.repo.hu/freestd32/trunk/lib/ext/secpan*

Néhány részlet

- Handshake: RSA
  - PKCS#1 v1.5 digital envelope
  - 1500-2000 bites kulcs: nagyobb nem lehet, mert túllépné a maximális csomag méretet
  - (Protocol version v1.1: EC)
  - Peer oldalon (tipikusan MCU) csak RSA public encrypt
- Adatkapcsolat: AES-128-CTR
  - Időszakos kulcs frissítés

Packet header (1)



Packet header (2)

- `CRC`: a teljes csomag CRC-je a CRC mező nélkül
- `Type`: control (0) vagy adat (1)
- `TTL`: mindig 0 (v1.0 szerint)
- `Length`: a teljes csomag hossza (minimum 8)
- `Flags`: irány (bit 0), a többi reserved
- `PO`: payload offset (általában 0)
- `Packet ID`: ACK-hoz és CTR-hez
- `Address`: mindig a peer címe van itt

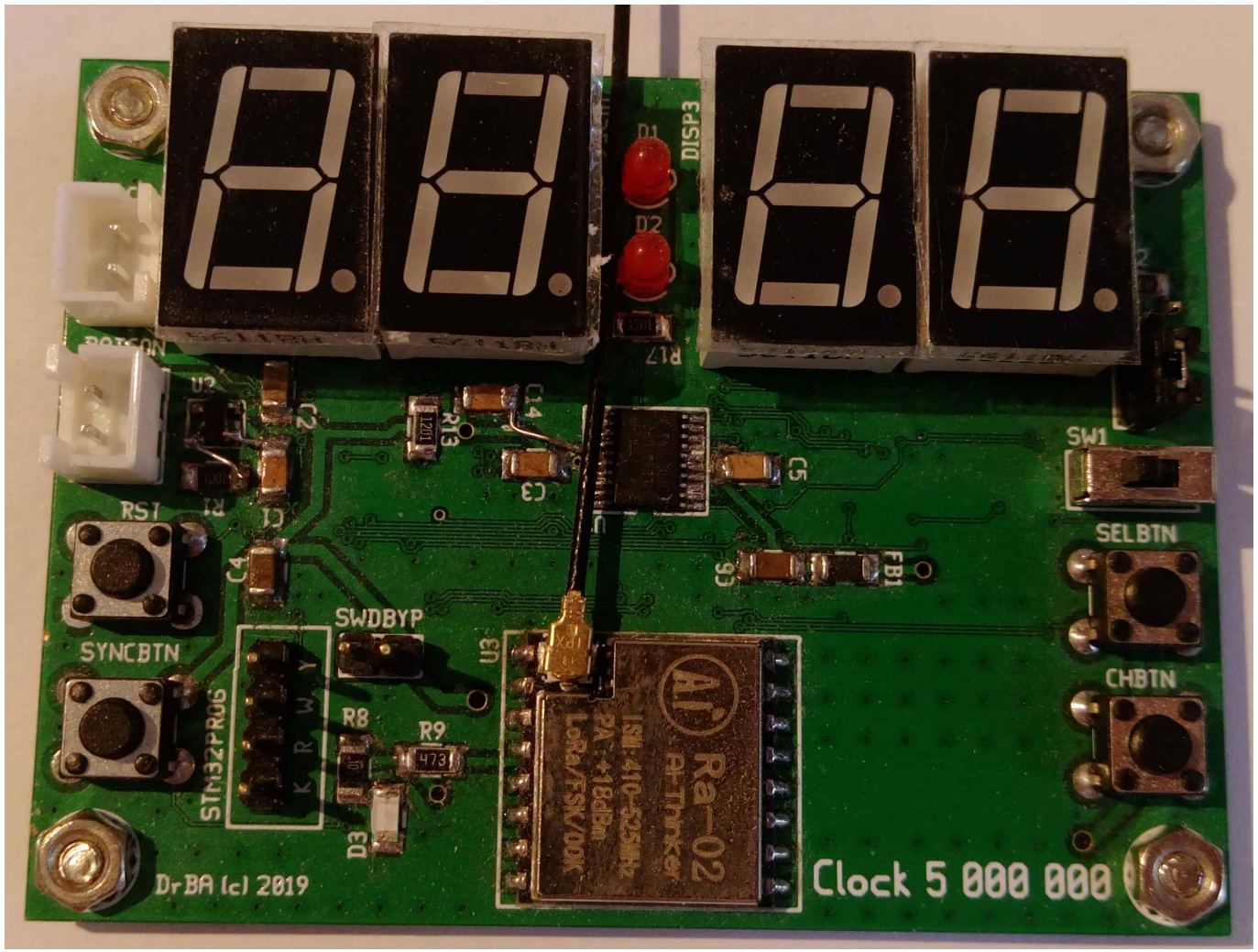
Küldés

1. Választunk egy Packet ID-t, ezzel titkosítjuk (AES-CTR)
   - Ha kifogy a Packet ID, akkor új kulcs generálása
2. Elkészített csomag elküldése, várakozunk az ACK-ra
3. Ha nem érkezik meg az ACK időn belül, akkor újraküldjük
   - Maximum 3 újraküldés, utána sikertelennek tekintjük a küldést
4. Ha megérkezik az ACK, akkor sikeresnek tekintjük a küldést, és foglalkozhatunk a következő csomaggal
   - Tehát maximum 1 függőben lévő csomag lehet (MCU erőforrásai szűkösek)

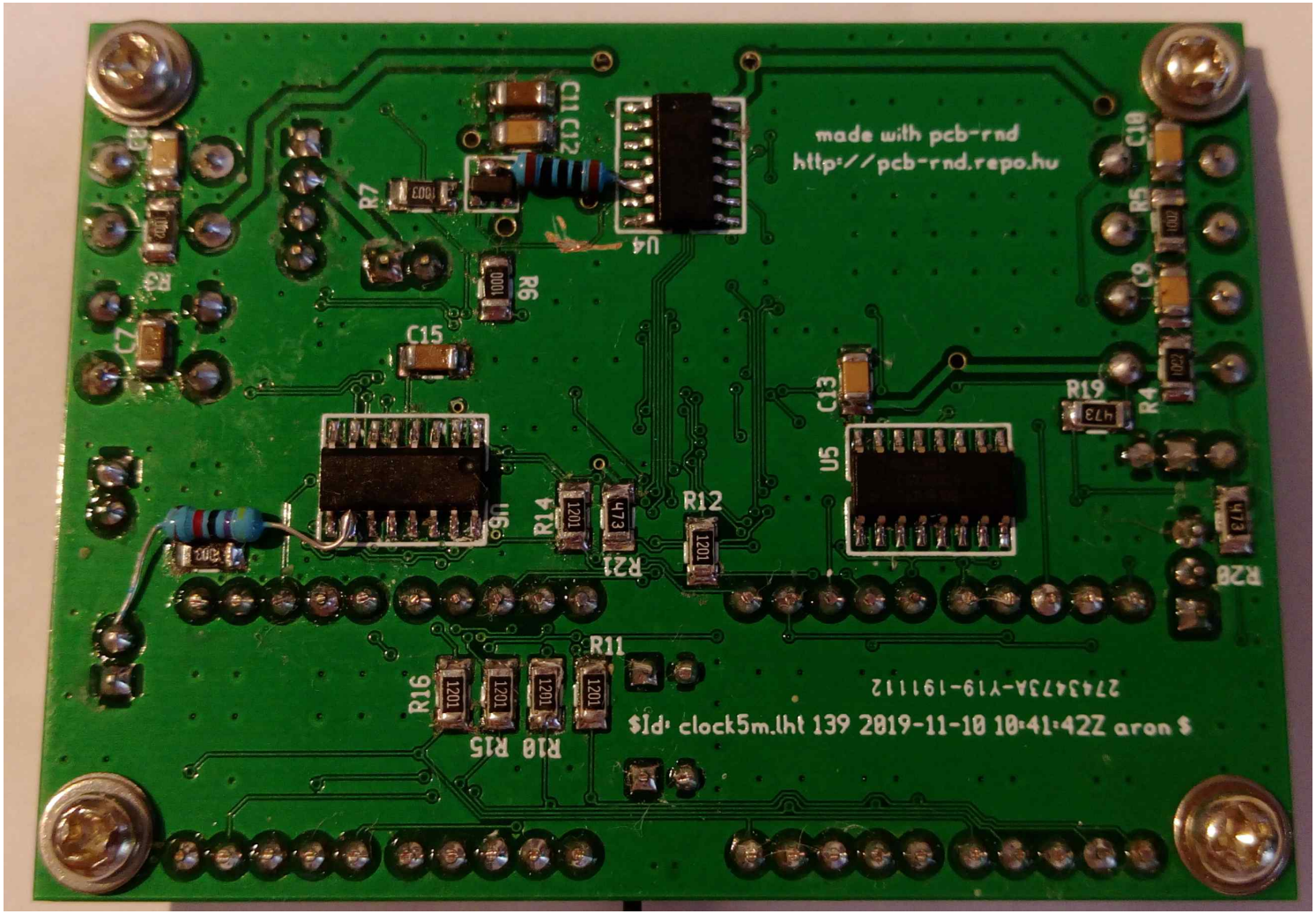
Clock 5 000 000

- Tesztelésre készült prototípus
  - Azért lett óra, hogy valami látványos, de egyszerű funkció legyen rajta
- STM32L031
  - 8 KiB RAM, 32 KiB FLASH
  - Felhasznált: 1388 RAM (16%), 14648 FLASH (45%)
  - Kb. 700 byte stack
- mbed-os-example-lorawan: "Currently the application takes about 15K of static RAM, ... So if you reduce the application stack size, you can barely fit into the 20K platforms"

Clock 5 000 000



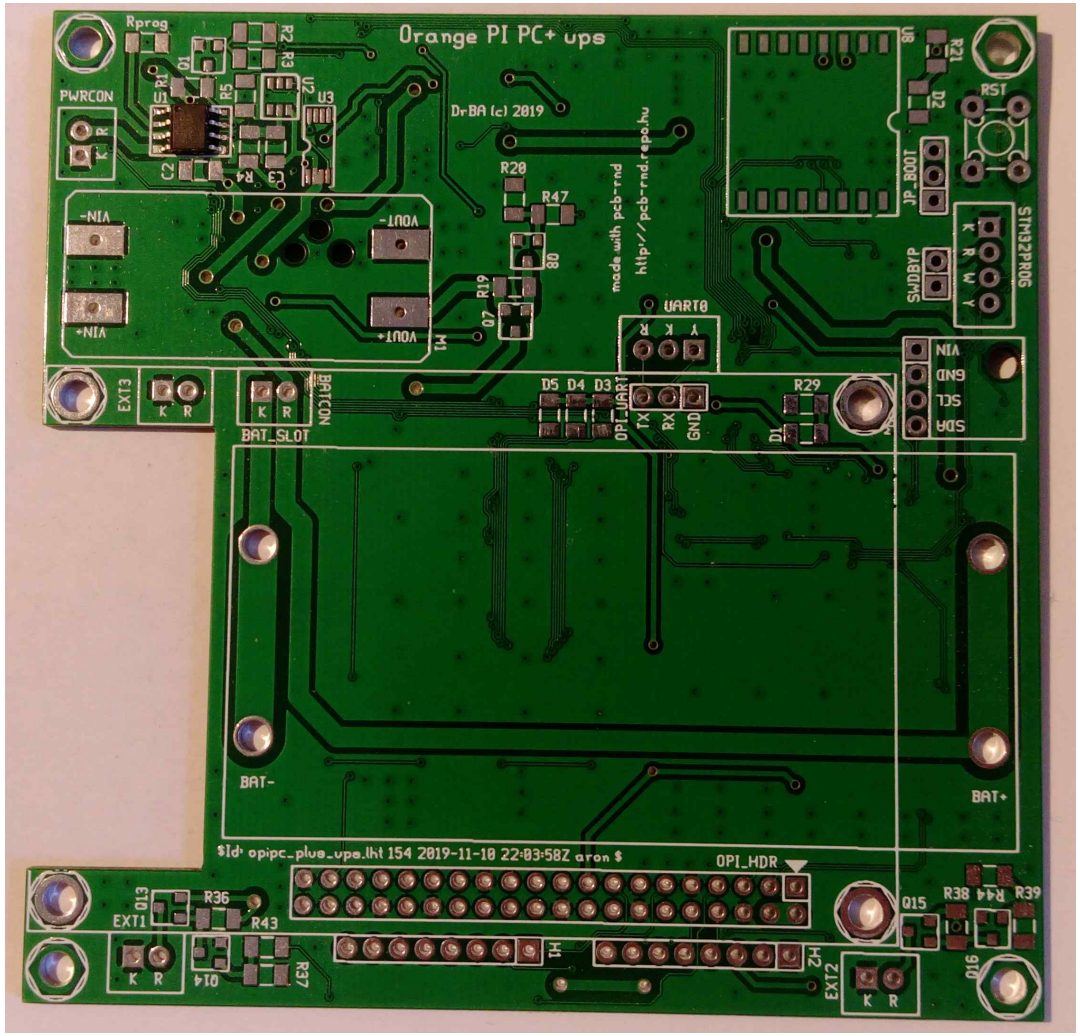
Clock 5 000 000



Gateway

- Orange PI PC+-hoz készült kiegészítő board-ra került
- Jelenleg csak egy "NTP" server
- A felhasználó döntése, akár relay-ezhet is az internet felé
- Ugyanazt a kódot használja (más libc-vel), mint ami az MCU-ra is kerül
  - Minimális glue réteg kell az SPI-hoz

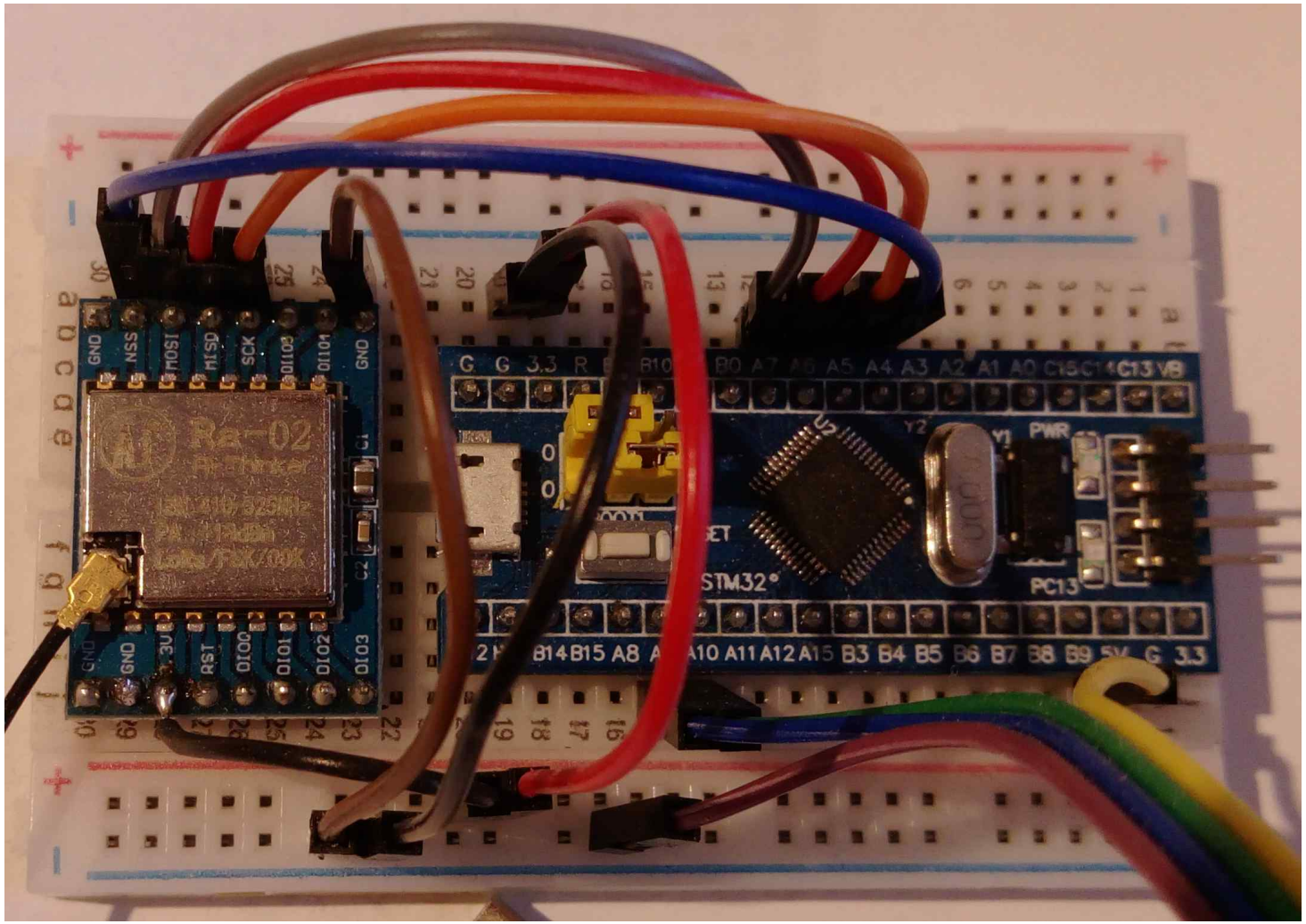
Gateway


Orange PI PC+ ups
DrBA (c) 2019
made with pcb-rnd
http://pcb-rnd.repo.hu
PWRCON
-VIN
+VIN
-VOUT
+VOUT
M1
R20
R47
R19
D7
UART0
SWDBYP
JP_BOOT
RST
STM32PROG
VIN
GND
SCL
SDA
R29
D1
D5 D4 D3
OPI_UART
TX
RX
GND
EXT3
BATCON
BAT_SLOT
BAT-
BAT+
OPI_HDR
H1
H2
EXT1
EXT2
R36
R43
R37
Q13
Q14
Q15
Q16
R38
R44
R39

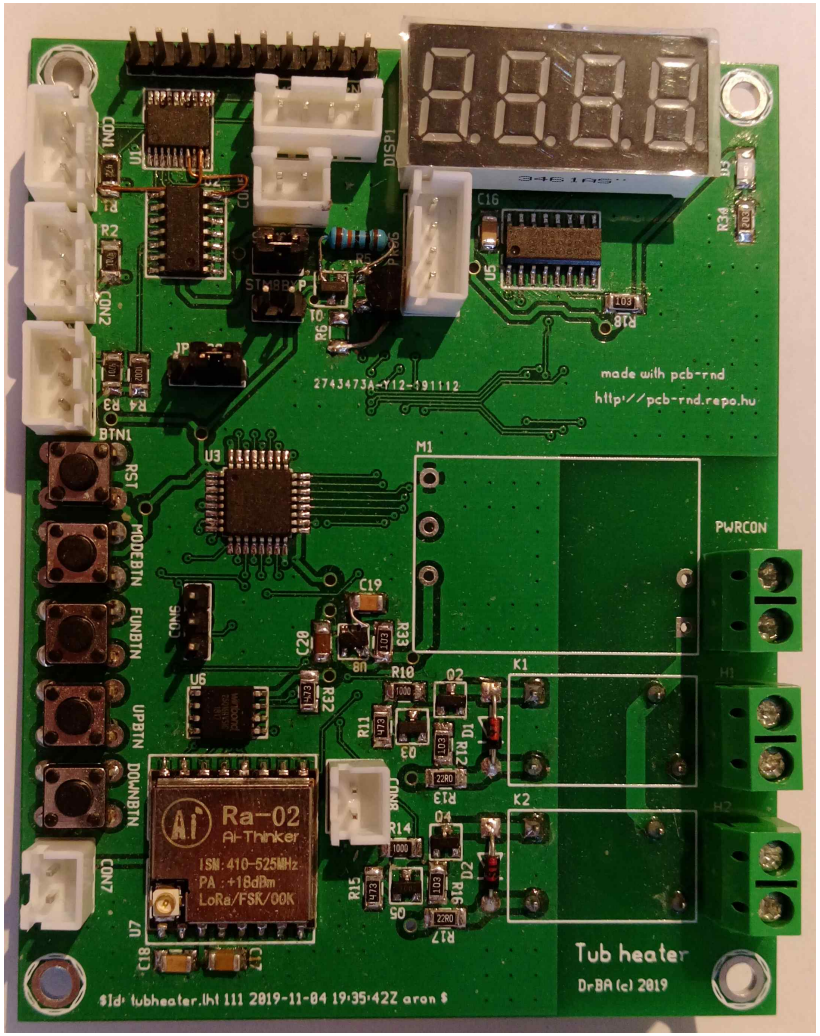
LoRa Sniffer

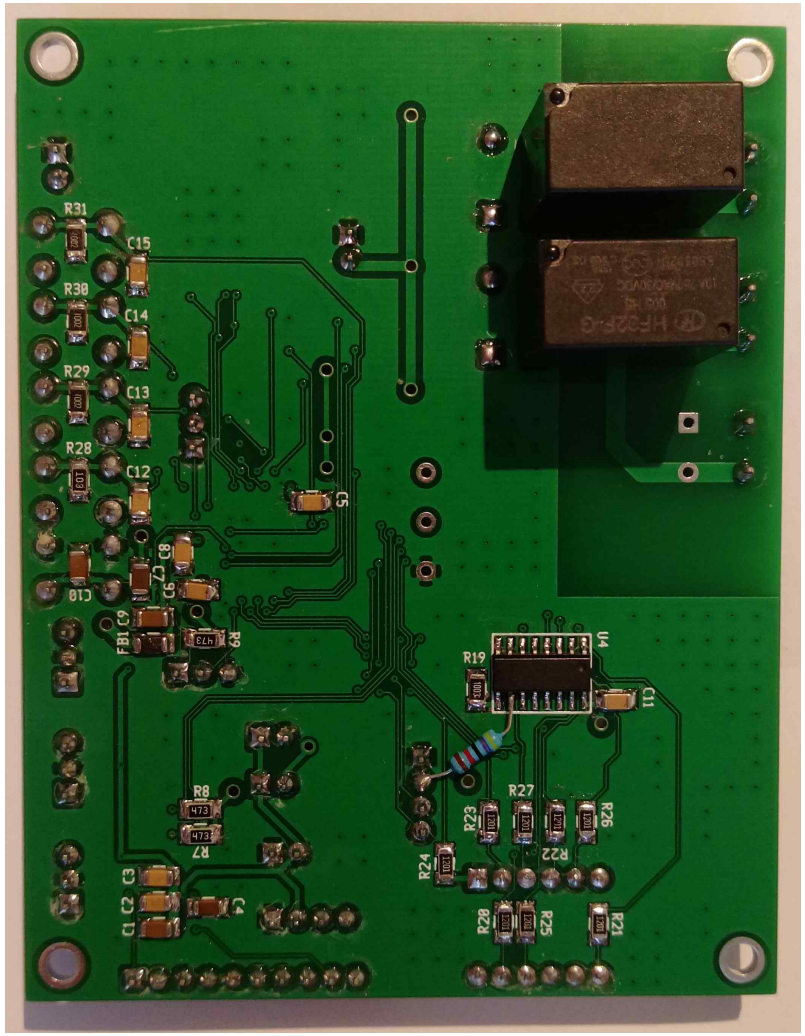
- Debug-olási célból jött létre
- Minden fogadott csomagot hexdump-ol az UART-ra
  - RXCONTINUOUS mód
- STM32F103
  - 20 KiB RAM, 64 KiB FLASH
  - Felhasznált: 636 RAM (3%), 6188 FLASH (9%)

LoRa Sniffer



Tubheater


made with pcb-rnd
http://pcb-rnd.repo.hu
Ra-02
AI-Thinker
ISM: 410-525MHz
PA: +18dBm
LoRa/FSK/OOK
Tub heater
DrBA (c) 2019



Welltype

- Szükségem lenne script-elhetőségre, hogy ne csak az előre meghatározott funkciókat tudja végrehajtani a végpont
- MCU-n szöveges script nem lehet, mert a parse-olása rengeteg erőforrást használna el fölöslegesen
- Compiled script-ek kellenek: kisebbek és a végrehajtásuk biztonságos tud lenni
- Open source: svn://svn.repo.hu/welltype

Welltype

- A meglévő virtuális gép refaktorálása után a Welltype egy jó megoldás
  - Utasításkészlet újratervezése
  - Kisebb memória-igény
  - Új üzemmód („asztali” és „embedded”)
  - Kisebb overhead
  - Opcionális SW-MMU (külső SPI flash-ből futtatás)

Köszönöm a figyelmet!

(baratharon@caesar.elte.hu)