MCU, LoRa, LoRaWAN and Data Processing

www.dupedup.cz

Petr Dušek, 3/2020 – revision 11/2021

Here you find the summary paper related to Internet of Things devices using Microcontroller Units and radio networks LPWAN in the ISM radio bands.

I design electronic devices using embedded small processor modules for new product development. The main part of modules is the MCU (8 or 32b Microcontroller Unit) provided by Atmel-Microchip, STMicroelectronics or others. The **MCU** contains processor AVR or ARM family (RISC architecture) and also on the same chip the operation SRAM and FLASH memory for custom firmware. Some other electronic function (WiFi, Bluetooth, senzors, etc.) may be integrated on the same chip and such embedded system is called **SoC** (System-on-Chip). The MCU or SoC are assembled with power and other auxiliary circuits and all is soldered on a small PCB (Printed Circuid Board) as the module. The frequent families of MCU are **AVR (Arduino), ESP8266 or ESP32 (Espressif Systems) and particulary ARM STM32 (STMicroelectronics**).

The I/O pins of the MCU or SoC modules are simply accessible and can be connected to senzors or other devices via UART, I2C or SPI data buses. After device devolopping the MCU can be used naturally without the module and can be soldered only with needed components on a custom designed PCB. The MCU/SoC have low consumption of power and can be powered from small Li-Ion, Li-Po acumulators or solars cells. MCUs or SoC can control local processes, display measured data on OLED, LCD, TFT etc. or transmit data via data networks to data servers for next processing.

Frequently is used air data connection LPWAN (**LoRaWAN** or SIGFOX) via ISM radio band (433/868 MHz), WiFi or GSM networks in my projects. In the suche case are the MCU modules connected via data buses (SPI, UART) to small radio modules. Using the LoRa radio has many advantages over the WiFi or GSM networks, particularly in low power consumption. The radio modules LoRa use mostly the radio chip by Semtech. The modules included auxiliary circuits and IPEX or SMA connector for 433/866 MHz antenna. Power is mostly 3,3V. Some modules included together the MCU (mostly ESP32 or ARM STM32) and the radio chip. The best solution is PSoC (Programmable System-on-Chip) where the ARM MCU and the radio are integrated on one chip (e.g. by Cypress)

The topology network is either end node (transmitter) connected via air to the node (receiver) within local network or some end nodes connected via air to a gateway controled by the network server (**LoRaWAN**). These ways can be output data from MCUs transmitted via air to data servers to control other processes (e.g. via MQTT broker) or can be stored and displayed via web interfaces. I use leased servers in cloudes or some cloude services. May be also used servers created within local networks e.g. lightweight servers created on Raspberry Pi.

The **firmware** I develop in the Arduino IDE or MS Visual Studio using the Wiring or C++ development platform. The support (linker, compiler etc.) for proper type processor of MCU/SoC can be added to IDE. The compiled firmware is uploaded to flash memory of the MCU via Serial-USB converter (e.g. FTDI) or ST-Link in case of STM32. The firmware can be uploaded also via network (OTA). I develop own firmware and use appropriate open software libraries from GitHub. Tools STM32CubeIDE/CubeMX or ARM Keil can be used for developping, debugging and to compile the firmware of STM32 MCU family. Web user interfaces I develop in PHP and JavaScript. May be also used Python as the programming language.

The MCU or LoRa devices are capable to observe any physical data where exist appropriate data senzors and whre is radios signal to send data to proccesing. I use these devices e.g. for UAV (Unmanned Aerial Vehicle) in the event of interruption of the control signal. At the moment is activated IoT device and via LoRa/LoRaWAN is received telemetry data from UAV on a Ground Station in particular the actual GPS, board voltage or temperature.

1. LoRa

Pro datové radiové spojení bod-bod je použita modulace **LoRa** (Long Range) v bezlicenčním pásmu (pásmo ISM - Industrial, Scientific, Medical) na 433 MHz nebo 868 MHz. Jsou vždy použity minimálně dva LoRa moduly, jeden primárně v módu vysílače (TX) a druhý přijímače (RX). Parametry radiového spojení, zejména datová rychlost, závisí na nastavení parametru **Spreading Factor** (SF) LoRa. Parametr SF je v rozsahu hodnot 7-12. Čím větší je parametr SF, tím větší je dosah (až desítky km), nižší dosažitelná datová rychost (při SF 12 je kolem 300 bps, při SF=7 je až 5 kbps), větší Time On Air a tím spotřeba radiomodulu a menší maximální payload (při SF=12 je 51 B, při SF=7 je 220 B). Šířka pásma je standardně 125 kHz, ale je nastavitelná od 62,5 do 500 kHz a též významně ovlivňuje zejména datovou rychlost.

V ČR/EU je **omezen výkon vysílání** EIRP v ISM na 10 až 25 mW (14 dBm) a využití pásma (Duty Cycle) na 1 %, což odpovídá radiovému provozu 36 s za hodinu. Tomu je nutno uzpůsobit intervaly mezi vysíláním a objemy přenášených dat při programování firmware a SF při nastavení radiového mudulu.

K pinům MCU jsou přes přes sběrnice UART (Rx, Tx), I2C (SDA, SCL) či SPI (MISO, MOSI, SCLK, SS) připojeny **datové snímače** nebo akční moduly (motory, spínače). Pro snímání teploty používám např. snímač DS18B20 (Arduino knihovna Wire a DallasTemperature), DHT11/22 (Arduino knihovna DHT) nebo BMP280 (knihovna Adafruit_BMP280_Library) pro teplotu a tlak. Pro snímání souřadnic pomocí GPS lze použít např. moduly V-KEL, Beitian BN280/880 či u-blox NEO (Arduino knihovna „TinyGPS++“), pro snímání polohy např. MPU6050.

Napájení MCU a komunikačních modulů je z 1S-3S Li-Po/Li-Ion akumulátorů (1S = 4,2 V) s využitím LDO regulátorů či Step UP / Down měniči pro 3.3 V nebo 5 V. Napájení je vhodné dimenzovat alespoň pro 500 mA. Pro komunikační moduly v modu příjmu, kdy je odběr nižší, lze možno používat napájení z regulátorů 3.3/5 V desek MCU (max. do 100 – 150 mA), které jsou napájeny přes USB nebo na RAW napájecí piny.

Při použití zařízení v praktickém provozu, je nutné snížit spotřebu modulů v době mezi měřením ze senzorů. MPU a komunikační moduly je proto nutno mimo dobu provozu či měření přepínat do módů „sleep“ nebo použít časované napájení (např. TPL5110 Power Timer Breakout).

AKU články je vhodné dobíjet z alespoň 2 W (6 V/330 mA) fotovoltaických článků. Možné je použit nabíjecí moduly (např. TP4056). Některé desky mají nabíjecí modul integrován npř. TTGO, HELTEC. S využitím dobíjení ze solárního článku a při použití sleep modů je možné zařízení provozovat nepřetržitě po dlouhou dobu (rok a více) bez výměny akumulátoru.

LoRa Point-to-Point Devices :

- **LoRa-E32-TTL100** https://www.ebyte.com - obsahuje čip SX1278, napájení je 3.3-5V, pro připojení k MCU se používá **UART**. Modul má nastavitelný výkon 20, 17, 14 a 10 dBm, 32 radiových kanálů v rozsahu 410-441MHz a datovou air rychlost od 0.3 do 19.2 kbps (defaultně 2.4 kbps). To odpovída změně SF 12-7 a šířky pásma 125-500 kHz. UART rychlost lze nastavit od 1.2 do 115.2 kbps. Pomocí pinů M0 a M1 lze nastavit mody Normal, Wake on Radio, Sleep nebo režim Settings. Lze nastavit topologii broadcast nebo adresování příjemce (mód transparent/fixed). Odběr vysílání/příjem/standbay je 120/14/0.01 mA. Pro kód je možné použít i knihovny např. „E32_TTL“ od Renzo Mischianti (github.com/xreef/LoRa_E32_Series_Library) nebo knihovna z github.com/Bob0505/E32-TTL-100.

- **LoRa RA-02** http://www.ai-thinker.com – obsahuje čip SX1278, 433MHz, napájení je 3.3V, pro připojení k MCU se používá **SPI. N**astavitelné je SF 7-12 a šířka pásma (62.5,125,250 kHz). Výkon je pevný 17 dBm. Odběr vysílání/příjem/standby je 90/12/1 mA. Použita nativní Arduino knihovna SPI a pro LoRa Arduino knihovna (github.com/sandeepmistry/arduino-LoRa).

- **TTGO ESP32 ver.1.0** http://www.lilygo.cn - deska obsahuje ESP32 (MCU Tensilica LX6 dual-core processor, 240MHz, 520 KB SRAM, 8MB FLASH) s podporou WiFi a Bluetooth a radiový čip LoRa SX1276 (868MHz, 148 dBm high sensitivity, 20 dBm max. power output), lithium battery charging circuit a CP2102. Napájení modulu je přes pin BATT z 3,5 – 7V nebo přes USB (do USB lze přímo zapojit dobíjení z fotočlánku). Programování je přes Arduino IDE s doplněnou podporou desek ESP32. Pro LoRa Arduino knihovna (github.com/sandeepmistry/arduino-LoRa).

2. LoRaWAN

Pro datové radiové spojení se sítí LoRaWAN v pásmu 868 MHz byly použity moduly (koncová zařízení v síti) obsahující na jedné desce MCU i radiový čip LoRa. Jako MCU je použit ESP32 nebo STM32 a pod. Programový kód pro MCU koncového zařízení LoRaWAN musí obsahovat procedury pro obsluhu LoRaWAN (join, re-join, atd).

Koncové zařízení připojují přes bránu LoRaWAN Gateway, která je dále spojena se síťovým řídícím serverem. Je možno si pořídit vlastní LoRaWAN Gateway od levných dvou kanálových pro domácí použití až po osmikanálové pro tisíce koncových nodů a vytvořit si vlastní síť a řídící server, nicméně vhodnější je použít komerční LoRaWAN síť.

Koncové zařízení se většinou připojuje do LoRaWAN pomocí **OTAA** (Over-the-Air Activation). To vyžaduje parametry **Device Ientifikátoru (DevEUI), Device application EUI (AppEUI) a šifrovací aplikační klíč (AppKey).** Tyto parametry musí splňovat jistá pravidla a je možno je vygenerovat např. na portálu LoRaWAN sítě TTN https://thethingsnetwork.org. Musí se párově použít v zdrojovém kódu pro nastavení koncového zařízení a pro nastavení LoRaWAN Gateway resp. LoRaWAN síťového serveru.

Používám komerční síť LoRaWAN provozovanou Českými Radiokomunikacemi, která je v ČR poměrně dobře dosažitelná. Po registraci na ČRA LoRaWAN https://sso.cra.cz lze v portále nakonfigurovat parametry svých LoRaWAN koncových zařízení – pro OTAA doplnit Device EUI, Application EU a AppKey pro každé koncové zařízení. Je možný provoz zdarma při počtu max. 360 zpráv denně a malém počtu zařízení.

Data z koncového zařízení po prvotním OTAA přijmou nejbližší LoRaWAN brány v okolí. Pro uplink (směr dat node -> gateway) je používáno až 9 kanálů v pásmu 868 MHz, šířka pásma kanálu je standardně 125 kHz. Datová rychlost závisí na parametru SF (viz. popis LoRa výše). Protokol umožňuje adaptivně přizpůsobovat parametr SF podmínkám přenosu.

Nasnímaná data před převádím firmwarem v MCU do 10 - 12B formátu a odesílám je jednou za 5 min do sítě LoRaWAN. Při SF7 a délce zprávy 12 B (payload) + 13 B (LoRa header) je TimeOnAir 61 ms, což odpovídá datové rychlosti min. 3,5 kbps. V ČR je omezeno využití pásma (Duty Cycle) na 1% denně (radiový provoz 36 s za hodinu). Při TimeOnAir 61 ms a četnosti 12 zpráv za hodinu to zcela vyhovuje.

LoRaWAN Devices :

- **Heltec Cube Cell HTCC-AB01** https://docs.heltec.cn https://heltec-automation-docs.readthedocs.io/en/latest/ - obsahuje integrovaný čip ASR6501 (radio LoRa SX1262 a ARM CortexM0+ 48 MHz, 16 KB RAM, 128 KB FLASH), IPEX pro připojení antény a IO piny. Programování se provádí přes miniUSB - deska obsahuje Serialto USB čip CP2102. Napájení je z konektorů pro BATT, dobíjení fotovoltaického článku na pin VS a napájení senzorů z pinu Vext 3,3V. Softwarová podpora od Heltec obsahuje potřebné knihovny i příklady kódu. Modul lze ovádat i přes AT příkazy. Lze použít IDE pro Adruino po doplnění podpory desek ASR650x - https://github.com/HelTecAutomation/ASR650x-Arduino.
  Dle mne se jedná o nejvhodnější a levné (deska lze získat do 300 Kč) řešení pro LoRaWAN zařízení.

- **Heltec LoRa Node 151** – deska obsahuje MCU STM32 L151CCU6 (ARM Cortex M3 48 MHz, 32 KB RAM, 8 KB EEPROM, 256 KB FLASH), čip LoRa SX1276, IO piny, SMA přípoj na anténu a držák pro ½ AA AKU. Programování lze provádět v STMCubeIDE a firmware nahrávat pomocí STM Cube Programmer s použitím USB DFU nebo ST-LINK. V omezené míře lze použít IDE pro Adruino s doplněnou podporou desek STM32 (STMduino) https://github.com/stm32duino/Arduino_Core_STM32. Příklady kódu lze získat z Heltec po registraci ID čipu. Výhodou STM32 je výrazně nízká spotřeba v sleep modech, jistou nevýhodou omezená podpora z IDE Arduino a tedy nutnost používat vývojové nástroje pro STM32.

- **Heltec ESP32 LoRa** - obdobně jako TTGO obsahují ESP32 (MCU Tensilica LX6 dual-core processor, 240MHz, 520 KB SRAM, 8MB FLASH) s podporou WiFi i Bluetooth a čip LoRa Semtech SX1278. Obsahují OLED display. Programování je přes Arduino IDE s doplněnou podporou desek ESP32.

- **Arduino MKRWAN 1300** https://store.arduino.cc/arduino-mkr-wan-1300-lora-connectivity-1414 - deska obsahuje Arduino Zero (MCU Atmel SAMD21 ARM Cortex M0+, 48 MHz, 32 KB SRAM, 256 KB FLASH) a radiový čip LoRa Murata CMWX1ZZABZ na 868 MHz. Deska je napájena přes konektory BATT 3,5 – 6V nebo USB. Pro napájení snímačů lze použít Vout 3.3V z desky. Do IDE pro Arduino je potřeba přidat podporu desky s CPU SAMD21. Implemetace protokolu LoRaWAN je v knihovně MKRWAN.

- **TTGO ESP32 ver.1.0** http://www.lilygo.cn - Pro LoRaWAN je použita Arduino knihovna „MCCI LoRaWAN LMIC library“.

3. LPWAN SIGFOX

Zařízení tvoří modul **SIGFOX WiSOL SFM10R1** na 868 MHz, který je připojen přes rozhrní UART k MCU např. Arduino Nano či Mini. K MCU je připojen snímač.

Data jsou před odesláním pomocí firmware v MCU převedena do 6 -10 B (**max. payload je v Sigfox omezen na 12B**) a následně zasílána jednou za 15 min do sítě SIGFOX (868 MHz) přes nejbližší radiové SIGFOX gateway v okolí. Datová rychlost je max. 100 bps. Nevýhodou sítě Sigfox je omezení v počtu a velikosti odesílaných zpráv.

4. Data Processing

LoRa

Data z vysílacích LoRa/MCU modulů (operační data nebo data ze senzorů) přijímám přijímacím modulem LoRa. Ten je připojen (většinou přes SPI) k MCU typu ESP8266/ESP32 (např. Wemos Mini), který je přímo připojen do interní WiFi nebo pomocí ETH modulu a kabelu do Ethernet switche. Při zpracování se k přijmutím datům z LoRa přidají servisní data získaná z přijímacího radiového modulu (SNR, RSSI, délka paketu) a následně jsou všechna data zformátována do JSON nebo TXT věty. Pak jsou data pomocí HTTP POST zasílány na portál internetového webového datového serveru nebo na cloudovou službu. Další zpracování dat je stejné jako u LoRaWAN.

LoRaWAN

V portálu LoRaWAN i Sigfox lze data ukládat do DB nebo nastavit **callback** (v LoRaWAN ČRA se nazýva endpoint), který se aktivuje vždy při přijmutí datového paketu z koncového zařízení. Pomocí callback se data z portálu přepošlou ve formátu JSON v HTTP POST requestu na uživatelské URL podporující HTTP REST API. K datům z klientských zařízení se do JSON přidávají i metadata LoRaWAN. U sítě SIGFOX jsou volitelně hodnoty RSSI, SNR, seqNumber. U LoRaWAN ČRA jsou do JSON metadata přidávána automaticky (viz popis dat dále).

V mém způsobu zpracování data ze zařízení neukládám v LoRaWAN, ale přes callback je přeposílám na URL HTTP serveru v cloudu. Na volaném URL aktivuji např. PHP kód, kterým se z HTTP POST requestu přijmou JSON data. JSON se „rozbalí“ na jednotlivé datové segmenty, data se zkontrolují, převedou z úsporného bitového formátu, upraví se a jako text se zapisují do datového souboru ve formátu CSV na datový server. Při větším počtu dat je vhodné místo CSV/TXT použít SQL DB. Datový soubor obsahuje v každém řádku zaslaný datový paket – zejména datum, čas odeslání dat, metadata z LoRaWAN a samozřejmě vlastní data např. ze snímačů (teplota, tlak, atd.). První řádek v CSV je popis dat. Soubor nebo tabulky DB jsou plněné daty a ukládány na datastore webového nebo datového serveru. Použitím logiky Javascript Query API jsou části souborů načítány klientem z browseru. V kódu Javascriptu z načtených dat většinou vytvářím pole dat, jehož položky pak mohou být zobrazeny v prohlížeči např. ve formě textu a tabulky. Vytvořené datové 2D pole používám též jako datový objekt pro API Google Charts (https://developers.google.com/chart/). Nasnímaná data lze tak graficky, např v line grafu, zobrazovat v prohlížeči. V API pro Google Charts lze nastavit různé typy grafů, přizpůsobit nastavení os a mnoho dalších nastavení grafu. Použití API je dobře dokumnetované.

Pokud je **senzorem GPS** jsou hodnoty GPS (LAT/LON/ALT) kódovány do 9 B formátu a přes LoRaWAN analogicky způsobem popsaným výše zapisovány do souboru na http serveru. Z datového souboru jsou pak záznamy čteny klientem pomocí Javascript, zpětně rozkódovány na GPS pozice a zobrazeny v prohlížeči. Možné je použití API Google Chart pro grafické zobrazování lokací GPS na mapě nebo export GPS dat do KML souborů.

Lze použít i jiné způsoby zpracování dat z LoRaWAN. Testoval jsem využití cloudové služby **IoT HUB v cloudu Microsoft Azure**. Koncová zařízení (tedy MCU + radiový modul) lze v Azure IoT HUB registrovat a vygenerovat autentizační stringy/klíče. Ty pak musí obsahovat Connection String přidávaný do dat zasílaných pomocí callback z řídícího serveru radiové sítě na URL Azure HUB. Krom komunikace přes HTTP REST API lze používat i protokol MQTT a pomocí radio downlink zpětně ovládat vzdálený proces. Je možné si „naklikat“ cloudové služby, které budou data přijmutá z MCU přes LoRaWAN směrovat do různých úloh, nastavit si pravidla třídění dat pro jednotlivé úlohy, které přijatá data budou dle požadavků zpracovávat (Stream Analytisc Jobs). Např. jen data presentovat do datové struktury pro Power BI a vykreslovat je v grafu.

V případě Azure je většina cloudových služeb placených, krom registrace zařízení v IoT HUB. Pro pouhé vykreslování dat je proto vhodnější použít API jiných cloudových služeb např. IoT Analytics - ThingSpeak Internet of Things.

5. Přílohy

1) Příklady zdrojových kódů - v samostatné příloze.
2) Schema zapojení některých modulů - v samostatné příloze.
3) Formát datové zprávy JSON v LoRaWAN ČRA:

{"type": "D", "data":
"{\"cmd\":\"gw\",\"seqno\":200378730,\"EUI\":\"A8610A3237237901\",\"ts\":1585669308366,\"fcnt\":8,\"port\":2,\"freq\":868300000,\"toa\":51,\"dr\":\"SF7 BW125 4/5\",\"ack\":false,\"gws\":[{\"rssi\":-111,\"snr\":2.5,\"ts\":1585669308382,\"tmms\":1269704526356,\"time\":\"2020-03-31T15:41:47.357068000Z\",\"gweui\":\"B827EBFFFF93E544\",\"lat\":49.8717006,\"lon\":14.273884899999985},{\"rssi\":-113,\"snr\":-3.5,\"ts\":1585669308366,\"tmms\":1269704526357,\"time\":\"2020-03-31T15:41:47.357060357Z\",\"gweui\":\"B827EBFFFF1779D6\",\"lat\":49.86272349999999,\"lon\":14.261414999999943},{\"rssi\":-114,\"snr\":-2.8,\"ts\":1585669308378,\"tmms\":1269704526356,\"time\":\"2020-03-31T15:41:47.357063000Z\",\"gweui\":\"B827EBFFFFB95C6B\",\"lat\":49.872463793500934,\"lon\":14.254774807821605}],\"bat\":0,\"**data\":\"c6eb6a8a2467018126**\"}", "tech": "L", "tags": []}

Popis dat:
"seqno": sekvenční číslo generované Network Serverem CRA
"EUI": unikátní identifikátor zařízení DevEUI
"ts": časová značka vygenerované zprávy
"fcnt": sekvenční číslo zprávy, zaslané ze zařízení
"port": komunikační port na zařízení
"freq": pracovní frekvence, na které byla zpráva odeslána [Hz]
"toa": time on air [ms]
"dr": "oRaWAN parametry (spreading factor, šířka kanálu)
"ack": false, true - jako vyžádané potvrzení předchozí DL zprávy; false – všechny ostatní případy
"gws": [{ seznam gateways, které zprávu zachytily
"rssi": úroveň signálu [dBm] přijatého na GW dle GWEUI
"snr": úroveň odstupu signálu o šumu [dB] přijatého na GW dle GWEUI
"ts": lokalizovaný čas příchodu uplink zprávy na centrální LNS, formát Unix epoch timestamp [ms], https://www.freeformatter.com/epoch-timestamp-to-date-converter.html
"time": UTC čas příchodu uplink zprávy na GW dle GWEUI, formát ISO 8601
"gweui": identifikátor LoRaWAN GW, formát 16 HEX znaků
"lat": zeměpisná šířka GW dle GWEUI, formát decimal degrees (DD)
"lon": zeměpisná délka GW dle GWEUI, formát decimal degrees (DD)

"bat": stav baterie koncového zařízení (0=externí napájení, 255=stav baterie není přenášen, 1-254=odpovídá stavu baterie 0-100%
"data": payload / užitná hodnota. Pokud při importu zařízení zadáte i aplikační klíč AppKey, portál payload dešifruje. V opačném přijde zašifrovaný.

4) Formát datové zprávy JSON v SIGFOX

{ **"data" : "38a4",** "time" : "1585920821", "rssi" : "-139.00", "snr" : "12.31", "number" : "81" }