

ЗБІР ВИМІРЮВАНЬ ІЗ БАГАТОКАНАЛЬНИХ РАДІОТЕХНІЧНИХ СИСТЕМ НА FPGA ДЛЯ ЧАСТОТНИХ ПЕРЕТВОРЮВАЧІВ ФІЗИЧНИХ ВЕЛИЧИН ВИКОРИСТОВУЮЧИ LORA КОНЦЕНТРАТОР НА БАЗІ LILYGO LORA32

Вінницький національний технічний університет

Анотація

У роботі представлено систему збору та передачі вимірювань із багатоканальної радіотехнічної системи на базі FPGA та протоколу LoRa. Розроблено пристрій збору даних і концентратор на модулі Lilygo LORA32, який накопичує та передає вимірювання через WiFi і Bluetooth. Для передачі даних використано компактний формат CBOR, що забезпечує високу точність, надійність і бездротовість системи. Подальший розвиток включає вдосконалення обробки даних, продуктивності концентратора та рішень для реального часу, що відкриває широкі перспективи застосування в IoT, автоматизації та моніторингу.

Ключові слова: FPGA; ESP32; LoRa; ПЛІС; багатоканальна радіовимірювальна система; частотний перетворювач; транзисторна структура з від'ємним опором

Abstract

The study presents a system for collecting and transmitting measurements from a multichannel radio engineering system based on FPGA and the LoRa protocol. A data collection device and a hub based on the Lilygo LORA32 module have been developed, which accumulates and transmits measurements via WiFi and Bluetooth. The CBOR compact format is used for data transmission, ensuring high accuracy, reliability, and wireless capability of the system. Further development includes improving data processing, hub performance, and real-time solutions, opening up broad prospects for applications in IoT, automation, and monitoring.

Keywords: ESP32; LoRa; FPGA; multi-channel radiomeasuring system; frequency transducer; transistor structure with negative resistance

Вступ

Сучасний світ потребує точних вимірювань у моніторингу довкілля, автоматизації та IoT. Основна проблема таких систем — ефективний збір, обробка й передача великих обсягів даних із мінімальними витратами енергії. Традиційні проводові технології обмежують мобільність і масштабованість, що ускладнює використання в розподілених структурах і важкодоступних місцях. Технології FPGA [1] відкривають нові можливості завдяки своїй гнучкості й продуктивності, а сучасні мікропроцесори, як-от NIOS II [2], дозволяють створювати універсальні системи без глибокого знання апаратури. Інтеграція FPGA із бездротовими модулями, такими як LoRa [3], WiFi та Bluetooth, залишається складним завданням, яке потребує оптимізації. Розроблена система на основі FPGA та модуля Lilygo LORA32 вирішує цю проблему, забезпечуючи паралельні вимірювання й бездротову передачу даних. Використання компактного формату CBOR [4] знижує затримки, дозволяючи застосовувати систему в складних умовах, зокрема для мобільних пристроїв і віддалених мереж. Завдяки LoRa досягаються енергоефективність і дальність передачі, що розширює можливості вимірювальних систем, підвищуючи їх продуктивність, адаптивність і масштабованість.

Результати розробки та дослідження

Загальна структура розробленої системи зображена на рисунку 1. Надалі буде розглянуто лише розробку прототипу концентратора та бібліотеки *LoraTransport*. Першим кроком реалізації *LoraTransport* є визначення структури пакета для бездротової передачі. Пакет містить такі поля:

- 1) Загальний ідентифікатор пакета для фільтрування пакетів, згенерованих іншими системами.
- 2) MAC-адресу пристрою, який згенерував цей пакет.
- 3) Ідентифікатор пакета даних.
- 4) Кількість частин з яких складається пакет даних.
- 5) Номер поточної частини.

- 6) Кількість байт у корисному навантаженні.
- 7) Корисне навантаження.

Розмір LoRa-пакета фіксований і становить 119 байт, із яких 100 байт відводиться під корисне навантаження. Якщо розмір корисного навантаження перевищує 100 байт, дані передаються у кількох пакетах.

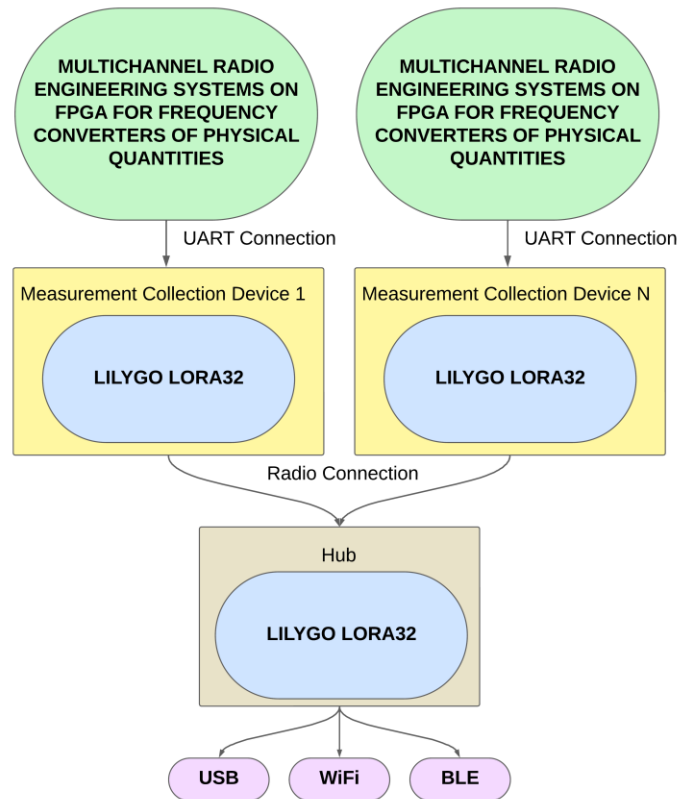


Рисунок 1 – Блок-схема системи збору та передачі вимірювань за допомогою протоколу LoRa

Код з описом LoRa-пакета у вигляді структури *LoraPkg*.

```
#pragma pack(push, 1)
struct LoraPkg {
    uint32_t magic; // Magic number to identify the package
    uint8_t srcMacAddr[LORA_PKG_SRC_MAC_ADDR_SIZE]; // Source MAC address
    uint32_t dataPkgId; // Payload package ID
    uint16_t partsCnt; // Total parts count, package can be splitted into multiple parts if
payload is more than LORA_PKG_PAYLOAD_SIZE_BYTES bytes
    uint16_t partId; // Current part ID
    uint8_t payloadSize; // Count of bytes in the payload
    uint8_t payload[LORA_PKG_PAYLOAD_SIZE_BYTES]; // Payload data, always fixed size, even
if payload is less than LORA_PKG_PAYLOAD_SIZE_BYTES, unused bytes are filled with 0
};
#pragma pack(pop)
```

Наступним етапом реалізації *LoraTransport* є конфігурація каналу зв'язку в протоколі LoRa. Першою важливою конфігурацією є відсутність додаткового заголовка (Implicit Header) у LoRa-пакеті. Цей заголовок додається модулем SX1276 [5] і обробляється приймальною стороною для визначення параметрів корисного навантаження. Переваги та недоліки режиму Implicit Header:

- 1) Швидша передача: скорочений час пакета, менше енергоспоживання.
- 2) Більше корисних даних: вища пропускна здатність.
- 3) Менша зайнятість каналу: зниження ймовірності колізій.
- 4) Фіксовані параметри: необхідність попереднього узгодження налаштувань.
- 5) Більший ризик помилок: менший захист даних без заголовка.
- 6) Обмеження для SF6: обов'язковий Implicit Header на високій швидкості.

Після налаштування режиму Implicit Header можна перейти до додаткових конфігурацій. Ці налаштування є другорядними за впливом на реалізацію і можуть змінюватися у майбутньому:

- 1) Центральна частота передачі – 915МГц.
- 2) Ширина смуги передачі – 500 кГц.

- 3) Фактор розширення – 6.
- 4) Рівень корекції помилки – 7.
- 5) Синхронізаційне слово – 0x99.
- 6) Довжина преамбули – 20 бітів.
- 7) Потужність передавача – 10дБм.

Частина коду який відповідає за встановлення параметрів.

```
#define LORA_FREQ_MHZ      915.0
#define LORA_BW_KHZ       500.0
#define LORA_SF           6
#define LORA_CR           7
#define LORA_SYNC_WORD    0x99
#define LORA_POWER_DBM    10
#define LORA_PREAMBLE_LEN 20
#define LORA_GAIN         0

LoraTransport::LoraTransport(const      std::vector<uint8_t>      &macAddr,      errorCallback
errorCallback) : _macAddr(macAddr),
                 _module(LORA_CS_PIN,      LORA_IO0_PIN,      LORA_RST_PIN,      LORA_IO1_PIN,      _spi,
RADIOLIB_DEFAULT_SPI_SETTINGS),      _radio(&_module),      _mode(Mode::IDLE),
_errorCallback(errorCallback) {
    if (macAddr.size() != LORA_PKG_SRC_MAC_ADDR_SIZE) return;
    _spi.begin(LORA_SCK_PIN, LORA_MISO_PIN, LORA_MOSI_PIN, LORA_CS_PIN);
    pinMode(LORA_IO1_PIN, INPUT);
    pinMode(LORA_IO2_PIN, INPUT);
    uint8_t status = _radio.begin(LORA_FREQ_MHZ, LORA_BW_KHZ, LORA_SF, LORA_CR,
LORA_SYNC_WORD, LORA_POWER_DBM, LORA_PREAMBLE_LEN, LORA_GAIN);
    _radio.implicitHeader(0);
}
```

Наступний крок — створення прототипу концентратора. Концентратор складатиметься з трьох основних блоків і драйверів для вихідних інтерфейсів. Деякі блоки, як-от *LoraTransport* і *DataPackage*, вже реалізовані для пристрою збору вимірювань, тому їх повторна розробка не потрібна. Інші компоненти, такі як *PackageManager* і драйвери для USB, Wi-Fi та BLE, виходять за межі цієї статті.

Нижче наведено приклад реалізації концентратора, який забезпечує отримання вимірювань від пристроїв збору, їх обробку та виведення інформації на UART.

Ініціалізація концентратора, переведення його у режим приймача і реєстрація зворотного виклику для знайдених пакетів.

```
uint8_t mac[6];
WiFi.macAddress(mac);
macAddr = std::vector<uint8_t>(mac, mac + 6);
loraTransport = new LoraTransport(macAddr, transportErrorCallback);
LOGI(TAG, "LoraTransport initialized");
loraTransport->startReceiving(receivedPkgCallback);
```

Пошук пакетів від пристроїв збору.

```
loraTransport->loop();
```

Обробка отриманих пакетів.

```
void receivedPkgCallback(const      std::vector<uint8_t>      &srcMacAddr,      uint32_t      pkgId,      const
std::vector<uint8_t>      &data) {
    DataPackage dataPackage;
    dataPackage.setRawData(data);
    if (dataPackage.isMacAddrExist()) {
        LOGI(TAG, "MAC address: %02X:%02X:%02X:%02X:%02X:%02X", dataPackage.getMacAddr()[0],
dataPackage.getMacAddr()[1], dataPackage.getMacAddr()[2], dataPackage.getMacAddr()[3],
dataPackage.getMacAddr()[4], dataPackage.getMacAddr()[5]);
    }
    if (dataPackage.isPkgIdExist()) {
        LOGI(TAG, "Package ID: %d", dataPackage.getPkgId());
    }
    if (dataPackage.isTimeStampExist()) {
        LOGI(TAG, "Time stamp: %d", dataPackage.getTimeStampMs());
    }
    if (dataPackage.isFmDataExist()) {
        FmData fmData = dataPackage.getFmData();
        LOGI(TAG, "Fm data freqs: %d, %d, %d, %d, %d, %d, %d, %d, %d, %d, %d, %d",
fmData.freqs[0], fmData.freqs[1], fmData.freqs[2], fmData.freqs[3], fmData.freqs[4],
fmData.freqs[5], fmData.freqs[6], fmData.freqs[7], fmData.freqs[8], fmData.freqs[9],
fmData.freqs[10], fmData.freqs[11]);
    }
```

```
LOGI(TAG, "Fm data: temp: %d, accel_x: %d, accel_y: %d, accel_z: %d, gyro_x: %d,
gyro_y: %d, gyro_z: %d", fmData.temp, fmData.accel_x, fmData.accel_y, fmData.accel_z,
fmData.gyro_x, fmData.gyro_y, fmData.gyro_z);
}
}
```

Висновки

У роботі представлено підхід до створення багатоканальної вимірювальної системи на основі FPGA з використанням модуля Lilygo LORA32, який поєднує високу продуктивність із бездротовою передачею даних. Система забезпечує надійний збір, обробку й передачу вимірювань завдяки протоколу CBOR і технології LoRa. Основні переваги розробки:

1. Гнучкість і масштабованість: Система адаптується до різних сценаріїв завдяки підтримці LoRa, WiFi і Bluetooth, що дозволяє працювати у віддалених місцях.
2. Ефективність обробки: Використання CBOR мінімізує обсяг даних, зменшуючи навантаження на канали зв'язку.
3. Енергоефективність: LoRa забезпечує тривалу роботу в умовах обмеженого живлення.
4. Універсальність: Поєднання FPGA й LoRa дає змогу додавати нові сенсори та алгоритми обробки.

Розробка стала важливим кроком у створенні сучасних вимірювальних систем із високою точністю та надійністю. Подальший розвиток включає вдосконалення обробки даних, підвищення продуктивності та створення рішень для моніторингу в реальному часі, відкриваючи нові перспективи в IoT, автоматизації та екологічному моніторингу.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. What is an FPGA? [Електронний ресурс]: - Режим доступу: <https://www.xilinx.com/products/silicon-devices/fpga/what-is-an-fpga.html>
2. Nios II Processor Reference Handbook. – San Jose: Altera, 2016. – 11 с.
3. LoRa Alliance. [Електронний ресурс]: - Режим доступу: <https://lora-alliance.org/>
4. CBOR. [Електронний ресурс]: - Режим доступу: <https://cbor.io/>
5. SX1276/77/78/79 - 137 MHz to 1020 MHz Low Power Long Range Transceiver. [Електронний ресурс]: - Режим доступу: https://cdn-shop.adafruit.com/product-files/3179/sx1276_77_78_79.pdf

Осадчук Олександр Володимирович — докт. техн. наук, проф., зав. кафедри інформаційних радіоелектронних технологій і систем, Вінницький національний технічний університет, osadchuk.av69@gmail.com

Скощук Валентин Костянтинович — аспірант кафедри інформаційних радіоелектронних технологій і систем, Вінницький національний технічний університет, skoschuk999@gmail.com

Alexander Osadchuk — Doc. Tech. Sc., prof. Head of Department of Information Radio Engineering Technologies and Systems, Vinnytsia National Technical University, Vinnytsia, Ukraine, osadchuk.av69@gmail.com

Valentyn Skoshchuk — graduate student of the Department of Information Radio Engineering Technologies and Systems, Vinnytsia National Technical University, Vinnytsia, Ukraine, skoschuk999@gmail.com