

Державний вищий навчальний заклад  
«Прикарпатський національний університет імені Василя Стефаника»  
Фізико-технічний факультет  
Кафедра комп'ютерної інженерії та електроніки

Магістерська робота

на тему:

“Система дистанційного моніторингу за станом немовлят”

Виконав: студент групи КІ(м)-21

напряму підготовки

123 Комп'ютерна інженерія

Сандалович В. В.

Керівник: к.т.н., доцент Грига В. М.

Рецензент: к.ф.м.н., проф. Никируй Л. І.

м. Івано-Франківськ

2019



Державний вищий навчальний заклад  
 «Прикарпатський національний університет імені Василя Стефаника»  
 Фізико-технічний факультет  
 Кафедра «Комп'ютерної інженерії і електроніки»

Пояснювальна записка  
 до магістерської роботи  
 на тему:  
 «Система дистанційного моніторингу за станом немовлят»

|                  |             |                   |               |             |                 |             |              |
|------------------|-------------|-------------------|---------------|-------------|-----------------|-------------|--------------|
|                  |             |                   |               |             | 123.КІ(м)-21.12 |             |              |
| <i>Змін.</i>     | <i>Арк.</i> | <i>№ докум.</i>   | <i>Підпис</i> | <i>Дата</i> |                 |             |              |
| <i>Розробив</i>  |             | Сандалович Василь |               |             |                 | <i>Арк.</i> | <i>Аркуш</i> |
| <i>Перевірів</i> |             | Грига В. М.       |               |             |                 |             | 3            |
| <i>Н. Контр.</i> |             |                   |               |             |                 |             |              |
| <i>Затверд.</i>  |             |                   |               |             |                 |             |              |

## АНОТАЦІЯ

В даній магістерській кваліфікаційній роботі розроблено систему дистанційного моніторингу стану немовлят. Така система призначена для виміру телеметричних показників дитини, таких як температура та пульс, і передачі їх, разом з голосом дитини, на батьківській пристрій для моніторингу батьками. Система орієнтована на використання в домашніх умовах.

В проєкті описується апаратна реалізація вимірювача температури та пульсу, побудованих на системі на кристалі CY8C24223. Реалізовано безпроводну завадостійку передачу голосу дитини в реальному часі з вкладеними вимірними телеметричними даними на мікроконтролері CYRF6936. Для пристроїв розміщених поряд з дитиною та батьками використано мікроконтролери сімейства CY8C29x66.

|     |      |          |        |      |                 |      |
|-----|------|----------|--------|------|-----------------|------|
|     |      |          |        |      | 123.KI(м)-21.12 | Арк. |
|     |      |          |        |      |                 | 4    |
| Зм. | Арк. | № докум. | Підпис | Дата |                 |      |













його випромінювання. Ця властивість дозволяє проводити вимірювання температури тіла без фізичного контакту з ним.

Медична термографія дає можливість вимірювати температуру різних ділянок тіла з точністю до декількох десятих кельвіна. Важливість отриманої інформації пояснюється тим, що температура шкіри в даній точці залежить від локальних клітинних процесів і від кровопостачання даної ділянки тіла. Термографія використовується для локалізації і визначення ступеня порушень при артриті, для визначення глибини ураження тканин при обмороженнях і опіках і для виявлення різних порушень периферійного кровообігу, таких як венозний тромбоз, оклюзія сонної артерії та ін [1].

У реальних умовах на достовірність результатів БВТ впливають неідеальність об'єкту вимірювань, середовище поширення випромінювання, наявність паразитних джерел та ін. У структурі радіометричного ланцюга (рис. 1.1) реальний об'єкт представлений у вигляді абсолютно чорного тіла (АЧТ), сигнал якого проходить через ланку з коефіцієнтом випромінювання  $\epsilon(\lambda, T, \varphi, x, y) < 1$ , що визначається станом поверхні, формою, матеріалом об'єкту і є складною функцією довжини хвилі, температури, напрямку поширення для даної координати поверхні.

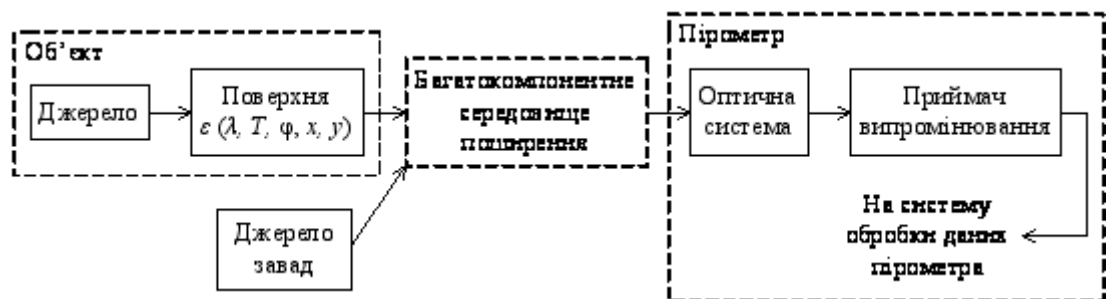


Рис. 2.1 – Структурна схема радіометричного ланцюга

При поширенні випромінювання з корисним сигналом сумується випромінювання фону і завад (широкосмугові при наявності інших нагрітих тіл або вузькосмугові від газів, що горять). Параметри середовища поширення, що вносить вибіркоче ослаблення сигналу в різних спектральних інтервалах, і їх флуктуації залежать від температури, тиску, компонентного складу і концентрації газів, ная-

вності дисперсних включень і т. п. Необхідно враховувати також характеристики оптичної системи і приймача випромінювання.

Через велику кількість чинників підходити до рішення задачі визначення температури найбільш доцільно зі статистичної точки зору [2].

#### 1.1.4 Оптоволоконні термодатчики

На рис. 1.3 зображено схему напівпровідникового зонда для вимірювання температури на основі арсеніда галію GaAs. Невеликий монокристал бездомішкового GaAs, виготовлений у вигляді призми, закріплений за допомогою епоксидної смоли на кінці двох сполучених разом оптоволоконних світлопроводів. Розміри датчика і світлопроводів повинні бути достатньо малими, щоб разом із захисним ковпачком їх можна було помістити в точку вимірювання. Один зі світлопроводів передає світло від світлодіода до датчика. Світло проходить через кристал GaAs і потім по другому світлопроводу передається у вимірювальний пристрій. При проходженні через напівпровідник частина енергії випромінювання поглинається, причому ця енергія передається електронам, які через це зможуть перестрибнути через заборонену зону з валентної зони в зону провідності. Ширина забороненої енергетичної зони сильно залежить від температури напівпровідника, тому при підвищенні температури кількість енергії, що поглинається, зростає.

Такі неметалічні зонди особливо зручні для вимірювання температури в сильних електромагнітних полях, які можуть використовуватися для нагрівання тканин при лікуванні раку або обігріві пацієнта [1].

|     |      |          |        |      |                 |      |
|-----|------|----------|--------|------|-----------------|------|
|     |      |          |        |      | 123.KI(м)-21.12 | Арк. |
|     |      |          |        |      |                 | 11   |
| Зм. | Арк. | № докум. | Підпис | Дата |                 |      |













ІЧ-датчик не потрібно. Через дві секунди (час, необхідний для стабілізації аналогової схеми) ви побачите мигання світлодіода синхронно з биттям серця.

Для перевірки точності і ефективності роботи плати сенсора автор використовував отладочную плату ChipKIT UNO32, до якої підключалася базова плата вводу / виводу IO Shield. Плата введення / виведення забезпечена невеликим матричним OLED дисплеєм, на якому буде відображатися виміряне значення пульсу. Сигнальні висновки плати ІЧ-сенсора VCC (харчування), EN (активація ІЧ-датчика), VOUT (вихідний сигнал), Gnd (Загальний, "земля") підключаються до контактів плати введення / виведення 3.3V, 5.0V, Pin 2 і Gnd , відповідно. Зауважте, при такому включенні ІЧ-датчик постійно знаходиться в активному стані.

|     |      |          |        |      |                 |      |
|-----|------|----------|--------|------|-----------------|------|
|     |      |          |        |      | 123.KI(м)-21.12 | Арк. |
|     |      |          |        |      |                 | 17   |
| Зм. | Арк. | № докум. | Підпис | Дата |                 |      |











-генератор випадкових псевдо послідовностей

Цифрові блоки можуть бути приєднані до будь-яких портів вводу/виводу через серію глобальних шин які можуть направляти будь-який сигнал на будь-який пін.

### 2.1.3. Аналогова підсистема мікроконтролерів сімейства CY8C29x66

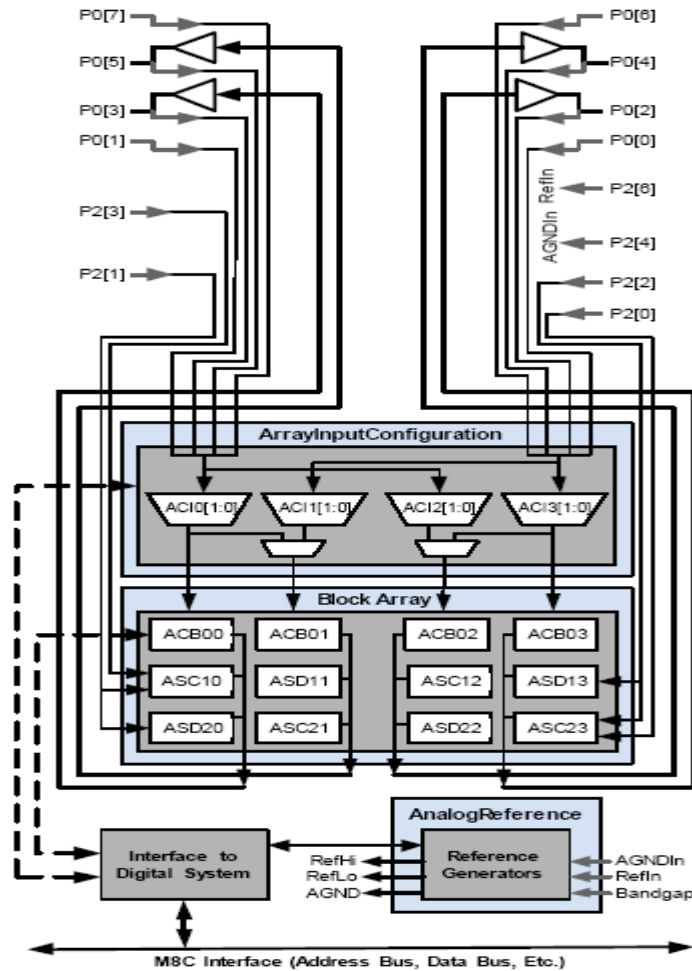


Рис. 2.3. Структурна схема аналогового блоку мікропроцесора CY8C29x66

Аналогова система складається з 12-ти блоків, кожен з яких складається з операційного підсилювача, який дозволяє створювати потік комплексних аналогових сигналів. Аналогові периферійні пристрої є дуже гнучкими і можуть бути зроблені для підтримки специфічних програмних потреб. Деякі з багатьох загальних функцій (найбільш підтримуваних, як модулів користувачів) є подані нижче.

-Аналогово-цифровий перетворювач (до 4-х, з 6-ти до 14-ти бітовим розширенням)

-Фільтри (2,4,6 і 8 дорожні)















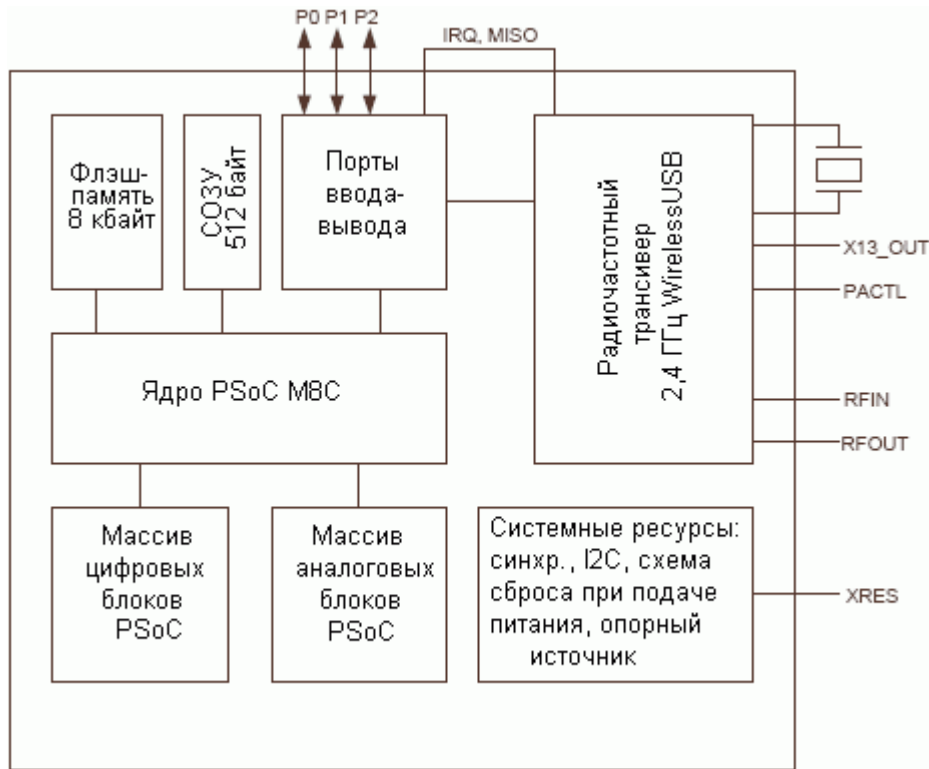


Рис 2.5. Структурна схема CYRF6936:

|     |      |          |        |      |
|-----|------|----------|--------|------|
|     |      |          |        |      |
| Зм. | Арк. | № докум. | Підпис | Дата |

### 3. ПРОЕКТНО-РОЗАХУНКОВИЙ РОЗДІЛ

#### 3.1. Опис структури системи

Поставлене завдання по проектуванню може вирішити мобільна система, призначена для контролю за станом здоров'я дитини в режимі реального часу.

Вона складається з трьох основних модулів:

- Модуль, розміщений на тілі дитини - On-Child-Device .
- Модуль, який розміщений біля дитини (до 5 метрів від On-Child-Device) - Child-Device.
- Модуль для батьків, які можуть знаходитись до 20 метрів від Child-Device-Parent-Device.

На рисунку 3.1. зображено загальну структурну схему проектованої системи.

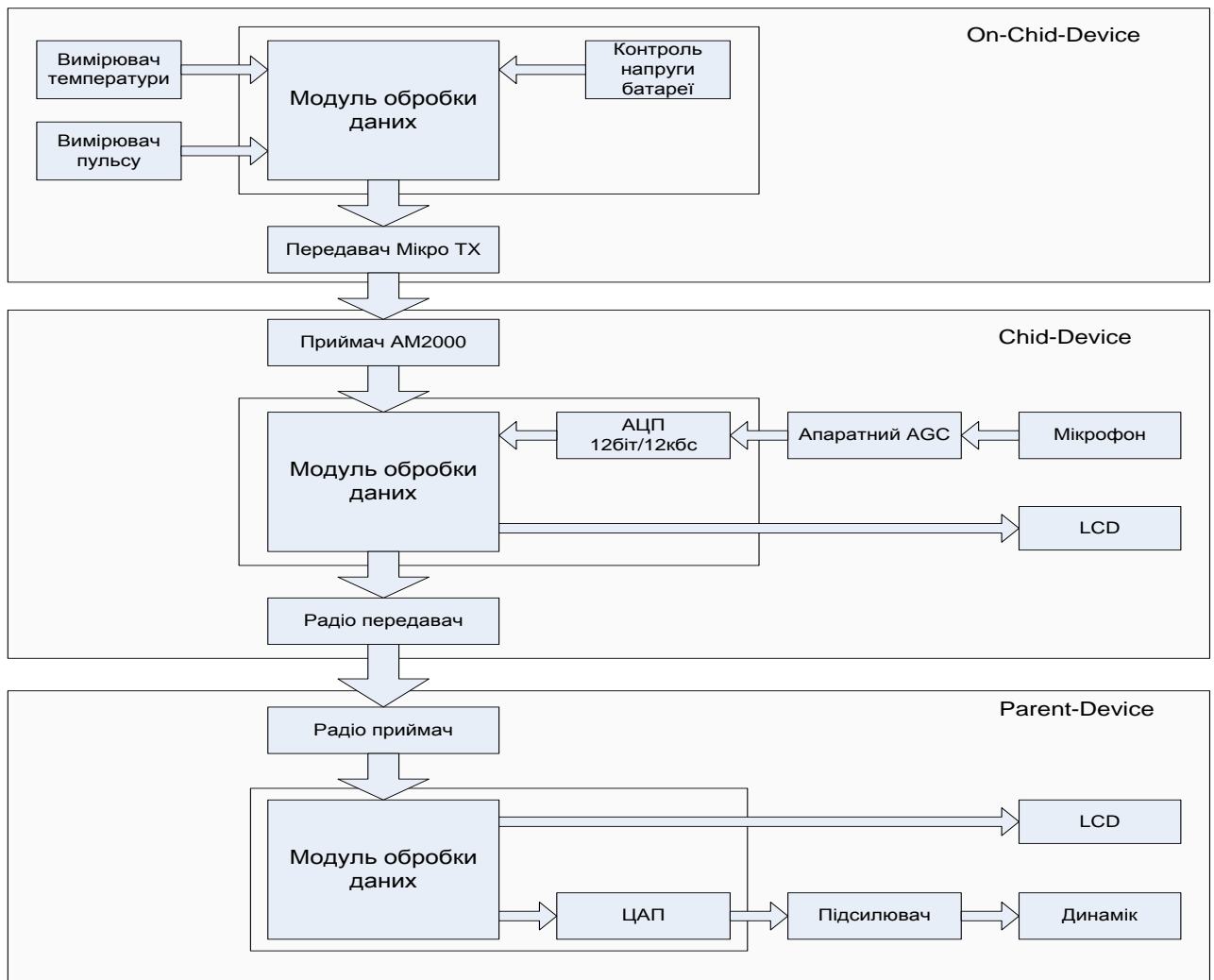


Рис. 3.1. Загальна структурна схема пристрою.

|     |      |          |        |      |
|-----|------|----------|--------|------|
|     |      |          |        |      |
| Зм. | Арк. | № докум. | Підпис | Дата |





лосового сигналу в цифрову форму використовується дельта-сігма аналого-цифровий перетворювач. Отриманий цифровий голосовий сигнал передається на Parent-Device через радіо-канал, побудований на устаткуванні WirelessUSB. Телеметричні пакети даних передаються одночасно з передачею голосових семплів. Для мінімізації рівня помилок під час передачі пакету, використовується перевірка цілісності пакету. Отримані телеметричні дані відображаються на рідкокристалічному індикаторі.

Батьківський пристрій отримує цифровий голосовий сигнал і вкладені телеметричні дані від Child-Device. Отримані телеметричні дані також відображаються на рідкокристалічному індикаторі.

У всіх радіо-каналів є здатність автоматичної пересинхронізації у разі втрати сигналу.

## **3.2. Проектування модуля On-Child-Device розташованого на тілі дитини**

### **3.2.1. Опис проектного пристрою**

|            |             |                 |               |             |                        |             |
|------------|-------------|-----------------|---------------|-------------|------------------------|-------------|
|            |             |                 |               |             | <i>123.KI(м)-21.12</i> | <i>Арк.</i> |
| <i>Зм.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> |                        | 33          |



























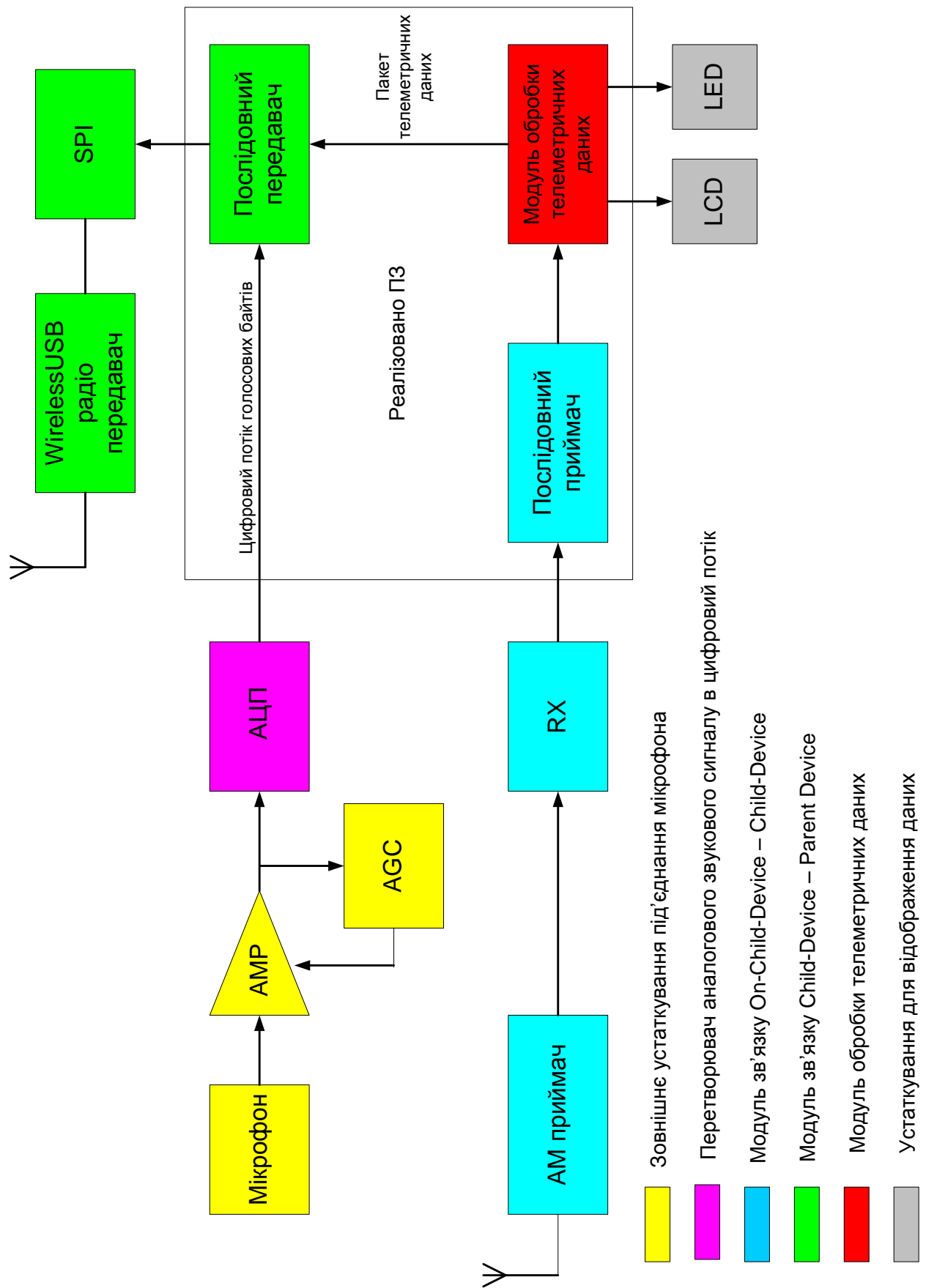


Рис. 3.6. Функціональна схема модуля Child-Device.

|     |      |          |        |      |
|-----|------|----------|--------|------|
| Зм. | Арк. | № докум. | Підпис | Дата |
|     |      |          |        |      |













Зміст буфера пакету не змінюється під час передачі або прийому пакетів АСК. В кожному випадку вся операція пакету відбувається без будь-якої потреби в мікропрограмній дії MCU (поки використовується пакети 16 байтів, або менше); щоб передати дані, MCU просто повинен завантажити пакет даних, який буде переданий, встановити довжину, і встановити TX GO біт

Так само отримуючи пакети, програмоване устаткування просто повинне відновити повністю отриманий пакет у відповідь на переривання, вказуюче на прийом пакету.

2.4 ГГц радіо передавач використовує підібрані каналом фільтри, щоб досягти високої ефективності у присутності сторонніх сигналів. Інтегрований Підсилювач (PA) забезпечує, потужність передаваного сигналу до +4 dBm з контролем за вихідною потужністю силою 34 децибел в семи кроках.

Пакети даних і реєстри конфігурації отримують доступ через інтерфейс SPI.

### 3.3.6. Розробка протоколу передачі даних

Формат пакету даних має наступну структуру:

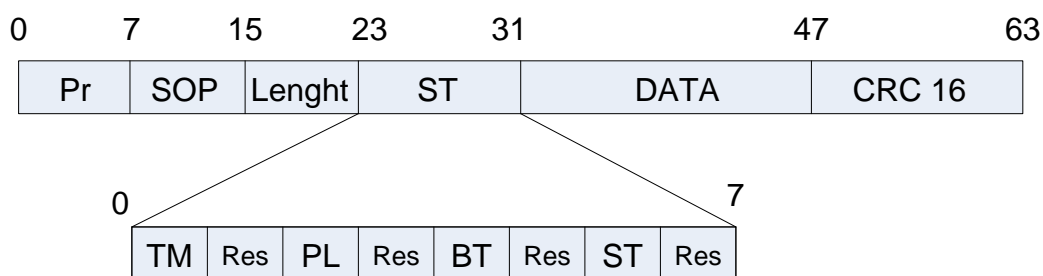


Рис. 3.9. Структура пакету передачі даних

Пакет даних складається з таких полів:

- Pr – байт преамбули, який потрібен для синхронізації корелятора приймача перед отриманням пакету даних.
- SOP - Пакети починаються з двох символів початкового маркера пакету. SOP повинен формуватися, щоб бути одної довжини з обох боків зв'язку.
- Lenght - є два варіанти для того, щоб виявити кінець пакету. Якщо SOP дозволений, то область довжини повинна бути дозволена. Це - перші вісім



- BT - якщо цей біт встановлений, батарея On-Child-Device розряджена. Для нормального функціонування пристрою тоді необхідно замінити батарею новою.
- ST – біт стану, якщо цей біт встановлений то це свідчить що сенсор датчика не спрацьовує, і телеметричні дані не можуть бути виміряні.
- DATA - корисні дані, які повинні бути передані через радіо-канал.

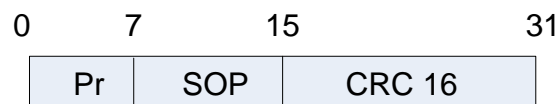


Рис. 3.10. Структура пакету підтвердження отриманих даних

### 3.4. Проектування модуля Parent-Device розташованого поряд з батьками

#### 3.4.1. Опис проектованого пристрою

Parent Device - модуль який батьки розташовують біля себе. Він відображає всі телеметричні дані і аудіо сигнал з дитячої кімнати, отриманий від Child-Device.

У Parent-Device також є два основні потоки даних: цифровий голосовий потік даних і телеметричний потік даних. Рисунок 3.11. зображає блок-схему батьківського пристрою.

Згідно системній блок-схемі модуль Parent-Device складається з наступних модулів:

- модуль зв'язку між Child-Device та Parent-Device.
- декодер даних.
- модуль обробки телеметричних даних.
- устаткування для відображення даних.
- звуковий ЦАП.

Дані, отримані через радіо-канал з Child-Device, декодовані декодером даних. Декодовані дані через модуль SPI, надходять у вбудоване програмне забезпечення здійснене послідовним приймачем і приймачем телеметричних даних.

Приймач телеметричних даних отримує телеметричні пакети даних для наступного декодування, синтаксичного аналізу і обробки. Отримані телеметричні дані відображені LCD і світлодіодами LEDs. Послідовний приймач відновлює цифровий потік голосових байтів. Дешевий звуковий ЦАП і звуковий підсилювач з регулюванням гучності забезпечують звукове відтворення сигналу динаміком.

Подробиці реалізації цих модулів окремо і Parent-Device взагалі розглянуті на рис. 3.11.

|     |      |          |        |      |                 |      |
|-----|------|----------|--------|------|-----------------|------|
|     |      |          |        |      | 123.KI(м)-21.12 | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата |                 | 55   |

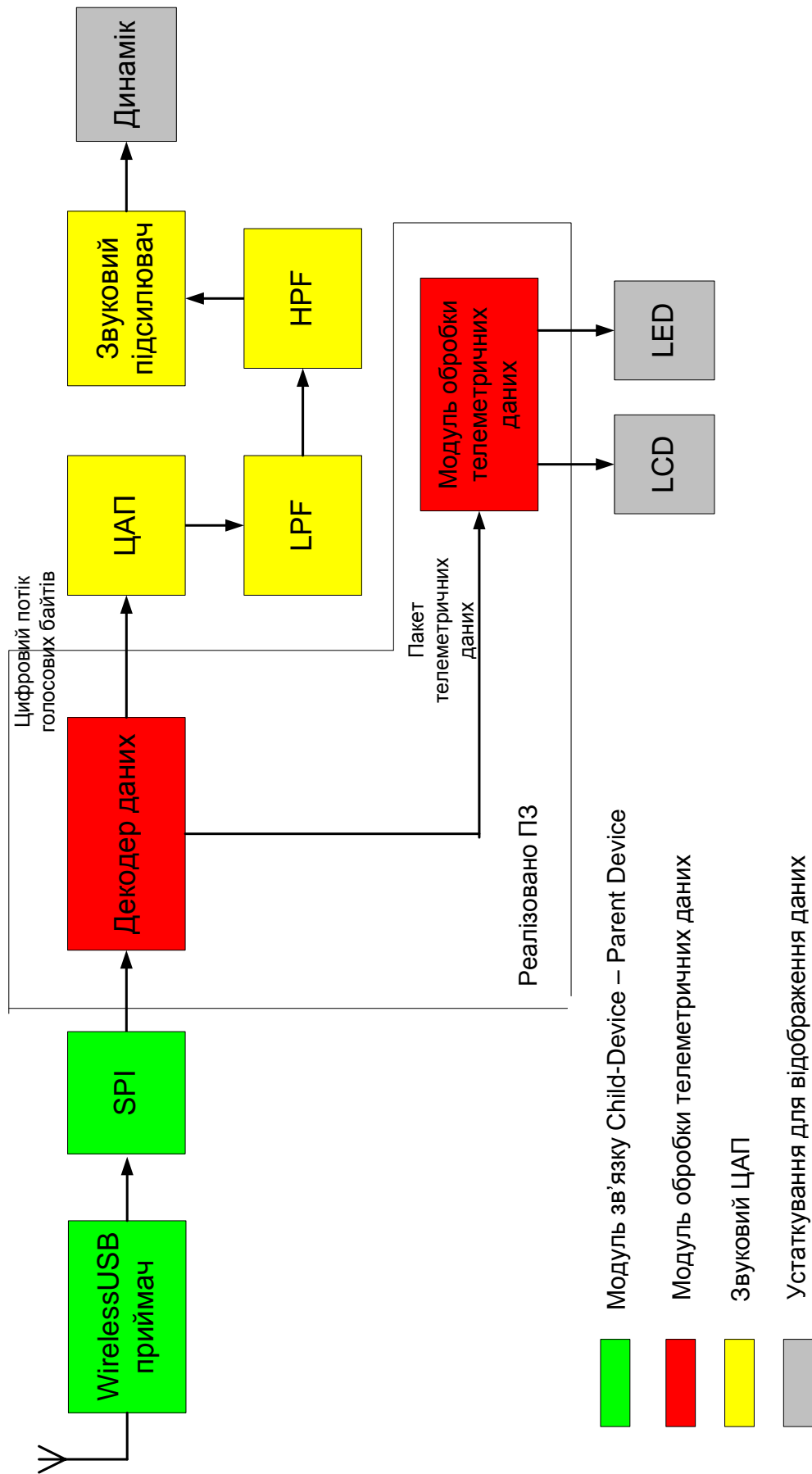


Рис. 3.11. Функціональна блок-схема модуля Parent-Device.

|     |      |          |        |      |
|-----|------|----------|--------|------|
|     |      |          |        |      |
| Зм. | Арк. | № докум. | Підпис | Дата |



### 3.4.2. Розробка модуля зв'язку між модулями Child-Device та Parent-Device

Цей модуль забезпечує отримання цифрового голосового сигналу і телеметричних даних через радіо-канал від Child-Device .

Апаратні засоби модуля комунікації і протокол передачі даних описані вище детальніше (див. опис модуля зв'язку Child-Device).

Оскільки сила переданого сигналу значно вища ніж потрібно для опрацювання, використовується керуючий сигнал RSSI (received signal strength indication) для визначення представлення потоку даних. Функціональні залежності між рівнем вхідного сигналу і вихідною напругою RSSI показані в таблиці 3.2.

Напруга RSSI порівнюється з набором порогів чутливості. Після цього приймається рішення, чи повинні отримані дані бути проаналізовані чи ні. Пороговий рівень чутливості визначає компроміс між радіо-діапазоном каналу і помилковим рівнем. Нижчий пороговий рівень збільшує діапазон каналу і помилковий рівень і навпаки.

Таблиця 3.2. RSSI вихідна напруга згідно рівня вхідного сигналу

| Вхідний рівень сигналу, dBm | RSSI напруга, V | Вхідний рівень сигналу, dBm | RSSI напруга, V |
|-----------------------------|-----------------|-----------------------------|-----------------|
| -10                         | 2.5             | -60                         | 1               |
| -20                         | 2.4             | -65                         | 0.7             |
| -30                         | 2.1             | -70                         | 0.6             |
| -40                         | 1.7             | -80                         | 0.3             |
| -50                         | 1.4             | -90                         | 0.1             |

Після отримання і демодуляції радіо-сигнал потрібно декодувати. Для декодування радіо-сигналу використовується вбудований декодер.

На один вхід декодера поданий безпосередньо вхідний сигнал даних; на інший повинен бути поданий вхідний сигнал через низькочастотний фільтр. Використання низькочастотного фільтру потрібне, щоб забезпечити середнє число на-

пруги вхідного сигналу і таким чином дає середню межу даних. Таким чином вхідний сигнал даних порівнюється з середнім значенням. Після цього рішення, вихідний сигнал повинен бути логічною одиницею або нулем.

Перед початком прийому, напруга низькочастотного фільтру повинна змінитись до середньої межі вхідного сигналу. Якщо цей процес є дуже повільним тоді, можуть бути помилки в даних в передній стороні вхідного пакету даних. Щоб уникнути цієї проблеми, напруга середньої межі повинна бути отримана швидко. Але завжди будуть деякі пульсації і в середньому напруга, особливо якщо вхідні дані не будуть збалансовані системою цифрового управління. Ці пульсації можуть викликати в процесі помилки. Ідеальна напруга середньої межі повинна бути отримана швидко, але повинні бути, наскільки можливо, такі ж низькі пульсації (константа довгого часу).

### **3.4.3. Розробка модуля відображення даних**

Всі отримані телеметричні дані після обробки повинні бути візуально відображені для контролю за дитиною батьків. У цьому проекті для відображення частоти пульсу дитини і температури використовується рідкокристалічний Volymn Inc. LCD 0801A. Це - галузевий стандарт фірми Hitachi HD44780, сумісний з двома рядками для 16 символів.

Як альтернатива, може використовуватися дешевий графічний LCD з SPI або інтерфейсом комунікації I2C, побудованим наприклад на Phillips PCD8544 або PCF8548.

## **3.5. Розробка внутрішньої структури мікроконтролерів PSoC**

### **3.5.1. Внутрішня організація PSoC модуля On-Child-device**

Внутрішня структура PSoC показана на рисунку 3.12.

Вхід PGA для вимірів температури знаходиться в ACB01. Аналоговий мультиплексор забезпечує перемикання між необхідними портами. Під час виміру температури вихід ADC перемикається до вихода підсилювача програмно.

|            |             |                 |               |             |                        |             |
|------------|-------------|-----------------|---------------|-------------|------------------------|-------------|
|            |             |                 |               |             | <i>123.KI(м)-21.12</i> | <i>Арк.</i> |
| <i>Зм.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> |                        | 58          |

Послідовний передавач, знаходиться в DCB03, разом з таймером швидкості передачі даних, розміщеним в DCB02, використовується для передачі даних LM-TXAM2433. SysClk, встановлений в 6 МГц.

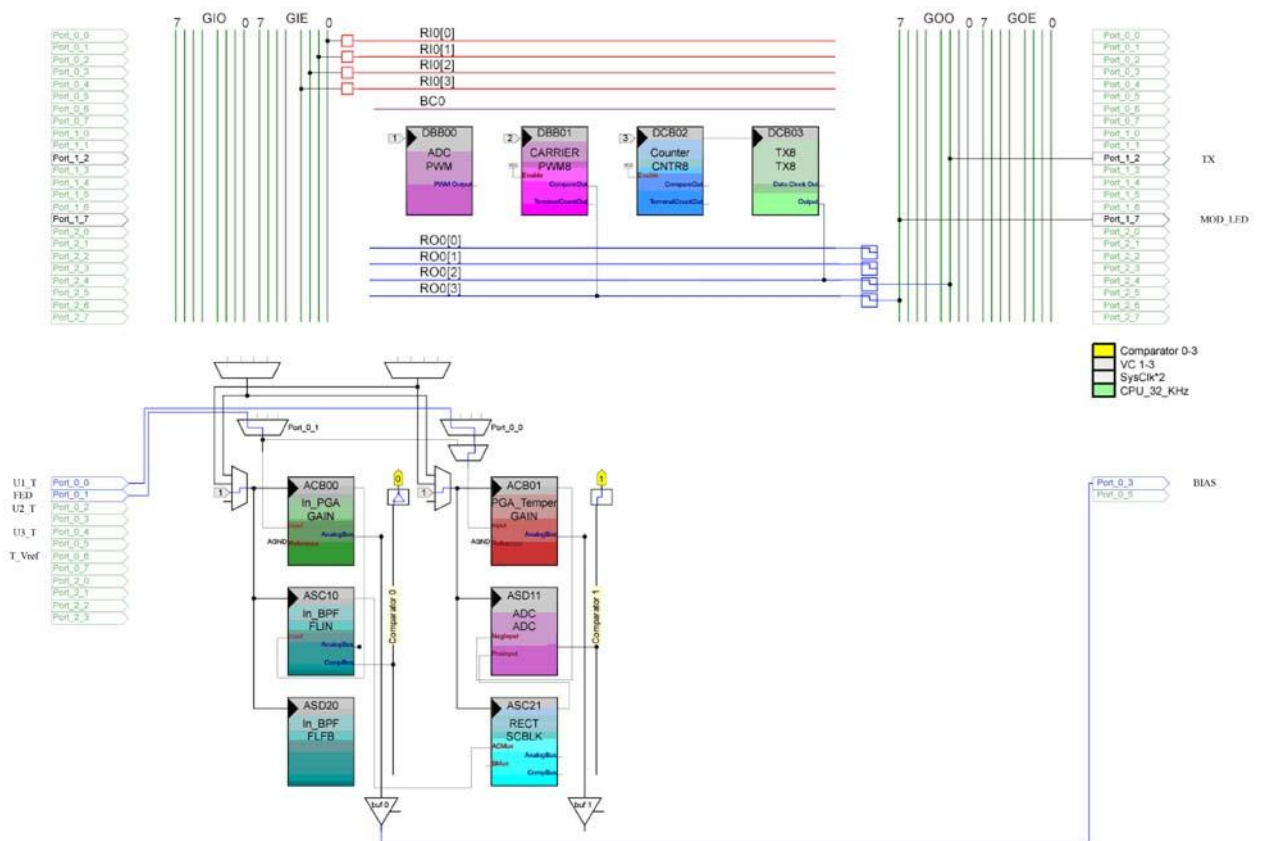


Рис. 3.12. Внутрішня структура PsoC On-Child-Device

### 3.5.2. Внутрішня організація PSoC модуля Child-device

Внутрішню структуру PSoC показано на рисунку 3.13. Корпус PGA, поміщений в ACB00, забезпечує підсилену буферизацію звукового сигналу. Вихідний сигнал PGA відсилається на P0 [3]. Для аналого-цифрового перетворення буферизованого сигналу використовується аналого-цифровий перетворювач сигма-дельта з 12-бітовою роздільною здатністю. Цей аналого-цифровий перетворювач поміщений в ASC10, ASD20, DBB00 і блоки DBB01. Тактовий генератор аналого-цифрового перетворювача встановлений в VC1 і рівний 6 МГц. Таким чином аналого-цифровим перетворювачем забезпечується частота семплів, рівна 11718,75 sps.

Послідовний передавач, поміщений в DCB12, разом з таймером швидкості пересилки даних , поміщеним в DBB10 і DBB11, використовується для прийому даних від LM-RXAM2433. Для ініціалізації LM-RXAM2433 використовується модуль SPI, який поміщений в DCB02. Послідовний передавач, поміщений в DCB13, забезпечує передачу даних до Parent-Device. Швидкість радіо-каналу рівна 62.5 кбит/сек. Послідовний приймач в DCB03 використовується для отримання даних від Радіо-модуля. Цей приймач не використовується в цьому проекті і зарезервований для майбутніх застосувань.

Корпус PGA, розміщений в ACB01, використовується для буферизації керуючого сигналу системи AGC. Вихідний сигнал PGA отриманий через порт P0 [5]. Слід зазначити що проект може бути перенесеним на групу PSoC CY8C27xxx зі звуковою роздільною здатністю АЦП зменшеного до 8 біт.

|     |      |          |        |      |                 |      |
|-----|------|----------|--------|------|-----------------|------|
|     |      |          |        |      | 123.KI(м)-21.12 | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата |                 | 60   |

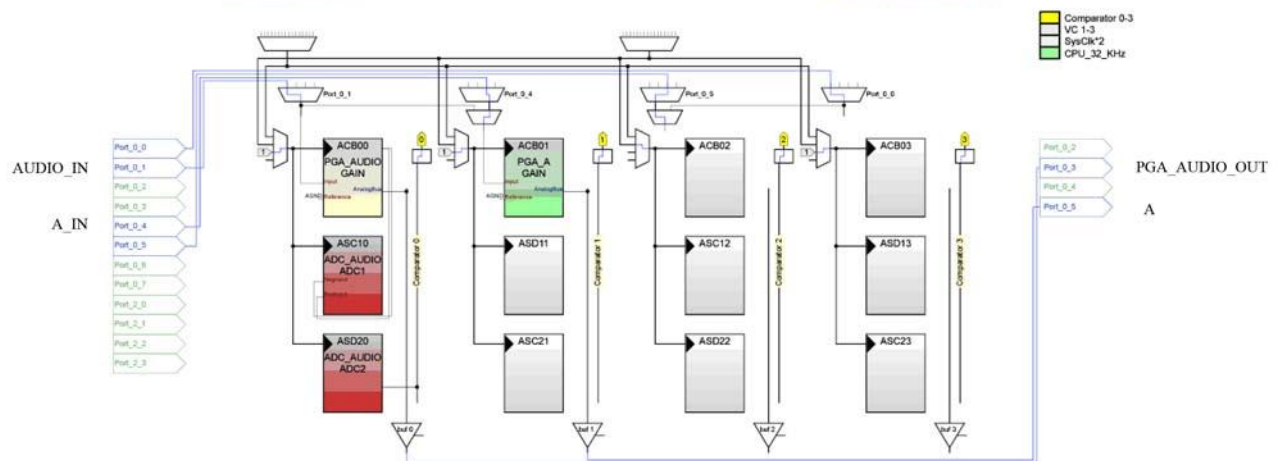
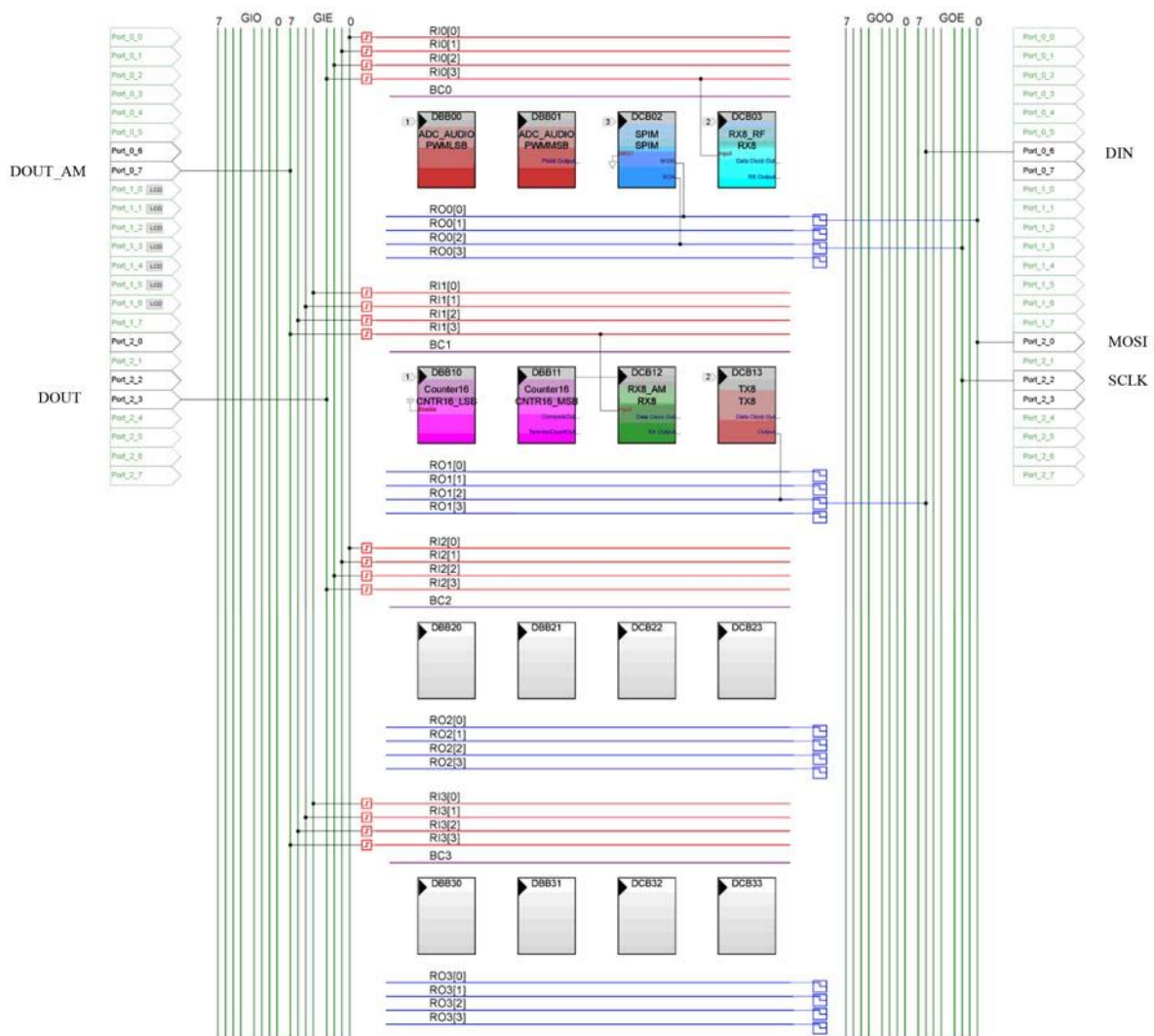


Рис. 3.13. Внутрішня структура PsoC Child-Device

|     |      |          |        |      |
|-----|------|----------|--------|------|
|     |      |          |        |      |
| Зм. | Арк. | № докум. | Підпис | Дата |

### 3.5.3. Внутрішня організація PSoC модуля Parent-Device

Внутрішня структура PSoC показана на рисунку 3.14. Аналоговий сигнал даних радіо модуля ML2724 через порт, P0 [7] буферизується PGA, який поміщений в АСВ00. Вихідний сигнал PGA надходить в порт P0 [2] для наступного декодування зовнішнім декодером. Модуль послідовного приймача для декодованого сигналу поміщений в DCB03. Як джерело тактового генератора використовується VC1 рівне 3 МГц.

Для ініціалізації радіо-передавача використовується SPI модуль, який поміщений в DCB12. Швидкість цього зв'язку рівна 115200 кбіт/сек. Пороговий компаратор, поміщений в АСВ03, використовується для оцінки рівня сили отриманого сигналу. Низька межа компаратора встановлена як VSS. Пороговий рівень - задається програмно. Вихід компаратора під'єднаний до шини 3 Компаратора для наступного програмного аналізу. Аналоговий вихід компаратора через аналогову шину і порт P0[2] під'єднаний до світлодіодів. Це забезпечує візуальну оцінку присутності радіорелейної лінії.

Для дешевого звукового ЦАП потрібна високочастотна 12-бітова PWM . Один PWM в цій роздільній здатності в змозі працювати на 12 кгц. PWM чергується місцями в DBB00, DBB01 і DBB02. Вихідний цифровий буфер, поміщено в DBB10. Всі джерела PWM синхронізовані тим же самим джерелом.

|     |      |          |        |      |                 |      |
|-----|------|----------|--------|------|-----------------|------|
|     |      |          |        |      | 123.KI(м)-21.12 | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата |                 | 62   |

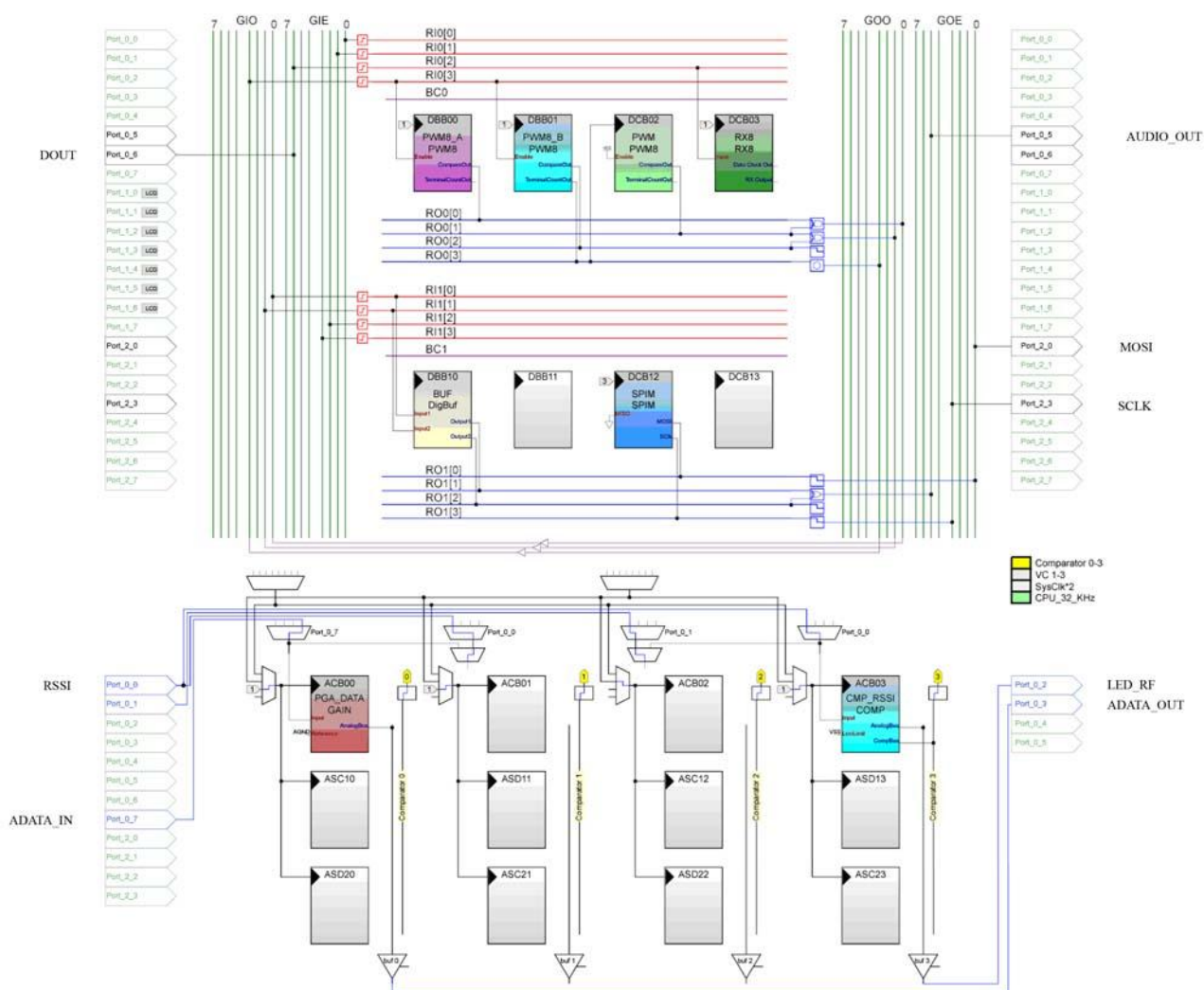


Рис. 3.14. Внутрішня організація PSoC Parent-Device

### 3.6. Розробка програмного забезпечення

#### 3.6.1. Вбудоване програмне забезпечення модуля On-Child-Device

Вбудоване програмне забезпечення On-Child-Device здійснює керування пристроєм і обробку даних. Воно складається з декількох модулів, які реалізують відповідні функції, такі як вимірювання температури і величини пульсу, попередня обробка отриманих телеметричних даних, величина напруги джерела живлення і управління розміщенням пристрою, і передачу даних до Child-Device.

Алгоритм вимірювання температури, показаний на рисунку 3.19. Він є досить простий для розуміння. Спочатку виміру програмне забезпечення перепідключає вхід аналого-цифрового перетворювача з виходом PGA\_Temper і запускає

ці модулі. Після цього значення напруги отримуються в трьох точках дільника опорів. На цьому процес вимірювання температури закінчений. Апаратні модулі зупинені і напруга перестає подаватись до вимірювальної установки. Для температури дитини визначено значення правої сторони рівняння 3.4. Всі обчислення здійснені в цілочисельному форматі. Результат збережений в пакеті яких відсилається, і біт який містить величину температури байта стану встановлений.

Управління напругою джерела живлення банальне. Для керування напругою використовується PSOC модуль датчика низької напруги. Система управління розміщення пристрою аналізує стан порту, підключеного до сенсорної кнопки.

Перед передачею, пакет що передається кодується Манчестерським кодером. Під час манчестерського кодування кожен байт даних перетворюється згідно табличної функції.

Основний цикл програми забезпечує загальне управління пристроєм. Для мінімізації споживаної потужності температурне вимірювання виконується один раз в п'ятнадцять секунд, вимірювання величини пульсу - один раз в 5 хвилин. Отримані телеметричні дані відсилаються через послідовний передавач. Весь інший час пристрій знаходиться в режимі очікування.

### 3.6.2. Вбудоване програмне забезпечення модуля Child-Device

Програмне забезпечення пристрою Child-Device здійснює контроль за пристроєм і обробку даних. Воно складається з декількох модулів, які присвоєні відповідним функціям, таким як отримання даних від On-Child-Device, обробка отриманих телеметричних даних (стан батареї, обчислення температури і частоти пульсу) і відображення, отримання семплу від звукового аналого-цифрового перетворювача, і обробивши їх передачу даних на Батьківський пристрій.

Послідовний приймач On-Child-Device - Child-Device повністю керований перериваннями. Він не потребує ніяких послідовних запитів. Кінцевий автомат блоку управління приймача зображений на рисунку 3.15.

|     |      |          |        |      |                 |      |
|-----|------|----------|--------|------|-----------------|------|
|     |      |          |        |      | 123.KI(м)-21.12 | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата |                 | 64   |



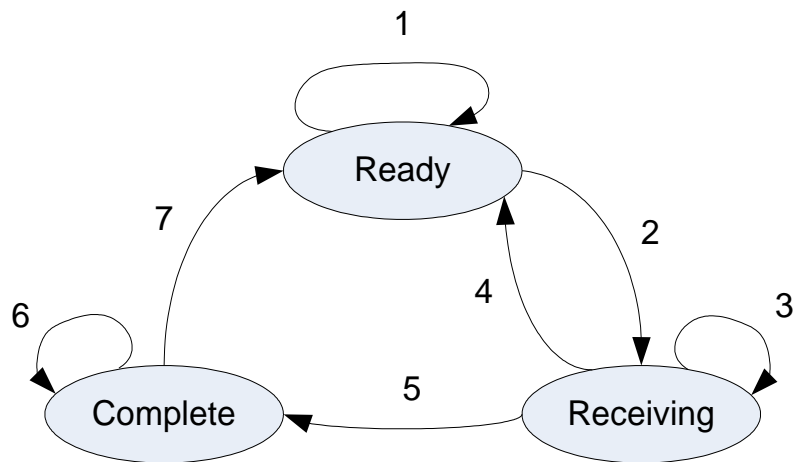


Рис. 3.15 Діаграма станів послідовного приймача.

Спочатку, одержувач знаходиться в стані готовності Ready. Приймач перебуває в цьому стані поки не отримає початковий байт пакету даних(1). Після початку приймання байтів,приймач переходить до стану передачі Receiving(2). Цей стан зберігатиметься, поки не буде отриманий пакет кінця передачі пакетів (3). Якщо це трапляється, приймач переходить в стан завершення Complete (5). Приймач може повернутись до стану готовності (4), якщо відбудеться будь-яка з наступних подій:

- отримано новий стартовий байт .
- байт отриманий з помилкою.
- було отримано більше байтів ніж розмір буферу без кінцевого байту.

Приймач перебуває в стані готовності, поки пакет не буде прочитаний і оброблений (6). Приймач буде скинутий (7), знову перейде в стан Ready, коли основний цикл програми очистить зайнятий прапорець послідовного приймача після вхідної обробки пакету даних.

Як описано вище потік даних між Parent-Device Child-Device складається з двох частин - звуковий аналого-цифровий перетворювач в реальному часі проводить передачу голосу і телеметричної передачі пакетів даних. Блок управління передавача здійснений в обробнику переривань звукового аналого-цифрового перетворювача. Це забезпечує установку біта пакету даних в отримані семпли аналого-цифрового перетворювача.

Після обчислення семпла аналого-цифровим перетворювачем і створення запису стартує блок управління передавача. Він аналізує придатність готових пакету даних. Якщо байти телеметричних даних не вбудовані, то байти голосу передають через радіо-канал. Прапорець готовності каналу передачі даних встановлений. Коли основний цикл програми встановлює прапорець готовності даних, передавач запускає порозрядну передачу закодованого пакету даних в наступних семплах аналого-цифрового перетворювача. Прапорець готовності каналу передачі даних очищений. Після монтування в семпл останнього біта останнього байта пакету блок управління передавача встановлюється в початковий стан. Прапорець каналу передачі даних встановлений знову, і передавач чекає наступного пакету даних.

Загальна потокова обробка здійснена в основному циклі програми. Цей модуль вбудованого програмного забезпечення обробляє пакети даних від послідовного приймача On-Child-Device - Child-Device, нарешті робить телеметричне обчислення параметрів, формує пакети даних для Parent-Device, і забезпечує загальну синхронізацію всіх модулів вбудованого програмного забезпечення.

Розглянемо функціонування основного блоку управління циклу програми (рис. 3.16).

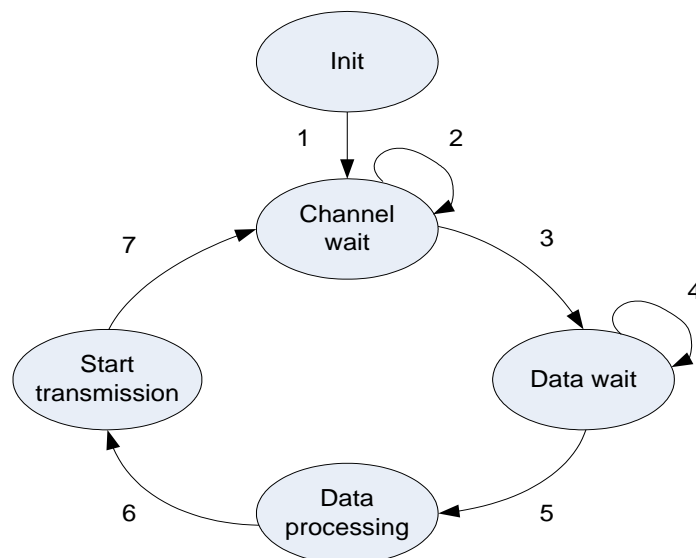


Рис. 3.16. Діаграма основного циклу програми модуля Child device

Спочатку, блок управління знаходиться в стані ініціалізації INIT. Вбудоване програмне забезпечення ініціалізувало всі необхідні змінні і запускає апаратні

модулі. Після того ініціалізується послідовний інтерфейс передавача. Тоді, радіо-передавач перемикається на необхідний канал. Тепер апаратні засоби радіо-каналу готові працювати. Приймач АМ АМ2000 не потребує ніякої ініціалізації програмним забезпеченням. Звуковий аналого-цифровий перетворювач починає перетворювати вхідний аналоговий звуковий сигнал. Отримані семпли передаються через приймач-передавач WirelessUSB. Блок управління переходить в СТАН ОЧІКУВАННЯ КАНАЛУ CHANNEL WAIT (1). З цього моменту модуль Child-Device, готовий до обробки отриманих від On-Child-Device пакетів даних.

СТАН ОЧІКУВАННЯ КАНАЛУ зберігатиметься, поки буфер каналу передачі даних Parent-Device Child-Device не буде порожнім (готовий до передачі) (2). Коли готовий прапорець каналу передачі даних встановлений блоком управління передавача, основний цикл програми переходить в СТАН ОЧІКУВАННЯ ДАНИХ DATA WAIT (3). Основний цикл перебуває в цьому стані поки не буде отриманий пакет даних від On-Child-Device (4).

Якщо прапорець готовності пакету даних встановлений блоком управління приймача, основний цикл програми переходить в стан ОБРОБКИ ДАНИХ DATA PROCESSING (5). Перш за все програмне забезпечення встановлює зайнятий прапорець послідовного приймача і очищує прапорець готовності пакету даних. Таким чином послідовний приймач блокуваний, до того часу поки вихідний пакет даних не готовий до передачі. Переданий пакет декодується, перевіряється контрольна сума і, якщо немає ніяких помилок, аналізуються.

Наступний крок - телеметрична обробка отриманих даних. Під час аналізу пакету визначаються набуті значення температури дитини і результат вимірювання величини пульсу. Значення байта стану On-Child-Device дає інформацію про розміщення пристрою, стан батареї і телеметричної законності даних.

Для обчислення температури дитини використовується права сторона рівняння 3.4. За цим значенням визначається відповідне співставлене йому температурне значення з таблиці пошуку. Згідно аналізу байта стану вся отримана інформація, зображає для батьків на LCD і світлодіодах.

|     |      |          |        |      |  |                 |      |
|-----|------|----------|--------|------|--|-----------------|------|
|     |      |          |        |      |  | 123.KI(м)-21.12 | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата |  |                 | 67   |

Після цього обчислені значення температури дитини і частоти пульсу запаковуються в пакет для відправлення . Після виконання кодування пакет готовий до передачі. Основний цикл програми переходить в стан початку передачі START TRANSMISSION (6). Зі встановленням прапорця готовності передачі пакету даних, починається передача пакетів. Зайнятий прапорець приймача очищений, і цикл обробки даних закінчений. Послідовний приймач дає можливість, переходити в стан готово READY. Після цього основний блок управління циклу програми скинутий в reset (7).

### 3.6.3. Вбудоване програмне забезпечення модуля Parent-Device

Вбудоване програмне забезпечення Parent-Device здійснює обробку даних і загальне управління пристроєм. Воно складається з декількох модулів, які присвоєнні відповідним функціям, таким як отримання байтового потоку сигналу, цифрового звукозапису від Child-Device, витягання телеметричних даних, отримання, обробку телеметричних даних і їх відображення

Для прийому байтового потоку цифрового аудіо сигналу використовується послідовний приймач Child-Device- Parent-Device. Цей приймач повністю керований перериванням. Він не потребує ніякого послідовного опитування. Діаграми станів блоку управління приймача показані на рисунку 3.17.

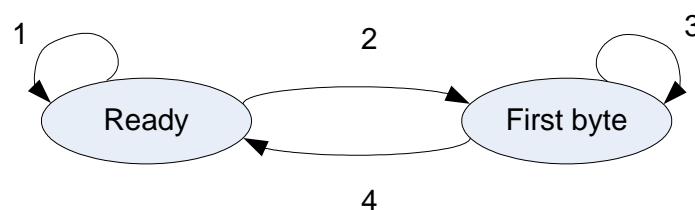


Рис. 3.17. Діаграма станів послідовного приймача

Представимо роботу приймача з потоком байтів показаному на рисунку 3.18.

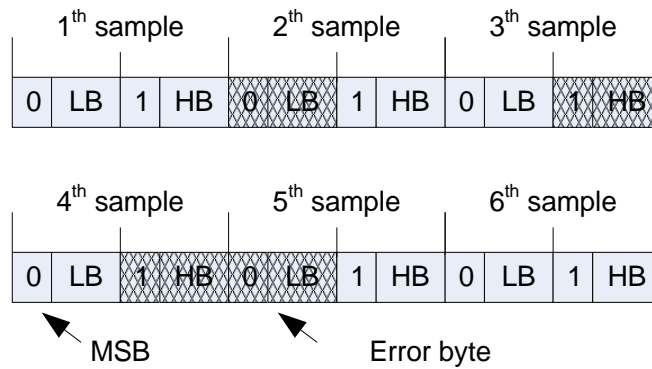


Рис. 3.18. Тестова передача байта послідовним приймачем

Спочатку, приймач знаходиться в стані готовності. Коли молодший байт першого семпла (LB) отриманий без помилок, кінцевий автомат переходить в стан First byte (2). Байт звукового семпла збережений в звуковому буфері. Після прийому коректного старшого байта (HB) автомат переходить в стан готово READY (4). Повний звуковий семпл після форматування використовується для звукової установки PWM.

Якщо молодший байт другого семпла отриманий з помилками, цей байт проігнорований і приймач перебуває в тому ж самому стані надалі (1). Звуковий семпл проігнорований в цьому випадку. Стан поточного блоку управління зберігається (1).

У третьому випадку молодший байт семпла отриманий без помилок, і блок управління знаходиться в стані першого байту FIRST BYTE. Якщо старший байт отриманий з помилками, цей байт проігнорований і приймач перебуває в тому ж самому стані (3). Коли молодший байт наступного семпла отриманий без помилок, приймач рухається в стан готовності READY.

Дві або більше послідовних втрати байтів мають як результат помилку отримання пакету даних. Кінцевий автомат повторно синхронізований на наступному правильному отриманому байті. Рисунок 3.23 зображає алгоритм приймача.

Обробка отриманих пакетів даних здійснюється в основному циклі програми. Ці програмні модулі опрацьовують пакети даних з Child-Device до Parent-Device приймача, виконують завершальну візуалізацію телеметричних даних і

забезпечують загальну синхронізацію всіх модулів вбудованого програмного забезпечення.

Алгоритм основного циклу програми показаний на рисунку 3.25. Розглянемо як він функціонує детальніше .

Перед початком основного циклу програми вбудоване програмне забезпечення ініціалізує всі змінні і запускає апаратні модулі. На початку кожного етапу рівень RSSI порівнюється із зумовленим пороговим рівнем. Якщо рівень сили отриманого сигналу нижче ніж необхідний, модуль послідовного приймача заблокований. Функціонування послідовного приймача відновлене, коли рівень RSSI буде відповідним для безпомилкового отримання даних.

Якщо рівень RSSI - нормальний, аналізується прапорець готовності пакету даних. Якщо цей прапорець встановлений тоді - в буфері приймача з вбудовуванням буде міститись новий пакет телеметричних даних. Основний цикл програми встановлює зайнятий прапорець приймача і починає обробку. Перш за все пакет декодується, і перевіряється цілісність пакету. Для цього обчислюється контрольна сума пакету і порівнюється з полем контрольної суми пакету. Якщо ці суми рівні то пакет аналізується, в іншому випадку він буде проігнорований. Якщо адреса доставки пакету не співпадає з адресою власного Parent-Device, отриманий пакет також буде проігнорований.

Під час аналізу пакету отримуються набуті значення температури дитини і частоти пульсу. Значення байта стану On-Child-Device дає інформацію про розміщення пристрою, стан батареї і законності телеметричних даних. Згідно аналізу байта стану вся інформація, зображається для батьків на LCD і світлодіодах.

Після закінчення обробки пакету даних зайнятий прапорець приймача очищується, і головний цикл програми чекає наступного пакету.

|     |      |          |        |      |                 |      |
|-----|------|----------|--------|------|-----------------|------|
|     |      |          |        |      | 123.KI(м)-21.12 | Арк. |
|     |      |          |        |      |                 | 70   |
| Зм. | Арк. | № докум. | Підпис | Дата |                 |      |

### 3.7. Блок-схеми алгоритмів

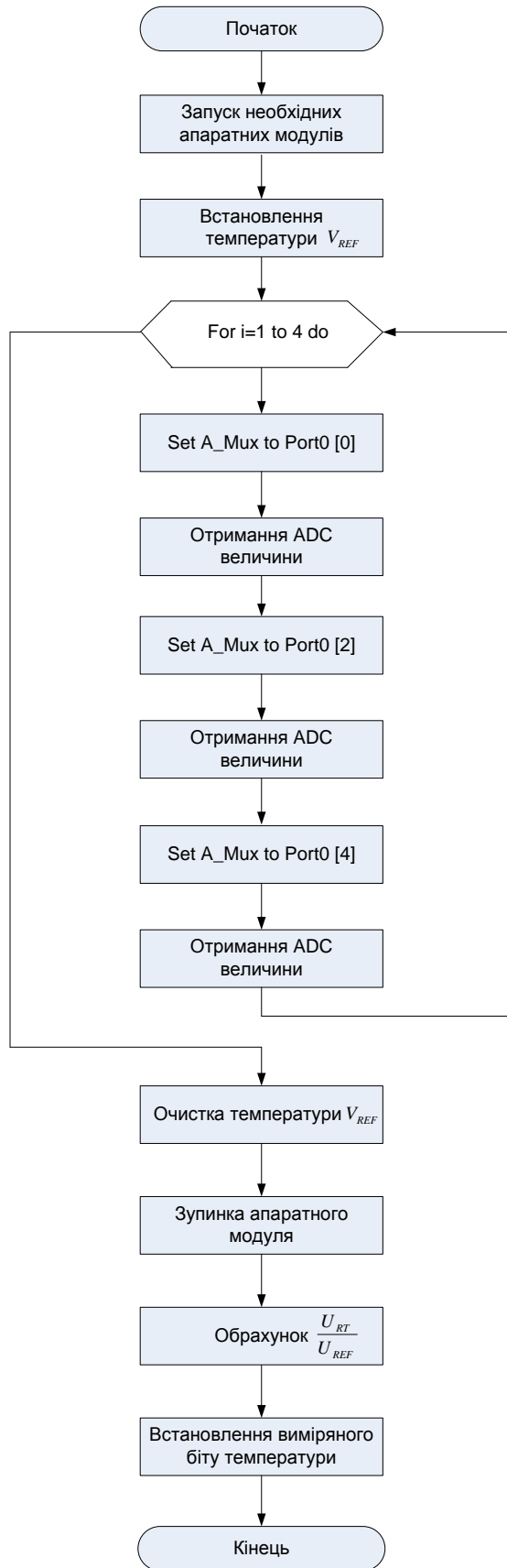


Рис. 3.19. Блок-схема алгоритму вимірювання температури.

| Зм. | Арк. | № докум. | Підпис | Дата |
|-----|------|----------|--------|------|
|     |      |          |        |      |

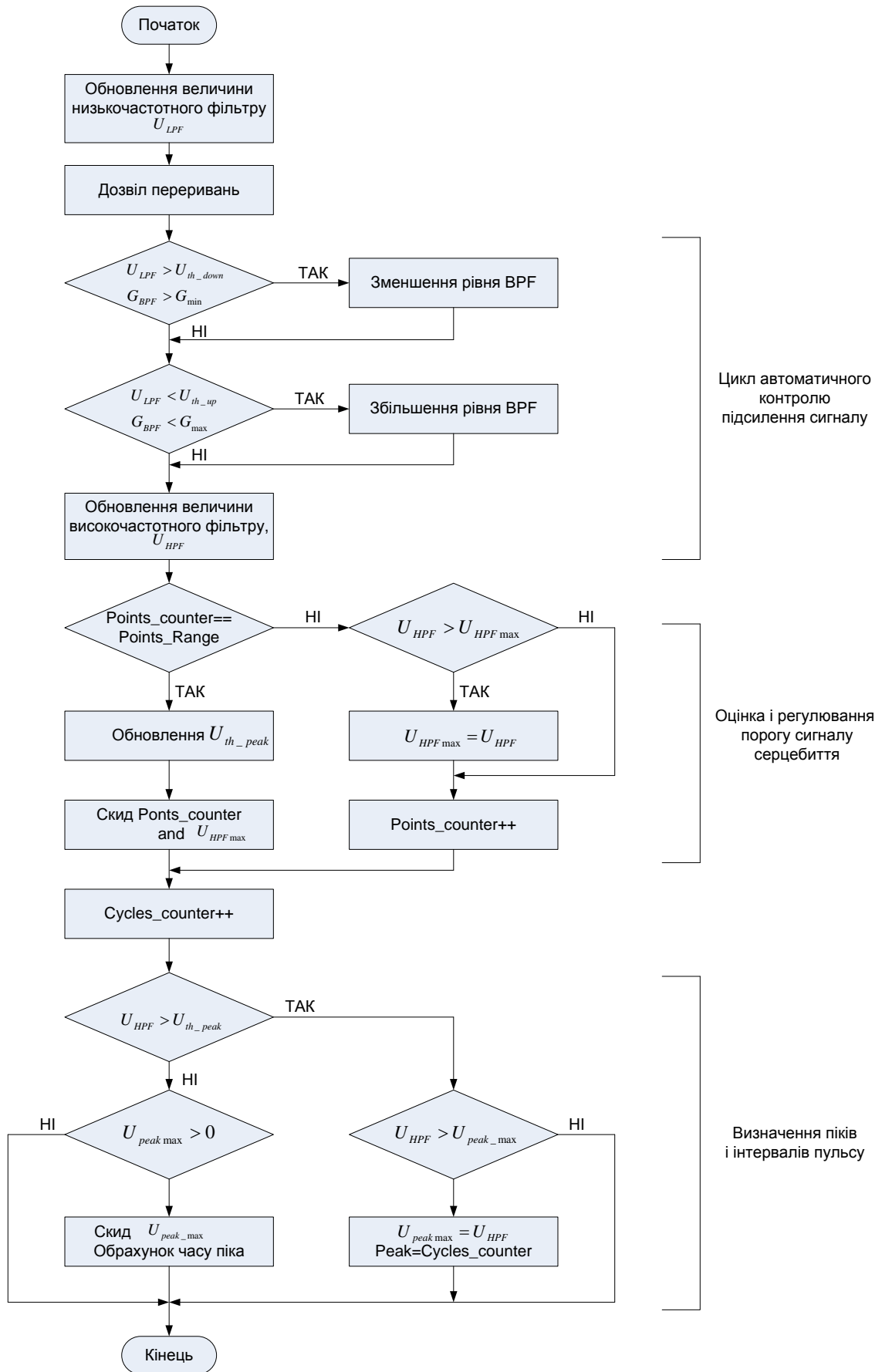
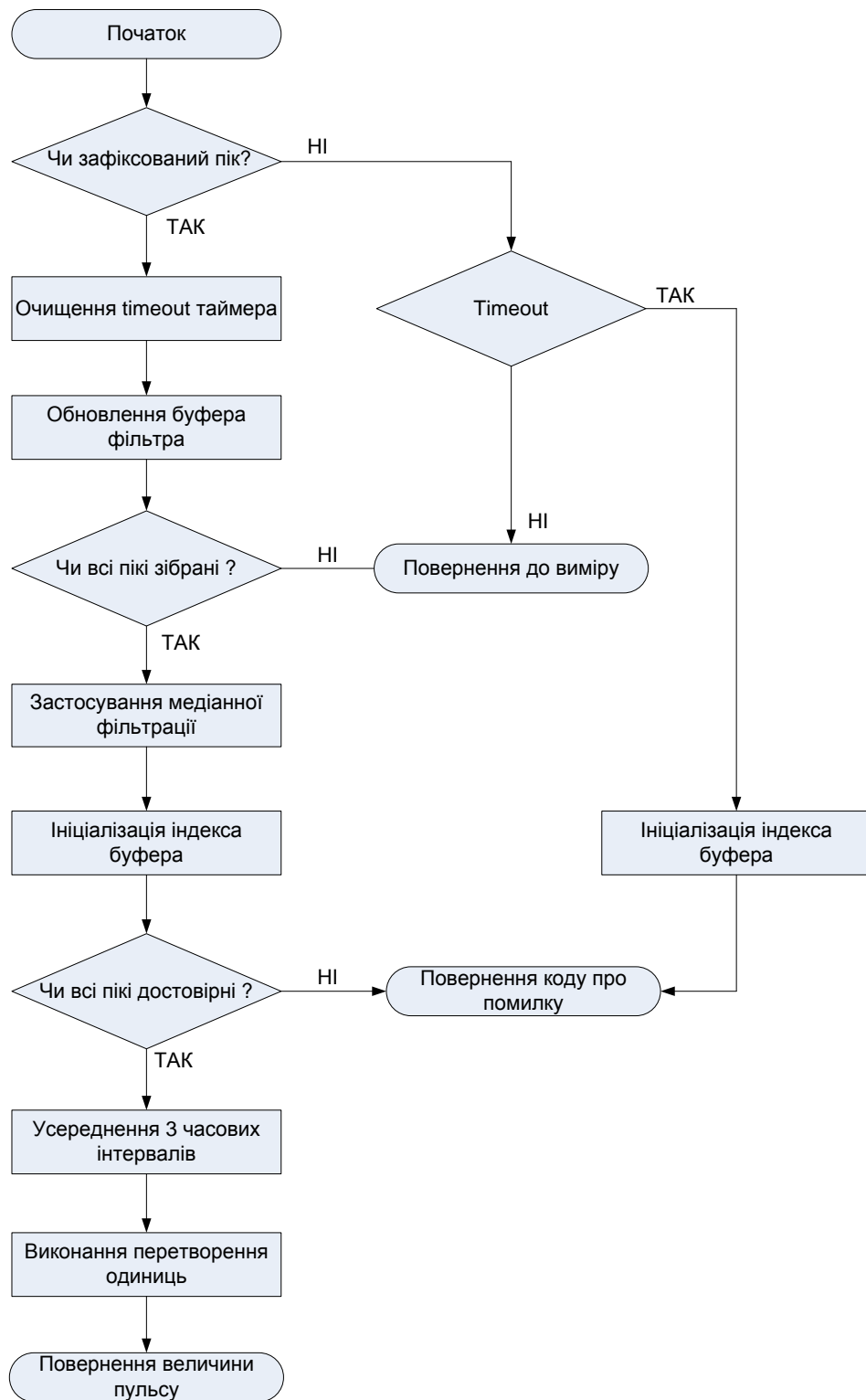


Рис. 3.20. Блок-схема алгоритму обробки сигналу пульсометра.

|     |      |          |        |      |
|-----|------|----------|--------|------|
|     |      |          |        |      |
| Зм. | Арк. | № докум. | Підпис | Дата |





-Рис. 3.21. Блок-схема алгоритму обчислення величини пульсу

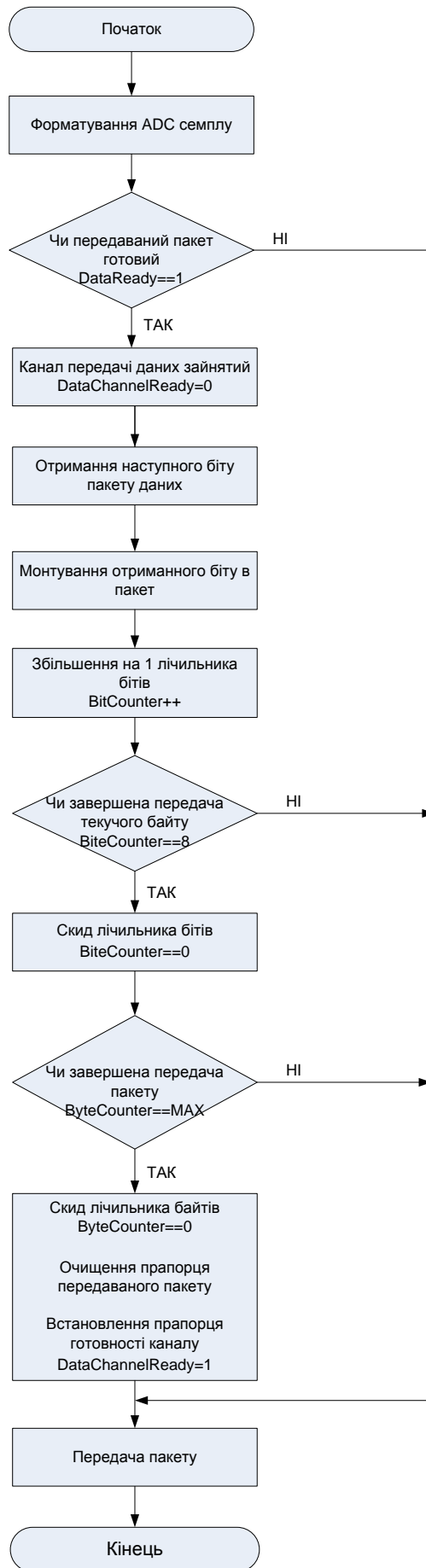


Рис. 3.22. Блок-схема алгоритму послідовного передавача.

|     |      |          |        |      |
|-----|------|----------|--------|------|
|     |      |          |        |      |
| Зм. | Арк. | № докум. | Підпис | Дата |

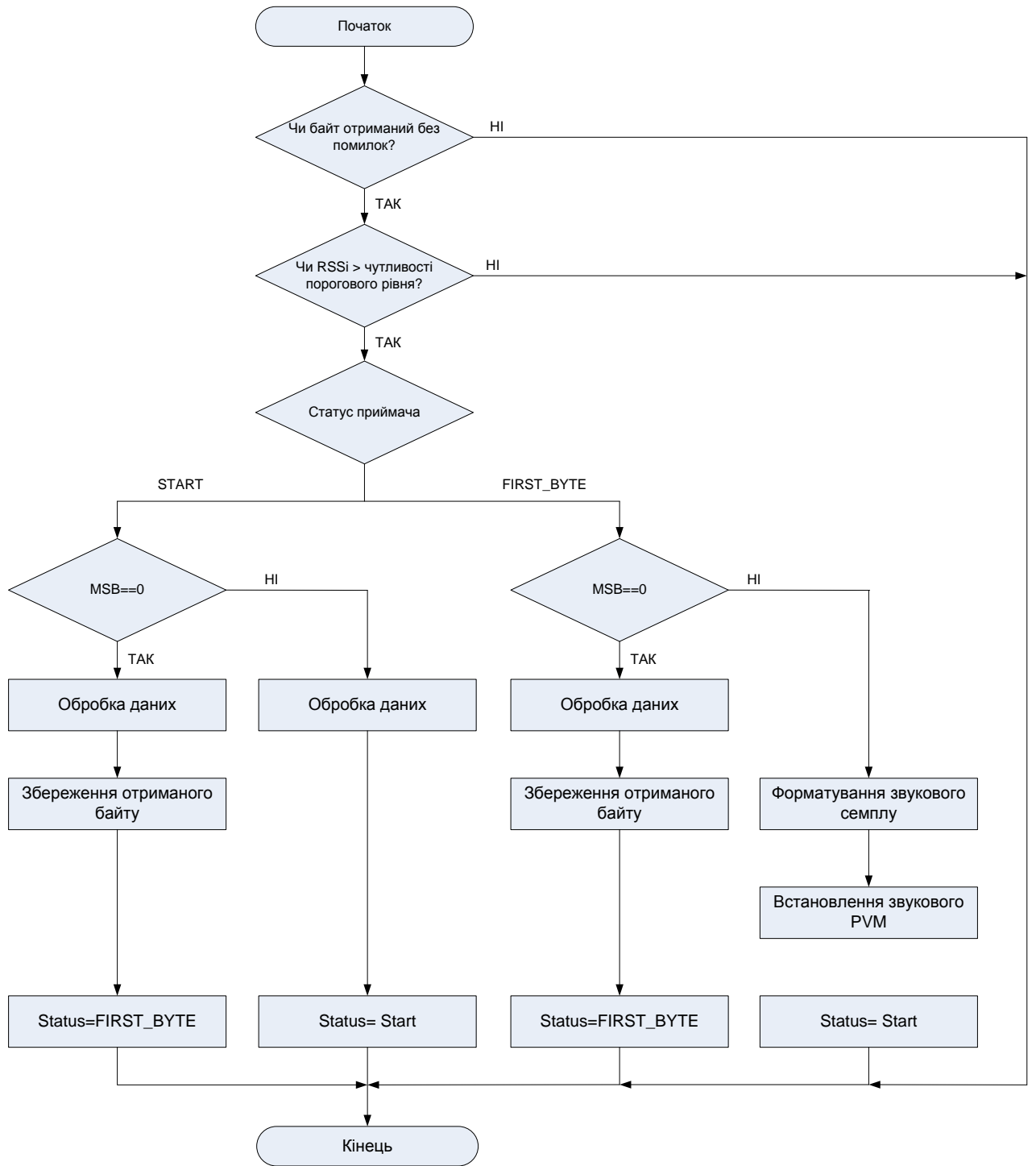


Рис. 3.23. Блок-схема алгоритму послідовного приймача.

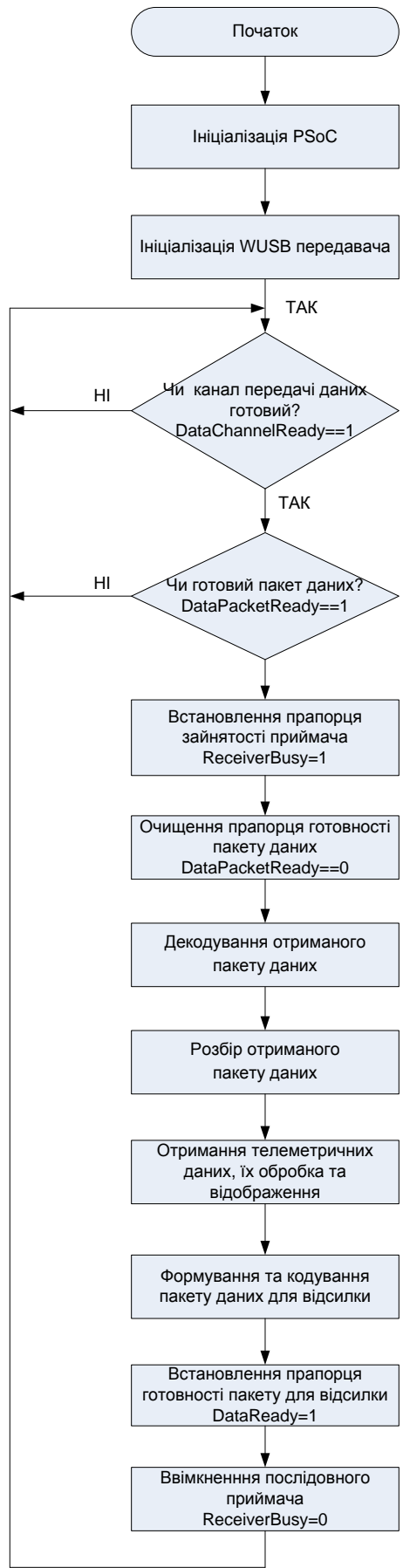


Рис. 3.24. Блок-схема алгоритму основного циклу програми модуля Child-Device

|     |      |          |        |      |
|-----|------|----------|--------|------|
|     |      |          |        |      |
| Зм. | Арк. | № докум. | Підпис | Дата |

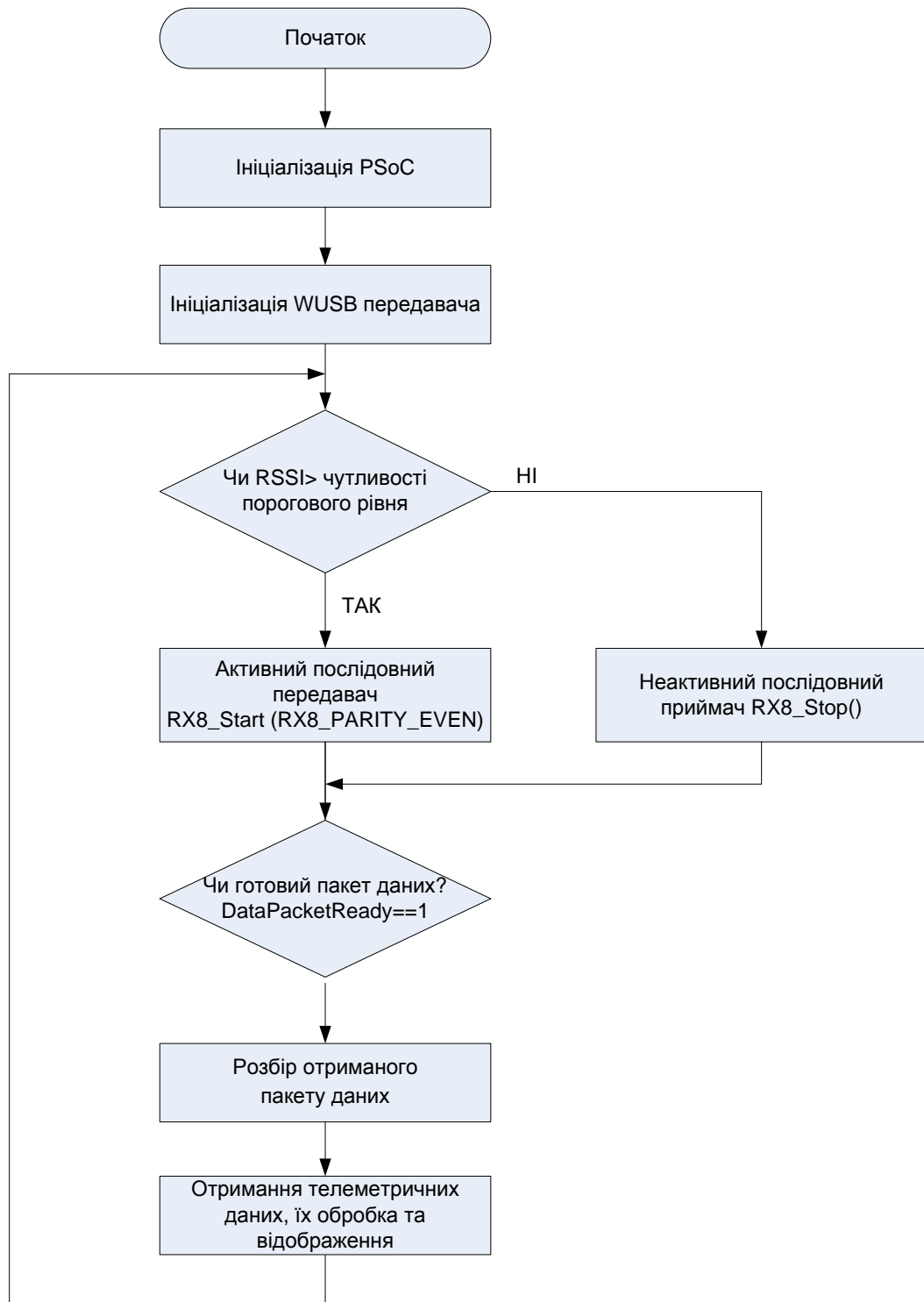


Рис. 3.25. Блок-схема алгоритму основного циклу програми модуля Parent-Device

|     |      |          |        |      |
|-----|------|----------|--------|------|
|     |      |          |        |      |
| Зм. | Арк. | № докум. | Підпис | Дата |

## 4. КОНСТРУКТОРСЬКИЙ РОЗДІЛ

Пристрій розроблено у вигляді 3-ох окремих плат:

- On-Child-device який живиться від двох батарей ААА, розташований в пластмасовому корпусі, який на резинці одівається на руку дитині.
- Child-Device, розташований в кімнаті з дитиною. Живиться від постійного джерела живлення напругою 9В.
- Parent device, розташований в кімнаті з батьками. Живиться від постійного джерела живлення напругою 9В.

При проектуванні було використано PSoC Designer 4.4 і програматор фірми Cypress.

Для перевірки роботи пристрою і вимірів споживаної потужності було змонтований тестовий стенд.

Параметри розводки проекрованої плати :

- крок сітки дюймовий і рівний 0,1";
- діаметр перехідних отворів рівний 0,8 мм;
- товщина доріжок рівна 0,4 мм;
- товщина доріжок VCC/GND рівна 1 мм.

### 4.1. Принципова схема модуля On-Child-device

Схема вимірювача температури складається з дільника опорів R1R2R3 із під'єднаним термістором J2. R2 – прецизійний резистор з похибкою вимірювання 0.1 %.

Резистор R4 і конденсатори C1C4 забезпечує підключення передавача - LQ-TX433 і реалізацію антени циклу PCB.

Пристрій живиться від однієї літєвої батареї. Для мінімізації споживаної потужності напруга живлення подається до портів вимірювання температури і пульсу тільки під час процесів вимірювання через порт PSoC P0 [6].

Вимірювач пульсу реалізований на інфрачервоному світлодіоді і фотодіоді.

|     |      |          |        |      |                 |      |
|-----|------|----------|--------|------|-----------------|------|
|     |      |          |        |      | 123.KI(м)-21.12 | Арк. |
|     |      |          |        |      |                 | 78   |
| Зм. | Арк. | № докум. | Підпис | Дата |                 |      |

D2- інфрачервоний світлодіод який спалахує кожного разу, коли удар виявлений.

Q1 - конвертер напруги в струм, який формує рівень зміщення DC для фотодіода D1. Генератор зміщення має низький імпеданс для постійної несучої або низькочастотних сигналів, і подавляє шумові сигнали, викликані різними зовнішніми джерелами світла. Для сигналів частотної модуляції імпеданс визначений перш за все R2. PGA підсилює вихід FED відносно PSoC AGND. Коли напруга FED іде до зменшення, це зменшує напругу на базі Q1 і зменшує струм до FED зміни його чутливості.

Генератор зміщення дозволяє скоротити до необхідного корисного рівня сигнал фотодіода, зменшивши FED несучу, і в кінці зменшити шумовий рівень у вихідному сигналі.

#### 4.2. Принципова схема модуля Child-Device

J4 це 16x1 конектор для радіо-приймача LM-RXAM2433. Для цього приймача використовується антена циклу PCB, підключена до пінів 1 і 2. Параметри такої реалізації антени описані детальніше в таблиці даних отримувача.

Для перетворення звукового сигналу в електричний використовується простий універсальний мікрофон. Отриманий сигнал підсилюється перепідсилювачем напруги мікрофона із змінним стисненням і автоматичним контролем підсилюваного сигналу U1 - SSM2167. Посилений сигнал поданий до КОРПУСУ PGA PSoC через порт P0 [1] для наступного аналого-цифрового перетворення.

Дільник опорів R1R3 з конденсаторами фільтрації C2C3 формує зовнішній рівень AGND. У цьому проекті дешевий 8x1 використовується дешевий символічний LCD. Для зміни контрасту LCD використовується підстроювальний резистор R12. Як інформаційні засоби відображення використовуються два світлодіоди D1D2.

Лінійний регулятор U4 забезпечує стабільних 3.3V живлення для компонентів всього пристрою за винятком LCD який потребує джерела живлення 5 V. Лі-

|     |      |          |        |      |                 |  |  |  |      |
|-----|------|----------|--------|------|-----------------|--|--|--|------|
|     |      |          |        |      |                 |  |  |  | Арк. |
|     |      |          |        |      |                 |  |  |  | 79   |
| Зм. | Арк. | № докум. | Підпис | Дата | 123.KI(м)-21.12 |  |  |  |      |

нійний регулятор U3 забезпечує окреме джерело живлення для цих модулів. Як зовнішнє стабілізоване джерело живлення для цього пристрою необхідно 9V.

### 4.3. Принципова схема модуля Parent device

Декодер даних складається з пікового датчика і компаратора. Оскільки швидкість бітової передачі інформації рівна 250 кбіт/сек, внутрішній пороговий компаратор PSoC не може використовуватися в цьому випадку. Для отримання сигналу даних, що декодує зовнішній компаратор MAX961 використовується U1. Це - малопотужний компаратор єдиного постачання від шини до шини з 40 ns затримкою розповсюдження. Піковий датчик спроектований на двох діодах Шоткі D1 і двох конденсаторах C1C2 для позитивного і негативного виявлення рівня і двох резисторів R1R4 аналогового суматора. Вихідний сигнал цього суматора дає середній рівень аналогового сигналу даних. Цей рівень використовується як пороговий рівень компаратора. Вихідний сигнал компаратора поданий в RX модуль PSoC через порт P0 [6] як декодований сигнал даних.

Вихідний сигнал звукового PWM від порту P0 [5] відсилають послідовно в фільтр по нижчому пріоритету LPF, спроектованому на резисторі R11 і конденсаторі C8, і фільтр по вищому пріоритету HPF, що спроектований на конденсаторі C7 і на резисторі R13. Після цього фільтру звуковий сигнал, підсилюється звуковим підсилювачем U3. В якості цього підсилювача використовується підсилювач Philips Semiconductors TDA7052A. Це 1 Wat BTL моно звуковий підсилювач з регулюванням гучності DC. Змінний резистор R14 регулює силу вихідного аудіо сигналу.

У цьому проекті використовується дешеве розміром 8x1 символне LCD . Для зміни контрасту LCD використовується змінний резистор R12. Як додаткові інформаційні засоби для відображення використовуються три світлодіоди D2-D4.

Лінійний регулятор U4 забезпечує 3.3V живлення для компонентів всього пристрою за винятком LCD який потребує джерела живлення 5 V. Лінійний регулятор U3 забезпечує окреме джерело живлення для цих модулів. Як зовнішнє стабілізоване джерело живлення для цього пристрою необхідно 9V.

|     |      |          |        |      |  |                 |      |
|-----|------|----------|--------|------|--|-----------------|------|
|     |      |          |        |      |  | 123.KI(M)-21.12 | Арк. |
|     |      |          |        |      |  |                 | 80   |
| Зм. | Арк. | № докум. | Підпис | Дата |  |                 |      |



## ВИСНОВКИ

Результатом проектування є розроблена система дистанційного моніторингу стану немовлят, яка працює в режимі реального часу. Система завадостійка, мобільна, компактна, з низьким споживанням енергії.

|     |      |          |        |      |                 |      |
|-----|------|----------|--------|------|-----------------|------|
|     |      |          |        |      | 123.KI(м)-21.12 | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата |                 | 81   |

Система побудована на PsoC мікроконтролерах фірми Cypress: CY8C24223 - на якому реалізовано модуль, який розташований на тілі дитини, CY8C29666 і CY8C29466 - для модулів Child-Device та Parent-Device відповідно, і CYRF6936 – який забезпечує завадостійку безпроводну передачу даних між цими модулями. Розроблено спеціалізовану структуру пакету, який відповідає поставленим умовам, зокрема забезпечує підтвердження отримання інформації, перевіряє її достовірність, ідентифікує вміст пакету.

Розроблено алгоритм вимірювання температури, який побудований на термісторі, та алгоритм вимірювання пульсу, побудований на оптичному методі індикації поглинання рівня людської тканини, під час заповнення кровоносного капіляра.

Реалізовано передачу голосового потоку по завадостійкому радіо-каналі, побудованому на устаткуванні Wireless USB LP, з можливістю пересинхронізації у випадку втрати сигналу.

Проведене економічне обґрунтування проектного рішення. Ціна та собівартість розробленого пристрою менші від розглянутого аналога, функціональність та комплексний показник якості вищі.

Описані заходи по охороні праці для приміщень, де використовуються комп'ютери. Також розглянуто правила роботи з електроінструментом.

Таким чином, за результатами комплексної оцінки технічного рівня і якості спроектованого пристрою, можна зробити висновок, що спроектований пристрій, в порівнянні з вибраним аналогом, має відчутні переваги, відносно економічного ефекту з точки зору виробництва і умов експлуатації.

|     |      |          |        |      |                 |      |
|-----|------|----------|--------|------|-----------------|------|
|     |      |          |        |      | 123.KI(м)-21.12 | Арк. |
|     |      |          |        |      |                 | 82   |
| Зм. | Арк. | № докум. | Підпис | Дата |                 |      |

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. <http://www.cypress.com> – офіційний сайт фірми Cypress.
2. <http://www.datasheet4.com> - спеціалізований сайт, який містить документації до мікроконтролерів.
3. <http://www.wireless.ru> – спеціалізований портал присвячений безпроводним технологіям.
4. <http://www.gav.ru>- сайт який містить багато інформації про мікроконтролери.
5. <http://www.compitech.ru> - сайт який містить багато інформації про мікроконтролери.
6. <http://www.atmel.com> - офіційний сайт фірми Atmel.
7. <http://www.freescale.com> - офіційний сайт фірми Freescale Semiconductor.
8. <http://www.microchip.com> – багато інформації по мікропроцесорам, документації і т.д.
9. <http://www.psocdeveloper.com> – сайт присвячений розробці систем на кристалі.
10. Немудров В., Мартин Г. “Проектирование систем на кристалле” видавництво “Техносфера”, 2004р.
11. Крайник О.П., Барвінська Є.С. Економіка підприємства. Львів: Національний ун-т "Львівська політехніка", 2003 - 208 с

|     |      |          |        |      |  |                        |      |
|-----|------|----------|--------|------|--|------------------------|------|
|     |      |          |        |      |  | <i>123.КІ(м)-21.12</i> | Арк. |
|     |      |          |        |      |  |                        | 83   |
| Зм. | Арк. | № докум. | Підпис | Дата |  |                        |      |

## Додаток А. Тексти програм.

```
//-----  
// C main line OnChildDevice  
//-----  
  
#include <m8c.h> // part specific constants and macros  
#include "PSoCAPI.h" // PSoC API definitions for all User Modules  
  
#include "data_exchange.h"  
#include "telemetric.h"  
#include "utils.h"  
#include "globalparams.h"  
  
#define STATUS_DEVICE_ERROR ( BIT(BATTERY_LOW_BIT) )  
static sleep_cnt = 0 ;  
  
void main()  
{  
    CLEARBIT (TEMPER_PORT, TEMPER_BIT) ;  
  
    glob.myAddress = 7 ;  
    glob.destAddress = 7 ;  
  
    M8C_EnableGInt ;  
    M8C_EnableIntMask(INT_MSK0, INT_MSK0_SLEEP) ;  
  
    while (1)  
    {  
        ARF_CR &= ~0x07 ; //turn off analog subsystem  
  
        M8C_ClearWDTAndSleep ;  
        INT_VC = 0 ;  
        M8C_Sleep;  
  
        sleep_cnt++ ;  
  
        if (TEMP_MEASURE_MASK == (sleep_cnt & TEMP_MEASURE_MASK))  
        {  
            ARF_CR |= ANALOG_POWER ;  
  
            CheckBatteryStatus () ;  
  
            if (0 == (status_byte & STATUS_DEVICE_ERROR))  
            {  
                TemperMeasure() ;  
            }  
  
            MakeAndSendDataPacket(glob.destAddress) ;  
            status_byte = 0 ;  
        }  
    }  
}
```

|     |      |          |        |      |  |  |  |  |  |                 |      |
|-----|------|----------|--------|------|--|--|--|--|--|-----------------|------|
|     |      |          |        |      |  |  |  |  |  | 123.KI(М)-21.12 | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата |  |  |  |  |  |                 | 84   |

```

}

}

/**-----
/** File Name      : data_exchange.c
/** Object         : Data exchange protocol functions library.
/** Creation       : Savchyn V (C) 2007
/**-----

#include "PSoCAPI.h" // PSoC API definitions for all User Modules
#include "data_exchange.h"
#include "data_exchange_func.h"
#include "telemetric.h"

// packet storage for transmit packets
T_PACKET txPacket ;

// buffer for holding post-encoded frames that are ready for transmission
BYTE txBuffer[ MAX_TX_BUFFER_SIZE ] ;

BYTE checksum = 0 ;
BYTE index = 0 ;

volatile T_GLOB glob ;

/**-----
/** Function Name   : MakeAndSendDataPacket
/** Object         : Creates an DATA packet from the information passed in,
/**               : and transmit the Packet.
/** Input Parameters : <destAddr> = the destination address for the packet
/** Output Parameters : none
/** Functions called : none
/** Creation       : Savchyn V (C) 2007
/**-----

void MakeAndSendDataPacket( BYTE destAddr )
{
    BYTE length, t = 2 ;
    // initialize checksum
    checksum = 0 ;

    M8C_DisableGInt ;

    txPacket.payload[0] = (BYTE)(TRES) ;
    txPacket.payload[1] = (BYTE)( (WORD)(TRES) >> 8) ;

    for ( index = 0; index < 7; index++ )
    {
        txPacket.payload[t++] = 0 ;
    }
}

```

|     |      |          |        |      |                 |      |
|-----|------|----------|--------|------|-----------------|------|
|     |      |          |        |      | 123.KI(М)-21.12 | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата |                 | 85   |

```

        txPacket.payload[t++] = 0 ;
    }
    txPacket.payload[t] = status_byte ;

M8C_EnableGInt ;

    // fill in packet

    checksum += ( txPacket.sourceAddr = glob.myAddress ) ;
    checksum += ( txPacket.destAddr = destAddr ) ;

    for( index = 0; index < MAX_TX_PAYLOAD; index++ )
    {
        checksum += txPacket.payload[index] ;
    }

    // fill in checksum
    txPacket.checksum = checksum ;

txBuffer[0] = TX_START_OF_PACKET ;

// encode the packet
    length = bManchesterEncode( sizeof( txPacket ) );

    txBuffer[MAX_TX_BUFFER_SIZE - 1] = TX_END_OF_PACKET ;

    // start TX8
    Counter_Start () ;
    TX8_Start ( TX8_PARITY_EVEN ) ;

    for( index = 0; index < 5; index++ )
    {
TX8_TX_BUFFER_REG = PML ;
        /* Wait for the data to start transmitting */
        while( !( TX8_CONTROL_REG & TX8_TX_COMPLETE ) );
    }

    for( index = 0; index < (length + 2); index++ )
    {
        TX8_TX_BUFFER_REG = txBuffer[index] ;
        /* Wait for the data to start transmitting */
        while( !( TX8_CONTROL_REG & TX8_TX_COMPLETE ) );
    }

    for( index = 0; index < 5; index++ )
    {
        TX8_TX_BUFFER_REG = PML ;
        /* Wait for the data to start transmitting */
        while( !( TX8_CONTROL_REG & TX8_TX_COMPLETE ) );
    }

// stop TX8

```

|     |      |          |        |      |  |  |  |  |                 |      |
|-----|------|----------|--------|------|--|--|--|--|-----------------|------|
|     |      |          |        |      |  |  |  |  | 123.KI(М)-21.12 | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата |  |  |  |  |                 | 86   |

```

TX8_Stop () ;
    Counter_Stop () ;

}

/**-----
/** File Name      : telemetric.c
/** Object        : Telemetric data procesing functions library.
/** Creation      : Savchyn V (C) 2007
/**-----

#include "PSoC_API.h" // PSoC API definitions for all User Modules
#include "telemetric.h"
#include "nmath.h"
#include "data_exchange.h"
#include "utils.h"
#include "globalparams.h"

LONG TRES = 0, Urt = 0, Urref = 0 ;
BYTE status_byte = 0 ;

INT get_adc_sample(void)
{
    ADC_GetSamples(1) ;
    while(ADC_fIsDataAvailable() == 0) ; // Loop until value ready
    return ADC_iClearFlagGetData() ;
}

/**-----
/** Function Name   : TemperMeasure
/** Object         : Obtain a temperature measuring data and calculate
/**               : its value
/** Input Parameters : none
/** Output Parameters : none
/** Functions called : get_adc_sample
/** Creation       : Savchyn V (C) 2007
/**-----
void TemperMeasure ()
{
    BYTE i = 0 ;
    INT U1 = 0, U2 = 0, U3 = 0 ;

    ADC_AtoDcr1 &= 0x1F ;

    PGA_Temper_Start ( PGA_Temper_MEDPOWER ) ;
    ADC_Start(ADC_MEDPOWER) ;
    ADC_fClearFlag() ;
    SETBIT (TEMPER_PORT, TEMPER_BIT) ;

    for ( i = 0 ; i < 4 ; i++)

```

|     |      |          |        |      |
|-----|------|----------|--------|------|
|     |      |          |        |      |
| Зм. | Арк. | № докум. | Підпис | Дата |

```

    {
        //select Port0[0]
        AMUX_T_InputSelect( AMUX_T_PORT0_0 );
        U1 += get_adc_sample() ;

        //select Port0[2]
        AMUX_T_InputSelect( AMUX_T_PORT0_2 );
        U2 += get_adc_sample() ;

        //select Port0[4]
        AMUX_T_InputSelect( AMUX_T_PORT0_4 );
        U3 += get_adc_sample() ;
    }

    CLEARBIT (TEMPER_PORT, TEMPER_BIT) ;
    ADC_Stop() ;
    PGA_Temper_Stop () ;

    Urt = (U2 - U1) ;
    Urt <<= 16 ;
    Urref = U3 - U2 ;

    TRES = DIV32s_32s_32s (Urt, Urref) ;
    TRES = TRES - 62042 ;

    if ( TRES > 39363 ) TRES = 50000 ;

    if ( (TRES & 0x80000000) == 0x80000000 ) TRES = 50001 ;

    SETBIT(status_byte, TEMP_MEASURED_BIT) ;
}

/**-----
/** Function Name      : CheckBatteryStatus
/** Object             : Check battery voltage and set correspondent bit
/**                   : in status byte
/** Input Parameters   : none
/** Output Parameters  : none
/** Functions called   : get_adc_sample
/** Creation           : Savchyn V (C) 2007
/**-----
void CheckBatteryStatus (void)
{
    if (VLT_CMP & 0x02) SETBIT(status_byte, BATTERY_LOW_BIT) ;
}

```

|            |             |                 |               |             |                        |             |
|------------|-------------|-----------------|---------------|-------------|------------------------|-------------|
|            |             |                 |               |             | <i>123.KI(М)-21.12</i> | <i>Арк.</i> |
| <i>Зм.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> |                        | 88          |





```

ADC_AUDIO_StartAD ();

    // init TX8
TX8_Start ( TX8_PARITY_EVEN );

    // init RX8
Counter16_Start ();
RX8_AM_EnableInt ();
RX8_AM_Start ( RX8_AM_PARITY_EVEN );

M8C_EnableGInt ;

// init RF
RadioInit ();

    // go to Tx mode
ClearRXON ();

    DelayCycles(ui16RxonToTxCycles[ui8TransceiverType]);
}

void main()
{
    OSC_CR2 |= 0x80;
    ECO_TR = 0x3F;

    PSoCInit ();
    DataChannelReady = 1 ;

    while (1)
    {
        if ( DataChannelReady == 1 )
        {
            if ( DataPacket_Ready == 1 )
            {
                Receiver_Busy = 1 ;
                DataPacket_Ready = 0 ;
                DecodePacket ();
                MakeAndSendDataPacket ( BROADCAST_ADDR );
                Receiver_Busy = 0 ;
            }
        }
    }
}

```

|            |             |                 |               |             |  |                        |             |
|------------|-------------|-----------------|---------------|-------------|--|------------------------|-------------|
|            |             |                 |               |             |  | <i>123.KI(M)-21.12</i> | <i>Арк.</i> |
| <i>Зм.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> |  |                        | 90          |

|            |             |                 |               |             |                        |             |
|------------|-------------|-----------------|---------------|-------------|------------------------|-------------|
|            |             |                 |               |             | <i>123.КІ(м)-21.12</i> | <i>Арк.</i> |
| <i>Зм.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> |                        | 91          |