**1. Cấu hình hoạt động cho AFE4420**

**1.1. Thiết lập độ lợi cho bộ khuếch đại TIA**

Thiết lập độ lợi cho bộ TIA của AFE4420, bảng dưới đây cho thấy các mức độ lợi có sẵn của AFE4420.

A picture containing text, receipt, number, screenshot

Description automatically generated

Các mức độ lợi từ mức 10 KOhm cho đến 2 MOhm, chọn độ lợi phù hợp giúp khuếch đại tín hiệu thu được từ photodiode để dễ dàng xử lý sau này khi thực hiện chuyển đổi ADC. Trong hệ thống đo đạc hiện tại mức độ lợi được chọn là **250 KOhm**, nằm ở khoảng giữa các mức, do đã được kiểm tra cùng các mức độ lợi khác cho thấy mức này là ổn định cho tín hiệu thu thập được.

**1.2. Cấu hình mặc định cho hoạt động của AFE4420**

Bước thiết lập này sẽ xây dựng các cấu hình ban đầu cho AFE để hoạt động, nó bao gồm các cấu hình về cách dùng xung đồng hồ của hệ thống, các vấn đề về thời gian cho các hoạt động trong một pha tín hiệu như là thời gian sáng của led, thời gian lấy mẫu trong một pha, hay băng thông của lọc RC.

Bảng dưới đây biểu diễn các cấu hình ban đầu quan trọng được thiết lập trong AFE.

(Xây dựng bản từ afe4420\_default\_cfg dòng 51 trong afe4420.c)

|  |  |
| --- | --- |
|  |  |
| Băng thông bộ lọc | 22.5 K |
| Thời gian lấy mẫu khi LED sáng | 23.4375 us |
| Thời gian chuyển đổi mẫu lấy được của LED ở pha trước đó | 7.8125 us |
| Dòng điểu khiển LED | Chế độ 2x tối đa lên đến 100mA |

Thiết lập cho ngưỡng cho phép đọc dữ liệu đã được chuyển đổi lưu trong FIFO là 100 mẫu.

**1.3. Kích chế độ huỷ nền tự động - AACM**

A picture containing line, plot, diagram, slope

Description automatically generatedTín hiệu PPG thu thập được sẽ bị nhiễu bởi thành phần dc gây ra từ ảnh hưởng của môi trường, nó làm giảm đi độ khuếch đại của bộ khuếch đại TIA, do đó cần phải loại bỏ thành phần này. Chế độ huỷ nền tự động – AACM trong AFE giúp thực hiện loại bỏ thành phần dc này, do đó ta cần thiết lập cho phép chế độ này hoạt động để nó tạo ra dòng điện huỷ đi thành phần dc, giúp làm tăng độ khuếch đại của bộ TIA làm cho tín hiệu rõ ràng và tốt hơn để chuyển đổi một cách chính xác.

Ở hình trên, đường sóng màu xanh thể hiện dòng điện môi trường xung quanh gây nhiễu, đường sóng màu đen thể hiện tín hiệu dòng điện đã được điều chỉnh kéo về dưới mức ngưỡng cho phép của tín hiệu bằng cách dùng dòng huỷ bỏ. Khi sự khác biệt giữa dòng điện môi trường xung bắt đầu vượt quá một ngưỡng (được đánh dấu là IAMB\_RECONV), AACM sẽ tự động điều chỉnh giá trị của Offset DAC, từ đó kéo dòng điện xung quanh gần với mức không.

**1.4. Hiệu chỉnh cho chế độ huỷ nền tự động**

Việc phát hiện dòng môi trường đạt đến ngưỡng được thực hiện bởi AACM thông qua quan sát mã đầu ra của pha môi trường được đặt làm mục tiêu để ước tính giá trị dòng huỷ (IFS\_OFFDAC) tạo ra. Nó ước tính bằng cách tự thực hiệu quá trình hiệu chuẩn dựa trên các thông số RF độ lợi và dòng huỷ tối đa để thiết lập giá trị tương ứng vào các thanh ghi bên trong AFE sau đó được thiết lập cho các photodiode cần thiết.

A picture containing text, screenshot, number, font

Description automatically generated

Hệ thống được thiết lập ở mức IFS\_OFFDAC là **1X mode** và RF là **250 KOhm** do đó giá trị **107** sẽ được ghi vào thanh ghi bên trong AFE để cấu hình cho các photodidoe.

**1.5. Thiết lập tần số lấy mẫu, ngưỡng dòng điện trong AFE4420**

Tần số lấy mẫu của tín hiệu PPG hay còn gọi là chu kỳ lặp lại hoạt động (PRF) bên trong AFE được đặt là **50 Hz**.

Dòng huỷ tối được tạo ra ở 1X mode tương ứng với dòng điện lúc hiệu chuẩn nên giá trị dòng điện tối đa có thể tạo ra là +(-) 15.875 uA.

A screenshot of a computer

Description automatically generated with low confidence

**1.6. Cấu hình các pha tín hiệu và các photodiode**

**1.6.1. Cấu hình các pha tín hiệu**

Các pha tín hiệu được chia thành ba pha:

* Pha dummy: pha này được dùng làm pha môi trường giả cho hoạt động hiệu chỉnh của bộ tạo dòng huỷ giúp tạo ra giá trị dòng huỷ chính xác.
* Pha môi trường: pha này tương ứng với môi trường thật bên ngoài, bộ tạo dòng huỷ sau khi được hiệu chỉnh có thể tạo dòng huỷ đúng pha môi trường thực tế ở pha này.
* Pha LED: đây là pha mà các LED sáng để thu thập tín hiệu và thực hiện chuyển đổi dữ liệu thu thập được.

AFE cho phép tối đa 16 pha trong một lần lấy mẫu, hệ thống hiện tại đang sử dụng thực tế là **12 pha**, dùng **4 LED** để phát sáng và dùng **2 photodidode** (PD1 và PD3) để thu thập tín hiệu.

Cấu hình pha dummy cho cả PD1 và PD3 ở các pha thứ 1 và pha thứ 7. Lúc này các LED sẽ không sáng, đối với pha 1, đầu PD1 sẽ thu tín hiệu và pha 7 thì đầu PD3 sẽ thu tín hiệu. Dữ liệu thu được ở các pha này sẽ không được lưu vào FIFO.

|  |  |  |  |
| --- | --- | --- | --- |
| **Pha** | **LED** | **PD1** | **PD3** |
| **1** | OFF | ON | OFF |
| **7** | OFF | OFF | ON |

Cấu hình các pha môi trường cho PD1 là ở phase 2 là PD3 là ở phase 8. Lúc này LED vẫn chưa được bật và dữ liệu thu được bởi các PD sẽ được lưu vào FIFO.

|  |  |  |  |
| --- | --- | --- | --- |
| **Pha** | **LED** | **PD1** | **PD3** |
| **2** | OFF | ON | OFF |
| **8** | OFF | OFF | ON |

Sau khi đã các hình các pha dummy và pha môi trường, tiến hành thiết lập các pha LED. Pha LED bao gồm 8 pha được chia cho cả 4 LED, 4 pha LED đầu tiên sẽ do PD1 thu tín hiệu, 4 pha LED còn lại do PD3 thu tín hiệu.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Pha** | **LED1** | **LED2** | **LED3** | **LED4** | **PD1** | **PD3** |
| **3** | ON | OFF | OFF | OFF | ON | OFF |
| **4** | OFF | ON | OFF | OFF | ON | OFF |
| **5** | OFF | OFF | ON | OFF | ON | OFF |
| **6** | OFF | OFF | OFF | ON | ON | OFF |
| **9** | ON | OFF | OFF | OFF | OFF | ON |
| **10** | OFF | ON | OFF | OFF | OFF | ON |
| **11** | OFF | OFF | ON | OFF | OFF | ON |
| **12** | OFF | OFF | OFF | ON | OFF | ON |

Bảng dưới đây mô tả các cấu hình các pha bên trong AFE4420.

|  |  |  |  |
| --- | --- | --- | --- |
| **Pha** | **LEDs được bật** | **PD lấy mẫu** | **Ghi chú** |
| **1** |  | PD1 | Pha dummy cho AACM hiệu chỉnh trên PD1 |
| **2** |  | PD1 | Pha môi trường |
| **3** | LED1 | PD1 |  |
| **4** | LED2 | PD1 |  |
| **5** | LED3 | PD1 |  |
| **6** | LED4 | PD1 |  |
| **7** |  | PD3 | Pha dummy cho AACM hiệu chỉnh trên PD3 |
| **8** |  | PD3 | Pha môi trường |
| **9** | LED1 | PD3 |  |
| **10** | LED2 | PD3 |  |
| **11** | LED3 | PD3 |  |
| **12** | LED4 | PD3 |  |

**2.** **Hoạt động đo đạc trong hệ thống**

Để thực hiện thu thập dữ liệu từ AFE4420, chương trình được chia thành nhiều luồng hay còn gọi là các thread được hỗ trợ bởi Zephyr RTOS để xây dựng các chương trình thời gian thực trên chip nRF5340.

Trong Zephyr, một thread (luồng) là một đơn vị thực thi độc lập, được sử dụng để thực hiện các công việc cùng một lúc trong hệ thống nhúng. Luồng có thể được xem như các "nhóm công việc" được chạy đồng thời trên một bộ xử lý.

Một thread trong Zephyr có thể có một luồng thực thi duy nhất hoặc nhiều luồng thực thi song song. Mỗi thread được xác định bởi một tập hợp các thông số như ưu tiên, độ ưu tiên và ngăn xếp (stack).

Các thread trong Zephyr có thể được tạo ra, tiếp tục chạy và kết thúc. Khi một thread đang chạy, nó có thể đợi hoặc chờ đợi các tác vụ hoặc sự kiện xảy ra. Khi các tác vụ hoặc sự kiện được kích hoạt, thread sẽ bắt đầu thực hiện và tiếp tục chạy cho đến khi hoàn thành hoặc có yêu cầu chuyển đổi ngữ cảnh.

Việc sử dụng thread trong Zephyr giúp tăng hiệu suất và đáp ứng của hệ thống nhúng. Bằng cách chia các công việc thành các thread riêng biệt, hệ thống có thể thực hiện nhiều công việc đồng thời mà không bị chặn hoặc tắc nghẽn. Điều này cũng cho phép quản lý và ưu tiên các tác vụ khác nhau một cách linh hoạt, đồng thời cải thiện khả năng phản hồi của hệ thống.

Trong một ứng dụng đa luồng, có nhiều luồng đang chạy đồng thời. Nếu nhiều hơn một luồng cố gắng truy cập vào cùng một phần mã đồng thời, thường được gọi là phần mã quan trọng (critical section), điều này có thể dẫn đến hành vi không mong muốn hoặc sai sót. Đó là lúc cần sử dụng đồng bộ hóa luồng; đó là một cơ chế để đảm bảo chỉ có một luồng thực thi phần mã quan trọng tại bất kỳ thời điểm nào.

Đồng bộ hóa luồng là một cách để giải quyết vấn đề xung đột khi nhiều luồng cùng truy cập và thay đổi dữ liệu chung. Khi một luồng đang thực thi trong phần mã quan trọng, các luồng khác phải đợi cho đến khi luồng trước đó hoàn thành trước khi được phép truy cập. Điều này đảm bảo rằng các thay đổi dữ liệu được thực hiện một cách an toàn và tránh gây ra các lỗi không đáng có.

Có một số cơ chế đồng bộ hóa luồng khác nhau mà các ứng dụng đa luồng có thể sử dụng, ví dụ như semaphore (cờ hiệu), mutex (khóa độc quyền). Các cơ chế này cho phép luồng giao tiếp và điều khiển trạng thái của nhau, đảm bảo rằng các tác vụ được thực thi một cách tuần tự và đúng đắn.

Việc sử dụng đồng bộ hóa luồng trong ứng dụng đa luồng giúp tránh các xung đột và lỗi dữ liệu không mong muốn. Nó đảm bảo rằng các tác vụ được thực thi một cách an toàn và đúng đắn, đồng thời giúp tối ưu hóa hiệu suất và sử dụng tài nguyên.

**Cơ chế đồng bộ semaphore:**

Semaphore là một cách để kiểm soát truy cập vào tài nguyên chung bằng cách sử dụng giá trị của biến semaphore. Khi một luồng muốn truy cập vào tài nguyên, nó sẽ kiểm tra giá trị của semaphore. Nếu giá trị là không âm, tức là còn các phiên bản của tài nguyên khả dụng, luồng sẽ tiếp tục và giảm giá trị semaphore. Nếu giá trị là âm, tức là không còn phiên bản của tài nguyên khả dụng, luồng sẽ phải đợi cho đến khi một phiên bản trở thành khả dụng bằng cách chờ đợi việc tăng giá trị của semaphore từ các luồng khác.

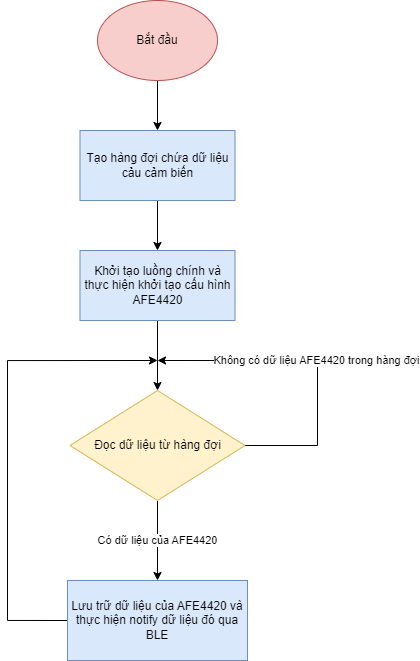
Semaphore có các thuộc tính sau:

* Khi khởi tạo, ta đặt một giá trị khởi tạo (lớn hơn 0) và một giới hạn tối đa.
* "Give" (tăng giá trị) sẽ tăng giá trị của semaphore cho đến khi giá trị đạt đến giới hạn tối đa, trong trường hợp đó, giá trị sẽ không tăng được nữa. "Give" có thể được thực hiện từ bất kỳ luồng nào.
* A picture containing text, screenshot, font, diagram

  Description automatically generated"Take" (giảm giá trị) sẽ giảm giá trị của semaphore cho đến khi bằng 0. Bất kỳ luồng nào cố gắng lấy semaphore khi đang bằng 0 thì phải chờ đợi cho đến khi một luồng khác làm cho nó khả dụng (bằng cách thực hiện "give" semaphore).

**2.1 Luồng chính thu thập dữ liệu các cảm biến**

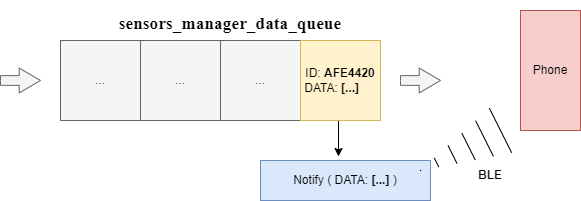
Khi hệ thống vừa được khởi động hay bật nguồn, luồng chính sẽ được khởi tạo để thực hiện việc đọc dữ liệu thu thập được của các luồng cảm biến khác trong hệ thống, ở đây khi luồng chính được khởi tạo nó sẽ thực hiện khởi tạo các cấu hình ban đầu cho AFE4420.

****

Ở luồng này, một hàng đợi tên là **sensors\_manager\_data\_queue** được tạo để lưu trữ các cấu trúc dữ liệu của các cảm biến khác nhau được thu thập trong hệ thống, bao gồm cả AFE4420. Cấu trúc này gồm có:

* ID: chỉ ra loại dữ liệu của cảm biến nào được đọc. VD: nếu là dữ liệu PPG của AFE4420 thì có ID là AFE4420.
* Dữ liệu của loại cảm biến: đây là nơi mà dữ liệu thu thập được của cảm biến được lưu trữ, đối với AFE4420 sẽ là một mảng kiểu số nguyên 32 bit gồm 500 phần tử

Khi luồng này được chạy, nó liên tục đọc các cấu trúc dữ liệu có ở trong hàng đợi và dựa vào ID của cấu trúc đó tương ứng với loại cảm biến nào thì sẽ thực hiện các tác vụ xử lý tương ứng với loại cảm biến đó. Ở đây, khi có dữ liệu của AFE4420 trong hàng đợi, luồng này sẽ thực hiện lưu trữ dữ liệu của AFE4420 vào trong một bộ đệm (buffer) để nhằm mục đích lưu trữ nó sau này vào trong SD card, sau đó nó sẽ thực hiện notify lần lượt các giá trị dữ liệu trong AFE4420 qua kết nối BLE để gửi dữ liệu đó qua ứng dụng di động.

****

**2.2. Luồng xử lý thu thập dữ liệu**

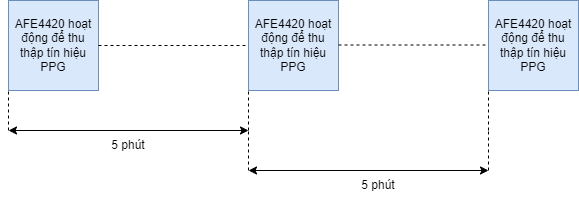
Ở bên trên là luồng hoạt động cho việc đọc và gửi dữ liệu đã được thu thập bởi các cảm biến cũng như dữ liệu của AFE4420. Còn việc bắt đầu cho AFE4420 tiến hành cho LED sáng và thu thập dữ liệu qua photodiode được thực hiện bên trong luồng xử lý dữ liệu thu thập này.

Luồng này sẽ thực hiện điều khiển việc cho cảm biến bắt đầu hoạt động và tiến hành thu thập dữ liệu và điều khiển kết thúc quá trình thu thập đó.

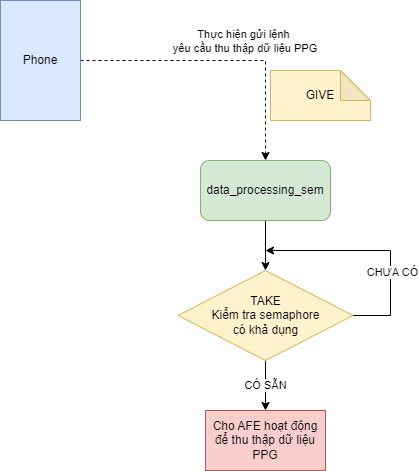
Luồng này được hoạt động dựa theo các chế độ hoạt động thu thập dữ liệu khác nhau, mỗi chế độ có cách thu nhập riêng biệt để đa dạng hoá khả năng thu thập và hiệu suất làm việc khác nhau cho thiết bị. Để tối ưu hiệu suất cũng như dễ dàng cho việc thu thập dữ liệu thì chế độ thu thập liên tục (continous mode) sẽ được thiết lập cho thiết bị.

Chế độ thu thập liên tục là chế độ mà khi có lệnh thực hiện thu thập dữ liệu, luồng này sẽ cho phép các cảm biến tiến hành thu thập dữ liệu mỗi năm phút một lần cho đến khi có lệnh yêu cầu ngừng thực hiện thu thập dữ liệu. Ở đây, AFE4420 sẽ thưc hiện phát sáng và thu thập dữ liệu qua photodiode mỗi năm phút một lần.

Trong mỗi lần thực hiện thu thập dữ liệu đó, AFE4420 sẽ thực hiện thu thập đủ số mẫu cần thiết để đảm bảo dữ liệu đáng tin cậy, số mẫu cần thiết để thực hiện đọc và gửi qua kết nối BLE được thiết lập là 500 mẫu, tương ứng với tích tốc độ lấy mẫu của AFE4420 đang được thiết lập là 50 Hz và số pha có nghĩa trong việc lấy mẫu của cảm biến là 10 pha, bao gồm 2 pha môi trường và 8 pha LED sáng.



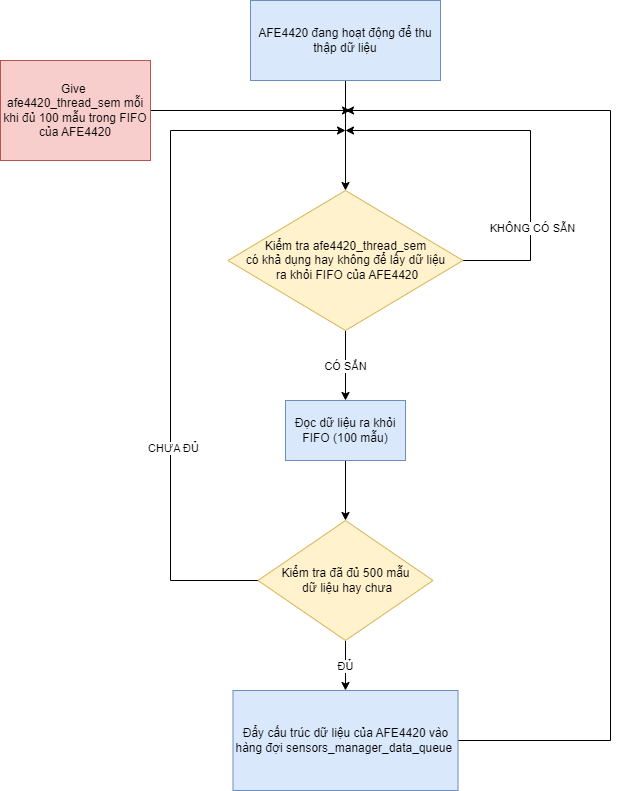
Khi có lệnh gọi thực hiện bắt đầu quá trình thu thập dữ liệu, tức là tiến hành đi vào chế độ thu thập liên tục. Lúc này hệ thống cho – “give” một semaphore gọi là data\_processing\_sem, luồng này sẽ liên tục lấy - “take” để kiểm tra xem semaphore này có có khả dụng hay không. Khi semaphore này khả dụng, tức là có tín hiệu yêu cầu bắt đầu cho AFE4420 hoạt động để thu thập dữ liệu, luồng này sẽ tiến hành cho các cảm biến hoạt động để thu thập dữ liệu.



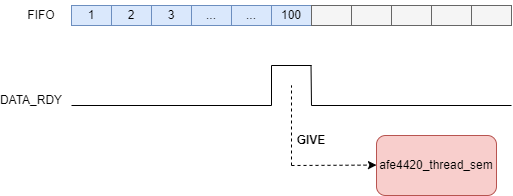
**2.3. Luồng thu thập dữ liệu của AFE4420**

Khi luồng xử lý thu thập dữ liệu hoạt động, nó cho phép các cảm biến hoạt động để thu thập dữ liệu của từng loại cảm biến. Mỗi loại cảm biến khác nhau lại có một luồng riêng biệt để xử lý cho việc đọc và ghi dữ liệu của chúng.

Đối với AFE4420, luồng thu thập dữ liệu của AFE4420 là luồng mà hoạt động của nó là đọc những dữ liệu mà nó thu thập được và lưu những giá trị thu thập được đó vào trong hàng đợi để cho luồng chính tiến hành đọc và gửi dữ liệu đó qua kết nối BLE.

****

Khi tiến hành khởi tạo các cấu hình ban đầu cho AFE4420 thì có thiết lập ngưỡng dữ liệu cho phép đọc ra khỏi FIFO là 100 mẫu, lúc này khi dữ liệu được thu thập đủ 100 mẫu thì một ngắt DATA\_RDY được tạo ra, tín hiệu ngắt này sẽ give cho một semaphore được thiết lập trong luồng này là **afe4420\_thread\_sem**. Do đó, mỗi khi có tín hiệu DATA\_RDY thì semaphore lại khả dụng, lúc này hoạt động của luồng là luôn kiểm tra semaphore này có khả dụng để tiến hành lấy dữ liệu.



Khi semaphore đã khả dụng thì 100 mẫu sẽ được đọc ra, do ta cần 500 mẫu nên phải chờ đến khi thu thập đủ thì mới đẩy 500 mẫu đó đã được cấu trúc lại vào hàng đợi **sensors\_manager\_data\_queue** để cho luồng chính thực hiện lưu trữ và gửi những dữ liệu này đi đến ứng dụng di động thông qua kết nối BLE.

