BÁO CÁO THỰC HÀNH BÀI Lab5\_VL53L01\_Flash

Môn học: **Chuyên đề thiết kế hệ thống nhúng 1** - Mã lớp: **CE437.P11**

Giảng viên hướng dẫn thực hành: **Phạm Minh Quân**

|  |  |  |
| --- | --- | --- |
| **Thông tin sinh viên** | **Mã số sinh viên** | **Họ và tên** |
| **22521472** | **Phạm Quốc Tiến** |
| **22521570** | **Trịnh Thành Trung** |
| **22521564** | **Nguyễn Đức Trung** |
| **Link các tài liệu tham khảo** *(nếu có)* |  | |
| **Đánh giá của giảng viên**:  *+ Nhận xét*  *+ Các lỗi trong chương trình*  *+ Gợi ý* |  | |

*[Báo cáo chi tiết các thao tác, quy trình sinh viên đã thực hiện trong quá trình làm bài thực hành. Chụp lại hình ảnh màn hình hoặc hình ảnh kết quả chạy trên sản phẩm. Mô tả và giải thích chương trình tương ứng để cho ra kết quả như hình ảnh đã trình bày.]*

Mục lục

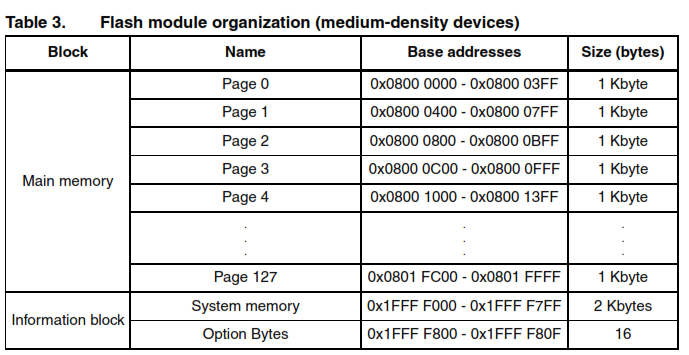
[1) Giao tiếp với bộ nhớ Flash 3](#_Toc184806537)

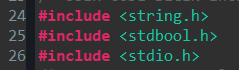
[2) Giao tiếp I2C 5](#_Toc184806538)

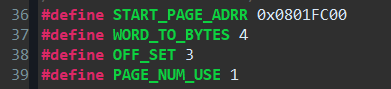
[3) DEMO 7](#_Toc184806539)

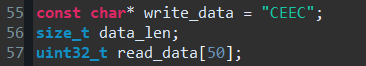
1. Giao tiếp với bộ nhớ Flash

Trong bộ nhớ flash của stm32f103c8t6 được phân bổ thành 127 page, mỗi page có kích thước là 1Kbyte. Do trong bộ nhớ flash có lưu trữ cả code nạp xuống vậy nên nhóm sẽ thực hiện đọc ghi ở page 127 với địa chỉ bắt đầu là 0x08001000

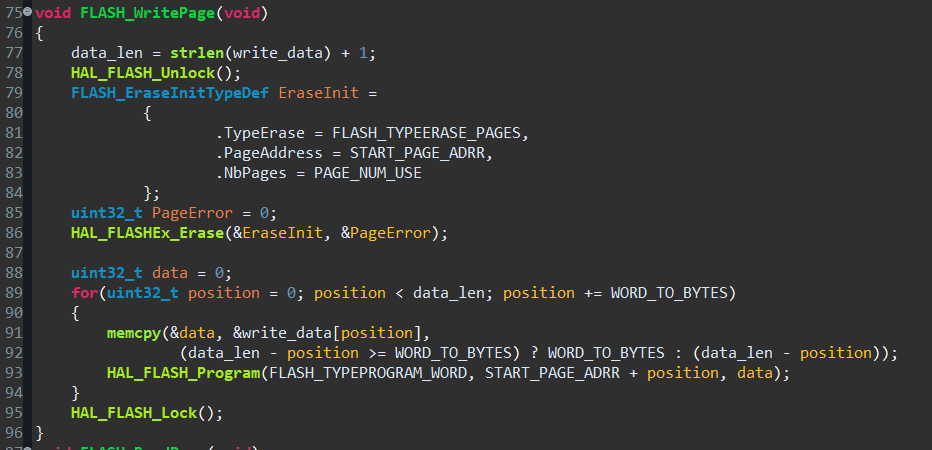
Thêm thư viện và khai báo



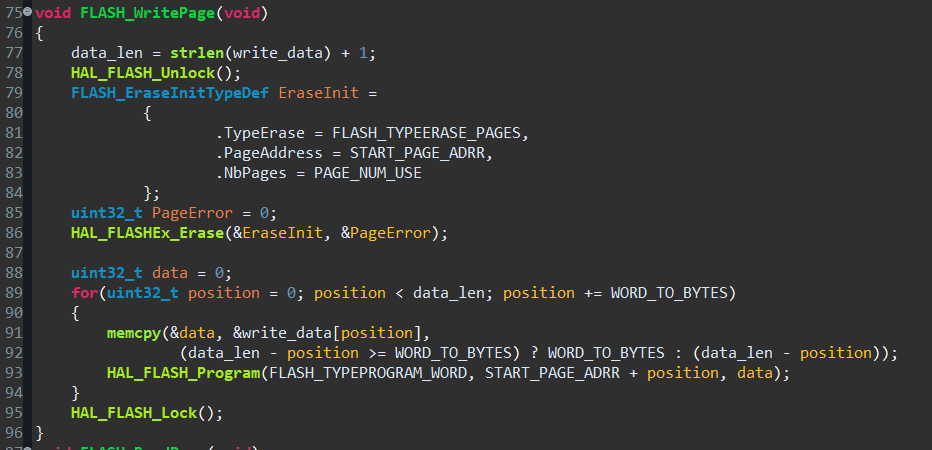




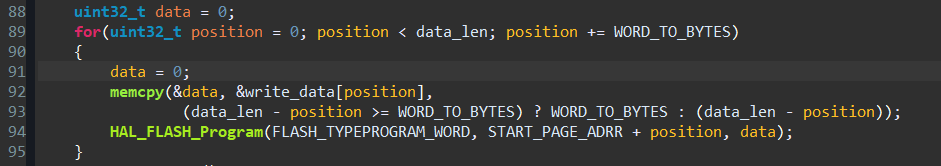
Trong hàm FLASH\_WritePage có chức năng viết dữ liệu xuống vùng nhớ FLASH thì nhóm sẽ thực hiện gán data\_len độ dài của chuỗi cần ghi cộng thêm 1 ký tự kết thúc ‘\0’. Sau đó nhóm tiến hành mở khóa vùng nhớ FLASH



Do đặc điểm của bộ nhớ FLASH nên nhóm sẽ xóa dữ liệu ở Page 127 trước khi tiến hành ghi dữ liệu. Biến PageError sẽ có địa chỉ cũa vùng nhớ bị lỗi (nếu có) khi đang xóa và phần này nhóm sẽ mặc định rằng không có vùng nhớ lỗi



Tiếp theo nhóm sẽ tiến hành covert data từ chuỗi ký tự sang dạng uint32\_t, sau đó tiến hành ghi xuống vùng nhớ FLASH

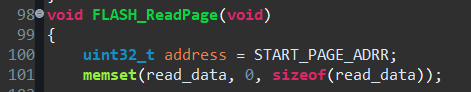


Cụ thể do chương trình sẽ ghi xuống vùng nhớ một word (4 bytes) mỗi lần ghi. Tuy nhiên, dữ liệu tron write\_data có thể không chia hết cho 4 bytes. Vậy nên điều kiện trong memcpy đảm bảo không ghi quá mức dữ liệu sẵn có trong write\_data và dữ liệu cuối cùng nếu nhỏ hơn 4 bytes thì vẫn được ghi đầy đủ mà không gây lỗi.

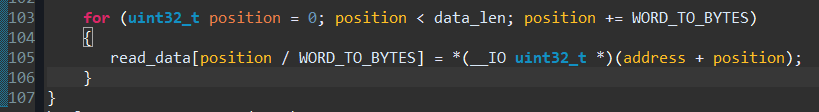
Sau đó nhóm sẽ khóa bộ nhớ FLASH lại



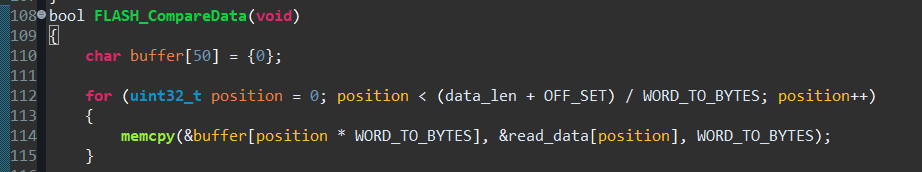
Trong hàm FLASH\_ReadPage() có chức năng đọc dữ liệu trong bộ nhớ FLASH thì nhóm sẽ tiến hành gán địa chỉ ban đầu, reset biến read\_data



Sau đó nhóm sẽ tiến hành đọc giá trị từ bộ nhớ FLASH



Trong hàm FLASH\_CompareData() có chức năng so sánh chuỗi ghi và chuỗi đọc được có giống nhau không thì nhóm sẽ tiến hành chuyển đổi dữ liệu đọc được từ uint32\_t sang chuỗi char

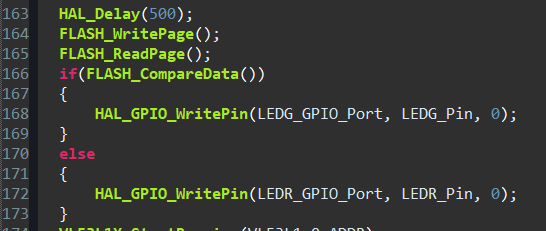


Lý đo phải cộng thêm OFF\_SET là để đảm bảo lấy các byte cuối cùng (không đủ 4 bytes). Lý do lấy OFF\_SET = 3 là do phần byte dư tối đa 3 bytes.

Sau đó nhóm sẽ return kết quả so sánh write\_data và buffer



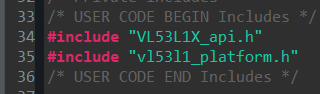
Trong hàm main nhóm sẽ bật led green khi hai chuỗi giống nhau và bật led red khi hai chuỗi khác nhau



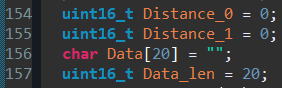
1. Giao tiếp I2C

Trong phần code này nhóm sẽ sử dụng thư viện để giao tiếp với VL53L1X

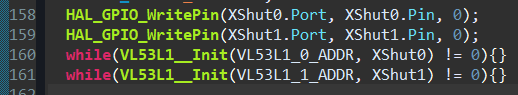
Thêm thư viện vào main.h



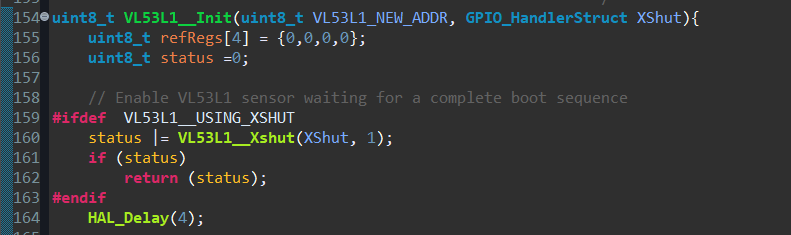
Khai báo trong hàm main



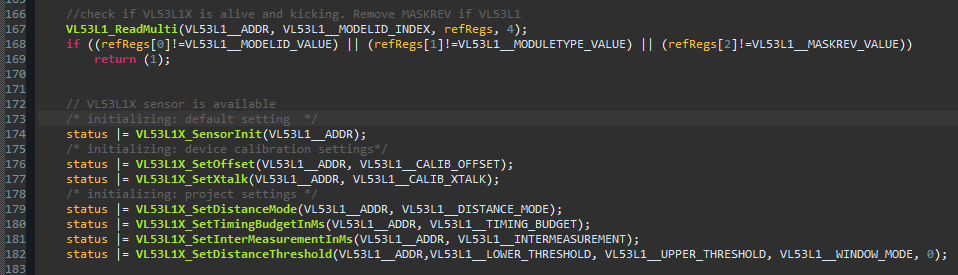
Tiếp theo nhóm sẽ kéo chân Xshut của hai cảm biến về 0 để hai cảm biến quay về chế độ standby. Sau đó nhóm sẽ gọi hàm khởi tạo



Trong hàm khởi tạo nhóm sẽ kéo chân XShut của cảm biến khởi tạo để cảm biến chạy chế độ boot. Khi đó nhóm có thể lập trình cho cảm biến được



Sau đó nhóm sẽ cài đặt các giá trị để cảm biến hoạt động



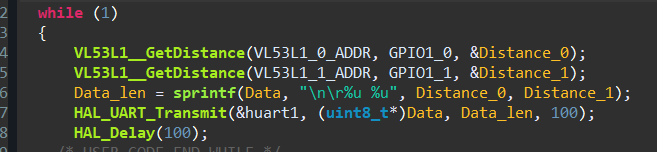
Sau cùng nhóm sẽ set lại địa chỉ của cảm biến (do khi mới khởi động lại thì cảm biến sẽ chạy giá trị mặc định) và return lại lỗi. Nếu không có lỗi thì status = 0



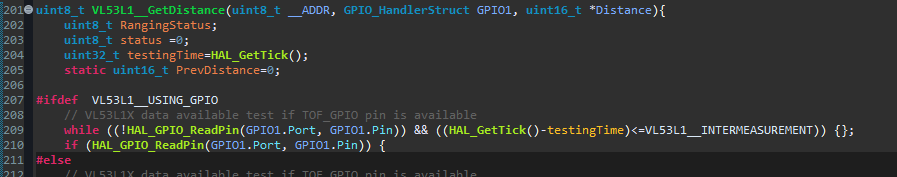
Tiếp đến trong hàm main nhóm sẽ tiến hành điều khiển cảm biến bắt đầu đo



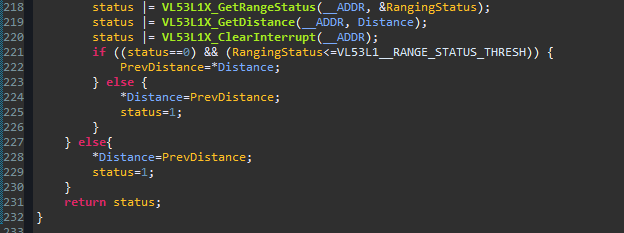
Trong vòng while nhóm sẽ tiến hành đọc và xuất data qua UART



Trong hàm VL53L1\_GetDistance() thì hàm sẽ thực hiện kiểm tra chân GPIO1 đã được kéo lên chưa (kéo lên tương đương dữ liệu sẵn sàng để đọc). Chương trình sẽ chở đến khi hết thời gian đo mà chương trình đã lập trình cho cảm biến



Nếu mà chân GPIO1 không kéo lên thì hàm sẽ set distance bằng giá trị trước đó đo được và trả về status = 1 (tương đương lỗi). Còn không thì chương trình sẽ thực hiện lấy data.



Nếu giá trị mà GetRangeStatus mà trả về lớn hơn 2 thì chương trình sẽ trả về lỗi còn không thì sẽ phụ thuộc vào status. Lý do mà nhóm lấy giá trị 2 thì trong datasheet họ nói rằng số 2 tương ứng với việc warning: low return signal level hay là mức tín hiệu phản xạ trở lại từ đối tượng được đo là quá thấp để đảm bảo phép đo chính xác tức là nhóm chấp nhận lấy dữ liệu từ trường hợp này đến trường hợp tốt nhất.

1. DEMO

[Code](https://github.com/kamuisi/CDHTN/tree/main/lab_5)

[Video demo](https://drive.google.com/file/d/1rF45ODXkMUbsL6kPxsRTW9gbR__Bn3pk/view?usp=sharing)