BÁO CÁO THỰC HÀNH BÀI 1

Môn học: **CHUYÊN ĐỀ THIẾT KẾ HỆ THỐNG NHÚNG 1**- Mã lớp: **CE437.N11**

Giảng viên hướng dẫn thực hành: Phạm Minh Quân

|  |  |
| --- | --- |
| **Thông tin sinh viên** | Mã số sinh viên: 19521924  Họ và tên: Trương Bảo Nguyên |
| Mã số sinh viên: 19522111  Họ và tên: Đồng Quang Quyền |
| Mã số sinh viên: 19522437  Họ và tên: Trịnh Trấn Trung |
| Mã số sinh viên: 19520571  Họ và tên: Tô Quang Huấn |
| **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. Sinh viên xuất ra file .pdf và đặt tên theo cấu trúc: MSSV\_HoTen\_Labx\_Report.pdf (Trong đó: MSSV là mã số sinh viên, HoTen là họ và tên, x trong Labx là chỉ số của bài thực hành tương ứng]*

Mục lục

[Mục lục 2](#_Toc118816304)

[PHẦN 1: KHỞI TẠO 3](#_Toc118816305)

[1. Khởi tạo GPIO 3](#_Toc118816306)

[2. Cấu hình tham số của Timer 4](#_Toc118816307)

[PHẦN 2: VIẾT CHƯƠNG TRÌNH 5](#_Toc118816308)

[Câu 1. Viết chương trình hiện thực Delay 5](#_Toc118816309)

[Câu 2. Viết chương trình tạo hiệu ứng LED 6](#_Toc118816310)

[Câu 3. Viết chương trình kiểm tra nút nhấn. 7](#_Toc118816311)

[PHẦN 3: KẾT QUẢ: 8](#_Toc118816312)

PHẦN 1: KHỞI TẠO

Đầu tiên chúng ta thực hiện các thao tác để khởi tạo Led in, Key button, Crystal, Timer:

Graphical user interface

Description automatically generated

Hình : Sơ đồ chân được khởi tạo

1. Khởi tạo GPIO

Theo như sơ đồ nguyên lý, thực hiện cấu hình 2 chân PA0 và PA1 là GPIO\_Input cho User Key. Tiếp theo khởi tạo 4 chân GPIO Output lần lượt là PB0, PB1, PB2, PB3 tương ứng với LED0, LED1, LED2, LED3 để phục vụ cho các công việc điều khiển bật tắt LED sau này.

Table

Description automatically generated

Hình : Khởi tạo GPIO cho hệ thống

1. Cấu hình tham số của Timer

Thực hiện cấu hình thạch anh như trong tài liệu hướng dẫn thực hành Lab1, ta có kết quả như sau:

Diagram

Description automatically generated

Hình : Sơ đồ giá trị tần số của hệ thống

Tiếp theo thực hiện cấu hình cho Timer cần sử dụng. Ở đây nhóm sử dụng Timer 3 với tần số của hệ thống là 84MHz. Rồi sau đó thực hiện tính toán Prescaler để có giá trị tần số của timer như mong muốn. Ta sử dụng công thức:

Nhóm muốn sử dụng tần số của timer là 10MHz nên chọn Prescaler là 8399. Chọn mode là Counting Up và để Period AutoReload là mặc định.

Graphical user interface, application

Description automatically generated

Hình : Các tham số được thiết lập cho Timer 3

# PHẦN 2: VIẾT CHƯƠNG TRÌNH

Yêu cầu:

* Sinh viên viết chương trình để tạo ra ít nhất 2 hiệu ứng chớp/tắt trên các LED sử dụng timer để điều khiển.
* Sinh viên đọc trạng thái nút nhấn tại chân PA0: nếu nút PA0 được nhấn thì sinh viên thực hiện giảm thời gian chu kỳ của các hiệu ứng LED ở trên.
* Sinh viên đọc trạng thái nút nhấn tại chân PA1: nếu nút PA1 được nhấn thì sinh viên thực hiện tăng thời gian chu kỳ của các hiệu ứng LED ở trên.
* Nếu cả 2 nút cùng được nhấn thì sinh viên sẽ thay đổi hiệu ứng LED mới sau khi cả 2 nút nhấn được thả ra.

**Diagram

Description automatically generated**

Hình : Lưu đồ giải thuật của hệ thống

Nhìn vào lưu đồ của hệ thống, đầu tiên là thực hiện khởi tạo các giá trị cần thiết của hệ thống như Timer, các GPIO cho Led và Key, các biến phục vụ cho chương trình,.... Sau khi khởi tạo hoàn tất, trong vòng while, ta kiểm tra luân phiên giá trị của 2 key là KEY0 và KEY1. Khi KEY0 được nhấn thì tiếp tục kiểm tra KEY1 có được nhấn cùng lúc luôn hay không. Nếu có thì thực hiện gán các cờ và chuyển hiệu ứng, nếu không thì thực hiện giảm thời gian của chu kỳ led và thực hiện chạy hiệu ứng. Trường hợp còn lại tương tự với việc kiểm tra KEY1. Nếu khi KEY1 được nhấn và có KEY0 được nhấn cùng thì thực hiện chuyển hiệu ứng Led. Còn nếu không thì thực hiện tăng thời gian chu kỳ Led lên và hiển thị hiệu ứng. Khi thực hiện xong thì quay lại để thực hiện kiểm tra như lúc ban đầu.

1. Viết chương trình hiện thực Delay

Đây là chương trình tự định nghĩa hàm Delay.



Đầu tiên, thực hiện viết chương trình cho hàm Delay\_100ms với mục đích sử dụng Timer để tạo ra delay 100ms. Bởi vì ta đã thực hiện cấu hình cho tần số của Timer là 10MHz tức là thời gian thực hiện một lần đếm là . Do đó cần đến 1000 lần đếm thì mới đủ thời gian là

. Tiếp tục hiện thực hàm Delay\_X00ms với input là số lần thực hiện hàm Delay\_100ms, có nghĩa là nếu ta muốn thực hiện delay 1s thì cần 10 lần delay 100ms nên giá trị cần truyền vào hàm là 10.

1. Viết chương trình tạo hiệu ứng LED

Ta thực hiện xây dựng 2 hàm tương ứng với 2 hiệu ứng LED khác nhau:



Bắt đầu với hiệu ứng đầu tiên, hiệu ứng này thực hiện bật cùng lúc 4 Led, sau đó đợi trong một khoảng thời gian và tắt Led. Tham số đầu vào của hàm là period tức là thời gian phải chờ của hiệu ứng Led. Sau khi thực hiện chớp tắt Led nhiều lần thì thực hiện Reset lại bộ Led. Ta sử dụng hàm Reset\_LEDs() để tắt hết Led.



Tiếp theo là viết chương trình cho hiệu ứng Led thứ 2. Hiệu ứng này sẽ thực hiện bật tắt từng Led luân phiên nhau với khoảng thời gian chờ được nhận từ tham số period. Để dễ quan sát thì video demo sẽ được đính kèm bên dưới bài báo cáo.



1. Viết chương trình kiểm tra nút nhấn.

Sau khi hoàn tất viết chương trình cho Timer và hiệu ứng Led, ta đến phần chính của chương trình. Tại hàm main thực hiện các khởi tạo cần thiết, trong loop while, ta thực hiện kiểm tra luân phiên giá trị của 2 nút nhấn đã được thiết lập ở trên. Đầu tiên thực hiện khai báo các biến như sau:



Sau khi khai báo biến và thực hiện khởi tạo các chức năng của hệ thống, ta thực hiện kiểm tra nút nhấn.



Trên đây là hàm hiểm tra tại nút nhấn PA0 được label là KEY0. Bởi vì khi thiết lập thì đã đặt giá trị của 2 nút nhấn này là Pull up nên khi nhấn sẽ nhận được giá trị là 0. Khi KEY0 được nhấn, ta thực hiện luôn kiểm tra xem KEY1 có được nhấn hay không để xử lý trường hợp chuyển trạng thái khi 2 nút được nhấn cùng lúc. Nếu KEY1 cũng được nhấn đồng thời với KEY0 thì thực hiện đổi state và gán flag = 1, điều này sẽ thông báo là đã có tín hiệu chuyển hiệu ứng và hiệu ứng sẽ được chuyển. Hàm Effect() sẽ thực hiện chuyển đổi giữa 2 hiệu ứng dựa trên giá trị biến state.



Trong trường hợp 2 nút nhấn không được nhấn cùng lúc thì thực hiện giảm giá trị thời gian của chu kỳ led đi 1 tương ứng với 100ms, nếu giá trị count giảm xuống bằng 1 thì khi thực hiện trừ tiếp tục thì sẽ bằng 0 và gây ra lỗi. Do đó thực hiện gán lại về 10 để tránh xảy ra lỗi. Thực hiện viết chương trình tương tự cho nút nhấn KEY1, thay vì giảm thì ta thực hiện tăng count lên 1.



PHẦN 3: KẾT QUẢ:

Link video demo: <https://youtu.be/GxAS6Vzpuss>

Kết quả của chương trình là ban đầu sẽ thực hiện hiển thị hiệu ứng 1. Khi nhấn KEY0 sẽ thực hiện giảm thời gian delay giữa mỗi lần chuyển trạng thái trong một hiệu ứng Led đi 100ms, làm cho hiệu ứng chạy với tốc độ chậm hơn. Nút nhất KEY1 có nhiệm vụ tăng lại thời gian đó lên 100ms. Còn trong trường hợp nhấn cùng lúc 2 nút nhấn thì sẽ chuyển qua hiệu ứng tiếp theo và vẫn giữ nguyên thời gian delay của hiệu ứng trước đó.