**BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP TP.HCM**

**----oOo----**

**Text, logo

Description automatically generated**

**VI ĐIỀU KHIỂN VÀ ỨNG DỤNG**

**BÀI BÁO CÁO 9**

**NHÓM 8**

Giảng viên: **Phạm Quang Trí**

Sinh viên: Hồ Bửu Sơn

MSSV: 21119571

**TP.HCM 2023**

**Bài tập mức độ 3:** Viết chương trình cho STM32 sử dụng chức năng PWM và chức năng ADC để tạo ra 2 xung PWM có **dạng ngược nhau**, có tần số như trong bảng bên dưới trình bày và chu kỳ nhiệm vụ (Active High) có thể thay đổi được từ 0% đến 100% tại một chân PWM (kết quả đo được và lấy minh chứng từ máy Oscilloscope, sai số tần số và chu kỳ nhiệm vụ mỗi 1% sẽ bị trừ 1 điểm). Việc thay đổi chu kỳ nhiệm vụ được thực hiện bằng cách sử dụng một biến trở. Lưu ý rằng 2 xung PWM này phải **có dead-time là 1µs**

**Nhóm 8**: fPWM = 16.000Hz.

**Yêu cầu 1**: **Trong quyển báo cáo, dựa vào yêu cầu bài tập đã chọn, phải vẽ sơ đồ kết nối phần cứng của toàn bộ hệ thống sử dụng vi điều khiển STM32.**

* Sơ đồ kết nối phần cứng

A yellow rectangular object with black text

Description automatically generated

**Yêu cầu 2: Trong quyển báo cáo, trình bày rõ ràng lưu đồ giải thuật và mã nguồn của chương trình điều khiển**

* Lưu đồ giải thuật

**A black screen with white text

Description automatically generated**

* Mã nguồn chương trình

#include "main.h"

ADC\_HandleTypeDef hadc1;

TIM\_HandleTypeDef htim1;

void SystemClock\_Config(void);

static void MX\_GPIO\_Init(void);

static void MX\_ADC1\_Init(void);

static void MX\_TIM1\_Init(void);

/\* USER CODE BEGIN 0 \*/

uint32\_t val;

uint32\_t dt\_cycle;

/\* USER CODE END 0 \*/

int main(void)

{

HAL\_Init();

SystemClock\_Config();

MX\_GPIO\_Init();

MX\_ADC1\_Init();

MX\_TIM1\_Init();

HAL\_TIM\_PWM\_Start(&htim1, TIM\_CHANNEL\_1);

HAL\_TIMEx\_PWMN\_Start(&htim1, TIM\_CHANNEL\_1);

while (1)

{

HAL\_ADC\_Start(&hadc1);

HAL\_ADC\_PollForConversion(&hadc1, 10);

val = HAL\_ADC\_GetValue(&hadc1);

dt\_cycle = val \* 900/4095;

\_\_HAL\_TIM\_SET\_COMPARE(&htim1, TIM\_CHANNEL\_1,dt\_cycle);

}

}

**Giải thích công thức tính toán**

**Chứng minh công thức chọn giá trị Prescaler và Counter Period:**

f PWM= 16000 Hz

Chọn Pre=4, = 72MHz,

Thế vào phương trình (1):

* Chọn Pre = 4, Counter period = 899

**Chứng minh giá trị Dead-time:**

Ta có: = 72 MHz = 0,0139 μs

* Chọn Dead-time = 72

**Chứng minh công thức ADC trong mã nguồn:**

- Module ADC của STM32F103RCT6 có 12 bit, giá trị tối đa mà có thể đọc được là 2^12 - 1 = 4095. Điều này tương ứng với điện áp mà ADC có thể đọc được là từ 0 đến 4095. Tương ứng với điện áp từ 0V đến 3.3V.

- Khi thay đổi chu kỳ nhiệm vụ (Active High) của PWM từ 0% đến 100%, khi giá trị ADC là 0 thì chu kỳ nhiệm vụ PWM là 0% và khi giá trị ADC là 4095 thì chu kỳ nhiệm vụ PWM là 100%.

Counter Period + 1 ⇔ 100%

**A diagram of a line graph

Description automatically generated**

Khi đường thẳng đi qua gốc tọa độ, dùng quy tắc tam suất:

Giá trị ADC ⇔ duty cycle

4095 ⇔ 900

-Trong đó:

Duty cycle: Giá trị chu kì nhiệm vụ của xung PWM.

Giá trị ADC : Giá trị chuyển đổi A/D của mô-đun ADC1.

Counter Period: Chu kỳ bộ đếm của PWM.

4095: Giá trị điện áp tối đa mà ADC 12-bit có thể đọc được.

**Yêu cầu 3: Nạp chương trình vào kit thí nghiệm STM32 và thực hiện cho chạy thử trên phần cứng trong phòng thực hành. Quay video clip minh chứng kết quả thực hiện, tải lên Youtube duy nhất 1 video clip ( trong trường hợp minh chứng có nhiều video clip nhỏ thì sinh viên phải tự ghép lại thành 1 video clip tổng hợp ) và ghi liên kết vào báo cáo.**

Liên kết video: <https://youtu.be/3UEl43kGx1U>