## 

BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH

khoa chất lượng cao- công nghệ thông tin

🙣🙣🕮🙡🙡

**BÁO CÁO**

**HỆ THỐNG NHÚNG**

**Đề tài: ĐIỀU KHIỂN ĐỘNG CƠ BƯỚC SỬ DỤNG CHIP ARM CORTEX-M4 STM32F407 DISCOVERY.**

GVHD: Thầy Đinh Công Đoan

SVTH: Nguyễn Thành Nam 14110312

Dương Cao Nhân 14110369

Nguyễn Thanh Nhân 14110404

Tp. Hồ Chí Minh – Ngày 24 Tháng 5/ 2017

[Phần 1 PHẦN MỞ ĐẦU 1](#_Toc483388408)

[1. Tóm tắt nội dung báo cáo 1](#_Toc483388409)

[2. Đặt vấn đề 1](#_Toc483388410)

[2.1 Nghiên cứu trong, ngoài nước và tính cấp thiết của đề tài 1](#_Toc483388411)

[2.1.1 Một số tài liệu liên quan 2](#_Toc483388412)

[2.2 Tính cấp thiết của đề tài 3](#_Toc483388413)

[2.3 Mục tiêu đề tài 3](#_Toc483388414)

[2.4 Đối tượng phạm vi nghiên cứu 3](#_Toc483388415)

[a. Đối tượng 3](#_Toc483388416)

[b. Phạm vi nghiên cứu 3](#_Toc483388417)

[2.5 Phương pháp nghiên cứu 3](#_Toc483388418)

[2.6 Nội dung đề tài 4](#_Toc483388419)

[PHẦN 2: NỘI DUNG 5](#_Toc483388420)

[Chương 1: Giới thiệu chung về dòng ARM cortex KIT STM32F4 DISCOVERY 5](#_Toc483388421)

[Chương 2 Cấu trúc chung của kit 6](#_Toc483388422)

[2. Điều khiển động cơ bước với Kit STM32F407- Discovery. 9](#_Toc483388423)

[2.1. Thiết kế mạch Motor Driver L298 Dual H-Bridge Motor Driver: 9](#_Toc483388424)

[2.2 Sơ đồ Motor Driver: 9](#_Toc483388425)

[2.3 Động cơ bước ( Stepper Motor) 10](#_Toc483388426)

[Chương 4 : Ứng dụng 13](#_Toc483388427)

[1. Phần mềm viết chương trình. 13](#_Toc483388428)

[2. Chương trình điều khiển Step Motor: 13](#_Toc483388429)

[Phần 3: Kết luận 18](#_Toc483388430)

[1.Ưu điểm 18](#_Toc483388431)

[2.Nhược điểm 18](#_Toc483388432)

[Tài liệu tham khảo 18](#_Toc483388433)

# **PHẦN 1 PHẦN MỞ ĐẦU**

## **Tóm tắt nội dung báo cáo**

* Giới thiệu vi xử lý board STM32F407- Discovery
* Giới thiệu driver motor L298
* Giới thiệu động cơ bước( Stepper motor)
* Ứng dụng vào để đóng mở chốt tự động

1. **Đặt vấn đề**
   1. **Nghiên cứu trong, ngoài nước và tính cấp thiết của đề tài**
2. **Nước ngoài**

Hệ thống nhúng được sản xuất hàng loạt đầu tiên là máy hướng dẫn cho tên lửa quân sự vào năm 1961. Nó là máy hướng dẫn Autonetics D-17, được xây dựng sử dụng những bóng bán dẫn và một đĩa cứng để duy trì bộ nhớ. Khi Minuteman II được đưa vào sản xuất năm 1996, D-17 đã được thay thế với một máy tính mới sử dụng mạch tích hợp. Tính năng thiết kế chủ yếu của máy tính Minuteman là nó đưa ra thuật toán có thể lập trình lại sau đó để làm cho tên lửa chính xác hơn, và máy tính có thể kiểm tra tên lửa, giảm trọng lượng của cáp điện và đầu nối điện.

Từ những ứng dụng đầu tiên vào những năm 1960, các hệ thống nhúng đã giảm giá và phát triển mạnh mẽ về khả năng xử lý. Bộ vi xử lý đầu tiên hướng đến người tiêu dùng là Intel 4004, được phát minh phục vụ máy tính điện tử và những hệ thống nhỏ khác. Tuy nhiên nó vẫn cần các chip nhớ ngoài và những hỗ trợ khác. Và những năm cuối 1970, những bộ xử lý 8 bit đã được sản xuất, nhưng nhìn chung chúng vẫn cần đến những chip nhớ bên ngoài.

1. Trong nước

Dòng ARM Cortex là một bộ xử lí thế hệ mới đưa ra một kiến trúc chuẩn cho nhu cầu đa dạng về công nghệ. Không giống như các chip ARM khác, dòng Cortex là một lõi xử lí hoàn thiện, đưa ra một chuẩn CPU và kiến trúc hệ thống chung. Dòng Cortex gồm có 3 phân nhánh chính: dòng A dành cho các ứng dụng cao cấp, dòng R dành cho các ứng dụng thời gian thực như các đầu đọc và dòng M dành cho các ứng dụng vi điều khiển và chi phí thấp. STM32 được thiết kế dựa trên dòng Cortex-M3, dòng Cortex-M3 được thiết kế đặc biệt để nâng cao hiệu suất hệ thống, kết hợp với tiêu thụ năng lượng thấp, CortexM3 được thiết kế trên nền kiến trúc mới, do đó chi phí sản xuất đủ thấp để cạnh tranh với các dòng vi điều khiển 8 và 16-bit truyền thống.

Thoạt nhìn thì các ngoại vi của STM32 cũng giống như những vi điều khiển khác, như hai bộ chuyển đổi ADC, timer, I2C, SPI, CAN, USB và RTC. Tuy nhiên mỗi ngoại vi trên đều có rất nhiều đặc điểm thú vị. Ví dụ như bộ ADC 12-bit có tích hợp một cảm biến nhiệt độ để tự động hiệu chỉnh khi nhiệt độ thay đổi và hỗ trợ nhiều mode chuyển đổi. Mỗi bộ timer có 4 khối capture compare, mỗi khối timer có thể liên kết với các khối timer khác để tạo ra một mảng các timer tinh vi.

* + 1. **Một số tài liệu liên quan**

- Giáo trình “Hệ thống nhúng” –Ths. Đinh Công Đoan

1 http://forum.cncprovn.com/threads/8692-Gioi-thieu-kit-phat-trien-STM32F4-Discovery-moi-nhat-cua-STMicroelectronics-STM32F4

2. http://www.proe.vn/stm32f407g-disc1-kit-stm32f4-discovery-discovery-kit-with-stm32f407vg\_p1\_1-1\_2-1\_3-602\_4-196.html

3. https://en.wikipedia.org/wiki/Stepper\_motor

4. http://www.st.com/en/evaluation-tools/stm32f4discovery.html

### **2.2 Tính cấp thiết của đề tài**

- Trong cuộc sống hiện nay, vi điều khiển đã trở nên quen thuộc trong các ngành kỹ thuật. Từ các dây chuyền sản xuất lớn đến các thiết bị gia dụng, chúng ta đều thấy sự hiện diện của vi điều khiển. Từ những thiết bị đơn giản như tivi, tủ lạnh, máy giặt đến các thiết bị cao cấp hơn như xe hơi, vệ tinh,… đều có các vi xử lí, vi điều khiển. Sự phát triển mạnh mẽ đó đã thay thế các tủ điều khiển to lớn, cồng kềnh bằng những thiết bị rất nhỏ, thậm chí là không thể thấy được. Từ đó, sự ra đời của hàng loạt thiết bị tối tân trong ngành viễn thông, truyền hình, đặc biệt là sự ra đời của mạng Internet, góp phần đưa con người đến đỉnh cao của nền văn minh nhân loại. Đó chính là một nhu cầu cần thiết và cấp bách đối với mỗi sinh viên, đề tài này được thực hiện chính là đáp ứng nhu cầu đó.

- Các bộ vi điều khiển theo thời gian cùng với sự phát triển của công nghệ bán dẫn đã tiến triển rất nhanh, từ các bộ vi điều khiển 4 Bit đơn giản đến các bộ vi điều khiển 32 Bit. Với công nghệ tiên tiến ngày nay các máy tính có thể đi đến việc suy nghĩ, tri thức các thông tin đưa vào, đó là các máy tính thuộc thế hệ trí tuệ nhân tạo. Mặc dù vi điều khiển đã đi được những bước dài như vậy nhưng để tiếp cận được với kỹ thuật này không thể là một việc có được trong một sớm một chiều.

- Vì vậy, việc tìm hiểu vi điều khiển một cách khách quan đến chi tiết, việc trang bị những kiến thức vi điều khiển là hết sức cần thiết. Chính từ đây, nhóm em đã quyết định tìm hiểu kit STM32F407-Discovery điều khiển động cơ bước đóng mở chốt.

### **2.3 Mục tiêu đề tài**

- Hiểu được dòng vi xử lý STM32F407- Discovery

- Hiểu được nguyên lý hoạt động và cấu tạo của driver motor L298 và Stepper Motor

- Điều khiển được motor đóng mở chốt.

### **2.4 Đối tượng phạm vi nghiên cứu**

#### **a. Đối tượng**

- Dòng vi xử lý ARM Cortex STM32F407- Discovery

- Motor Driver L298 Dual H-Bridge Motor Driver

- Stepper Motor

- Dây cắm test board

#### **b. Phạm vi nghiên cứu**

Nghiên cứu dựa trên những kiến thức lập trình nhúng đã học. Cách nạp và chạy chương trình, hiểu được cấu trúc và tính chất của vi xử lý và board mạch, sử dụng ngôn ngữ C để lập trình trong Keil

### **2.5 Phương pháp nghiên cứu**

* + Từ quá trình học môn Hệ Thống Nhúng
  + Tham khảo tài liệu trên mạng

### **2.6 Nội dung đề tài**

Sử dụng kit STM32F407VG đề điểu khiển động cơ bước đóng mở chốt( yêu cầu là làm luôn khóa).

# 

# 

# **PHẦN 2: NỘI DUNG**

## CHƯƠNG 1: GIỚI THIỆU CHUNG VỀ DÒNG ARM CORTEX [KIT STM32F4 DISCOVERY](http://banlinhkien.vn/goods-2480-kit-stm32f4-discovery.html)

* Dòng ARM Cortex [KIT STM32F4 DISCOVERY](http://banlinhkien.vn/goods-2480-kit-stm32f4-discovery.html)  
  là một bộ xử lí thế hệ mới đua ra một kiến trúc chuẩn  
  cho nhu cầu đa dạng về công nghệ. Không giống nhu các chip ARM khác, dòng Cortex là một lõi xử lí hoàn thiện, dua ra một chuẩn CPU và kiến trúc hệ thống chung.  
  Dòng Cortex [KIT STM32F4 DISCOVERY](http://banlinhkien.vn/goods-2480-kit-stm32f4-discovery.html) gồm có 3 phân nhánh chính:
* Dòng A dành cho các ứng dụng cao cấp.
* Dòng R dành cho các ứng dụng thời gian thực nhu các đầu đọc và dòng  
  M dành cho các ứng dụng vi điều khiển và chi phí thấp. STM32 đuợc thiết kế  
  dựa trên dòng Cortex-M3.
* Dòng Cortex-M3 đuợc thiết kế đặc biệt để nâng cao hiệu suất hệ thống, kết hợp với tiêu thụ nang luợng thấp, CortexM3 đuợc thiết kế trên nền kiến trúc mới, do đó chi phí sản xuất đủ thấp để cạnh tranh với các dòng vi điều khiển 8 và 16-bit truyền thống.  
  [KIT STM32F4 DISCOVERY](http://banlinhkien.vn/goods-2480-kit-stm32f4-discovery.html) Các chip ARM7 và ARM9 đuợc các nhà sản xuất bán dẫn thiết kế với giải pháp riêng của mình, đặc biệt là phần xử lí các các ngắt đặc biệt (exception) và  
  các ngắt thông thuờng (interrupt). Cortex-M3 đưa ra một lõi vi điều khiển chuẩn  
  nhằm cung cấp phần tổng quát, quan trọng nhất của một vi điều khiển, bao gồm hệ thống ngắt (interrupt system), SysTick timer (đuợc thiết kế cho hệ điều hành thời gian thực), hệ thống kiểm lỗi (debug system) và memory map.  
  Không gian địa chỉ 4Gbyte của Cortex-M3 đuợc chia thành các vùng cho mã  
  chương trình, SRAM, ngoại vi và ngoại vi hệ thống.  
  [KIT STM32F4 DISCOVERY](http://banlinhkien.vn/goods-2480-kit-stm32f4-discovery.html) Không giống với ARM7 đuợc thiết kế theo kiến trúc Von Neumann (bộ nhớ chương trình và bộ nhớ dữ liệu chung với nhau), Cortex-M3 đuợc thiết kế dựa  
  theo kiến trúc Harvard (bộ nhớ chương trình và bộ nhớ dữ liệu tách biệt với  
  nhau), và có nhiều bus cho phép thực hiện các thao tác song song với nhau, do  
  đó làm tang hiệu suất của chip. Không giống với các kiến trúc ARM truớc đó,  
  dòng Cortex cho phép truy cập dữ liệu không xếp hàng (unaligned data, vì chip  
  ARM là kiến trúc 32bit, do đó tất cả các dữ liệu hoặc mã chuong trình đều đuợc  
  sắp xếp khít với vùng bộ nhớ là bội số của 4byte). Ðặc điểm này cho phép sử  
  dụng hiệu quả SRAM nội. Dòng Cortex còn hỗ trợ việc đặt và xoá các bit bên  
  trong hai vùng 1Mbyte của bộ nhớ bằng phuong pháp gọi là bit banding. [KIT STM32F4 DISCOVERY](http://banlinhkien.vn/goods-2480-kit-stm32f4-discovery.html) Ðặc điểm này cho phép truy cập hiệu quả tới các thanh ghi ngoại vi và các cờ đuợc  
  dùng trên bộ nhớ SRAM mà không cần một bộ xử lí luận lí (Boolean processor).

## CHƯƠNG 2 CẤU TRÚC CHUNG CỦA KIT

## 

**Thông số kỹ thuật**

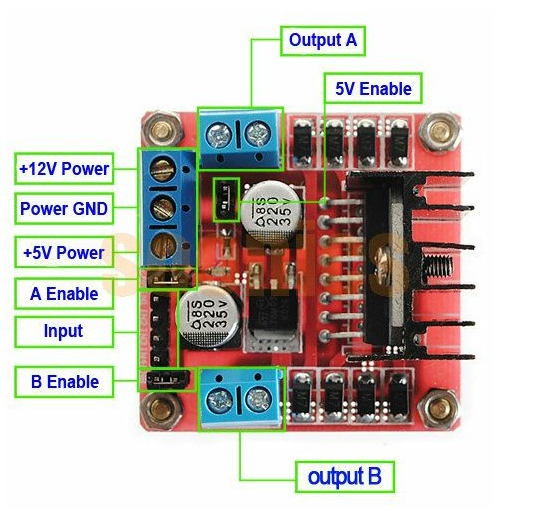
* Processor: Cortex M4, 32 bit ARM
* Memory: 1 MB
* Ram: 192 KB
* Power: USB bus or 5V, 3V electric
* Accelerometer: 3
* Button: user and reset
* LEDS: LD1(Red/Green), LD2(Red), LD3(orange), LD4(Green), LD5(Red), LD6(Blue), LD7(Green), LD8(Red).
* Sensor: MP45DT02 ST-MEMS audio sensor,
* USB OTG FS with micro-AB connector
* Interface: Debug port, Virtual com port, Mass storage.

GNO C 
PCI C 
STM32F4070VG 
24-bit Audin:DRC 
PCI S 
pot 1 
eoL3 
GNO 
Audlo 
-nttttlitiüüitttttiltn• 
MB997A 
GNO 
SV O sv 
PHO O O 
00 
O O Pts 
•n 00 
O woo 
Accelö,rometel 
POI O 
Chip Microphone 
O O PC9 
O GNO 
IJ Electronics 

円 iB 一 ) - 日 EOd 
目 ・ - 目 洋 
↑ 伺 ↓ 
ー 目 譽 - 目 4 目 
」 77 ー 当 「 一 き 斗 目 - ロ 朝 
HPZEX ( い 
( い 7 工 
」 ま っ 卩 、 目 、 目 . 卩 
日 . 目 、 -0 住 
0 ( 日 朝 
〕 、 J57 ミ 5 一 一 〔 〔 凵 
じ - に 】 Ⅵ Ova 、 一 工 国 国 当 
「 1 を 可 5 コ ArMd ー 
に El を 5 コ 

### 2. Điều khiển động cơ bước với Kit STM32F407- Discovery.

#### 2.1. Thiết kế mạch Motor Driver L298 Dual H-Bridge Motor Driver:



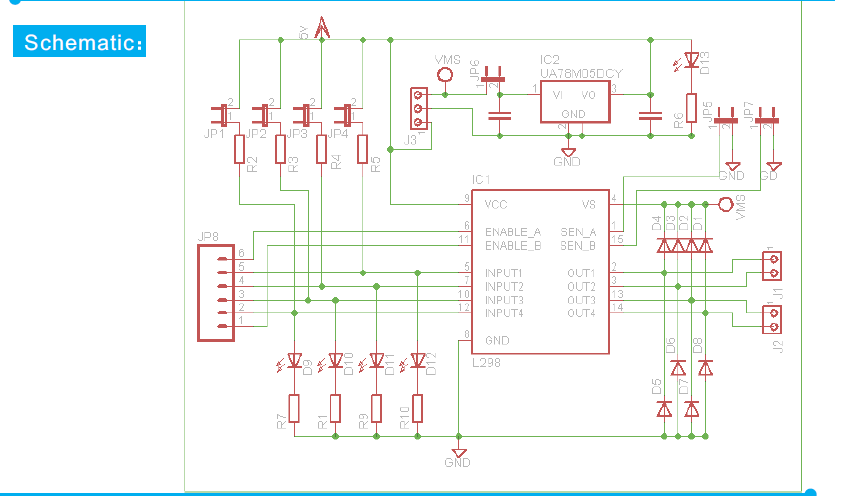
- Thông số kỹ thuật:

* Driver: L298N tích hợp hai mạch cầu H.
* Điện áp điều khiển: +5 V ~ +12 V
* Dòng tối đa cho mỗi cầu H là: 2A (=>2A cho mỗi motor)
* Điện áp của tín hiệu điều khiển: +5 V ~ +7 V
* Dòng của tín hiệu điều khiển: 0 ~ 36mA
* Công suất hao phí: 20W (khi nhiệt độ T = 75 ℃)
* Nhiệt độ bảo quản: -25 ℃ ~ +130 ℃

L298 gồm các chân:

* 12V power, 5V power. Đây là 2 chân cấp nguồn trực tiếp đến động cơ. Có thể cấp nguồn 9-12V ở 12V.
* Bên cạnh đó có jumper 5V, nếu để như hình ở trên thì sẽ có nguồn 5V ra ở cổng 5V power, ngược lại thì không.
* Power GND chân này là GND của nguồn cấp cho Động cơ.
* 2 Jump A enable và B enable
* Gồm có 4 chân Input. IN1, IN2, IN3, IN4.
* Có thể điều khiển động cơ bước 6 dây hoặc 4 dây

#### 2.2 Sơ đồ Motor Driver:

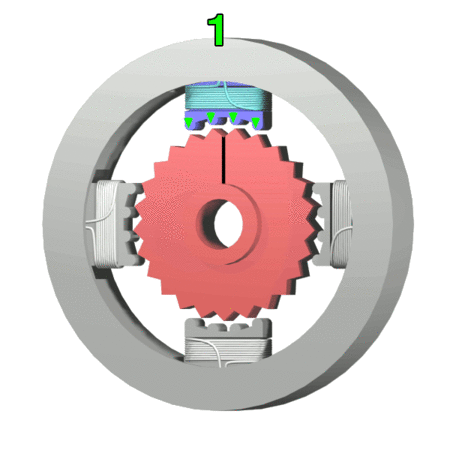
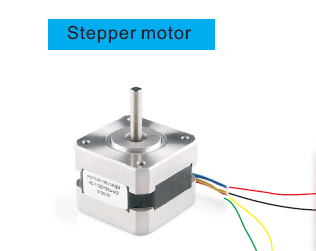


#### 2.3 Động cơ bước ( Stepper Motor)

 Là một loại động cơ điện có nguyên lý và ứng dụng khác biệt với đa số các động cơ điện thông thường. Chúng thực chất là một động cơ đồng bộ dùng để biến đổi các tín hiệu điều khiển dưới dạng các xung điện rời rạc kế tiếp nhau thành các chuyển động góc quay hoặc các chuyển động của rôto có khả năng cố định rôto vào các vị trí cần thiết.

**Nguyên lý hoạt động**

Động cơ bước không quay theo cơ chế thông thường, chúng quay theo từng bước nên có độ chính xác rất cao về mặt điểu khiển học. Chúng làm việc nhờ các bộ chuyển mạch điện tử đưa các tín hiệu điều khiển vào stato theo thứ tự và một tần số nhất định. Tổng số góc quay của rôto tương ứng với số lần chuyển mạch, cũng như chiều quay và tốc độ quay của rôto phụ thuộc vào thứ tự chuyển đổi và tần số chuyển đổi.



- Sử dụng Step Motor đơn cực 4 dây có góc bước 0,90/nguồn cấp 12V

Ráp mạch:

Ta nối dây Enable A của Driver với chân PA0 của Kit

Enable B của Driver với chân PA1 của Kit

INT1 của Driver với chân PB0 của Kit

INT2 của Driver với chân PB1 của Kit

INT3 của Driver với chân PB2 của Kit

INT4 của Driver với chân PB3 của Kit

Cực (+) của nguồn nối với chân VMS của Kit

Cực(-) của nguồn nối với chân GND của Kit

Nối các chân của motor với chân motor A và motor B của driver

## 

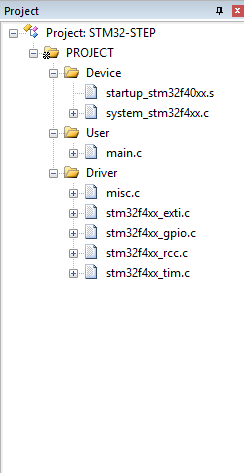
## **CHƯƠNG 4 ỨNG DỤNG**

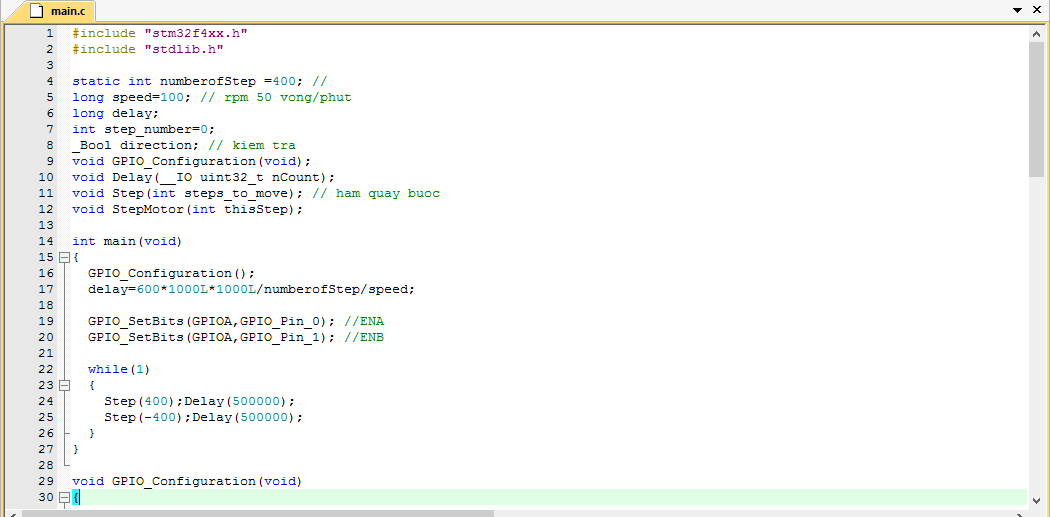
### 1. Phần mềm viết chương trình.

Chương trình được viết trên Keil C v5

Phần mềm có giao diện như sau:







### 2. Chương trình điều khiển Step Motor:

Sử dụng thư viện

#include "stm32f4xx.h"

#include "stdlib.h"

int main(void)

{

GPIO\_Configuration();

delay=600\*1000L\*1000L/numberofStep/speed;

GPIO\_SetBits(GPIOA,GPIO\_Pin\_0); //ENA set chân lên mức cao 3.3V

GPIO\_SetBits(GPIOA,GPIO\_Pin\_1); //ENB

while(1)

{

Step(200);Delay(500000); //quay thuận

Step(-200);Delay(500000); //quay ngược

// step ở đây là số bước. 200 bước tương đương với xoay 180 độ

}

}

// còn cái hàm này là set bus clock, dùng port B nên set cho bus AHB1

void GPIO\_Configuration(void)

{

RCC\_AHB1PeriphClockCmd(RCC\_AHB1Periph\_GPIOB | RCC\_AHB1Periph\_GPIOA, ENABLE); GPIO\_InitTypeDef GPIO\_InitStructure;

/\* Configure PB0 PB1 in output pushpull mode \*/

GPIO\_InitStructure.GPIO\_Pin = GPIO\_Pin\_0|GPIO\_Pin\_1|GPIO\_Pin\_2|GPIO\_Pin\_3;

GPIO\_InitStructure.GPIO\_Mode = GPIO\_Mode\_OUT;

GPIO\_InitStructure.GPIO\_OType = GPIO\_OType\_PP;

GPIO\_InitStructure.GPIO\_Speed = GPIO\_Speed\_100MHz;

GPIO\_InitStructure.GPIO\_PuPd = GPIO\_PuPd\_NOPULL;

GPIO\_Init(GPIOB, &GPIO\_InitStructure);

/\* Configure PA0 PA1 in output pushpull mode \*/

GPIO\_InitStructure.GPIO\_Pin = GPIO\_Pin\_0|GPIO\_Pin\_1;

GPIO\_InitStructure.GPIO\_Mode = GPIO\_Mode\_OUT;

GPIO\_InitStructure.GPIO\_OType = GPIO\_OType\_PP;

GPIO\_InitStructure.GPIO\_Speed = GPIO\_Speed\_100MHz;

GPIO\_InitStructure.GPIO\_PuPd = GPIO\_PuPd\_NOPULL;

GPIO\_Init(GPIOA, &GPIO\_InitStructure);

}

// hàm này đếm giờ gian delay

void Delay(\_\_IO uint32\_t nCount)

{

while(nCount--){}

}

// hàm này là điều kiện để động cơ di chuyển từng bước

void Step(int steps\_to\_move)

{

int steps\_left = abs(steps\_to\_move); // how many steps to take

// determine direction based on whether steps\_to\_mode is + or -:

if (steps\_to\_move > 0) { direction = 1; }

if (steps\_to\_move < 0) { direction = 0; }

while(steps\_left>0)

{

Delay(delay);

if(direction==1){

step\_number++;

if(step\_number==numberofStep){step\_number=0;}

}

else{

if(step\_number==0){step\_number=numberofStep;}

step\_number--;

}

steps\_left--;

StepMotor(step\_number % 4); // gọi hàm phía dưới

}

}

// hàm quay theo thứ tự các pha trong motor(4 pha)

void StepMotor(int thisStep)

{

switch (thisStep) {

case 0: // 1010 GPIO\_Write(GPIOB,0x0A);

break;//

case 1: // 0110

GPIO\_Write(GPIOB,0x06);

break;

case 2: //0101

GPIO\_Write(GPIOB,0x05);

break;

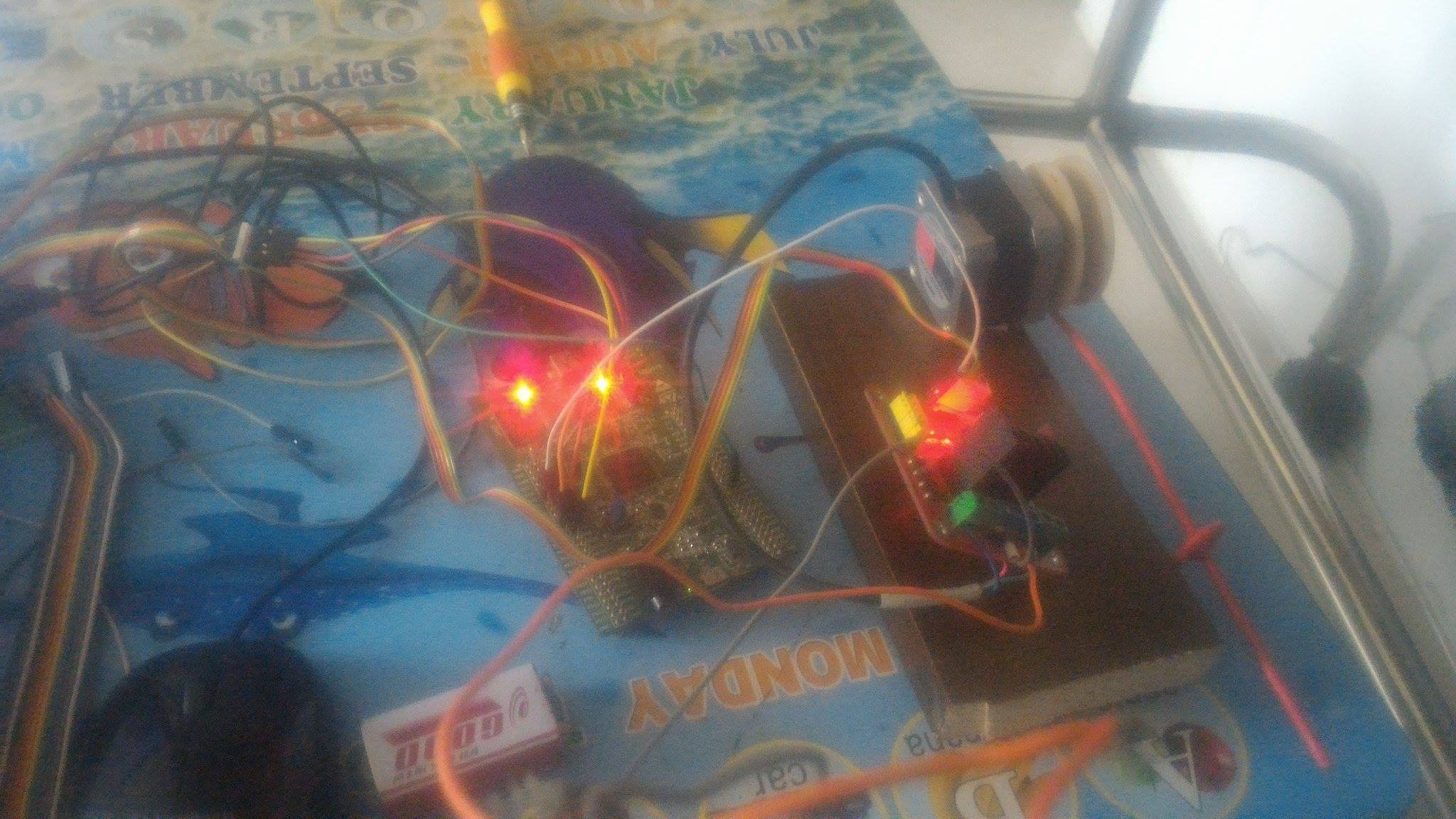
case 3: //1001

GPIO\_Write(GPIOB,0x09);

break;

}

}



# **Phần 3: Kết luận**

* Hệ thống nhúng đang là làn sóng đổi mới thứ ba trong công nghệ thông tin và truyền thông, trở thành quốc sách của nhiều nước trên thế giới trong giai đoạn hậu PC hiện nay.
* Sau sự phát triển của máy tính lớn mainframe và máy tính mini của thời kỳ 1960-1980, đến PC-Internet giai đoạn 1980-2000, thế giới đang bước vào giai đoạn các thiết bị đều được thông minh hoá và kết nối với nhau. Hệ thống nhúng đang làm nên làn sóng đổi mới thứ ba trong sự phát triển của công nghệ thông tin (CNTT).
* Thuật ngữ "hệ thống nhúng" (embedded system) mới xuất hiện ở Việt Nam một vài năm trở lại đây và nhiều người không hiểu nhúng là gì. Hệ thống nhúng giống như máy tính được thiết kế để thực hiện một vài chức năng đặc thù nào đó. Nó được “nhúng” vào một thiết bị hoàn chỉnh và thường là cả phần mềm và phần cứng.

1.Ưu điểm

* Qua đề tài, cho thấy dòng ARM Cortex là một bộ xử lí thế hệ mới đưa ra một kiến trúc chuẩn cho nhu cầu đa dạng về công nghệ. Không giống như các chip ARM khác, dòng Cortex là một lõi xử lí hoàn thiện, đưa ra một chuẩn CPU và kiến trúc hệ thống chung.
* Từ đề tài board STM32F407- Discovery giao tiếp cảm biến nhiệt độ, giúp sinh viên có nhiều hướng suy nghĩ đến ứng dụng thực tế vào cuộc sống hiện tại, hoàn thiện các kĩ năng kể cả phần cứng và phần mềm, phát triển cả về tư duy và kĩ năng tìm hiểu, ráp mạch, lập trình và còn nhiều hơn nữa, …

2.Nhược điểm

- Vì chuyên ngành là IT nên đọc các tài liệu chuyên ngành Điện tử khá khó khăn

- Chỉ mô phỏng mạch ở một số giao tiếp cơ bản nhất chứ không làm những ứng dụng phong phú hơn vì quy trình làm mạch thủ công khá khó khăn đối với 1 sinh viên IT chập chững làm quen với thiết bị vi điều khiển.

## 

# **Tài liệu tham khảo**

<http://www.st.com/en/evaluation-tools/stm32f4discovery.html>

<https://www.youtube.com/watch?v=bngx2dKl5jU>