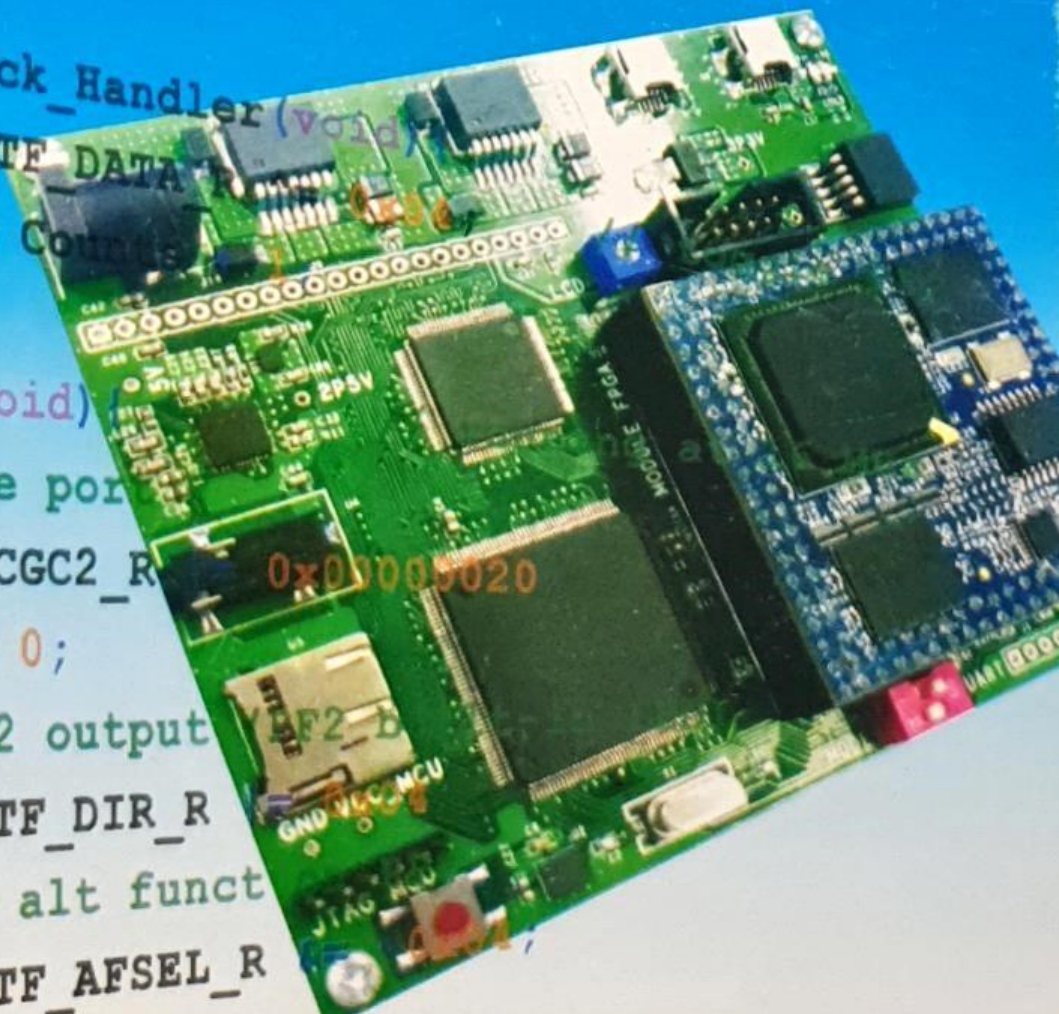


HOÀNG TRẠNG - BÙI QUỐC BẢO

# LẬP TRÌNH HỆ THỐNG NHÚNG

```
void SysTick_Handler(void)
{
    GPIO_PORTF_DATA_OUT_R = Counts;
    Counts = Counts + 1;
}

int main(void)
// activate port F
SYSCTL_RCGC2_R = 0x00000020;
Counts = 0;
// make PF2 output
GPIO_PORTF_DIR_R |= 0x00000004;
// disable alt function
GPIO_PORTF_AFSEL_R = 0;
```



NHÀ XUẤT BẢN

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH



**ĐẠI HỌC QUỐC GIA TP HỒ CHÍ MINH  
TRƯỜNG ĐẠI HỌC BÁCH KHOA**

**Hoàng Trang - Bùi Quốc Bảo**

**LẬP TRÌNH  
HỆ THỐNG NHÚNG**

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN VÀ MÔI TRƯỜNG THÔNG TIN THƯ VIỆN  
01 12  
01269

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN VÀ MÔI TRƯỜNG THÔNG TIN THƯ VIỆN  
01 12  
04033

**NHÀ XUẤT BẢN ĐẠI HỌC QUỐC GIA  
TP HỒ CHÍ MINH - 2016**



# MỤC LỤC

<b>Chương 1</b>	<b>TỔNG QUAN</b>	<b>7</b>
1.1	Mở đầu	7
1.2	Yêu cầu phần cứng	7
1.3	Cài đặt phần mềm	8
<b>Chương 2</b>	<b>CƠ BẢN VỀ LẬP TRÌNH C CHO HỆ THỐNG NHÚNG</b>	<b>9</b>
2.1	Giới thiệu	9
2.2	Các vấn đề cơ bản về C cho hệ thống nhúng	10
<b>Chương 3</b>	<b>CẤU TRÚC CHƯƠNG TRÌNH</b>	<b>53</b>
3.1	Vai trò của các hàm (function)	53
3.2	Sử dụng lưu đồ để thiết kế cấu trúc chương trình	54
3.3	Giả mã (pseudo code)	56
3.4	Chia chương trình thành các module	57
3.5	Quá trình biên dịch chương trình	58
3.6	Chi dẫn tiền biên dịch	59
<b>Chương 4</b>	<b>GIỚI THIỆU DÒNG VI ĐIỀU KHIỂN LM4F</b>	<b>69</b>
4.1	Kiến trúc lõi Cortex M3 và Cortex M4F	70
4.2	Lõi CM3	72
4.3	Họ vi điều khiển LM4F	75
<b>Chương 5</b>	<b>LẬP TRÌNH VỚI LM4F120</b>	<b>79</b>
5.1	Giới thiệu kit Stellaris LaunchPad	79
5.2	Giới thiệu về các ngoại vi của LM4F120	81
5.3	Sử dụng phần mềm KEIL để viết và thử nghiệm chương trình	89
5.4	Mô phỏng chương trình với KEIL	99
5.5	Lập trình GPIO cho LM4F120	102
5.6	Cấu trúc xung nhịp trên chip LM4F120 và trên kit LaunchPad	109
5.7	Thư viện StellarisWare	112
5.8	Bit-bang	114
5.9	Đồng bộ I/O	115



<b>Chương 6 NGẮT</b>	119
6.1 Các khái niệm cơ bản về ngắt	119
6.2 Thông tin và đồng bộ giữa các thread	120
6.3 Tạo xung dừng ngắt timer 0	123
6.4 Mô hình ngắt của lõi ARM Cortex M4	126
6.5 Ngắt System Tick	140
6.6 Ngắt ngoài	148
6.7 Chạy đua	153
<b>Chương 7 TRUYỀN THÔNG NỘI TIẾP BẤT ĐỒNG BỘ (UART)</b>	155
7.1 Cấu trúc UART của LM4F120	156
7.2 Sử dụng UART0 của Kit Stellaris Launchpad	157
7.3 Lập trình điều khiển UART của LM4F120	159
7.4 Truyền nhận dữ liệu với UART0	162
7.5 Lập trình UART với thư viện StellarisWare	164
7.6 Lập trình UART sử dụng ngắt	166
7.7 Sử dụng UART trong lập trình và gỡ rối	169
<b>Chương 8 LẬP TRÌNH VỚI MÁY TRẠNG THÁI</b>	
8.1 Khái niệm về máy trạng thái	173
8.2 Lập trình hệ thống sử dụng máy trạng thái	175
8.3 Máy trạng thái có sự kiện thời gian	177
8.4 Xử lý đa tác vụ với máy trạng thái	183
8.5 Máy trạng thái có ngõ vào là ngõ ra máy trạng thái khác (hierachy state machine)	189
8.6 Thời gian đáp ứng của chương trình	198
<b>Chương 9 HỆ ĐIỀU HÀNH THỜI GIAN THỰC</b>	201
9.1 Giới thiệu FreeRTOS	201
9.2 Thread trong FreeRTOS	205
9.3 Quản lý tác vụ trong FreeRTOS	206
9.4 Quản lý hàng đợi (queue) trong FreeRTOS	233
9.5 Quản lý ngắt trong FreeRTOS	243
9.6 Quản lý tài nguyên trong FreeRTOS	264
9.7 Quản lý bộ nhớ	282
9.8 Cấu trúc file và thư mục của FreeRTOS	285



## LỜI MỞ ĐẦU

Tựa đề cuốn sách này là Lập trình hệ thống nhúng. Vậy vấn đề quan tâm đầu tiên có lẽ là hệ thống nhúng là gì? Thực ra rất khó định nghĩa chính xác, cô đọng khái niệm về hệ thống nhúng, một khái niệm được nhắc đến và sử dụng phổ biến trên thế giới khoảng hơn mười năm nay. Trong các định nghĩa, xác định về hệ thống nhúng, một định nghĩa sau có lẽ là phổ biến, dễ chấp nhận hơn cả là: hệ thống nhúng là một thuật ngữ chuyên môn để chỉ một hệ thống có tích hợp phần cứng và phần mềm phục vụ cho mục đích chuyên dụng trong nhiều lĩnh vực công nghiệp, tự động, ... và có khả năng tự hoạt động một chức năng nhất định và có khả năng được "nhúng" vào trong một môi trường, một hệ thống lớn hơn khác. Khác với các máy tính đa chức năng, hệ thống nhúng chỉ thực hiện: một hoặc một vài chức năng chuyên dụng nhất định và thường cần phải đạt yêu cầu cụ thể, cần được xây dựng nhằm giảm thiểu chi phí, kích thước phù hợp với chức năng của nó.

Khi phát triển các hệ thống nhúng phục vụ các mục đích khác nhau, tựu chung có một số cách tiếp cận chính như sau: tiếp cận dựa trên chip chuyên dụng ASIC; tiếp cận dựa trên các dòng vi xử lý, vi điều khiển; tiếp cận dựa trên các dòng FPGA, CPLD; tiếp cận dựa trên các dòng DSP. Các đội ngũ phát triển hệ thống nhúng, hay nói rộng hơn là phát triển sản phẩm điện tử thông thường sẽ dựa trên các tiêu chí sau để chọn lựa, quyết định cách tiếp cận nào là phù hợp: thời gian phát triển sản phẩm, số lượng sản phẩm, chi phí phát triển sản phẩm, thời gian đáp ứng sự thay đổi nhu cầu của người sử dụng, thay đổi cấu hình-tính năng sản phẩm dễ dàng, các ngoại vi liên quan để thực hiện sản phẩm ....

Trong các cách tiếp cận ở trên, cách dựa trên các dòng vi xử lý, vi điều khiển được chọn lựa chủ yếu dựa trên yếu tố thời gian phát triển sản phẩm-hệ thống nhanh chóng, đáp ứng nhanh sự thay đổi liên tục của người dùng, sự dễ dàng trong chọn lựa các dòng vi xử lý, vi điều khiển rất đa dạng trên thế giới. Cách tiếp cận này có thể nói là cân bằng khi xét đến hầu hết các tiêu chí so với những cách tiếp cận còn lại; đặc biệt trong điều kiện tại Việt Nam hiện tại, khi chưa có nhà máy sản xuất chip.

Hệ thống nhúng dựa trên vi xử lý, vi điều khiển để hoạt động được cần thiết phải xây dựng phần mềm nhúng bên trong các vi xử lý, vi điều khiển. Như đã nói ở trên, bởi hệ thống nhúng là phục vụ cho một hoặc vài chức năng chuyên dụng, nhỏ gọn; nên phần mềm được xây dựng và nhúng trong hệ thống thường chạy được với ràng buộc tài nguyên phần cứng hạn chế, yêu cầu thời gian đáp ứng nhanh. Có thể nói rằng, xây dựng phần mềm nhúng trong hệ thống nhúng hay còn gọi là lập trình hệ thống nhúng đóng vai trò rất quan trọng cho sự thành công của một hệ thống nhúng bất kỳ.



Cuốn sách này trình bày về lập trình hệ thống nhúng dựa trên vi xử lý, vi điều khiển. Vi xử lý, vi điều khiển với phần mềm nhúng bên trong đóng vai trò trung tâm, kết nối các ngoại vi và xử lý dữ liệu. Cuốn sách này nhằm mục đích cung cấp các kiến thức cần thiết cho việc lập trình cho các dòng vi xử lý, vi điều khiển thông dụng, đặc biệt là ARM, tập trung vào kỹ năng sử dụng ngôn ngữ lập trình C cho hệ thống nhúng. Các ví dụ trong cuốn sách này sử dụng vi xử lý LM4F của TI và board Stellaris EK-LM4F120XL LaunchPad. Tuy nhiên, các kiến thức và kỹ thuật lập trình được cung cấp có thể được ứng dụng cho các dòng vi xử lý khác như 8051, PIC hay các dòng ARM Cortex của các hãng khác như ST, NXP.

Cuốn sách nhằm phục vụ cho các sinh viên, học viên cao học chuyên ngành Kỹ thuật điện tử và cho cả các ngành khác liên quan bởi tính liên ngành-đa ngành trong lĩnh vực kỹ thuật hiện nay là rất phổ biến. Các lĩnh vực, ứng dụng của hệ thống nhúng là rất đa dạng, cuốn sách này được viết không có tham vọng trình bày hết mọi vấn đề liên quan, mà chỉ để phục vụ cho người đọc trong phạm vi cơ bản ban đầu, với thời lượng giảng tại lớp khoảng 45-60 tiết, thời lượng thực hành và tự học khoảng 90 tiết.

Tác giả chúng tôi chân thành cảm ơn sự hỗ trợ, đóng góp quý báu của công ty Texas Instruments Việt Nam đối với khoa Điện-Điện Tử, trường Đại Học Bách Khoa - Đại học Quốc gia TP. Hồ Chí Minh. Chúng tôi chân thành cảm ơn những ý kiến đóng góp, giúp đỡ quý giá của thầy Hồ Trung Mỹ, của đồng nghiệp, của các bạn nghiên cứu viên tại bộ môn Kỹ thuật Điện Tử, khoa Điện - Điện Tử. Bên cạnh đó, chân thành cảm ơn tô giáo trình của trường Đại học Bách khoa - Đại học Quốc gia TP. Hồ Chí Minh đã hỗ trợ để hoàn chỉnh cuốn sách. Và cuối cùng, không thể không nhắc đến sự tài trợ kinh phí nghiên cứu qua đề tài KC.01/11-15 của Bộ Khoa học và Công nghệ Việt Nam.

Trong quá trình viết cuốn sách này, chúng tôi nhận thức rằng không thể tránh khỏi các sai sót. Rất mong nhận được sự lượng thứ của bạn đọc và mong nhận được các góp ý, phản hồi để cuốn sách hoàn thiện hơn.

Mọi chi tiết góp ý, xin vui lòng liên hệ bộ môn Kỹ thuật Điện Tử, khoa Điện - Điện Tử, trường Đại học Bách khoa - Đại học Quốc gia TP. Hồ Chí Minh, 268 Lý Thường Kiệt, quận 10, TP. Hồ Chí Minh; hoặc email: hoangtrang@hcmut.edu.vn, buiquochao@hcmut.edu.vn.