

BỘ GIÁO DỤC VÀ ĐÀO TẠO
ĐẠI HỌC ĐÀ NẴNG

BÙI VĂN THÀNH TRUNG

**THIẾT KẾ THI CÔNG HỆ THỐNG GIÁM SÁT
VÀ ĐIỀU KHIỂN THIẾT BỊ BẰNG GPRS
ỨNG DỤNG CÔNG NGHỆ FPGA**

Chuyên ngành: KỸ THUẬT ĐIỆN TỬ
Mã số: 60.52.70

TÓM TẮT LUẬN VĂN THẠC SĨ KỸ THUẬT

Đà Nẵng – Năm 2013

Công trình được hoàn thành tại
ĐẠI HỌC ĐÀ NẴNG

Người hướng dẫn khoa học: TS NGUYỄN VĂN CƯỜNG

Phản biện 1: TS. HUỖNH VIỆT THẮNG

Phản biện 2: PGS. TS. NGUYỄN HỮU THANH

Luận văn đã được bảo vệ trước Hội đồng chấm Luận văn tốt nghiệp Thạc sĩ Kỹ thuật họp tại Đại học Đà Nẵng vào ngày 02 tháng 06 năm 2013.

Có thể tìm hiểu luận văn tại:

– Trung tâm Thông tin - Học liệu, Đại học Đà Nẵng

MỞ ĐẦU

1. TÍNH CẤP THIẾT CỦA ĐỀ TÀI

Hệ thống nhúng sử dụng vi điều khiển có những ưu thế về tính linh hoạt, độ tin cậy cao và giá thành rẻ. Nhưng đối với những yêu cầu khắt khe đòi hỏi hiệu suất cao thì phần lớn những vi điều khiển không đáp ứng nổi về tốc độ tính toán cũng như tính linh hoạt. Đồng thời, vi điều khiển thực thi nhiệm vụ một cách tuần tự thế nên cần đến thời gian lâu hơn để hoàn thành những nhiệm vụ.

FPGA với khả năng xử lý song song và khả năng tái lập trình cho phép thiết kế hệ thống trên FPGA linh hoạt và tối ưu. Do các yêu cầu khắt khe về thời gian, về việc sử dụng tài nguyên và sự quan trọng của việc lập lịch, các hệ điều hành thời gian thực (RTOS) đóng vai trò rất quan trọng trong phát triển hệ thống nhúng. Hệ điều hành thời gian thực FreeRTOS là một hệ điều hành nhỏ gọn nhưng vẫn đầy đủ các tính năng của một hệ điều hành thời gian thực. Đề tài hướng đến xây dựng một hệ thống nhúng trên nền FPGA với MCU Microblaze Xilinx chạy trên hệ điều hành thời gian thực FreeRTOS để giải quyết bài toán điều khiển và giám sát thiết bị qua GPRS.

2. MỤC TIÊU NGHIÊN CỨU

- Nghiên cứu nhúng MCU Microblaze và hệ điều hành thời gian thực FreeRTOS xuống KIT Xilinx Spartan-6 FPGA SP605.
- Nghiên cứu, xây dựng bảng mạch điện tử cho hệ thống giám sát và điều khiển thiết bị bằng GPRS ứng dụng công nghệ FPGA.
- Thi công phần mềm hệ thống trên KIT Xilinx Spartan-6 FPGA SP605.

3. ĐỐI TƯỢNG VÀ PHẠM VI NGHIÊN CỨU

- Vi xử lý nhúng MicroBlaze.
- Hệ điều hành thời gian thực FreeRTOS.
- Công nghệ FPGA và ngôn ngữ mô tả phần cứng.
- Thực hiện trên KIT FPGA của hãng Xilinx.
- Thi công bảng mạch điện tử và phần mềm hệ thống.

4. PHƯƠNG PHÁP NGHIÊN CỨU

- Thu thập tài liệu, chọn lọc và phân tích các thông tin liên quan đến nội dung nghiên cứu của đề tài.
- Nghiên cứu tài liệu về FreeRTOS để viết chương trình và chạy mô phỏng và nạp lên FPGA KIT.
- Thi công và kiểm tra thực nghiệm trên mạch in cho các khối điều khiển và giám sát.

5. BỐ CỤC ĐỀ TÀI

Luận văn gồm có 4 chương:

- Chương 1: CÔNG NGHỆ FPGA CỦA XILINX VÀ VI XỬ LÝ MỀM MICROBLAZE

Mục đích của phần này là để cung cấp cho chúng ta một cái nhìn tổng quan toàn diện về công nghệ FPGA, về ngôn ngữ mô tả phần cứng và vi xử lý mềm MicroBlaze của hãng Xilinx. Giới thiệu về kiến trúc cơ bản, về giao tiếp tín hiệu của vi xử lý mềm MicroBlaze.

Trên cơ sở lý thuyết đã trình bày để ứng dụng vào phần thiết kế hệ thống cụ thể ở chương 4.

- Chương 2: HỆ ĐIỀU HÀNH THỜI GIAN THỰC FREERTOS

Trình bày các kiến thức cơ sở lý thuyết liên quan đến khái niệm, về đặc điểm, về tầm quan trọng và về các hệ điều hành thời gian thực phổ biến.

Trình bày về quản lý tác vụ, về hàng đợi, về timer trong hệ điều hành thời gian thực FreeRTOS.

- Chương 3: THIẾT KẾ PHẦN CỨNG ĐIỀU KHIỂN

Nội dung của chương này sẽ tập trung vào việc thiết kế và thi công phần cứng của hệ thống.

- Chương 4: THIẾT KẾ PHẦN MỀM HỆ THỐNG

Trong chương này trình bày về sơ đồ thuật toán, về chương trình của đề tài.

Ngoài ra, trong chương này sẽ thực hiện quá trình tổng hợp thiết kế và nhúng hệ thống xuống KIT SP605.

6. TỔNG QUAN TÀI LIỆU NGHIÊN CỨU

Tài liệu nghiên cứu được tham khảo là những bài giảng, bài báo, các luận văn thạc sĩ từ các trường đại học, cùng với tài liệu từ các nhà sản xuất và các trang web.

CHƯƠNG 1 CÔNG NGHỆ FPGA CỦA XILINX VÀ VI XỬ LÝ MỀM MICROBLAZE

1.1. GIỚI THIỆU VỀ CÔNG NGHỆ FPGA

1.1.1. Tổng quan về công nghệ FPGA

Công nghệ FPGA cho phép cấu hình lại cấu trúc mạch logic để thay đổi chức năng hệ thống. Khả năng này cho phép nhà thiết kế có thể dễ dàng phát triển, thay đổi chức năng phần cứng giống như đang thực hiện trên phần mềm. Thiết kế và kiểm tra phần cứng nhanh chóng.

1.1.2. FPGA của hãng Xilinx

Xilinx là hãng cung cấp FPGA hàng đầu thế giới với nhiều dòng sản phẩm khác nhau. Kiến trúc của chip FPGA Spartan-6 của hãng Xilinx được mô tả như hình 1.1 (xem cuốn luận văn).

a. Khối Logic

b. Kết nối cấu hình được

c. Mạch xung clock hệ thống

d. Các phân tử tích hợp sẵn

1.1.3. Ứng dụng của FPGA

FPGA được ứng dụng điển hình trong các lĩnh vực như: xử lý tín hiệu số, tiền thiết kế mẫu ASIC (ASIC prototyping), các hệ thống điều khiển trực quan, phân tích nhận dạng ảnh, nhận dạng giọng nói, mật mã học, mô hình phần cứng máy tính, mô phỏng (emulation)... FPGA đặc biệt mạnh trong các lĩnh vực hoặc ứng dụng mà kiến trúc của nó yêu cầu một lượng rất lớn xử lý song song, đặc biệt là mã hóa và giải mã. FPGA cũng được sử dụng trong những ứng dụng cần thực thi các thuật toán như FFT, nhân chập, thay thế cho vi xử lý.

1.2. NGÔN NGỮ MÔ TẢ PHẦN CỨNG VERILOG

1.2.1. Tổng quát về Verilog

Verilog được sử dụng phổ biến trong ngành công nghiệp thiết kế phần cứng, có khả năng mô tả mạch số theo những mức độ khác nhau: mức thuật toán, mức thanh ghi và mức cổng logic.

1.2.2. Cấu trúc chương trình và kiểu dữ liệu của ngôn ngữ

Verilog

a. Cấu trúc một chương trình Verilog

b. Kiểu dữ liệu

c. Câu lệnh trong ngôn ngữ Verilog

d. Câu lệnh gán assign

e. Câu lệnh always

1.3. LỖI VI XỬ LÝ MỀM MICROBLAZE

1.3.1. Giới thiệu về MicroBlaze

MicroBlaze là lõi vi xử lý mềm 32 bit phát triển bởi hãng Xilinx [11]. Người thiết kế có thể thiết lập các thông số cho vi xử lý MicroBlaze và kết nối với các ngoại vi (UART, GPIO, Ethernet MAC...) thông qua phần mềm EDK.

Lõi vi xử lý MicroBlaze được xây dựng theo kiến trúc Harvard, với tập lệnh thu gọn RISC. MicroBlaze có các bus riêng biệt để truy xuất dữ liệu và lệnh từ bộ nhớ on-chip và bộ nhớ ngoài tại cùng một thời điểm [11].

1.3.2. Kiến trúc cơ bản của MicroBlaze

a. Kiến trúc lõi MicroBlaze

Lõi của vi xử lý MicroBlaze (hình 1.8) gồm có:

- 32 thanh ghi mục đích chung kích thước 32 bit.
- Các thanh ghi mục đích đặc biệt kích thước 32 bit.
- Bộ đệm lệnh, bộ giải mã lệnh, bộ đếm chương trình.

➤ Khối quản lý bộ nhớ MMU.

➤ Giao tiếp Bus.

➤ Các khối xử lý: Đơn vị dấu phẩy động, bộ chia, bộ nhân, các khối dịch và khối logic và số học ALU thực hiện các phép toán số học và logic.

b. Kiến trúc bộ nhớ

Bộ xử lý MicroBlaze được xây dựng theo kiến trúc Harvard. Trong kiến trúc Harvard, sự truy xuất lệnh và truy xuất dữ liệu được tách riêng biệt với nhau trong vùng không gian địa chỉ. Mỗi không gian địa chỉ là 32 bit (MicroBlaze cung cấp tới 2^{32} byte = 4Gbyte địa chỉ truy xuất lệnh và dữ liệu). Vùng nhớ lệnh và vùng nhớ địa chỉ được ánh xạ tới cùng một ô nhớ vật lý.

MicroBlaze không phân biệt truy cập dữ liệu tới thiết bị ngoại vi và tới bộ nhớ.

c. Kiểu dữ liệu trong MicroBlaze

d. Hoạt động ngắt trong MicroBlaze

1.3.3. Các giao tiếp tín hiệu của MicroBlaze

a. Giới thiệu chung

b. Bus xử lý nội PLB

c. Bus nhớ nội LMB

d. Bus ngoại vi on-chip OPB

e. Bus AXI

f. Bus XCL

KẾT LUẬN CHƯƠNG

CHƯƠNG 2

HỆ ĐIỀU HÀNH THỜI GIAN THỰC FREERTOS

2.1. GIỚI THIỆU VỀ HỆ ĐIỀU HÀNH THỜI GIAN THỰC

2.1.1. Hệ điều hành thời gian thực

Hệ điều hành thực chất chính là một giao diện quan trọng, giao tiếp trực tiếp với từng phần cứng cấp thấp phục vụ cho cả người sử dụng cũng như các chương trình ứng dụng thực thi trên nền phần cứng hệ thống. Hơn nữa hệ điều hành còn có vai trò quan trọng trong việc đảm nhiệm 3 tác vụ nguyên lý chính: (1) Quản lý quá trình, (2) Quản lý tài nguyên, (3) Bảo vệ tài nguyên khỏi sự xâm phạm của các quá trình thực thi sai [4].

a. Khái niệm hệ điều hành thời gian thực

b. Các đặc điểm của hệ điều hành thời gian thực

c. Các loại hệ điều hành thời gian thực

d. Tầm quan trọng hệ điều hành thời gian thực

e. Các hệ điều hành thời gian thực phổ biến

2.1.2. Hệ điều hành FreeRTOS

FreeRTOS là lõi của hệ điều hành thời gian thực miễn phí. Hệ điều hành này được Richard Barry công bố rộng rãi từ năm 2003, phát triển mạnh đến nay và được cộng đồng mạng mã nguồn mở ủng hộ [18]. Mục đích của FreeRTOS là khả năng linh động, mã nguồn mở, là một hệ điều hành thời gian thực nhỏ có thể được thao tác trong chế độ ưu tiên cũng như phối hợp.

2.2. QUẢN LÝ TÁC VỤ TRONG FREERTOS

2.2.1. Tác vụ trong FreeRTOS

❖ Các đặc điểm của FreeRTOS [18]:

- Lõi FreeRTOS hỗ trợ cả chế độ ưu tiên (preemptive) và phối hợp (cooperative) và các lựa chọn cấu hình lai giữa hai phần.
- SafeRTOS là sản phẩm dẫn xuất, cung cấp mã nguồn riêng ở mức độ cao.
- Được thiết kế nhỏ, đơn giản và dễ sử dụng.
- Cấu trúc mã nguồn rất linh động được viết bằng ngôn ngữ C.
- Hỗ trợ cả tác vụ và đồng thủ tục.
- Có lựa chọn nhận biết tràn ngăn xếp.
- Không giới hạn số tác vụ có thể tạo ra, phụ thuộc vào tài nguyên của chip.
- Không giới hạn số mức ưu tiên được sử dụng.
- Không giới hạn số tác vụ cùng một mức ưu tiên.
- Hỗ trợ truyền thông và đồng bộ giữa các tác vụ hoặc giữa tác vụ và ngắt: queues, binary semaphores, counting semaphores và recursive mutexes.
- Mutexes với ưu tiên kế thừa.
- Hỗ trợ 33 kiến trúc vi xử lý khác nhau.
- Miễn phí mã nguồn phần mềm nhúng.
- Tiền cấu hình cho các ứng dụng demo, từ đó dễ dàng tìm hiểu và phát triển.

a. Kernel của hệ điều hành

Thuật ngữ kernel được dùng để chỉ đến một thành phần lõi bên trong của một hệ điều hành. Mỗi một chương trình đang thi hành là một nhiệm vụ được phân chia điều khiển bởi hệ điều hành. Nếu một hệ điều hành có khả năng thi hành nhiều tác vụ thì được gọi là đa nhiệm.

b. Tác vụ

Thường thì các bộ vi xử lý chỉ có thể thực hiện một tác vụ duy nhất trong một thời điểm nhưng với sự chuyển đổi một cách rất nhanh giữa các tác vụ của một hệ điều hành đa nhiệm làm cho chúng dường như được chạy đồng thời với nhau.

c. Lập lịch

Bộ lịch trình là một phần của nhân hệ điều hành chịu trách nhiệm quyết định nhiệm vụ nào sẽ được thi hành tại một thời điểm. Nhân có thể cho dừng một tác vụ và phục hồi lại tác vụ sau đó nhiều lần trong suốt quá trình sống của tác vụ đó.

d. Chuyển đổi ngữ cảnh

Khi một tác vụ đang thi hành, nó sẽ sử dụng các thanh ghi và truy cập vào ROM, RAM như các tác vụ khác. Những tài nguyên này bao gồm : thanh ghi, RAM, ROM, ngăn xếp... gọi là ngữ cảnh thực thi nhiệm vụ của một tác vụ. Một tác vụ là một đoạn mã liên tục, nó sẽ không biết và không được báo trước nếu bị dừng hoặc được phục hồi bởi kernel.

e. Bộ lập lịch thời gian thực

Hệ điều hành RTOS trước tiên tự tạo cho nó một tác vụ gọi là Idle Task, tác vụ này chỉ thực thi khi không có tác vụ nào có khả năng thực thi. Tác vụ Idle của hệ RTOS luôn ở trạng thái sẵn sàng hoạt động.

2.2.2. Các hàm API**2.3. HÀNG ĐỢI TRONG FREERTOS****2.3.1. Hàng đợi**

Hàng đợi là phương thức chính để giao tiếp giữa các tác vụ với nhau trong FreeRTOS, hàng đợi có thể được sử dụng để gửi các thông điệp giữa các tác vụ, và giữa ngắt với tác vụ. Trong hầu hết

trường hợp, hàng đợi được sử dụng như một luồng dữ liệu an toàn theo kiểu vào trước, ra trước [7][18].

Việc sử dụng hàng đợi giúp hệ thống đơn giản và linh hoạt, các thông điệp được gửi bằng cách sao chép, có nghĩa là dữ liệu sẽ tự sao chép vào trong hàng đợi mà không phải luôn luôn lưu ở một vị trí biết trước.

2.3.2. Các hàm API**2.4. TIMER TRONG FREERTOS****2.4.1. Giới thiệu về timer**

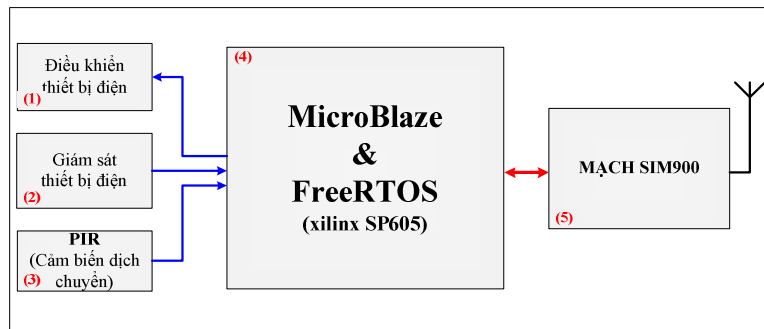
Timer được sử dụng trong FreeRTOS là timer mềm (software timer), timer này cho phép một hàm được thực thi ở một thời điểm xác định trong tương lai, tác vụ được thực thi được gọi thông qua hàm callback, thời gian tính từ khi timer bắt đầu thực thi đến khi hàm callback được thực thi được gọi là chu kỳ của timer. Có thể hiểu đơn giản là hàm callback sẽ được gọi khi timer đã hết một chu kỳ [7][18].

2.4.2. Các hàm API**2.5. QUẢN LÝ TÀI NGUYÊN TRONG FREERTOS****2.5.1. Semaphore và Mutex****2.5.2 Quản lý bộ nhớ****KẾT LUẬN CHƯƠNG**

CHƯƠNG 3 THIẾT KẾ PHẦN CỨNG ĐIỀU KHIỂN

3.1. SƠ ĐỒ MẠCH HỆ THỐNG

Đề tài hướng đến xây dựng một hệ thống nhúng trên nền FPGA với vi xử lý mềm MicroBlaze và hệ điều hành thời gian thực FreeRTOS để giải quyết bài toán điều khiển và giám sát thiết bị qua GPRS. Hệ thống được mô tả như hình 3.1:



Hình 3.1: Sơ đồ hệ thống

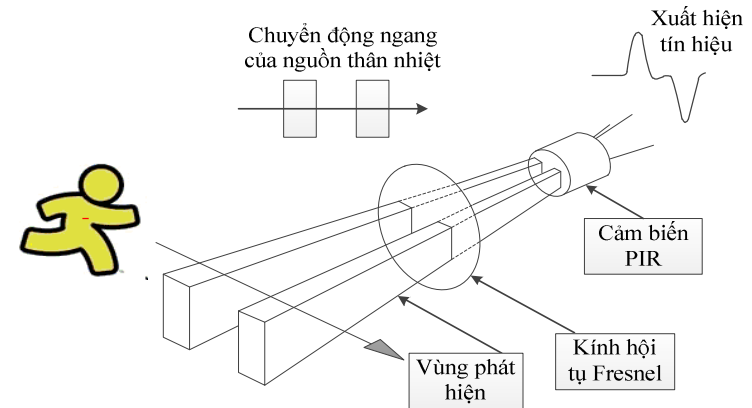
3.2. THIẾT KẾ MẠCH GIÁM SÁT VÀ ĐIỀU KHIỂN THIẾT BỊ ĐIỆN

Mạch vừa điều khiển tại chỗ thông qua công tắc, vừa điều khiển từ xa qua role (như hình 3.2).

3.3. MẠCH CẢM BIẾN PIR

Nguyên lý làm việc của cảm biến PIR (hình 3.4): Các nguồn nhiệt (với người và con vật là nguồn thân nhiệt) đều phát ra tia hồng ngoại, qua kính Fresnel, qua kích lọc lấy tia hồng ngoại, nó được cho tiêu tụ trên 2 cảm biến hồng ngoại gắn trong đầu dò, và tạo ra điện áp được khuếch đại với transistor FET. Khi có một vật nóng đi ngang

qua, từ 2 cảm biến này sẽ cho xuất hiện 2 tín hiệu và tín hiệu này sẽ được khuếch đại để có biên độ đủ cao và đưa vào mạch so áp để tác động vào một thiết bị điều khiển hay báo động.

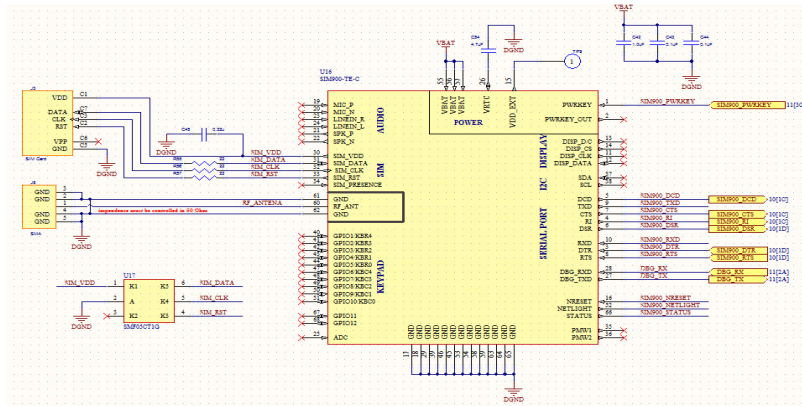


Hình 3.4: Nguyên lý chuyển động ngang của nguồn thân nhiệt

3.4. THIẾT KẾ MẠCH GIAO TIẾP MODULE GPRS SIM900

Sim900 của hãng SIMCom là một module GSM/GPRS cực kỳ nhỏ gọn, được thiết kế cho thị trường toàn cầu. Sim900 hoạt động được ở 4 băng tần GSM 850MHz, EGSM 900MHz, DCS 1800MHz và PCS 1900MHz như là một loại thiết bị đầu cuối với một Chip xử lý đơn nhân dựa trên nền vi xử lý ARM926EJ-S [14].

❖ Sơ đồ mạch SIM900 được thiết kế như Hình 3.6.



Hình 3.6: Sơ đồ mạch SIM900

- ❖ Nguồn cung cấp cho SIM900 (V_{BAT}) [14]:
 - ✓ Điện áp: $V_{cực\ đại} = 4.5V$, $V_{cực\ tiểu} = 3.4V$, $V_{thường} = 4.0V$,
 - ✓ Điện năng tiêu thụ ở chế độ chờ (ngủ) là: 1.5mA
 - ✓ Dòng cực đại: $I_{cực\ đại} = 3A$

3.5. KẾT QUẢ THỰC HIỆN

3.5.1. Kết quả layout

3.5.2. Mạch PIR

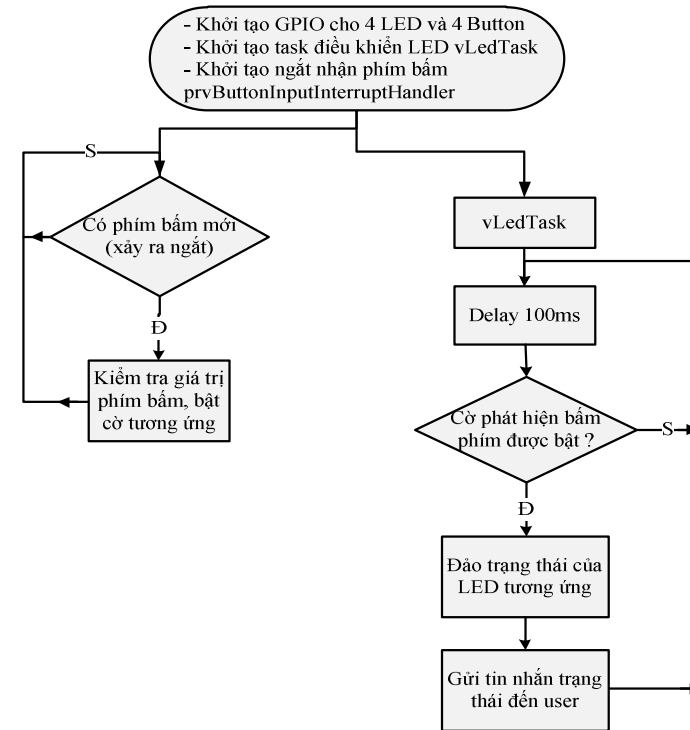
3.5.3. Mạch SIM900

CHƯƠNG 4 THIẾT KẾ PHẦN MỀM HỆ THỐNG

4.1. KHỐI ĐIỀU KHIỂN

4.1.1. Lưu đồ thuật toán

Lưu đồ thuật toán của khối điều khiển được trình bày như Hình 4.1



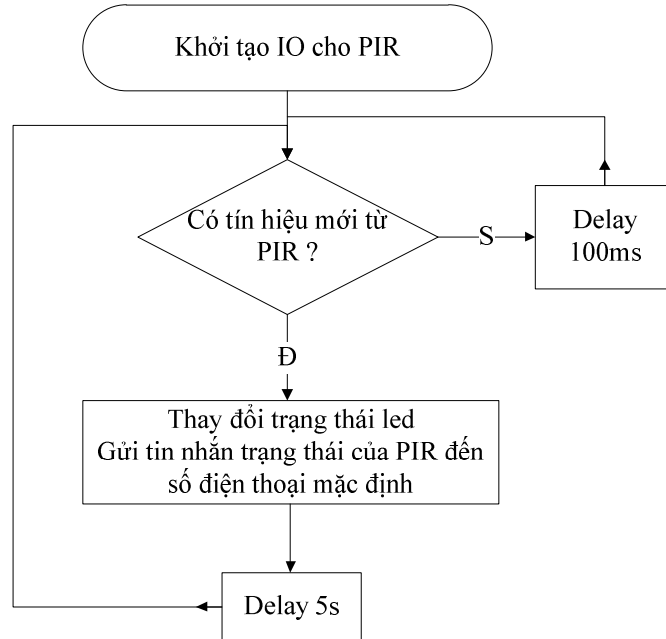
Hình 4.1: Lưu đồ thuật toán khối điều khiển

4.1.2. Chương trình

4.2. KHỐI PIR

4.2.1. Lưu đồ thuật toán

Lưu đồ thuật toán của khối PIR được trình bày như hình 4.2



Hình 4.2: Lưu đồ thuật toán khối PIR

4.2.2. Chương trình

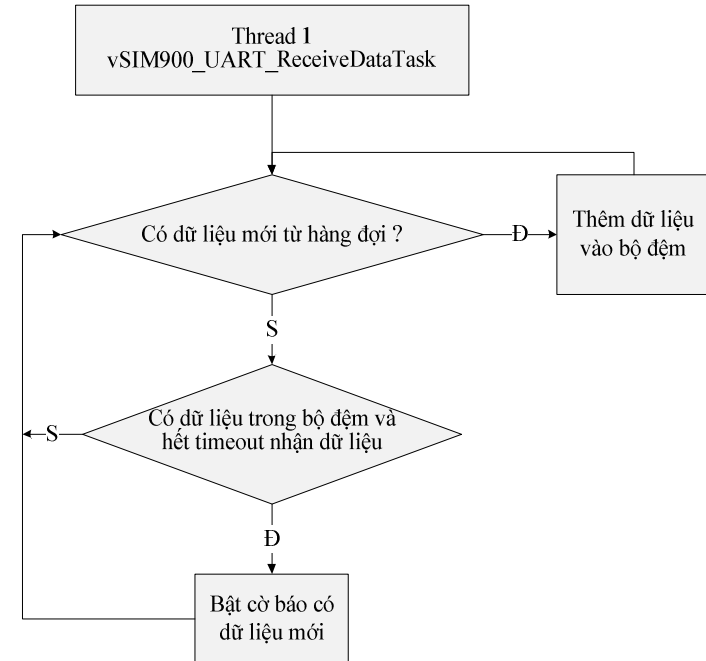
4.3. KHỐI GPRS SIM900

4.3.1. Lưu đồ thuật toán

- ❖ Lưu đồ thuật toán của tác vụ nhận dữ liệu từ UART (hình 4.3):

Tác vụ này chờ và nhận dữ liệu từ UART do mạch SIM900 gửi lên, mỗi khi nhận được 1 kí tự, tác vụ sẽ bật cờ timeout đếm lùi, khi

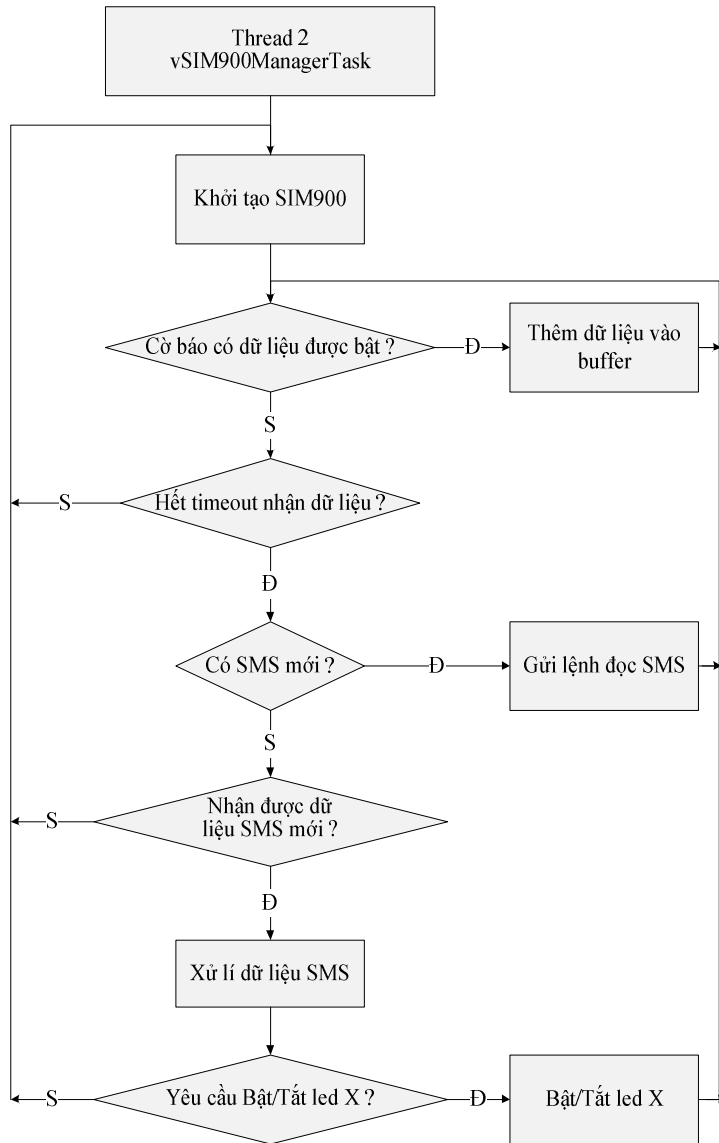
hết timeout mà chưa nhận thêm được dữ liệu mới, tác vụ sẽ bật cờ báo có dữ liệu mới.



Hình 4.3: Lưu đồ thuật toán nhận dữ liệu từ UART

- ❖ Lưu đồ thuật toán của tác vụ nhận dữ liệu (hình 4.4):

Tác vụ nhận dữ liệu chờ cờ báo có dữ liệu mới từ module SIM900, sau đó xử lý dữ liệu. Khi nhận được dữ liệu là tin nhắn với yêu cầu tắt/mở led, tác vụ sẽ bật, tắt led tương ứng và gửi tin nhắn về số điện thoại đã được cài đặt sẵn.



Hình 4.4: Lưu đồ thuật toán khối nhận dữ liệu của SIM900

4.3.2. Chương trình

4.4. GHEP NOI CAC CORE VAO HE THONG

Một hệ thống bao gồm: Vi xử lý mềm MicroBlaze, bộ nhớ trong, bộ nhớ ngoài, UART và các cổng vào/ra. Tất cả được kết nối với bus AXI để hệ thống nhận biết và cấp phát không gian bộ nhớ. Và kết nối với xung clock để vi xử lý điều khiển.

4.4.1. Gán địa chỉ cho các ngoại vi

Địa chỉ cho các thiết bị ngoại vi có thể được gán tự động hoặc bằng tay. Địa chỉ của các thiết bị ngoại vi như hình 4.6.

4.4.2. Biên dịch hệ thống

Biên dịch toàn bộ thiết kế gồm tạo netlist cho hệ thống (hình 4.7) và tạo bitstream cho hệ thống (hình 4.8).

4.4.3. Tích hợp hệ điều hành FreeRTOS vào hệ thống

4.5. PHÂN TÍCH VÀ ĐÁNH GIÁ KẾT QUẢ

4.5.1. Tài nguyên sử dụng của hệ thống

4.5.2. Tài nguyên sử dụng của MicroBlaze và RAM

4.5.3. Tài nguyên sử dụng của khối UART

4.5.4. Tài nguyên sử dụng của các khối vào/ra

4.5.5. Thực nghiệm và phân tích kết quả

a. Kích bản kiểm tra

b. Kết quả thực nghiệm và đánh giá

- ❖ Kết quả thực nghiệm đối với kích bản 1:

Bảng 4.7: Kết quả thực nghiệm đối với kịch bản 1

Trạng thái	Công tắc		Tin nhắn phản hồi	Số lần kiểm tra	Tỷ lệ thành công
	Bật	Tắt			
LED 1	Sáng	Tắt	Có	10	100%
LED 2	Sáng	Tắt	Có	10	100%
LED 3	Sáng	Tắt	Có	10	100%
LED 4	Sáng	Tắt	Có	10	100%

Nhận xét:

- ✓ Dựa vào kết quả ở bảng 4.7 nhận thấy, phần điều khiển bằng công tắc hoạt động tốt.
- ✓ Tin nhắn phản hồi đạt 100%, chứng tỏ phần SIM900 hoạt động đúng yêu cầu thiết kế về giám sát thiết bị bật/tắt.
- ❖ Kết quả thực nghiệm đối với kịch bản 2:

Bảng 4.8: Kết quả thực nghiệm đối với kịch bản 2

Trạng thái	SIM900		Tin nhắn phản hồi	Số lần kiểm tra	Tỷ lệ thành công
	Bật	Tắt			
LED 1	Sáng	Tắt	Có	10	100%
LED 2	Sáng	Tắt	Có	10	100%
LED 3	Sáng	Tắt	Có	10	100%
LED 4	Sáng	Tắt	Có	10	100%

Nhận xét:

- ✓ Dựa vào kết quả ở bảng 4.8 nhận thấy, phần điều khiển bằng tin nhắn điện thoại di động hoạt động chính xác.
- ✓ Tin nhắn phản hồi đạt 100%, chứng tỏ phần SIM900 hoạt động đúng yêu cầu thiết kế về giám sát thiết bị bật/tắt.

❖ Kết quả thực nghiệm đối với kịch bản 3:

Bảng 4.9: Kết quả thực nghiệm đối với kịch bản 3

Trạng thái	SIM900		Tin nhắn phản hồi	Số lần kiểm tra	Tỷ lệ thành công
	Bật	Tắt			
LED 1	Sáng		Có	2	100%
LED 3	Sáng		Có	2	100%
LED 2	Sáng		Có	2	100%
LED 4	Sáng		Có	2	100%
LED 3		Tắt	Có	2	100%
LED 2		Tắt	Có	2	100%
LED 1		Tắt	Có	2	100%
LED 4		Tắt	Có	2	100%
LED 2	Sáng		Có	2	100%
LED 3	Sáng		Có	2	100%
LED 4	Sáng		Có	2	100%
LED 1	Sáng		Có	2	100%
LED 1		Tắt	Có	2	100%
LED 4		Tắt	Có	2	100%
LED 3		Tắt	Có	2	100%
LED 2		Tắt	Có	2	100%

❖ Kết quả thực nghiệm đối với kịch bản 4:

Bảng 4.10: Kết quả thực nghiệm đối với kịch bản 4

Trạng thái		Trạng thái của LED 1		Tin nhắn phản hồi	Số lần kiểm tra	Tỷ lệ thành công
		Bật	Tắt			
Ban đầu cấp nguồn		Sáng		Có	10	100%
Có chuyển động của nguồn thân nhiệt	Có	Sáng		Có	10	100%
	Không		Tắt	Có	10	100%
	Có	Sáng		Có	10	100%

Nhận xét: Dựa vào kết quả ở bảng 4.10 nhận thấy, mạch PIR và chương trình hoạt động chính xác và đúng với yêu cầu của thiết kế ban đầu.

4.5.6. Ưu điểm và khuyết điểm của hệ thống

❖ Ưu điểm:

- ✓ Hệ thống sử dụng FPGA cho phép cấu hình lại với sự tùy biến của cả phần cứng và phần mềm với chi phí thấp. Đề tài tận dụng lợi thế của công nghệ FPGA để thiết kế điều khiển thiết bị và được tích hợp trên một chip duy nhất.
- ✓ Nhờ ứng dụng vi xử lý mềm MicroBlaze và hệ điều hành FreeRTOS nên hệ thống hoạt động được linh hoạt, đáp ứng thời gian thực cao. Hệ thống thừa hưởng được những ưu điểm của FPGA và cả vi điều khiển.

- ✓ Phần cứng có thể mở rộng thêm nhiều cổng vào/ra để điều khiển thêm được nhiều thiết bị bên ngoài. Các thiết bị ngoại vi không hạn chế nhà sản xuất bởi giao tiếp với mạch chính qua UART và ngõ vào/ra phổ biến.
 - ✓ Hệ thống sử dụng mạng viễn thông GSM trong truyền thông mang tính công nghiệp. Và sử dụng được tất cả các loại SIM của các nhà mạng GSM hiện nay như Mobifone, Vinaphone, Viettel...
 - ✓ Bên cạnh đó, có thể giám sát và cảnh báo chuyển động theo phương ngang như giám sát người, động vật đi vào những vùng bị cấm.
- ❖ Khuyết điểm: Mặc dù có nhiều ưu điểm như trình bày trên, nhưng hệ thống vẫn tồn tại một số điểm hạn chế như sau:
- ✓ Hệ thống chỉ hoạt động ở vùng có phủ sóng điện thoại di động.
 - ✓ Hệ thống chưa trang bị hệ thống pin dự phòng trong trường hợp mất điện.
 - ✓ Hệ thống được lập trình chỉ điều khiển thiết bị dùng tin nhắn SMS dạng text. Vì vậy, cần nghiên cứu và ứng dụng thêm điều khiển qua internet ...

4.5.7. Khả năng ứng dụng thực tế của đề tài

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN CỦA ĐỀ TÀI

1. KẾT LUẬN

Sau khoảng thời gian nghiên cứu và thực hiện đề tài, cùng với sự giúp đỡ, hướng dẫn và định hướng của các thầy, tác giả đã hoàn thành luận văn này và đã đạt những kết quả nhất định sau:

- ✓ Nghiên cứu lý thuyết về vi xử lý mềm của Xilinx và hệ điều hành thời gian thực FreeRTOS.
- ✓ Nghiên cứu kỹ thuật lập trình và nhúng hệ thống xuống FPGA của hãng Xilinx. Sử dụng phần mềm Xilinx Platform Studio của hãng Xilinx.
- ✓ Viết chương trình và nhúng được MicroBlaze & hệ điều hành FreeRTOS xuống KIT Xilinx Spartan-6 FPGA SP605. Sử dụng phần mềm Xilinx Software Development Kit của hãng Xilinx.
- ✓ Thiết kế, viết chương trình hệ thống và xây dựng bảng mạch điện tử cho hệ thống điều khiển thiết bị bằng GPRS.

Đề tài đã xây dựng được một hệ thống nhúng gồm có vi xử lý mềm MicroBlaze và hệ điều hành thời gian thực FreeRTOS thực hiện chức năng điều khiển & giám sát thiết bị điện và giám sát chuyển động theo phương ngang bằng PIR bằng GPRS. Hệ thống hoạt động ổn định.

Tuy nhiên vì lý do hạn chế về thời gian thực hiện luận văn, cũng như giới hạn về điều kiện nghiên cứu thực nghiệm, làm mạch in. Tác giả chỉ dừng lại ở phạm vi điều khiển được 8 thiết bị ngoài và giám sát chuyển động theo phương ngang dùng cảm biến PIR.

2. HƯỚNG PHÁT TRIỂN CỦA ĐỀ TÀI

Đề tài này đã đáp ứng được cơ bản yêu cầu đề ra. Tuy nhiên, cần nghiên cứu phát triển thêm:

- ✓ Tích hợp thêm nhiều ứng dụng khác ngoài giám sát và điều khiển thiết bị như là đo dòng, đo áp của thiết bị. Tính công suất hiệu dụng của thiết bị điện.
- ✓ Nghiên cứu thêm phần điều khiển thiết bị qua internet.
- ✓ Xây dựng cơ sở dữ liệu về công suất của thiết bị điện dân dụng để có thể tự động xác định được từng thiết bị đang hoạt động.
- ✓ Mở rộng hệ thống để điều khiển được nhiều thiết bị hơn.
- ✓ Thiết kế lại mạch FPGA để thay thế và giảm giá thành khi dùng KIT FPGA hiện tại. Mạch mới sẽ gồm có FPGA Spartan 6 hoặc Spartan 3, SIM900, mạch nguồn và các cổng vào/ra để nối đến các thiết bị bên ngoài ... để có thể ứng dụng thuận tiện hơn và đóng gói thành một sản phẩm hoàn thiện.