

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
TRƯỜNG ĐIỆN – ĐIỆN TỬ



BÁO CÁO BÀI TẬP LỚN
HỆ THỐNG NHÚNG
VÀ THIẾT KẾ GIAO TIẾP NHÚNG

Đề tài: Hệ thống điều khiển bồn cầu tự động

Nhóm sinh viên thực hiện:

Tên sinh viên	MSSV	Mã lớp
Phạm Ngọc Lâm	20182628	133385
Nguyễn Việt Thi	20182798	133385
Nguyễn Huy Nam	20182695	133385

Giảng viên hướng dẫn: TS. Phạm Văn Tiến

Hà Nội, 7-2022

LỜI NÓI ĐẦU

Thời gian gần đây, các Hệ thống nhúng – Thời gian thực được quan tâm nhiều hơn ở Việt Nam, và trên thế giới thì các hệ thống này đã và đang được phát triển mạnh mẽ và là xu hướng thịnh hành ở các nước Công nghiệp vì những lợi ích to lớn, thiết thực mà nó mang lại. Chương trình học môn Hệ thống nhúng và thiết kế giao tiếp nhúng là một phần quan trọng giúp hiểu rõ quy trình thiết kế, đánh giá hệ thống nhúng. Trong báo cáo này, nhóm chúng em triển khai thiết kế và mô phỏng hệ thống nhúng cho Điều khiển bồn cầu tự động. Thiết kế được triển khai bằng ngôn ngữ mô tả phần cứng Verilog, System Verilog và ngôn ngữ C, mô phỏng kiểm thử trên phần mềm QuestaSim và tổng hợp trên phần mềm Vivado. Cho ra kết quả hoạt động đúng với yêu cầu bài toán. Chúng em sẽ đi trình bày cụ thể những gì chúng em đã làm được thông qua 6 chương sau:

Chương 1: Giới thiệu chung (Introduction)

Chương 2: Đặc tả hệ thống (System Specification)

Chương 3: Thiết kế hệ thống (System Design)

Chương 4: Triển khai trên FPGA (Implement FPGA)

Chương 5: Kiểm thử (Verification)

Chương 6: Kết luận

Nhóm chúng em xin chân thành cảm ơn thầy Phạm Văn Tiến đã tận tâm hướng dẫn nhóm chúng em trong quá trình thực hiện bài tập lớn cũng như hoàn thiện báo cáo này.

MỤC LỤC

LỜI NÓI ĐẦU	ii
MỤC LỤC	iii
DANH MỤC HÌNH VẼ	i
DANH MỤC BẢNG BIỂU	iii
CHƯƠNG 1. GIỚI THIỆU CHUNG (INTRODUCTION)	1
1.1 Giới thiệu chung	1
1.2 Hệ thống thời gian thực (Real-time operating system)	1
1.3 Hệ thống điều khiển bồn cầu tự động	2
CHƯƠNG 2. MÔ HÌNH HÓA HỆ THỐNG	4
2.1 Mô hình hóa hệ thống sử dụng UML	4
2.1.1 Use case diagram	4
2.1.2 Class Diagram	5
2.1.3 State Diagram	6
2.1.4 Sequence Diagram	6
2.2 Mô hình hóa hệ thống sử dụng SystemC	10
2.3 Kết luận	13
CHƯƠNG 3. THIẾT KẾ HỆ THỐNG (SYSTEM DESIGN)	14
3.1 Thiết kế phần cứng	14
3.1.1 Mô tả chức năng	14
3.1.2 Thiết kế sơ đồ ASMD (Algorithm State Machine and Datapath)	14
3.1.3 Thiết kế khối register_block cho system_controller	17
3.1.4 Thiết kế rtl	17
3.2 Thiết kế phần mềm	17
3.2.1 Mục tiêu	17
3.2.2 Lý do lựa chọn phần mềm	17
3.2.3 Thiết kế	18
3.2.4 Hướng triển khai	19
3.2.5 Sơ đồ thuật toán	19

3.2.6 Giải quyết các vấn đề gặp phải trong quá trình triển khai.....	21
CHƯƠNG 4. TRIỂN KHAI TRÊN FPGA (IMPLEMENT FPGA).....	22
4.1 Tổng quan về triển khai hệ thống trên FPGA.....	22
4.2 Các IP dùng trong hệ thống.....	23
4.2.1 Smart toilet system controller (system_controller)	23
4.2.2 Oled controller (oled_controller).....	24
4.2.3 DHT11 controller (dht11_controller)	25
4.3 Tổng hợp và triển khai trên phần mềm Vivado	27
4.3.1 Tạo project, thêm các file source code mô tả phần cứng	27
4.3.2 Tạo Block design.....	27
4.3.3 Tạo constraint file.....	28
4.3.4 Tổng hợp (synthesis).....	28
4.3.5 Triển khai (implement).....	29
4.3.6 Gen bitstream và tạo SDK (software development kit)	30
4.3.7 Kết quả	31
CHƯƠNG 5. KIỂM THỬ (VERIFICATION).....	32
5.1 Kế hoạch kiểm thử.....	32
5.1.1 Kịch bản test.....	32
5.1.2 Kiểm thử thiết kế phần cứng bộ điều khiển bồn cầu sử dụng QuestaSim.....	32
CHƯƠNG 6. BÁO CÁO CÁ NHÂN	34
6.1 Đưa ra ý tưởng thiết kế và kiểm tra hoạt động của hệ thống.....	34
6.2 Triển khai hệ thống trên ZedBoard.....	34
6.2.1 Tìm kiếm và triển khai các IP để điều khiển OLED và DHT11.....	34
6.2.2 Đóng gói system_controller	34
6.2.3 Thiết kế block design.....	35
6.2.4 Tạo constraint file.....	36
6.2.5 Tổng hợp và implement thiết kế trên vivado.....	37
6.2.6 Tạo bitstream và viết SDK	37
6.2.7 Nạp bitstream và phần mềm lên kit.....	38
6.3 Kiểm thử các chức năng bằng kit.....	39
KẾT LUẬN.....	41
TÀI LIỆU THAM KHẢO	42

DANH MỤC HÌNH VẼ

Hình 2.1 Use case diagram của hệ thống bồn cầu tự động	4
Hình 2.2 Class diagram của hệ thống bồn cầu tự động	5
Hình 2.3 State diagram của hệ thống bồn cầu tự động	6
Hình 2.4 Sequence diagram của hệ thống xịt vệ sinh	7
Hình 2.5 Sequence diagram của hệ thống sấy khô	8
Hình 2.6 Sequence diagram của hệ thống xả nước tự động	8
Hình 2.7 Sequence diagram của hệ thống làm ấm tự động	9
Hình 2.8 Sequence diagram của hệ thống chiếu sáng tự động	10
Hình 2.9 Sơ đồ hệ thống bồn cầu tự động sử dụng SystemC	11
Hình 3.1 Sơ đồ ASMD	15
Hình 3.2 Sơ đồ khối Control Unit và Datapath.....	16
Hình 3.3 Các thanh ghi trong system_controller	17
Hình 3.4 Sơ đồ phần mềm hệ thống.....	18
Hình 3.5 Sơ đồ thuật toán	20
Hình 3.6 Mô hình giao tiếp giữa vi xử lý và các module	20
Hình 4.1. Sơ đồ khối của hệ thống.....	22
Hình 4.2 Sơ đồ khối của smart toilet system controller.....	23
Hình 4.3 Sơ đồ khối của oled_controller	24
Hình 4.4 Sơ đồ khối của DHT11 controller	26
Hình 4.5 Sơ đồ thời gian quá trình gửi tín hiệu điều khiển từ master và phản hồi tín hiệu từ slave của cảm biến DHT11.....	27
Hình 4.6 Tạo project và thêm các file source code	27
Hình 4.7 Block design của hệ thống	28
Hình 4.8 Báo cáo về timing	29
Hình 4.9 Báo cáo về công suất của thiết kế	30
Hình 4.10 Thiết kế trên SDK	30
Hình 4.11 Kết quả trên kit FPGA.....	31

Hình 5.1 Kết quả mô phỏng khối clock_generator.....	32
Hình 5.2 Kết quả mô phỏng khối controller.....	33
Hình 5.3 Kết quả mô phỏng khối core.....	33
Hình 6.1. Block design của hệ thống.....	35
Hình 6.2. Các thông số của đã cài đặt trên vivado	35
Hình 6.3. Mô hình giao tiếp giữa vi xử lý và các module	36
Hình 6.4 Kết quả của quá trình implement.....	37
Hình 6.5 Report của hệ thống.....	37
Hình 6.6 Sơ đồ thuật toán	38
Hình 6.7 Giao diện xsdk	38
Hình 6.8 Cấu hình để nạp hệ thống vào ZedBoard	39
Hình 6.9. Hình ảnh thực tế khi đã nạp hệ thống lên kit.....	39

DANH MỤC BẢNG BIỂU

Bảng 1.1 Bảng thông số kỹ thuật	2
Bảng 4.1 Ràng buộc các tín hiệu của system_controller	23
Bảng 4.2 Tổ chức thanh ghi trong oled_controller	25
Bảng 4.3 Ràng buộc các tín hiệu của oled_controller	25
Bảng 4.4 Tổ chức thanh ghi trong dht11_controller	26
Bảng 4.5 Ràng buộc các tín hiệu của dht11_controller	26
Bảng 4.6 Số lượng LUT và Register của thiết kế	29

CHƯƠNG 1. GIỚI THIỆU CHUNG (INTRODUCTION)

Chương này giới thiệu khái quát và các kiến thức cơ bản về hệ thống nhúng bài toán Điều khiển bồn cầu tự động.

1.1 Giới thiệu chung

Hệ thống nhúng (Embedded Systems) là sự kết hợp của phần cứng và phần mềm máy tính, và có thể bổ sung phần cơ khí hoặc một số bộ phận khác, được thiết kế để thực hiện một chức năng cụ thể. Điều này trái ngược hoàn toàn với máy tính cá nhân như Laptop, Computer, ... Nó cũng bao gồm phần cứng, phần mềm các thành phần cơ khí. Tuy nhiên, máy tính cá nhân không được thiết kế để thực hiện một chức năng cụ thể. Đúng hơn là nó thực hiện nhiều công việc khác nhau, hay có thể sử dụng thuật ngữ máy tính đa năng để phân biệt với máy tính hệ nhúng.

Thông thường, một hệ thống nhúng là một thành phần trong một số hệ thống lớn hơn. Như bài toán ta đang triển khai Điều khiển bồn cầu tự động, một hệ thống nhúng điều khiển xả nước, xịt rửa và làm ấm tự động.

1.2 Hệ thống thời gian thực (Real-time operating system)

Trong các bài toán điều khiển chúng ta hay bắt gặp các thuật ngữ “Thời gian thực”. Thời gian thực không phải là thời gian phản ánh một cách trung thực, chính xác thời gian hay yêu cầu hệ thống phải trùng với thời gian thực tế.

Hệ thống thời gian thực được hiểu là các hoạt động của hệ thống phải thỏa mãn về tính tiên định. Tính tiên định là hành vi của hệ thống phải được thực hiện trong một khung thời gian cho trước hoàn toàn xác định, khung thời gian này được quyết định bởi đặc điểm và yêu cầu của hệ thống.

Thực tế cho thấy rằng hầu hết các hệ thống nhúng là các hệ thống thời gian thực và ngược lại các hệ thống thời gian thực là hệ thống nhúng.

1.3 Hệ thống điều khiển bồn cầu tự động

Bồn cầu tự động là một bồn cầu được tự động hóa và cải tiến để trở nên thông minh hơn phục vụ nhu cầu của người sử dụng. Trong gia đình đặc biệt là các gia đình có người cao tuổi thì bồn cầu tự động chứng tỏ được giá trị của mình.

Hệ thống của một bồn cầu tự động bao gồm:

- Bồn cầu
- Nắp bồn cầu
- Cảm biến nhiệt độ phòng
- Cảm biến nhiệt độ nước
- Cảm biến phát hiện đi nặng/nhẹ dựa vào thời gian
- Cảm biến trọng lực
- Oled
- Processor controller
- Button/Switch
- Bộ phận làm nóng nước
- Vòi xịt và sấy khô
- Bộ phận xả nước

Các thông số kỹ thuật được mô tả qua bảng sau:

Bảng 1.1 Bảng thông số kỹ thuật

Item	Parameter
Processor	Xilinx Zynq®-7000 All Programmable SoC
Temperature threshold (°C)	20
Temperature sensor	DHT11
Detect defecate/urinate threshold (sec)	60
Spray time	20 sec
Drying time	20 sec
Discharge time	3 sec
Power Supply	AC 220V
Power of Processor	100MHz
Frequency of Processor	1.679 W

Yêu cầu về công nghệ:

- Dễ điều khiển
- An toàn tuyệt đối cho người và thiết bị
- Xử lý các tình huống phát sinh chuẩn xác
- Tiết kiệm điện năng
- Dễ dàng nâng cấp chỉnh sửa, bảo trì
- Đảm bảo tính thời gian thực

CHƯƠNG 2. MÔ HÌNH HÓA HỆ THỐNG

Chương này mô tả tổng quan về hệ thống bồn cầu tự động mà nhóm thực hiện.

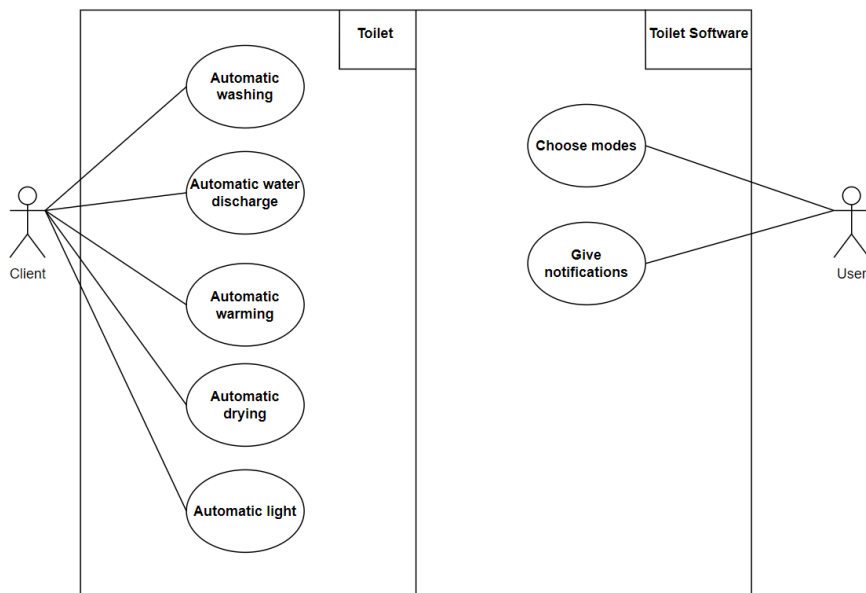
2.1 Mô hình hóa hệ thống sử dụng UML

UML (Unified Modeling Language) là một ngôn ngữ được sử dụng để mô tả các đặc điểm của hệ thống. Dựa vào chỉ tiêu kỹ thuật đã được đề cập trong chương 1, nhóm thực hiện thiết kế hệ thống bồn cầu tự động sử dụng UML.

2.1.1 Use case diagram

Tất cả các phần của hệ thống tương tác với con người hoặc các tác nhân tự động sử dụng hệ thống cho một số mục đích và cả con người và tác nhân đều mong đợi hệ thống hoạt động theo những cách có thể dự đoán được. Trong UML, use case diagram được sử dụng để mô hình hóa các hành vi của một hệ thống hoặc một phần của hệ thống.

Dựa theo các yêu cầu kỹ thuật đã được nêu trước đó, use case diagram của hệ thống bồn cầu tự động được thể hiện ở hình dưới.



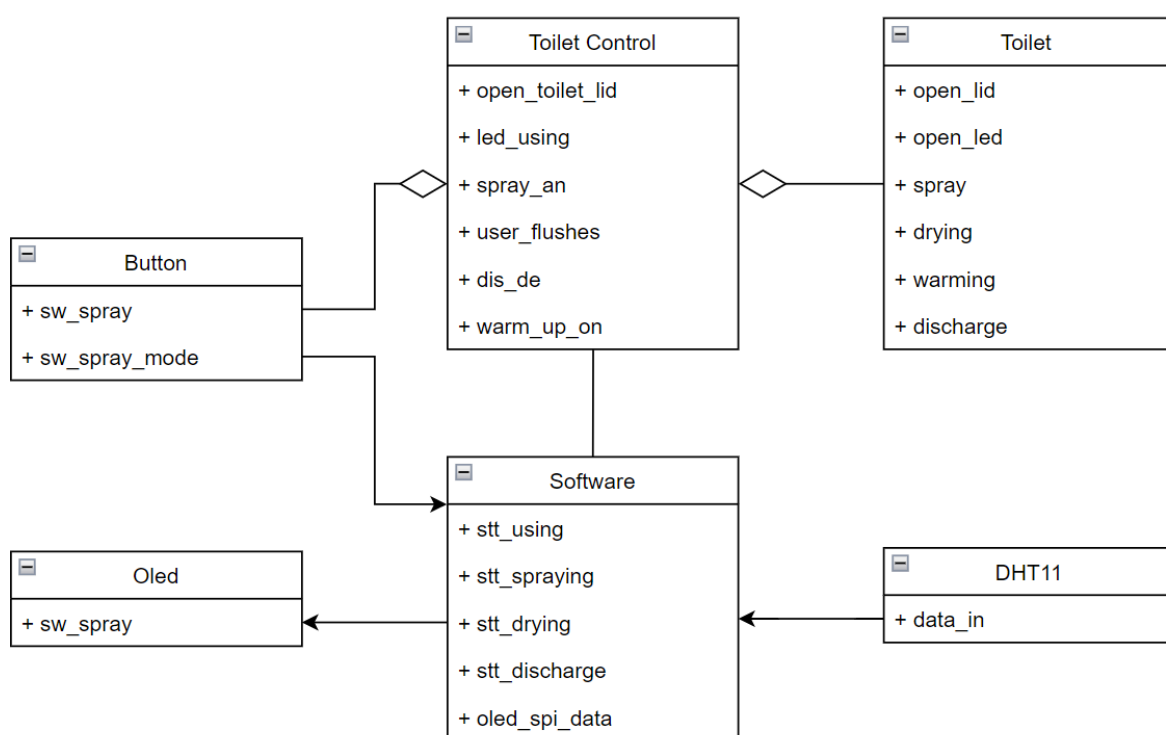
Hình 2.1 Use case diagram của hệ thống bồn cầu tự động

Tác nhân duy nhất trong hệ thống bồn cầu tự động là người sử dụng, đây là vai trò của con người khi tương tác với hệ thống. Người sử dụng tương tác với hệ thống. Do đó, biểu đồ use case cho thấy tác nhân có mối quan hệ với các chức năng của hệ thống:

Đóng/mở nắp bồn cầu, làm ấm nước, tự động xịt, sấy khô, xả nước, tự chiếu sáng. Người dùng hoàn toàn có thể tắt hoặc bật chức năng tự động nhằm mục đích vệ sinh hay bảo trì hệ thống.

2.1.2 Class Diagram

Biểu đồ lớp (Class Diagram) sẽ làm rõ các lớp, giao diện và mối quan hệ giữa chúng. Biểu đồ lớp liên quan đến mô tả hệ thống một cách tổng quan bao gồm các thuộc tính, hoạt động trong một lớp. Các thuộc tính và hoạt động có thể có của một hệ thống bồn cầu tự động được thể hiện trên hình dưới đây.

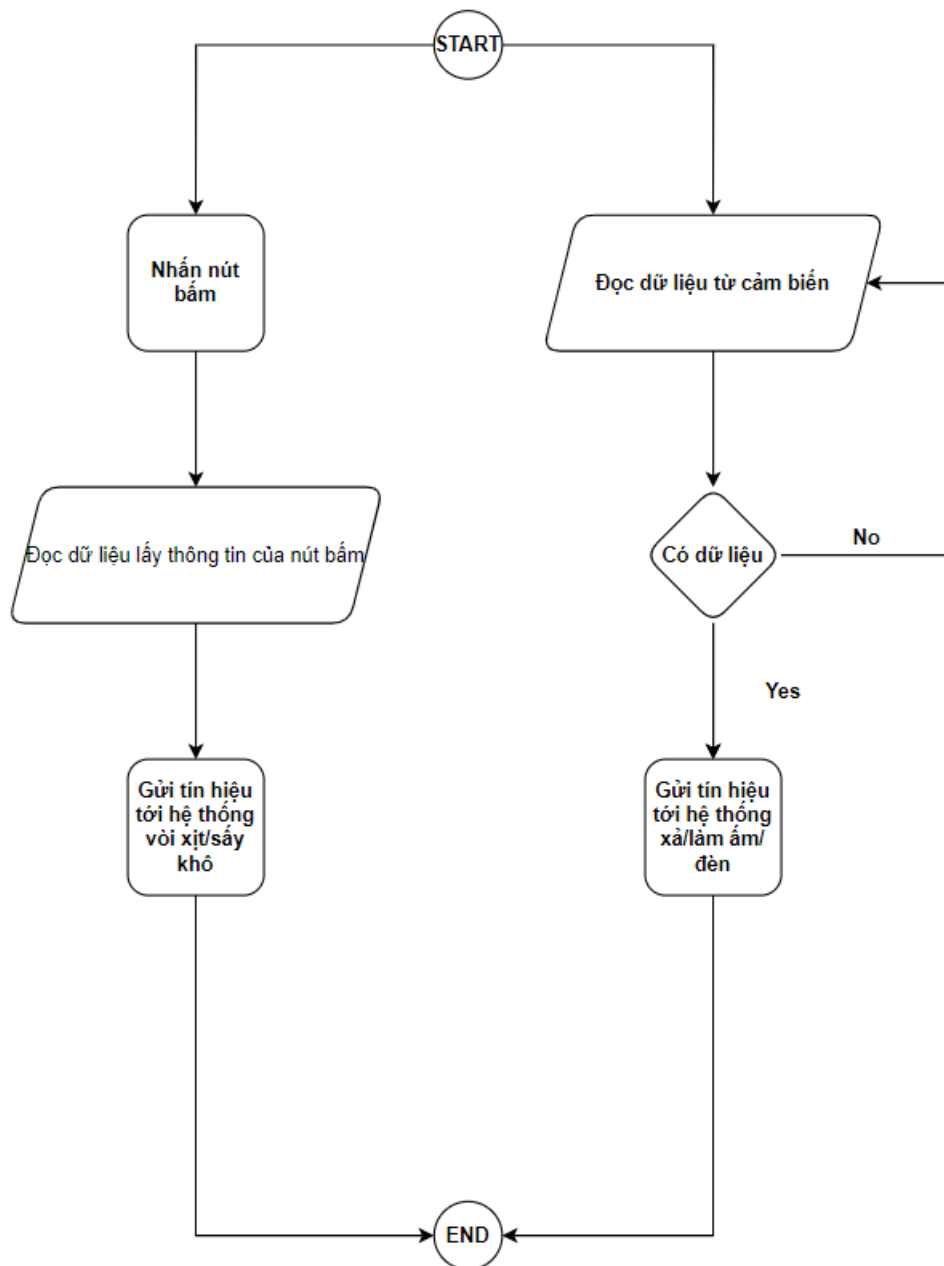


Hình 2.2 Class diagram của hệ thống bồn cầu tự động

Hệ thống bồn cầu tự động gồm hai phần chính là Toilet control và Software. Toilet control là thành phần chính của hệ thống, có nhiệm vụ nhận tín hiệu từ các đầu vào như nút bấm,... đồng thời tương tác với phần mềm để điều khiển toilet. Phần mềm xử lý các trạng thái, nhận tín hiệu từ các cảm biến để đưa ra các tín hiệu gửi tới màn hình, toilet control,...

2.1.3 State Diagram

Biểu đồ trạng thái (state diagram) để hiện luồng hoạt động của một thành phần hệ thống. Dưới đây là biểu đồ trạng thái của phần mềm hệ thống.

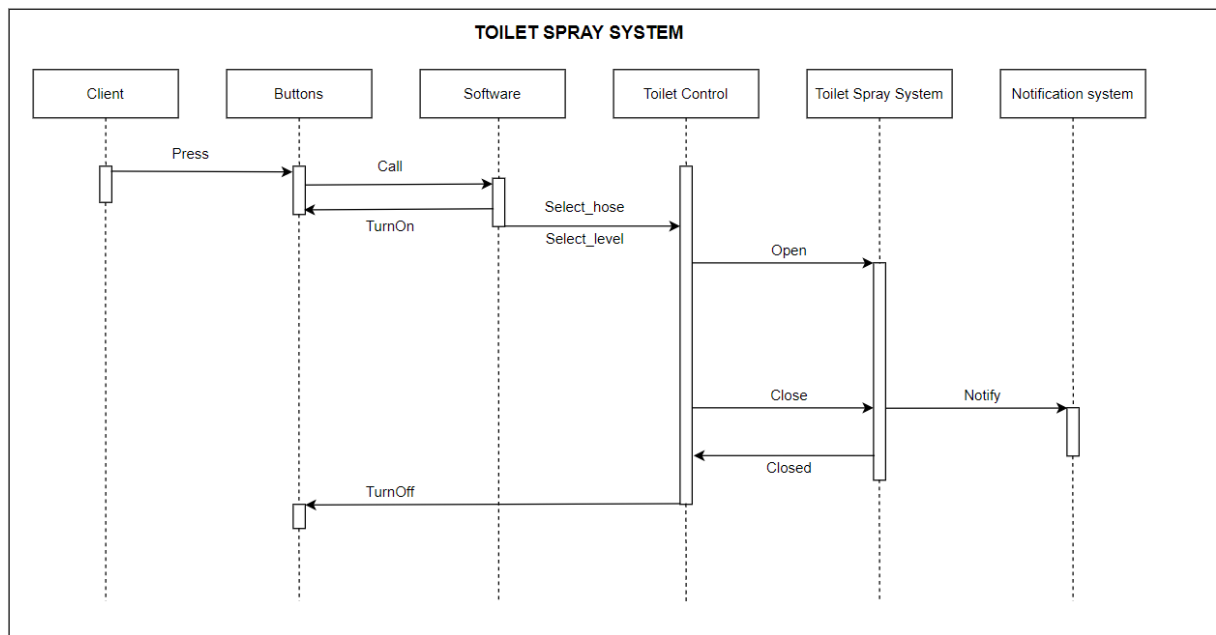


Hình 2.3 State diagram của hệ thống bồn cầu tự động

2.1.4 Sequence Diagram

Sơ đồ trình tự (Sequence diagram) thể hiện tương tác giữa một tập hợp các đối tượng của hệ thống. Sơ đồ trình tự bao gồm tất cả các thông báo cho một phần hoặc một trường hợp sử dụng. Ở đây, nhóm sử dụng sơ đồ trình tự để thể hiện sự tương tác giữa phần cứng và phần mềm, và các tương tác cho trường hợp thông báo của hệ thống.

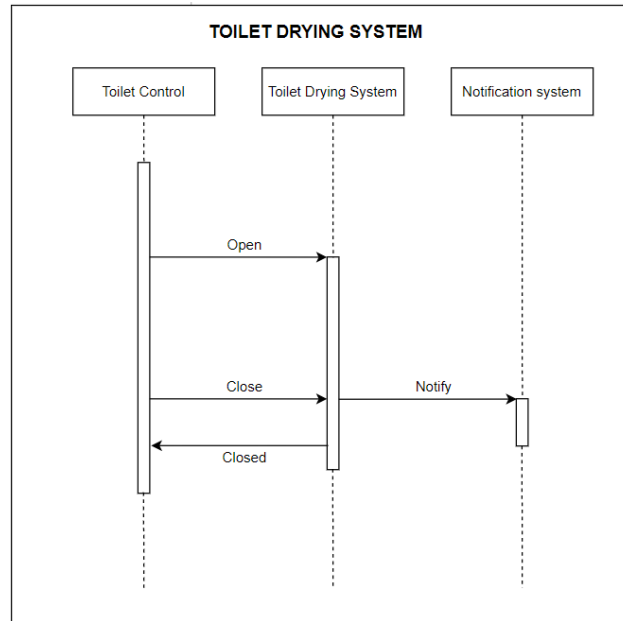
2.1.4.1 Sequence Diagram cho hệ thống xịt vệ sinh



Hình 2.4 Sequence diagram của hệ thống xịt vệ sinh

Dựa vào sơ đồ trên, khi người dùng bấm nút xịt vệ sinh, phần mềm hệ thống sẽ nhận tín hiệu. Sau đó toilet control thực hiện điều khiển hệ thống vòi xịt vệ sinh cho người dùng. Sau khi xịt vệ sinh xong sẽ có hệ thống đèn thông báo đã thực hiện xong việc xịt vệ sinh cho người dùng.

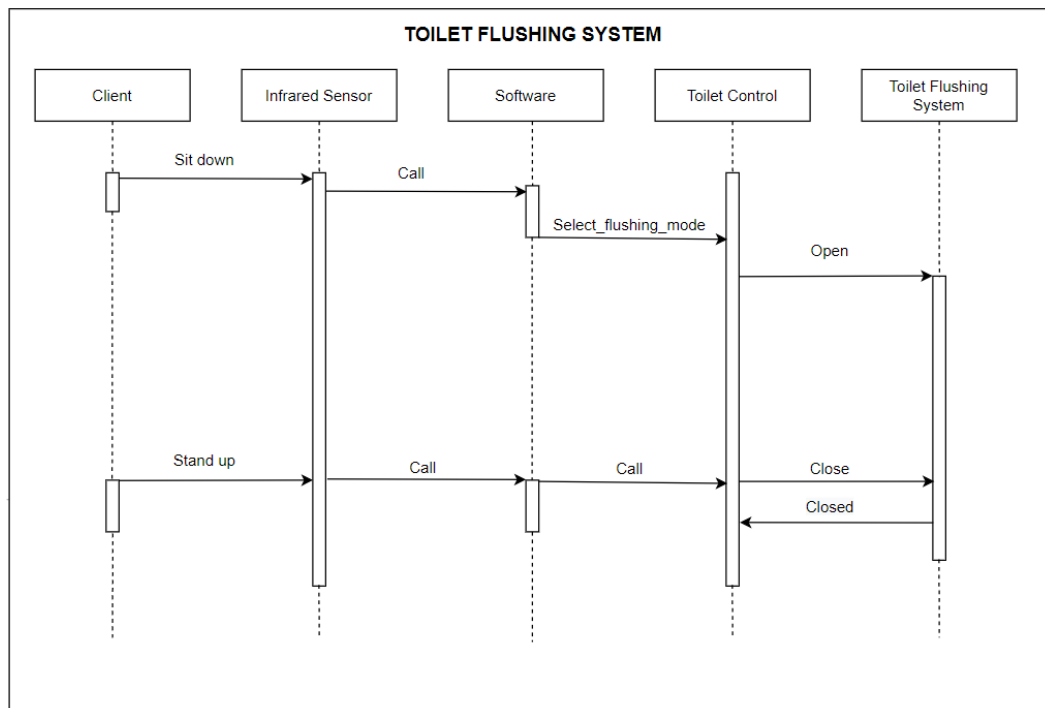
2.1.4.2 Sequence Diagram cho hệ thống sấy khô



Hình 2.5 Sequence diagram của hệ thống sấy khô

Sau khi kết thúc quá trình xịt, toilet control thực hiện điều khiển hệ thống sấy khô vệ cho người dùng. Sau khi sấy khô xong sẽ có hệ thống đèn thông báo đã thực hiện xong việc sấy khô cho người dùng.

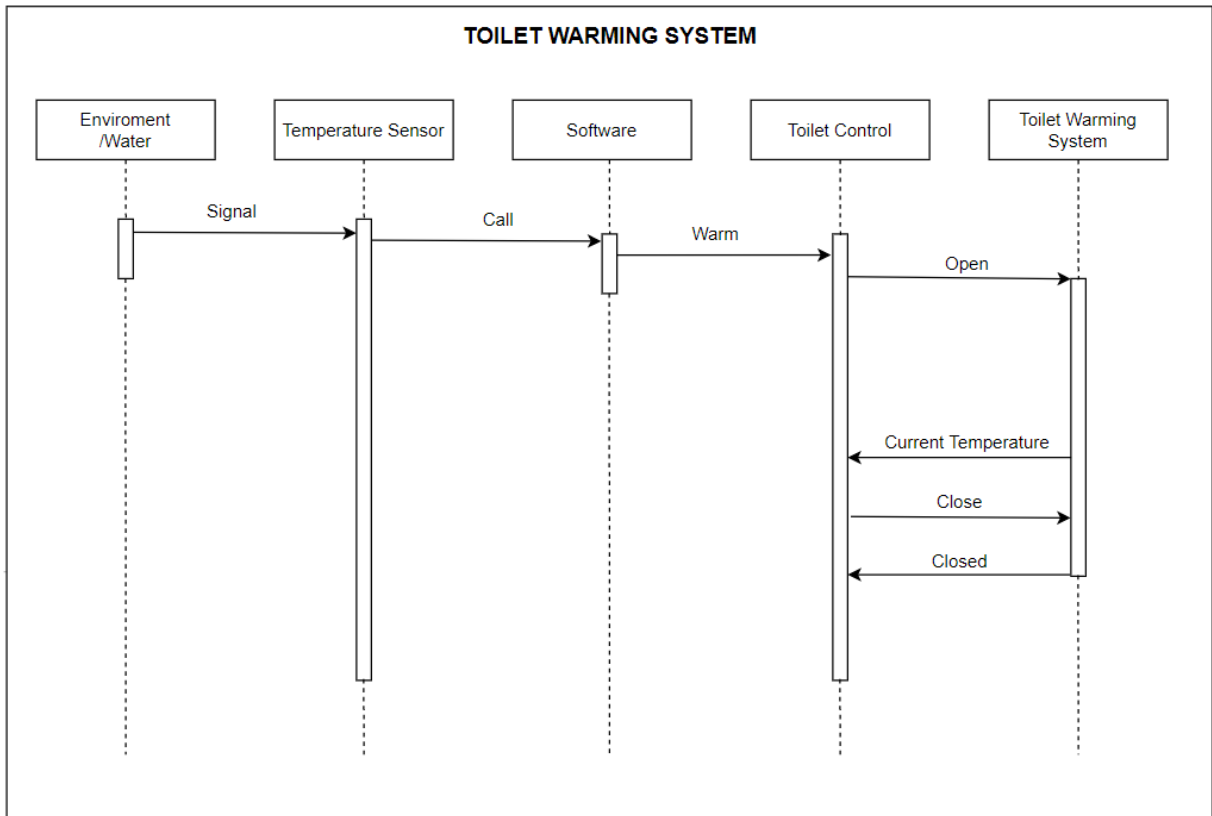
2.1.4.3 Sequence Diagram cho hệ thống xả nước tự động



Hình 2.6 Sequence diagram của hệ thống xả nước tự động

Khi phát hiện người dùng ngồi xuống sử dụng bồn cầu, cảm biến sẽ gửi tín hiệu tới phần mềm. Lúc này hệ thống bắt đầu thực hiện đếm thời gian tới khi cảm biến gửi tín hiệu người dùng đứng dậy để đưa ra chế độ xả nước phù hợp để gửi tới toilet control. Sau đó, toilet control gửi tín hiệu xuống hệ thống xả nước của bồn cầu.

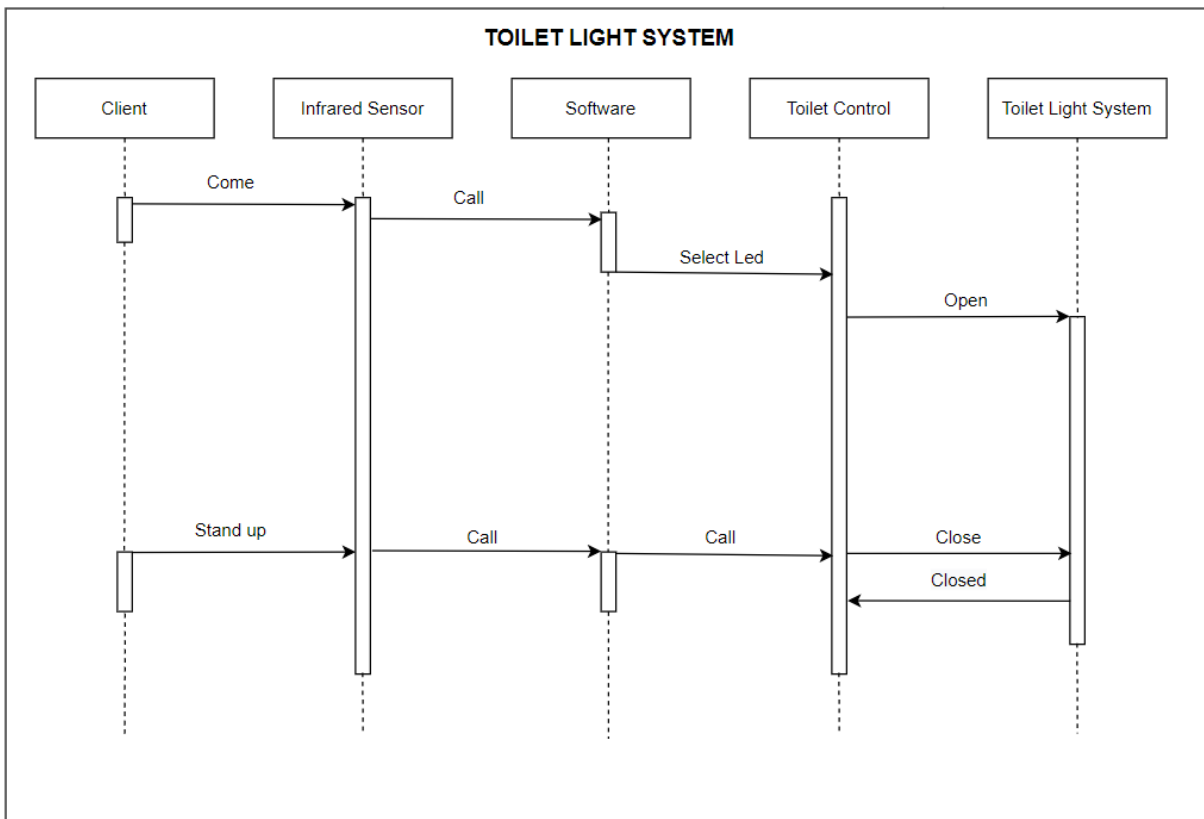
2.1.4.4 Sequence Diagram cho hệ thống làm ấm tự động



Hình 2.7 Sequence diagram của hệ thống làm ấm tự động

Cảm biến cảm nhận nhiệt độ từ môi trường rồi gửi tín hiệu tới phần mềm. Lúc này phần mềm sẽ so sánh nhiệt độ nhận được từ cảm biến với các ngưỡng đã thiết lập để đưa ra điều chỉnh về nhiệt độ phù hợp. Sau đó, tín hiệu này được gửi tới toilet control để điều khiển hệ thống làm ấm của bồn cầu.

2.1.4.5 Sequence Diagram cho hệ thống chiếu sáng tự động

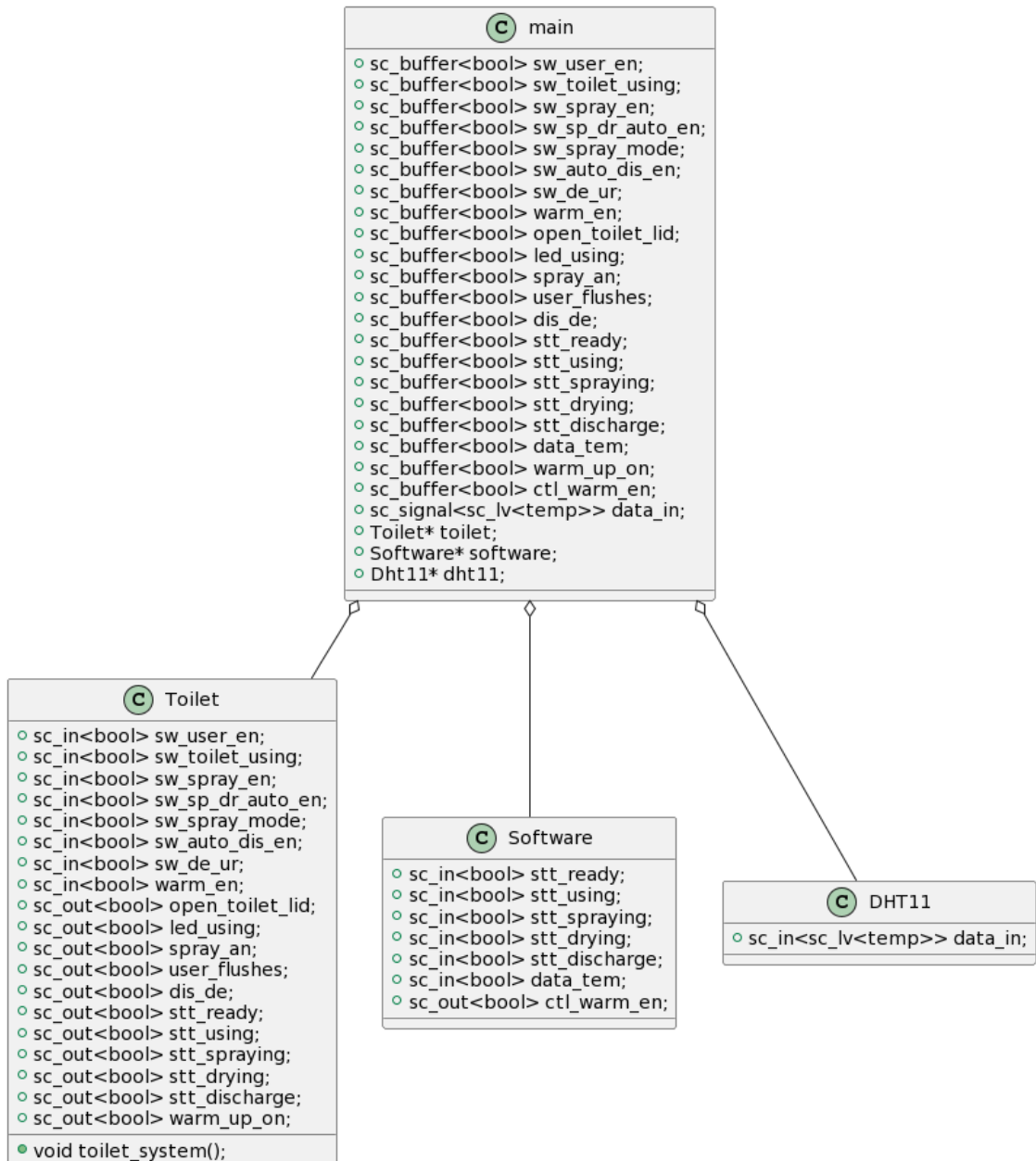


Hình 2.8 Sequence diagram của hệ thống chiếu sáng tự động

Khi có người dùng tiến lại gần bồn cầu sẽ, cảm biến hồng ngoại sẽ gửi tín hiệu tới phần mềm để xử lý. Sau đó phần mềm gửi tín hiệu khởi động hệ thống chiếu sáng tới toilet control. Sau khi nhận được tín hiệu toilet control sẽ điều khiển việc bật đèn. Tương tự, khi người dùng không xuất hiện trong cảm biến hồng ngoại, hệ thống sẽ xử lý để tắt đèn chiếu sáng.

2.2 Mô hình hóa hệ thống sử dụng SystemC

Ngoài việc sử dụng UML, nhóm còn chọn systemC là ngôn ngữ thứ hai để thiết kế và mô hình hóa hệ thống bồn cầu tự động. Sơ đồ hệ thống của hệ thống bồn cầu tự động được mô tả trên hình dưới đây.



Hình 2.9 Sơ đồ hệ thống bồn cầu tự động sử dụng SystemC

Dựa vào yêu cầu kỹ thuật ở phần trước, nhóm sử dụng SystemC để thiết kế các đầu vào và đầu ra của hệ thống bồn cầu tự động được thể hiện như trên hình gồm các class sau:

- Class main chứa các tín hiệu đầu vào đầu ra của hệ thống phục vụ cho việc mô phỏng.
- Class Toilet (Toilet control) chứa phần điều khiển hệ thống.
- Class Software chứa tín hiệu về trạng thái của các chân chức năng.
- Class DHT11 chứa tín hiệu về nhiệt độ của môi trường.

Bảng 2.1 Tín hiệu đầu vào/đầu ra của Toilet control

Tên	Input/Output	Mô tả
sw_user_en	input	Cảm biến khoảng cách vật
sw_toilet_using	input	Cảm biến trọng lực
sw_spray_en	input	Nút bấm xịt nước
sw_sp_dr_auto_en	input	Nút bấm bật/tắt chế độ tự động xịt sấy
sw_spray_mode	input	Nút bấm chọn kiểu xịt
sw_auto_dis_en	input	Nút bấm bật/tắt tự động xả
sw_de_ur	input	Cảm biến xác định đi nặng/đi nhẹ
warm_en	input	Cảm biến nhiệt độ
open_toilet_lid	output	Tín hiệu mở nắp bồn cầu
led_using	output	Tín hiệu thông báo đang có người sử dụng
spray_an	output	Tín hiệu chọn chế độ xịt: 1: xịt hậu môn 0: xịt cho phụ nữ
user_flushes	output	Tín hiệu thông báo người dùng tự xả
dis_de	output	Tín hiệu chọn chế độ xả nước: 1: đi nặng 0: đi nhẹ
stt_ready	output	Tín hiệu trạng thái sẵn sàng
stt_using	output	Tín hiệu trạng thái đang sử dụng
stt_spraying	output	Tín hiệu trạng thái đang xịt vệ sinh
stt_drying	output	Tín hiệu trạng thái đang sấy khô
stt_discharge	output	Tín hiệu trạng thái đang xả nước
warm_up_on	output	Tín hiệu mở hệ thống làm ấm nước

Bảng 2.2 Tín hiệu đầu vào/đầu ra của phần mềm hệ thống

Tên	Input/Output	Mô tả
stt_ready	input	Tín hiệu trạng thái sẵn sàng
stt_using	input	Tín hiệu trạng thái đang sử dụng
stt_spraying	input	Tín hiệu trạng thái đang xịt vệ sinh
stt_drying	input	Tín hiệu trạng thái đang sấy khô
stt_discharge	input	Tín hiệu trạng thái đang xả nước
data_tem	input	Dữ liệu nhiệt độ
ctl_warm_en	output	Đưa tín hiệu vào khối thanh ghi

Bảng 2.3 Tín hiệu đầu vào/đầu ra của DHT11

Tên	Input/Output	Mô tả
data_in	inout	Tín hiệu từ cảm biến nhiệt độ

2.3 Kết luận

Trong chương này, nhóm đã trình bày việc mô hình hóa hệ thống bồn cầu tự động sử dụng hai công cụ là UML và SystemC. Từ đó, nhóm có thể hình dung tổng quan về hệ thống và các phần cần thiết để đáp ứng chỉ tiêu kỹ thuật đã nêu ra ở CHƯƠNG 1. Công việc triển khai hệ thống được trình bày ở chương tiếp theo.

CHƯƠNG 3. THIẾT KẾ HỆ THỐNG (SYSTEM DESIGN)

Chương này mô tả thiết kế chi tiết từng khối và toàn hệ thống bồn cầu tự động.

3.1 Thiết kế phần cứng

3.1.1 Mô tả chức năng

Hệ thống điều khiển bồn cầu tự động có các chức năng sau:

- Tự động làm ấm nước và bệ ngồi khi nhiệt độ thấp.
- Tự động mở nắp bồn cầu khi có người đến gần.
- Tự động xịt, sấy, xả nước.
- Các chế độ xịt nước:
 - Xịt dành cho phụ nữ
 - Xịt hậu môn
- Các chế độ xả:
 - Đi nặng
 - Đi nhẹ

Mô tả hoạt động của bồn cầu khi có người sử dụng:

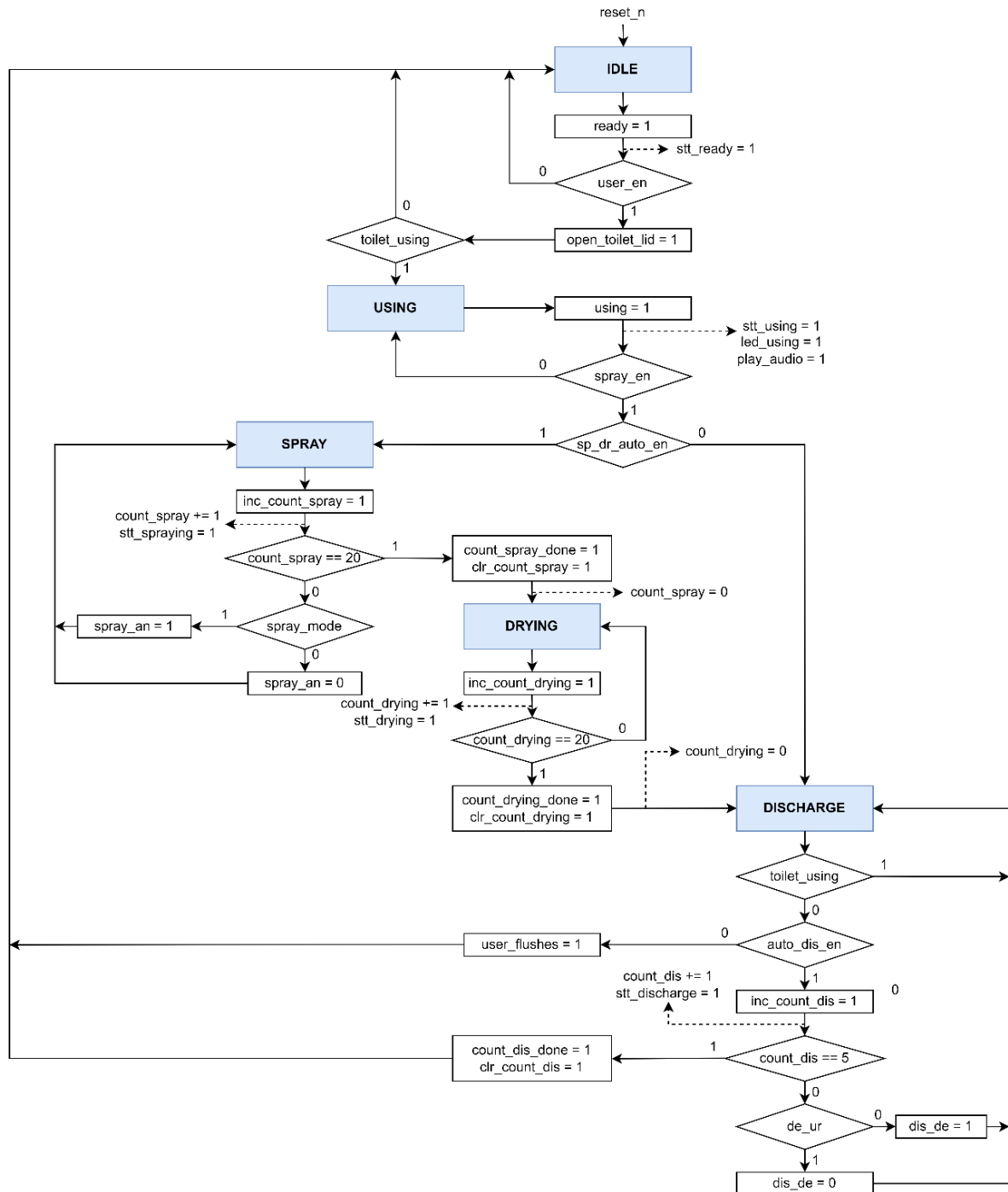
- Khi có người đến gần nắp bồn cầu sẽ tự động mở lên. Hệ thống làm nóng sẽ tự động làm nóng khi nhiệt độ thấp.
- Người sử dụng ngồi vào. Khi đi vệ sinh xong, người dùng sẽ chọn kiểu xịt và bấm nút xịt nước. Nếu tự động xịt, sấy được chọn, bồn cầu sẽ đưa vòi xịt và xịt theo kiểu xịt mà người dùng đã chọn trong 20 giây. Sau đó chuyển sang chức năng sấy khô trong 20 giây.
- Sau khi sấy khô hoặc người dùng không chọn chức năng tự động xịt, sấy thì sẽ chuyển sang bước xả nước. Người dùng đứng lên khỏi bệ ngồi thì bồn cầu sẽ tự động xả nước theo một trong hai chế độ đi nặng hoặc đi nhẹ.

3.1.2 Thiết kế sơ đồ ASMD (*Algorithm State Machine and Datapath*)

Từ mô tả chức năng và hoạt động của hệ thống điều khiển bồn cầu tự động, xác định được năm trạng thái:

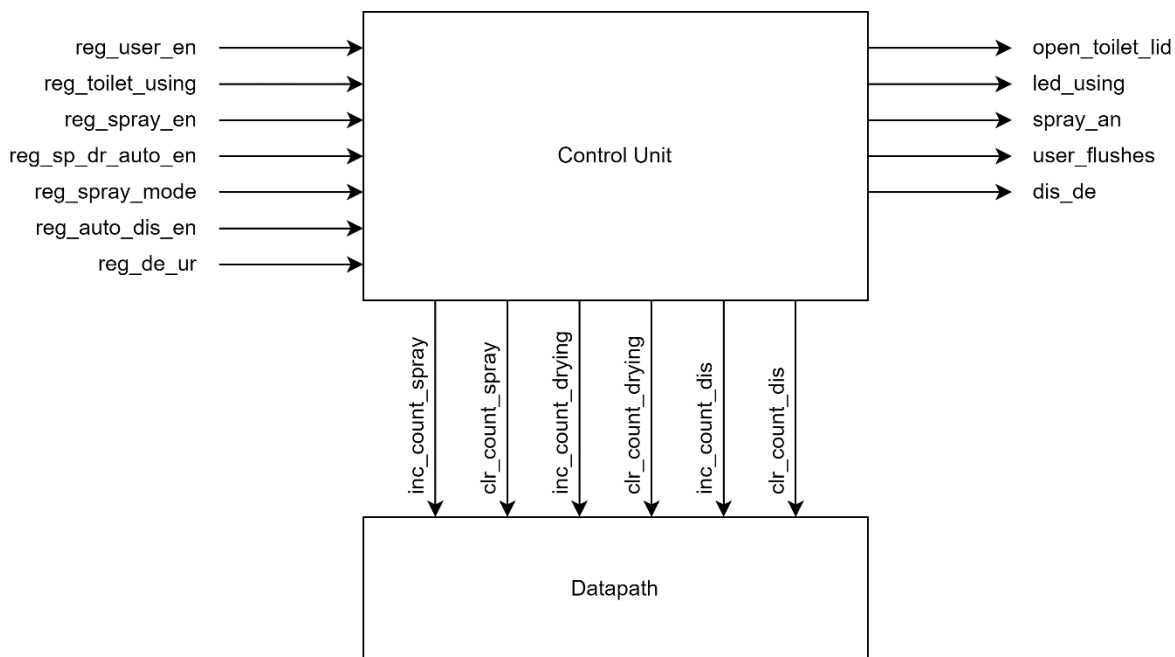
- IDLE: trạng thái IDLE, hệ thống sẽ trong trạng thái chờ người sử dụng.

- USING: trạng thái cos người sử dụng.
- SPRAY: trạng thái bồn cầu xịt.
- DRYING: trạng thái sấy khô.
- DISCHARGE: trạng thái xả nước.



Hình 3.1 Sơ đồ ASMD

Từ sơ đồ ASMD ta xác định được sơ đồ khối giao tiếp giữa control unit và datapath của hệ thống:



Hình 3.2 Sơ đồ khối Control Unit và Datapath

STT	Tên tín hiệu	Độ rộng (bit)	Chiều	Mô tả
1	reg_user_en	1	input	Cảm biến khoảng cách vật
2	reg_toilet_using	1	input	Cảm biến trọng lực
3	reg_spray_en	1	input	Nút bấm xịt nước
4	reg_sp_dr_auto_en	1	input	Nút bấm bật/tắt chế độ tự động xịt sấy
5	reg_spray_mode	1	input	Nút bấm chọn kiểu xịt
6	reg_auto_dis_en	1	input	Nút bấm bật/tắt tự động xả
7	reg_de_ur	1	input	Cảm biến xác định đi nặng/đi nhẹ
8	open_toilet_lid	1	output	Tín hiệu mở nắp bồn cầu
9	led_using	1	output	Tín hiệu thông báo đang có người sử dụng
10	spray_an	1	output	Tín hiệu chọn chế độ xịt: 1: xịt hậu môn 0: xịt cho phụ nữ
11	user_flushes	1	output	Tín hiệu thông báo người dùng tự xả
12	dis_de	1	output	Tín hiệu chọn chế độ xả nước: 1: đi nặng 0: đi nhẹ

3.1.3 Thiết kế khối register_block cho system_controller

Register Group	Register Name	Register Description	Address	# Instance	Suffix	Register Width	Reset Value	Register SW Access	Field Name	Field Description	Field Offset	Field Width	Field Reset Value	Field SW Access	Field HW Access	Field I/O Type	Simulation Value	Derivation
data_reg	data	Data Register	0	1	NA	32	0	RW	tem	Temperature Data	0	2	0	RO	WO	Input	0	
									rfu	Reserved for Future Use	2	30	0	NA	NA	NA	NA	
ctrl_reg	ctrl	Control Register	4	1	NA	32	0	RW	warm_en	Warm up enable 0: Disable 1: Enable	0	1	0	RW	RO	Output	0	
									rfu	Reserved for Future Use	1	31	0	NA	NA	NA	NA	
stt_reg	stt	Status Register	8	1	NA	32	0	RO	ready	Ready for using	0	1	0	RO	WO	Input	0	
									using	Status Using	1	1	0	RO	WO	Input	0	
									spraying	Status Spraying	2	1	0	RO	WO	Input	0	
									drying	Status Drying	3	1	0	RO	WO	Input	0	
									discharge	Status Discharge	4	1	0	RO	WO	Input	1	
									rfu	Reserved for Future Use	5	27	0	NA	NA	NA	NA	

Hình 3.3 Các thanh ghi trong system_controller

Thiết kế khối register block có giao tiếp APB được mô tả bằng ngôn ngữ verilog.

3.1.4 Thiết kế rtl

Phần thiết kế được mô tả trong source code tại thư mục *hdl*.

Các thiết kế bao gồm:

- clock_generator
- counter
- warm_up
- controller
- core
- smart_toilet_system_top

3.2 Thiết kế phần mềm

Trình bày tổng quát về phần mềm thiết kế, cũng như có 1 cái nhìn tổng thể về hệ thống, những vấn đề phát sinh và hướng giải quyết khi gặp vấn đề và đưa ra giải pháp thay thế với những bài toán đã đặt ra.

3.2.1 Mục tiêu

Nhóm đề xuất việc áp dụng phần mềm vào việc hiển thị thông báo và xử lý dữ liệu nhận về từ các sensors.

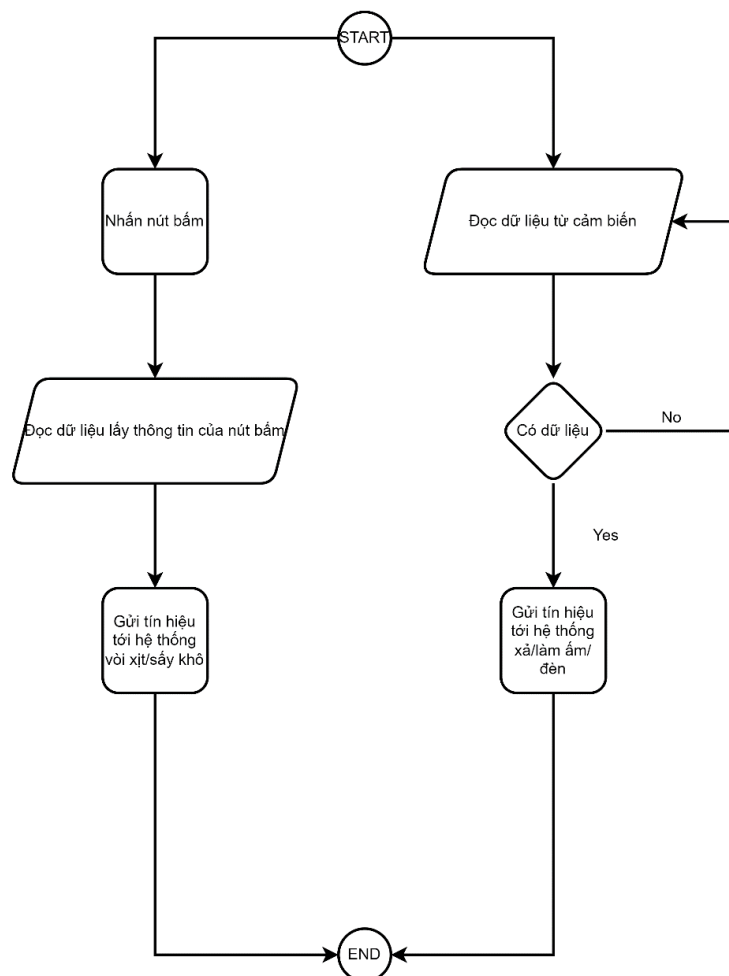
3.2.2 Lý do lựa chọn phần mềm

Lý do nhóm lựa chọn phần mềm để thực hiện mục đích này gồm những ý chính sau:

- Thứ nhất: Việc để phần mềm xử lý logic sẽ dễ thực hiện hơn phần cứng, mặc dù về chi phí thì cũng không quá tốn kém hơn việc sử dụng phần cứng.

- Thứ hai: Việc sử dụng phần mềm giúp chúng ta có thể bảo trì hệ thống dễ dàng hơn khi sử dụng nhiều cảm biến.
- Cuối cùng: Tính tùy biến cho phần mềm cao hơn, thay vì việc thiết kế FPGA thì đối với việc thay đổi thuật toán trở nên vô cùng khó khăn, gần như việc đổi thuật toán nếu hệ thống đã thương mại hóa thì gần như chúng ta phải thay thế con chip FPGA, nhưng đối với phần mềm, khi đưa vào vận hành, nếu thấy thuật toán đang có vấn đề, chúng ta cần tùy biến thì chúng ta chỉ cần nạp lại code, là hệ thống có thể thay đổi các vận hành. Điều này làm nên sức mạnh của hệ thống phần mềm.

3.2.3 Thiết kế



Hình 3.4 Sơ đồ phần mềm hệ thống

Giải thích:

- Phần mềm sẽ liên tục đọc các dữ liệu qua các thanh ghi từ khối điều khiển cảm biến để xác định trạng thái của môi trường.
- Dựa vào các dữ liệu đọc được đưa ra tín hiệu điều khiển cho khối điều khiển bồn cầu.
- Đọc các trạng thái từ các thanh ghi của khối điều khiển bồn cầu, hiển thị lên màn hình OLED.

3.2.4 Hướng triển khai

Việc triển khai được sử dụng ngôn ngữ lập trình C, lập trình trên phần mềm Vivado sử dụng board ZedBoard (Xilinx Zynq®-7000 All Programmable SoC).

Sử dụng hệ điều hành FreeRTOS (Real-time operating system). Đồng thời việc triển khai được viết trên ngôn ngữ C.

Các hàm cơ bản sử dụng:

```
static INLINE u32 Xil_In32 (UINTPTR Addr);
```

Hàm đọc 32 bit dữ liệu từ địa chỉ Addr

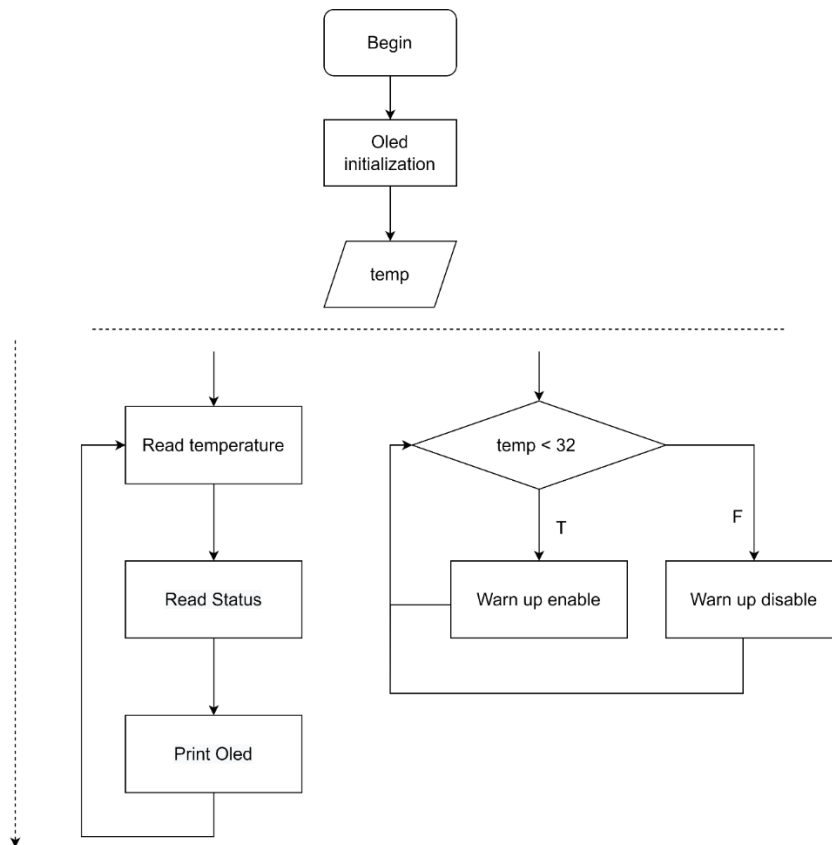
```
static INLINE void Xil_Out32 (UINTPTR Addr, u32 Value);
```

Hàm ghi 32 bit Value vào địa chỉ Addr

3.2.5 Sơ đồ thuật toán

Khi sử dụng hệ điều hành FreeRTOS, chúng ta sẽ chạy được đồng thời nhiều task và AXI sẽ chia ra các kênh đọc ghi riêng nên ở đây chúng ta sẽ chia làm 2 task:

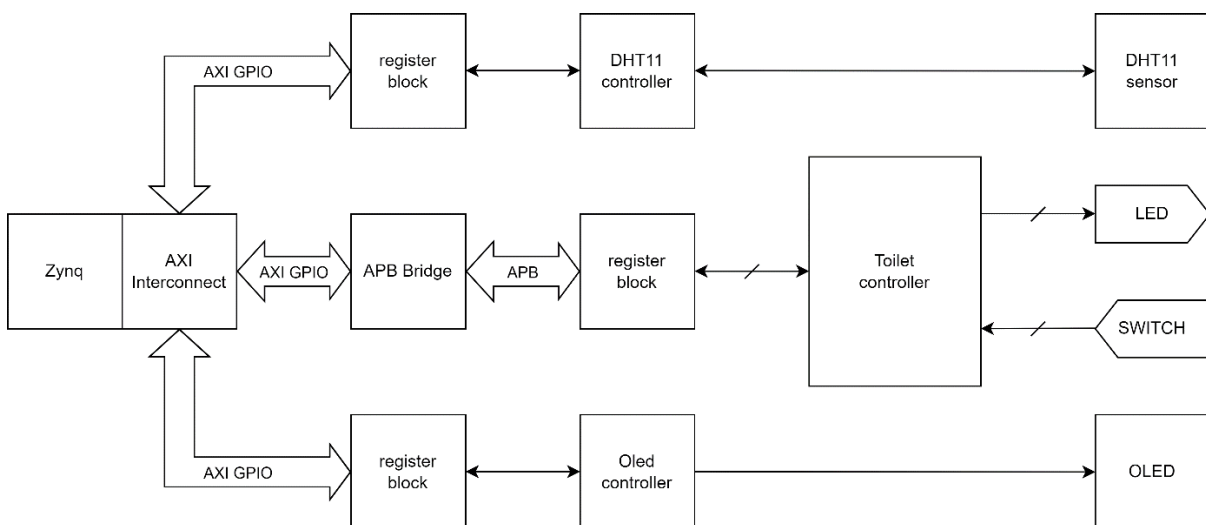
- Đọc dữ liệu (xReadTask)
- Ghi dữ liệu (xWriteTask)



Hình 3.5 Sơ đồ thuật toán

Hình 3.5 mô tả sơ đồ thuật toán của phần mềm.

Ở đây chúng ta sử dụng các hàm đọc ghi cấu hình các thanh ghi thông qua giao tiếp AXI để điều khiển các module. Các thanh ghi có địa chỉ và công dụng đã được mô tả chi tiết ở phần **Error! Reference source not found..**



Hình 3.6 Mô hình giao tiếp giữa vi xử lý và các module

Hình 3.6 mô tả sơ bộ toàn hệ thống giao tiếp giữa phần vi xử lý và phần cứng tự thiết kế. Trong project này, chúng em chọn giao tiếp GPIO để truyền thông giữa phần mềm và phần cứng. Để đồng bộ giữa 2 phần, chúng em thiết kế thêm một khối trung gian là register block hoạt động cùng tần số với tần số của controller, để capture tín hiệu điều khiển được gửi từ phần mềm, gửi đầu vào ổn định cho controller và lưu trạng thái hiện tại của controller để phần mềm xử lý hiển thị.

3.2.6 Giải quyết các vấn đề gặp phải trong quá trình triển khai

3.2.6.1 Giải quyết vấn đề về kit lập trình

Do nhóm chỉ có 1 bộ kit của một thành viên nên trong quá trình phát triển, gặp đôi chút khó khăn trong việc nạp code lên kit và chạy chương trình. Nhóm đã giải quyết vấn đề bằng cách lập trình chương trình C/C++ độc lập chạy khối logic và chương trình bên ngoài bằng “build-essential” ở trên Ubuntu.

Cách chạy thử:

1. *Clone git repository*
2. *Install build-essential*
3. *Build source bằng cách sử dụng lệnh gcc*
4. *Chạy chương trình bằng cách chạy file đã build ở bước 3*

Sau khi kiểm tra chương trình đã chạy thỏa mãn yêu cầu, chúng ta tiến hành ghép code vào SDK Vivado. Về mặt cơ bản source code sẽ không khác nhiều, chỉ có một vài thư viện cần phải thay đổi và đầu vào sẽ chuyển qua sử dụng thư viện của Xilinx.

3.2.6.2 Giải quyết vấn đề về phần cứng các cảm biến

Do không có cảm biến mô tả việc phát hiện người, cảm biến trọng lực và cảm biến phát hiện đi nặng hay đi nhẹ nên nhóm đã quyết định sử dụng các switches, buttons để mô phỏng tín hiệu từ cảm biến. Vì vậy, việc nhận tín hiệu sẽ được nối trực tiếp vào phần cứng.

3.2.6.3 Giải quyết vấn đề liên quan đến cảm biến nhiệt độ

Do mới chỉ sử dụng cảm biến nhiệt độ phòng nên việc xác định nhiệt độ nước chưa được tính toán. Để tiện cho việc mô phỏng, ngưỡng làm ấm nước là 30 (°C).

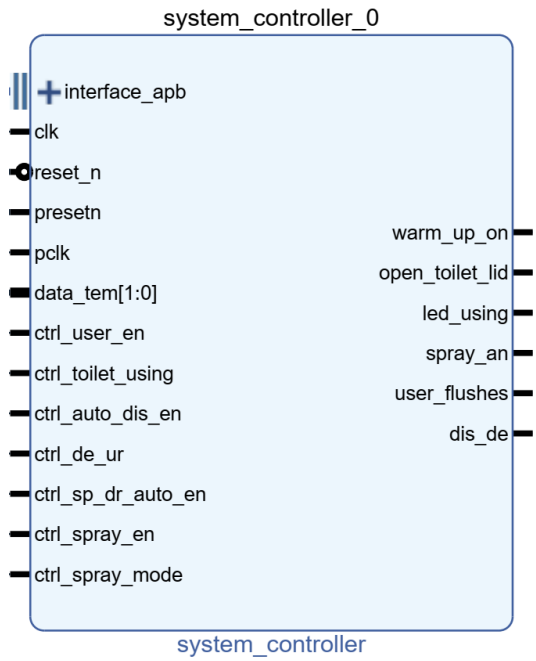
Bus Architecture - Advanced Peripheral Bus) nên chúng ta cần AXI APB Bridge để giao tiếp với vi xử lý.

Các chân in, out sẽ được gán ràng buộc vào các LED, SWITCH, BUTTON và GPIO có sẵn trên kit để thuận tiện mô phỏng cũng như kiểm thử.

4.2 Các IP dùng trong hệ thống

4.2.1 Smart toilet system controller (system_controller)

Dựa vào phân thiết kế ở chương 3 bao gồm: khối Register block và khối System controller được triển khai bằng ngôn ngữ verilog. Sử dụng phần mềm Vivado để đóng gói thành IP system_controller.



Hình 4.2 Sơ đồ khối của smart toilet system controller

Dưới đây là bảng gán ràng buộc cho các tín hiệu của system_controller lên các chân của FPGA

Bảng 4.1 Ràng buộc các tín hiệu của system_controller

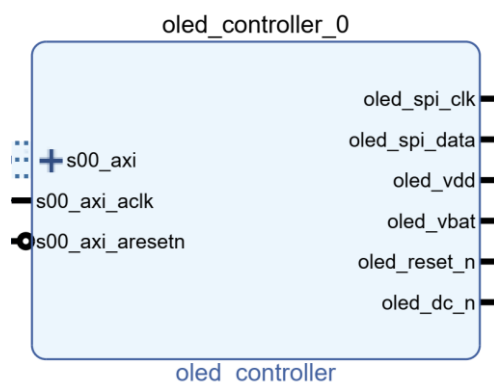
Tín hiệu	Cổng	Mô tả
warm_up_on	T22	LED0
open_toilet_lid	T21	LED1
led_using	U22	LED2
spray_an	U21	LED3

dis_de	V22	LED4
ctrl_toilet_using	H22	SW2
ctrl_spray_mode	F21	SW3
ctrl_spray_en	H19	SW4
ctrl_sp_dr_auto_en	H18	SW5
ctrl_de_ur	H17	SW6
ctrl_auto_dis_en	M15	SW7
user_flushes	Y11	JA1

Bảng 4.1 Bảng 4.1 Ràng buộc các tín hiệu của system_controller chỉ ra các tín hiệu được constraint với các cổng trên kit zedboard. Các mô tả sẽ được chú thích trên kit.

4.2.2 Oled controller (oled_controller)

IP oled_controller sử dụng giao thức SPI để giao tiếp với màn hình oled đã được gắn sẵn trên kit. Người dùng sẽ cấu hình các giá trị trong thanh ghi () để điều khiển cũng như hiển thị dữ liệu lên màn hình.



Hình 4.3 Sơ đồ khối của oled_controller

Địa chỉ cơ sở (Base address): 0x43C00000

Luồng hoạt động:

- Ghi dữ liệu vào thanh ghi data (0x08) là 1 byte tương ứng với ký tự muốn in lên màn hình
- Kích hoạt IP sử dụng thanh ghi ctrl (0x00)
- Kiểm tra thanh ghi status (0x04) nếu mà bằng 0 thì sẽ chuyển sang trạng thái “polling mode”, sau khi bằng 1 tiến hành cấu hình lại thanh ghi status về 0.

Màn hình hiển thị tối đa 64 ký tự.

Bảng 4.2 Tổ chức thanh ghi trong oled_controller

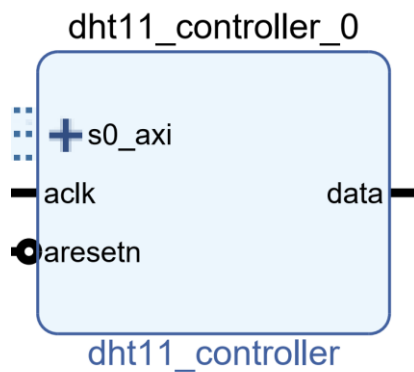
Thanh ghi	Mô tả	Địa chỉ	Mô tả của thanh ghi
data	Thanh ghi dữ liệu	0x08	Ký hiệu cần ghi lên màn hình (1 char – 8bit)
ctrl	Thanh ghi điều khiển	0x00	Kích hoạt bộ điều khiển 1: Bật 0: Tắt
stt	Thanh ghi trạng thái	0x04	Kiểm tra trạng thái để quyết định ghi dữ liệu vào thanh ghi data

Bảng 4.3 Ràng buộc các tín hiệu của oled_controller

Tín hiệu	Cổng	Mô tả
oled_dc_n	U10	OLED_DC
oled_reset_n	U9	OLED_RES
oled_spi_clk	AB12	OLED_SCLK
oled_spi_data	AA12	OLED_SDIN
oled_vbat	U11	OLED_VBAT
oled_vdd	U12	OLED_VDD (GND)

4.2.3 DHT11 controller (dht11_controller)

Cảm biến độ ẩm và nhiệt độ DHT11 Temperature Humidity Sensor là cảm biến rất thông dụng hiện nay vì chi phí rẻ và rất dễ lấy dữ liệu thông qua giao tiếp TTL - 1 wire (giao tiếp digital 1 dây truyền dữ liệu duy nhất). Bộ tiền xử lý tín hiệu tích hợp trong cảm biến giúp bạn có được dữ liệu chính xác mà không phải qua bất kỳ tính toán nào. So với cảm biến đời mới hơn là DHT22 thì DHT11 cho khoảng đo và độ chính xác kém hơn rất nhiều.



Hình 4.4 Sơ đồ khối của DHT11 controller

Địa chỉ cơ sở (base address): 0x43C20000

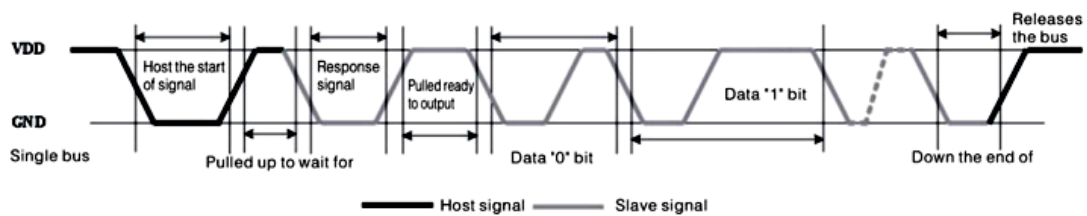
Bộ dht11_controller điều khiển và xử lý dữ liệu nhận được từ cảm biến DHT11 sau đó dữ liệu đã xử lý được ghi vào các thanh ghi. Từ đó, vi xử lý có thể đọc nhiệt độ, độ ẩm từ các thanh ghi qua giao thức AXI.

Bảng 4.4 Tổ chức thanh ghi trong dht11_controller

Thanh ghi	Mô tả	Địa chỉ
reg_ctrl	Thanh ghi điều khiển	0x00
reg_debug	Thanh ghi debug	0x04
reg_temperature	Thanh ghi dữ liệu nhiệt độ	0x08
reg_humidity	Thanh ghi dữ liệu độ ẩm	0x0C
reg_crc	Thanh ghi Cyclic Redundancy Check	0x10
reg_ack	Thanh ghi phản hồi	0x14

Bảng 4.5 Ràng buộc các tín hiệu của dht11_controller

Tín hiệu	Cổng	Mô tả
data	AA11	JA2

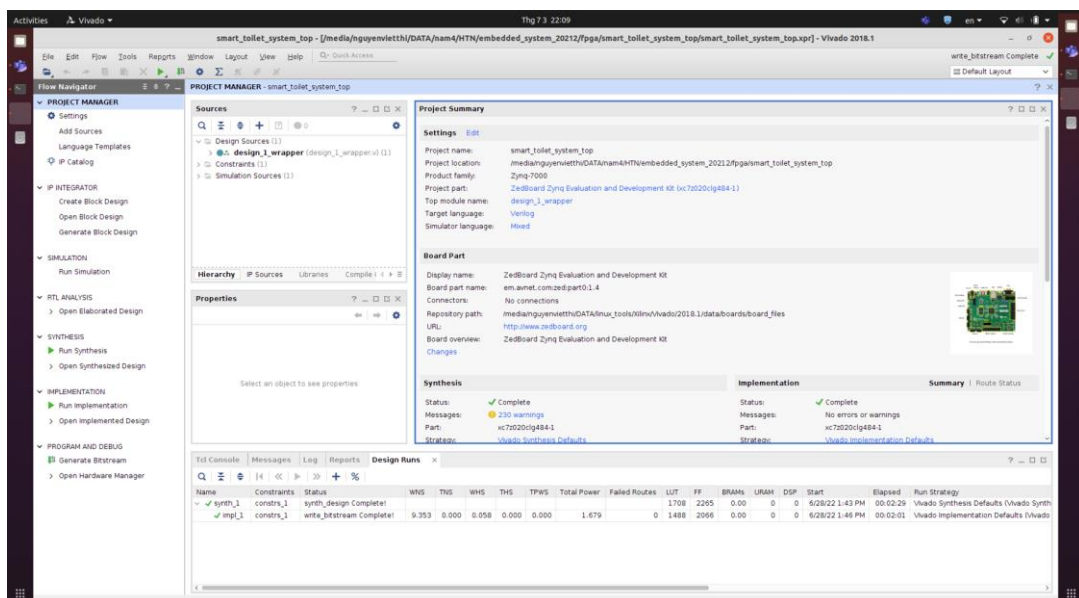


Hình 4.5 Sơ đồ thời gian quá trình gửi tín hiệu điều khiển từ master và phản hồi tín hiệu từ slave của cảm biến DHT11

4.3 Tổng hợp và triển khai trên phần mềm Vivado

Trong phần nhóm sử dụng phần mềm Vivado v2018.1 (64-bit).

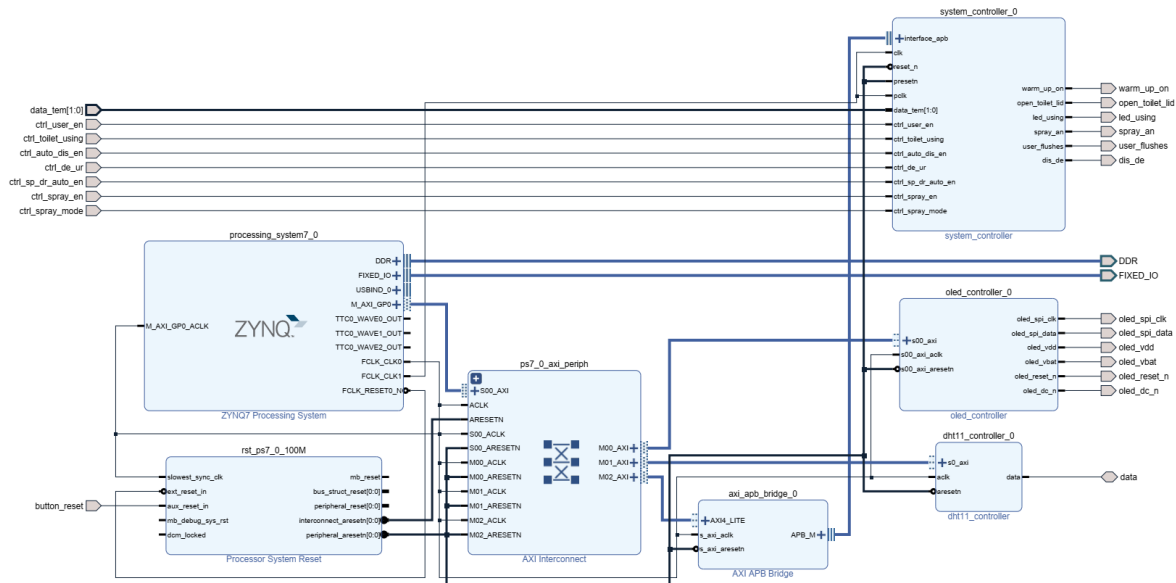
4.3.1 Tạo project, thêm các file source code mô tả phần cứng



Hình 4.6 Tạo project và thêm các file source code

4.3.2 Tạo Block design

Từ các IP được thiết kế và các IP có sẵn của Xilinx, block design được thiết kế như hình dưới.



Hình 4.7 Block design của hệ thống

4.3.3 Tạo constraint file

Tạo file ràng buộc nối các input và output của block design với các chân của kit FPGA (chi tiết xem tại ZedBoard constraint file)

Dưới đây là ví dụ constraint các chân của oled controller với port trên kit.

```
#OLED

set_property PACKAGE_PIN U10 [get_ports oled_dc_n];
set_property PACKAGE_PIN U9 [get_ports oled_reset_n];
set_property PACKAGE_PIN AB12 [get_ports oled_spi_clk];
set_property PACKAGE_PIN AA12 [get_ports oled_spi_data];
set_property PACKAGE_PIN U11 [get_ports oled_vbat];
set_property PACKAGE_PIN U12 [get_ports oled_vdd];
set_property IOSTANDARD LVCMOS33 [get_ports oled_dc_n];
set_property IOSTANDARD LVCMOS33 [get_ports oled_reset_n];
set_property IOSTANDARD LVCMOS33 [get_ports oled_spi_clk];
set_property IOSTANDARD LVCMOS33 [get_ports oled_spi_data];
set_property IOSTANDARD LVCMOS33 [get_ports oled_vbat];
set_property IOSTANDARD LVCMOS33 [get_ports oled_vdd];
```

4.3.4 Tổng hợp (synthesis)

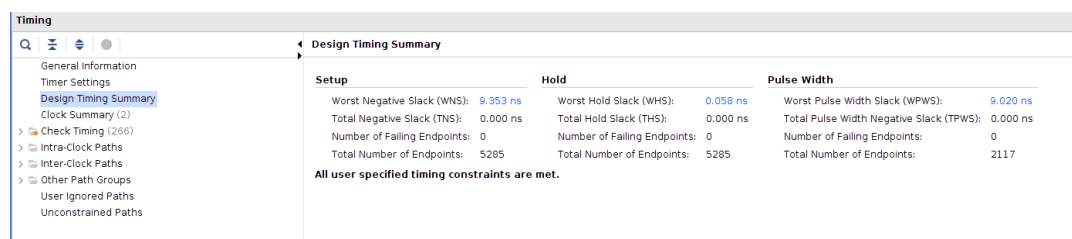
Kết quả của quá trình tổng hợp được thể hiện tổng bảng sau:

Bảng 4.6 Số lượng LUT và Register của thiết kế

Loại		Sử dụng	Tỷ lệ
Slice LUTs		1708	3.21
	LUT as Logic	1642	3.09
	LUT as Memory	LUT as Distributed RAM	66
		LUT as Shift Register	0
Slice Registers		2297	2.16
	Register as Flip Flop	2265	2.13
	Register as Latch	32	0.03
F7 Muxes		108	0.41
F8 Muxes		38	0.29

4.3.5 Triển khai (implement)

Hình 4.8 cho ta thấy, thiết kế thỏa mãn setup time và hold time để đảm bảo slack dương. Kết quả timing tốt.

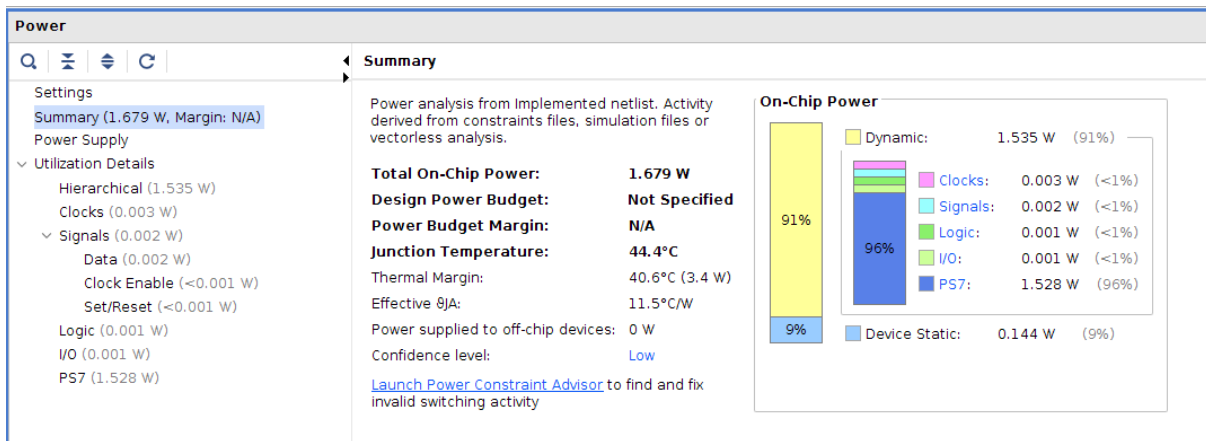


Setup	Hold	Pulse Width
Worst Negative Slack (WNS): 9.353 ns	Worst Hold Slack (WHS): 0.058 ns	Worst Pulse Width Slack (WPWS): 9.020 ns
Total Negative Slack (TNS): 0.000 ns	Total Hold Slack (THS): 0.000 ns	Total Pulse Width Negative Slack (TPWS): 0.000 ns
Number of Failing Endpoints: 0	Number of Failing Endpoints: 0	Number of Failing Endpoints: 0
Total Number of Endpoints: 5285	Total Number of Endpoints: 5285	Total Number of Endpoints: 2117

All user specified timing constraints are met.

Hình 4.8 Báo cáo về timing

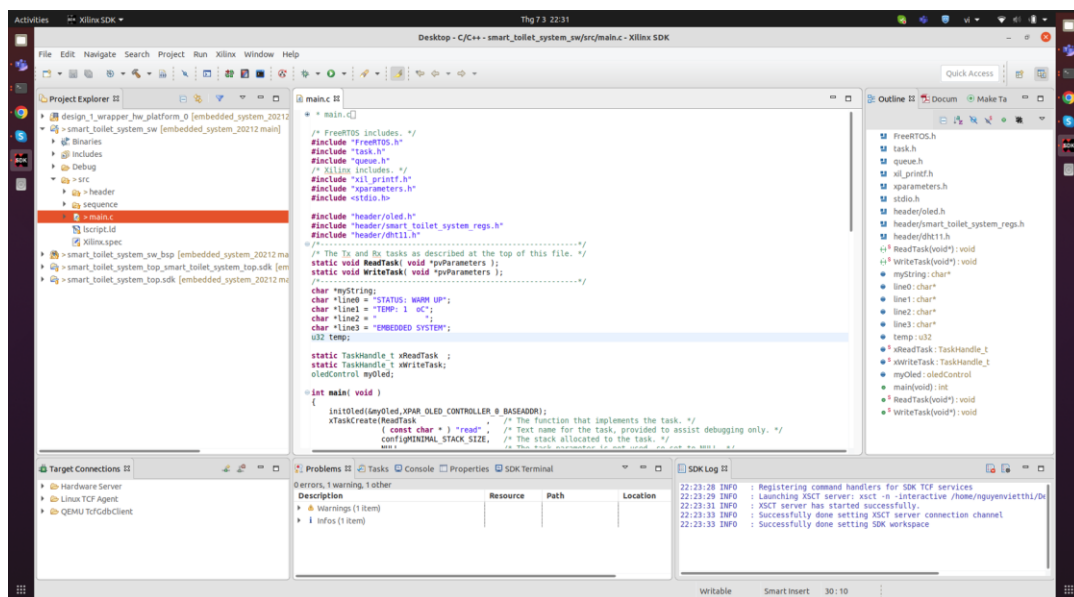
Hình 4.9 cho ta thấy kết quả về mặt công suất của thiết kế. Ở đây tổng công suất không tính công suất dự trữ (margin) là 1.679 W



Hình 4.9 Báo cáo về công suất của thiết kế

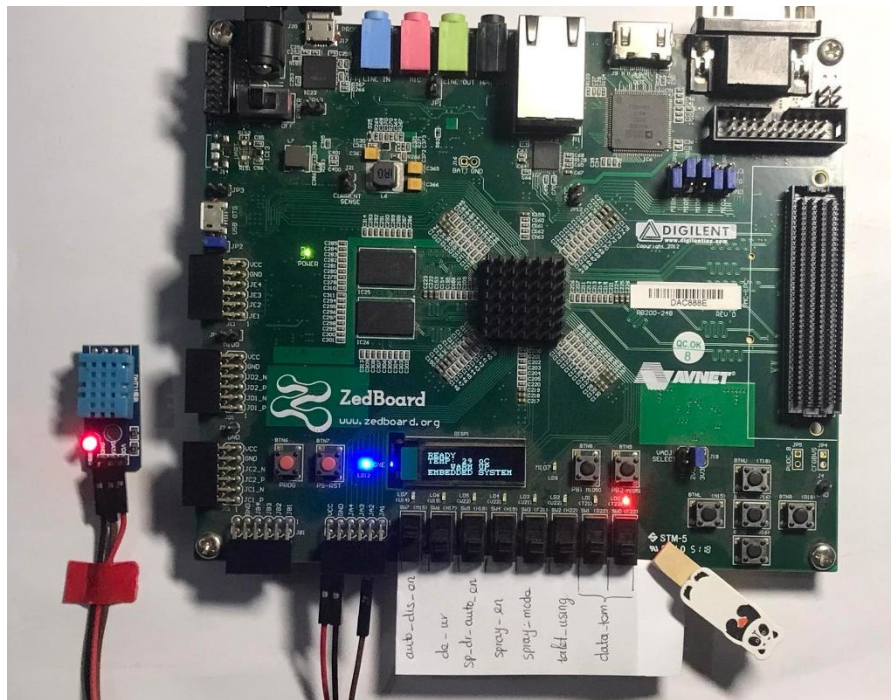
4.3.6 Gen bitstream và tạo SDK (software development kit)

Sau khi gen bitstream trên Vivado, ta tiến hành tạo project trên phần mềm SDK, thiết kế lập môi trường biên dịch, platform hardware và biên dịch code.



Hình 4.10 Thiết kế trên SDK

4.3.7 Kết quả



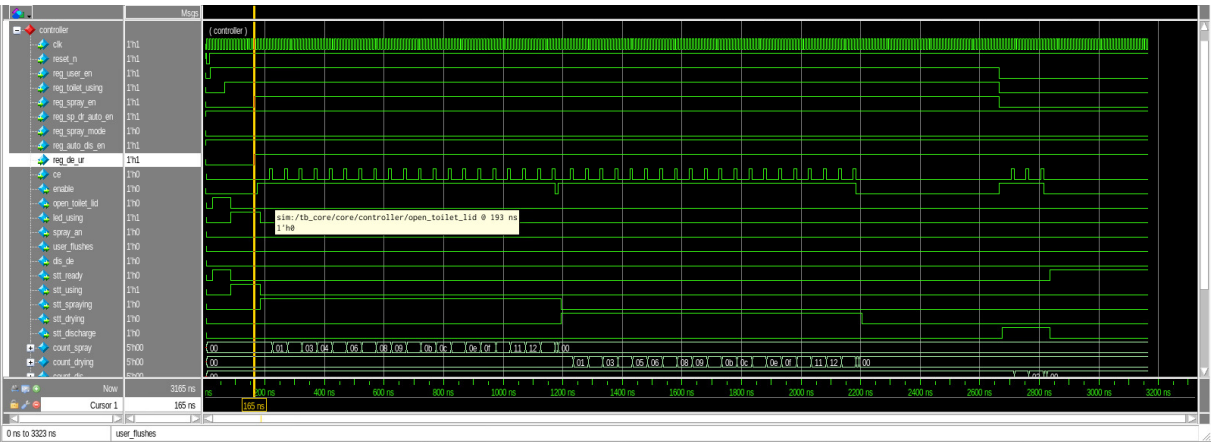
Hình 4.11 Kết quả trên kit FPGA

Sử dụng các botton, switch, led, oled đã được contrain sẵn. Tiến hành thử những kịch bản khác nhau.

Kết quả: hệ thống đã chạy đúng với function đề ra ở khâu thiết kế. Về mặt timing thỏa mãn (độ trễ) và năng lượng tiêu tốn thấp trong phạm vi phù hợp.

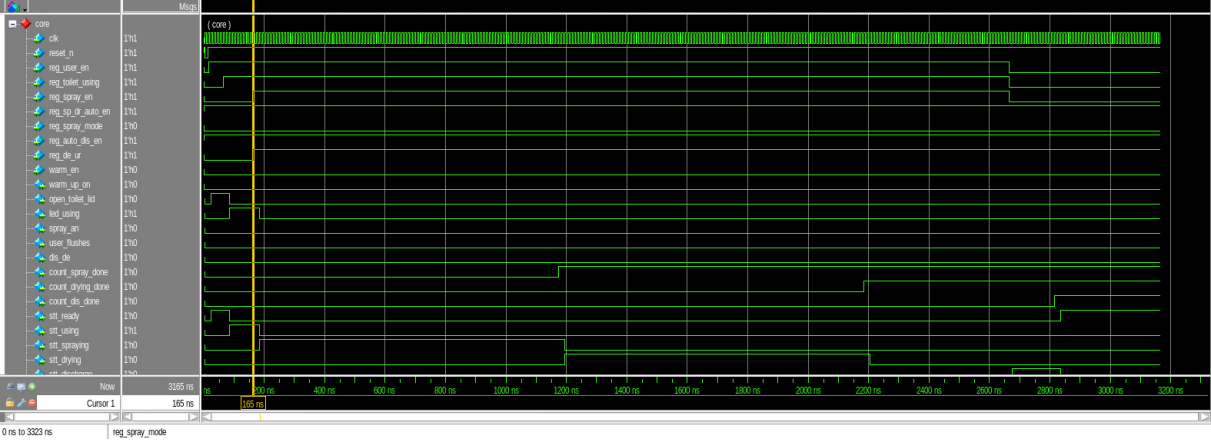
CHƯƠNG 5. KIỂM THỬ (VERIFICATION)

5.1.2.2 Khởi controller



Hình 5.2 Kết quả mô phỏng khối controller

5.1.2.3 Khởi core



Hình 5.3 Kết quả mô phỏng khối core

CHƯƠNG 6. BÁO CÁO CÁ NHÂN

Trong chương này em sẽ tổng hợp lại các phần mà em đã thực hiện và đóng góp và kết quả chung của nhóm.

6.1 Đưa ra ý tưởng thiết kế và kiểm tra hoạt động của hệ thống

Đưa ra và thống nhất các chức năng của hệ thống buồn cầu thông minh.

Kiểm tra và góp ý để ASMD thực hiện đúng các chức năng đã được đặt ra từ trước.

6.2 Triển khai hệ thống trên ZedBoard

6.2.1 Tìm kiếm và triển khai các IP để điều khiển OLED và DHT11

Tham khảo các IP có sẵn trên mạng, đọc datasheet.

Ở đây chúng em có tham khảo của:

- DHT11 controller: [kayas257/DHT11-zynq: Zynq Design Example \(github.com\)](https://github.com/kayas257/DHT11-zynq)
- OLED controller: [vipinkmenon/ZynqOLED: Source code for Zynq OLED controller \(github.com\)](https://github.com/vipinkmenon/ZynqOLED)

Đọc hiểu code VHDL của 2 repo trên và áp dụng vào hệ thống của mình.

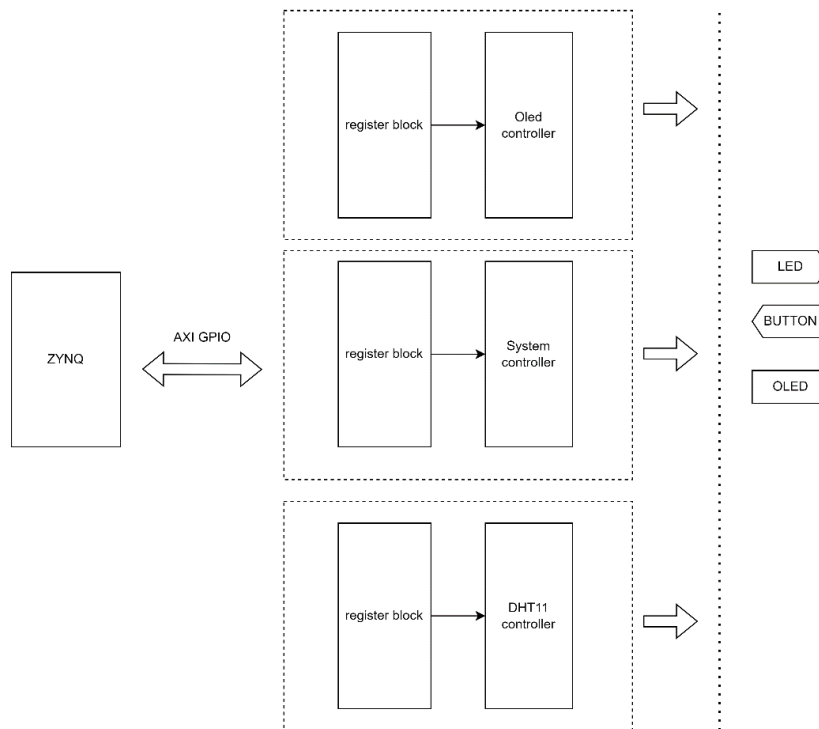
Hiểu sơ bộ các hoạt động cơ bản của 2 IP trên.

Tìm các thanh ghi và địa chỉ thanh ghi dùng để cấu hình cũng như đọc trạng thái của các IP. Từ đó có thể điều khiển cũng như đọc dữ liệu của oled và cảm biến DHT11.

Đóng gói các IP trên bằng phần mềm vivado và gom các cổng thành các interface tương ứng (AXI, APB) để dễ dàng đi dây trong khối tổng.

6.2.2 Đóng gói system_controller

Từ code verilog đã được triển khai từ ASMD, tiến hành đóng gói và gom các nhóm tín hiệu.



Hình 6.3. Mô hình giao tiếp giữa vi xử lý và các module

6.2.4 Tạo constraint file

Là file để kết nối các cổng inout của block design với các chân đã được định nghĩa sẵn trên ZedBoard.

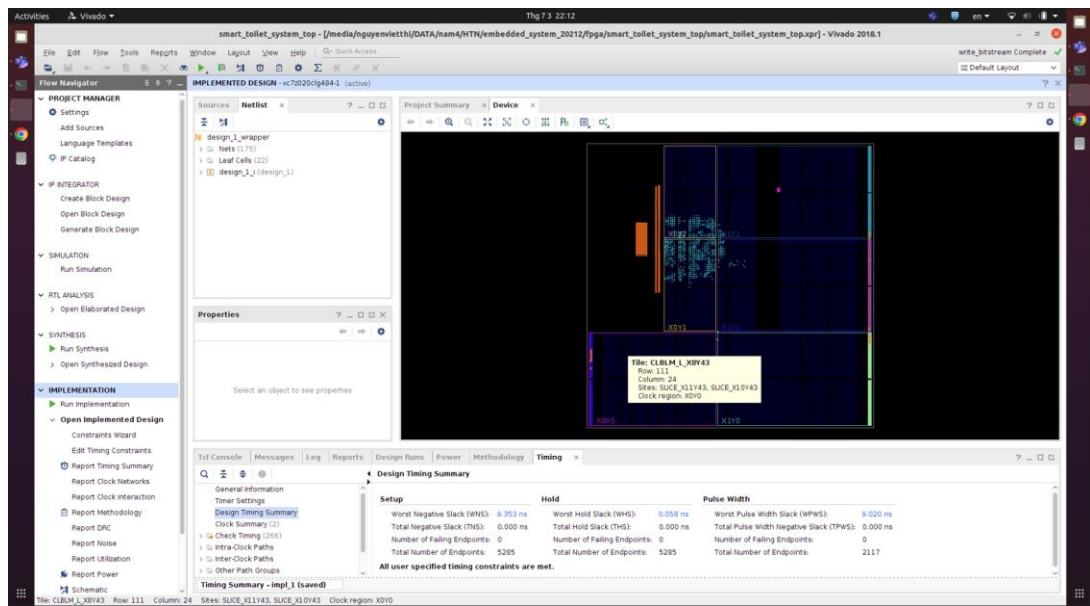
Dưới đây là ví dụ constraint các chân của oled controller với port trên kit.

```

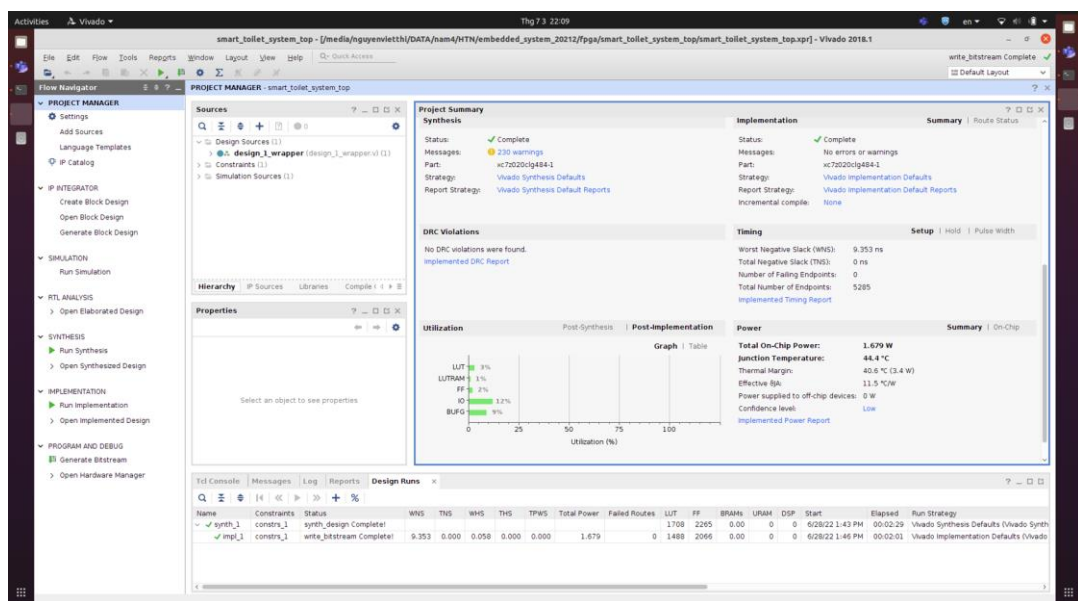
#OLED

set_property PACKAGE_PIN U10 [get_ports oled_dc_n];
set_property PACKAGE_PIN U9 [get_ports oled_reset_n];
set_property PACKAGE_PIN AB12 [get_ports oled_spi_clk];
set_property PACKAGE_PIN AA12 [get_ports oled_spi_data];
set_property PACKAGE_PIN U11 [get_ports oled_vbat];
set_property PACKAGE_PIN U12 [get_ports oled_vdd];
set_property IOSTANDARD LVCMOS33 [get_ports oled_dc_n];
set_property IOSTANDARD LVCMOS33 [get_ports oled_reset_n];
set_property IOSTANDARD LVCMOS33 [get_ports oled_spi_clk];
set_property IOSTANDARD LVCMOS33 [get_ports oled_spi_data];
set_property IOSTANDARD LVCMOS33 [get_ports oled_vbat];
set_property IOSTANDARD LVCMOS33 [get_ports oled_vdd];
  
```

6.2.5 Tổng hợp và implement thiết kế trên vivado



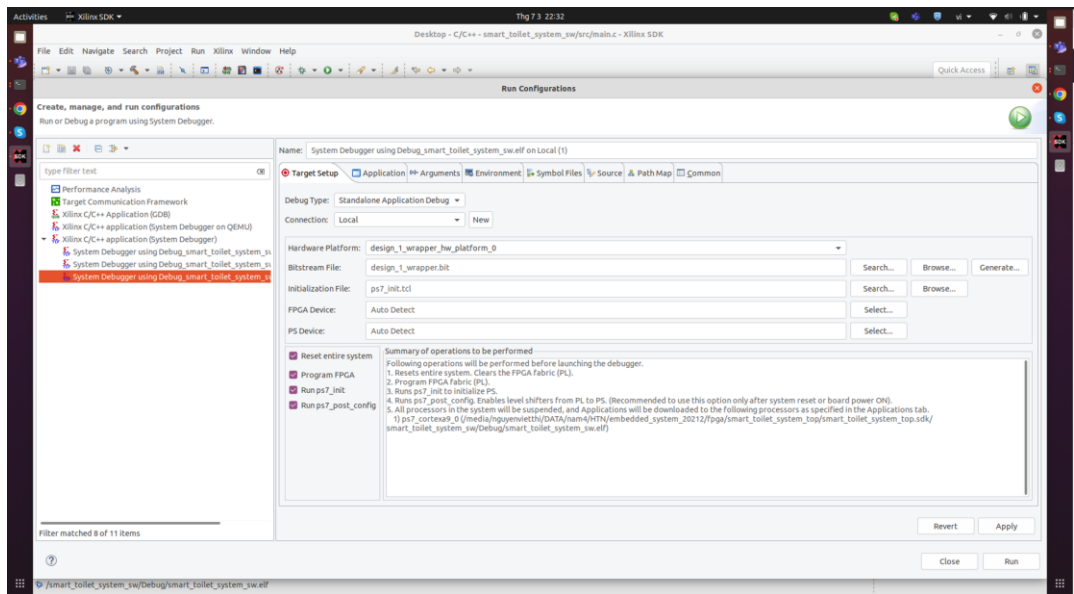
Hình 6.4 Kết quả của quá trình implement



Hình 6.5 Report của hệ thống

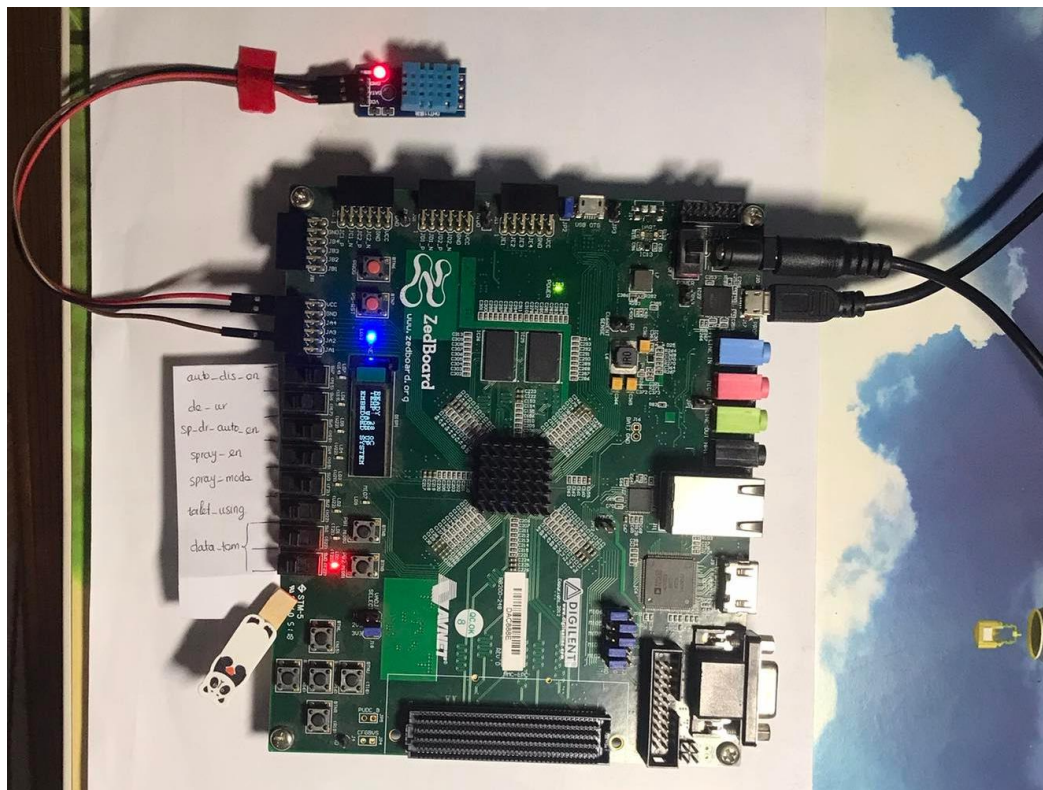
6.2.6 Tạo bitstream và viết SDK

Tiến hành tạo bitstream và export hardware, sau đó triển khai phần mềm trên xsdk.



Hình 6.8 Cấu hình để nạp hệ thống vào ZedBoard

6.3 Kiểm thử các chức năng bằng kit



Hình 6.9. Hình ảnh thực tế khi đã nạp hệ thống lên kit

Sử dụng các botton, switch, led, oled đã được constrain sẵn. Tiến hành thử những kịch bản khác nhau.

Kết quả: hệ thống đã chạy đúng với function đề ra ở khâu thiết kế. Về mặt timing thỏa mãn (độ trễ) và năng lượng tiêu tốn thấp trong phạm vi phù hợp.

KẾT LUẬN

Sau một thời gian nghiên cứu về hệ thống nhúng, đặc biệt là hệ thống nhúng điều khiển bồn cầu tự động, nhóm em đã nắm được các quy trình thiết kế một sản phẩm nhúng cũng như cấu tạo của bồn cầu tự động và hoạt động của nó. Đồng thời nhóm đã thiết kế hệ thống điều khiển bồn cầu tự động trên Xilinx ZedBoard. Hệ thống gồm các sensor, button sẽ nhận tín hiệu và đưa vào bộ điều khiển được viết bằng ngôn ngữ C, sau đó các tín hiệu sẽ được gửi tới bộ điều khiển được mô tả bằng ngôn ngữ Verilog để điều khiển hoạt động của bồn cầu.

Hệ thống điều khiển bồn cầu tự động hoạt động đúng như những chỉ tiêu kỹ thuật đã được đề ra đảm bảo hoạt động theo yêu cầu. Trong quá trình thực hiện do thời gian và kiến thức còn nhiều sai sót, mong thầy cô và các bạn đóng góp ý kiến để nhóm em hoàn thiện hơn.

Nhóm em xin chân thành cảm ơn TS. Phạm Văn Tiến đã giúp đỡ nhóm chúng em trong quá trình thực hiện bài tập lớn này.

TÀI LIỆU THAM KHẢO

- [1] kayas257, "DHT11-zynq: DHT11 Interfacing Zynq PL," Github, 2019. [Online]. Available: <https://github.com/vipinkmenon/ZynqOLED>.
- [2] vipinkmenon, "ZynqOLED: Source code for Zynq OLED controller," Github, 2020. [Online]. Available: <https://github.com/kayas257/DHT11-zynq>.
- [3] phamngoclam2628, nguyenvietthi and huynam1802, "Smart_Toilet_System_Release," Github, 2022. [Online]. Available: https://github.com/phamngoclam2628/Smart_Toilet_System_Release.