

Tóm tắt

Trong bản thuyết trình này, nhóm chúng tôi sẽ giới thiệu về công nghệ Bluetooth Low Energy (BLE) và khả năng ứng dụng thực tiễn và hiện thực thiết kế sản phẩm móc khóa thông minh dựa trên nền tảng công nghệ BLE và lập trình ứng dụng di động sử dụng công nghệ này.

Tài liệu sẽ gồm có 4 phần chính:

- Tổng quan: giới thiệu về công nghệ BLE, mục tiêu và phạm vi đề tài.
- Thiết kế và hiện thực sản phẩm móc khóa: xác định mục tiêu ứng dụng và lựa chọn thiết bị phần cứng, thiết kế mô hình thiết bị và giao thức với ứng dụng thiết bị di động, lập trình ứng dụng dành cho hệ điều hành Android.
- Thử nghiệm và kết quả đạt được: thử nghiệm khả năng ứng dụng thiết bị trong thực tế.
- Tổng kết: rút ra các kết luận nhận xét về kết quả đạt được và hướng phát triển trong tương lai.

Mục lục

Danh sách hình vẽ	v
Danh sách bảng	vi
1 Tổng quan	1
1.1 Mở đầu	1
1.1.1 Mục đích đề tài và tính cấp bách	1
1.1.2 Mô tả đề tài	2
1.1.3 Tình hình nghiên cứu	2
1.1.4 Mục tiêu - Phạm vi - Đối tượng nghiên cứu	3
1.2 Bluetooth Low Energy (BLE)	3
1.2.1 Khái niệm và lịch sử phát triển Bluetooth	4
1.2.2 Phân loại vai trò thiết bị BLE	6
1.2.3 Cách thức hoạt động của BLE	6
1.2.4 Các vi điều khiển tích hợp công nghệ BLE	8
2 Thiết kế và hiện thực	
sản phẩm móc khóa thông minh - Smart Keyring	9
2.1 Thiết kế tính năng sản phẩm	9
2.2 Các hướng tiếp cận vấn đề	10
2.2.1 Phát triển thiết bị chỉ dựa vào SoC CC2540/2541	10
2.2.2 Kết hợp MCU và Module BLE HM-10	11
2.3 Sơ đồ hoạt động	11
2.3.1 Tổng quát	11
2.3.2 Nhận lệnh báo từ thiết bị di động	12
2.3.3 Kích hoạt thiết bị di động bật chế độ báo hiệu	13
2.3.4 Sơ đồ trạng thái hoạt động khi ngắt kết nối	14

2.4	Hiện thực phần cứng	16
2.4.1	Lựa chọn linh kiện và thiết bị phần cứng	16
2.4.2	Lập trình firmware cho thiết bị	22
2.5	Hiện thực ứng dụng di động trên Android	25
2.5.1	Các khái niệm trong lập trình BLE trên Android	25
2.5.2	Phát triển ứng dụng Android	26
3	Thử nghiệm và đánh giá kết quả đạt được	32
3.1	Thử nghiệm	32
3.1.1	Khoảng cách hoạt động	32
3.1.2	Năng lượng tiêu thụ	33
3.2	Đánh giá kết quả ứng dụng thực tế	34
4	Tổng kết	35
4.1	Thành quả đạt được và khó khăn	35
4.1.1	Thành quả đạt được	35
4.1.2	Khó khăn	35
4.2	Tính ứng dụng thực tiễn của sản phẩm	36
4.3	Hướng phát triển	36
	Tài liệu tham khảo	37

Danh sách hình vẽ

1.1	Các ứng dụng Bluetooth	4
1.2	Sơ đồ hoạt động của BLE	7
1.3	Module HM-10	8
2.1	Sơ đồ hoạt động tổng quát	12
2.2	Sơ đồ hoạt động khi nhận lệnh báo từ thiết bị di động	13
2.3	Sơ đồ kích hoạt thiết bị di động bật chế độ báo hiệu	14
2.4	Sơ đồ trạng thái trên thiết bị Smart Keyring	15
2.5	Sơ đồ hoạt động trên thiết bị di động	16
2.6	Module BLE HM-10	17
2.7	MCU ATmega328P	18
2.8	Schematic của thiết bị Smart Keyring	20
2.9	Mạch thử nghiệm	21
2.10	Mặt trước mạch thiết bị	21
2.11	Mặt sau mạch thiết bị	22
2.12	Các chế độ sleep mode khác nhau và khả năng hoạt động của từng chế độ	22
2.13	Giao diện kiểm tra BLE	28
2.14	Giao diện scan thiết bị BLE	29
2.15	Giao diện chính của ứng dụng di động	30

Danh sách bảng

1.1	Bảng so sánh các công nghệ truyền không dây	5
3.1	Thử nghiệm khoảng cách hoạt động khi không có vật cản	33
3.2	Thử nghiệm khoảng cách hoạt động khi có vật cản	33

Chương 1

Tổng quan

1.1 Mở đầu

1.1.1 Mục đích đề tài và tính cấp bách

Trong cuộc sống bận rộn hiện nay, con người chúng ta hay có xu hướng bị xao nhãng và bỏ quên các thiết bị nhỏ. trong đó có thiết bị điện thoại di động và chùm chìa khóa là hai vật rất quan trọng và thường hay bỏ quên nhất. Và tìm kiếm chúng không hề dễ dàng, nhất là khi đang vội thì sẽ làm mọi thứ rối tung lên.

Và đồng thời hiện nay công nghệ phát triển cho các thiết bị kết nối không dây phát triển mạnh, có thể ứng dụng cho nhiều lĩnh vực. Bluetooth Low Energy là một trong những cái tên nổi bật nhất trong các công nghệ truyền dữ liệu không dây bởi đặc tính tiện lợi, phổ biến và tiết kiệm năng lượng. Và nhóm chúng tôi muốn tìm hiểu và ứng dụng công nghệ này vào cuộc sống thực tiễn.

Từ điều đó đã thúc đẩy nhóm chúng tôi tìm cách giải quyết và nảy ra ý tưởng tạo ra sản phẩm móc khóa thông minh - Smart Keyring có chức năng kết nối với thiết bị di động sử dụng công nghệ BLE để giải quyết vấn đề trên dựa trên các tính năng của BLE.

1.1.2 Mô tả đề tài

Đề tài sẽ chia làm 2 phần chính:

- **Sản phẩm móc khóa:** 1 thiết bị có kích thước nhỏ, năng lượng tiêu thụ ít.

Chức năng:

- Báo hiệu bằng âm thanh và ánh sáng khi có yêu cầu định vị.
- Điều khiển chức năng định vị trên thiết bị di động qua nút ấn.
- Báo hiệu khi mất kết nối với thiết bị di động.

- **Ứng dụng di động:** ứng dụng chạy trên nền tảng Android.

Chức năng:

- Gửi yêu cầu định vị tới sản phẩm móc khóa để kích hoạt tính năng báo hiệu trên móc khóa.

- Báo hiệu bằng âm thanh yêu cầu định vị.
- Báo hiệu khi bị mất kết nối với sản phẩm móc khóa.

1.1.3 Tình hình nghiên cứu

- Tình hình nghiên cứu ngoài nước:

- Thiết bị TrackR (www.thetrackr.com) : sử dụng công nghệ Bluetooth Low Energy, có chức năng tìm kiếm thiết bị và chống thất lạc để quên. Tuy nhiên hạn chế là chưa phân phối chính thức ở Việt Nam và giá thành còn khá cao so với mức sống của người Việt Nam.

- Tình hình nghiên cứu trong nước:

- Chưa thấy xuất hiện sản phẩm có chức năng tương tự sử dụng công nghệ BLE.
- Có vài sản phẩm có chức năng gần giống sử dụng 2 thẻ tag RF để tìm nhau, chưa có chức năng cảnh báo khi mất kết nối (trường hợp để quên hoặc trộm mất)

1.2 Bluetooth Low Energy (BLE)

- Chưa có tài liệu nghiên cứu về lập trình trên hệ thống System on Chip, chỉ có hướng dẫn sử dụng module theo firmware sẵn có từ nhà sản xuất. Cần tìm hiểu và nghiên cứu, so sánh về tính ứng dụng giữa 2 hướng phát triển trên.

1.1.4 Mục tiêu - Phạm vi - Đối tượng nghiên cứu

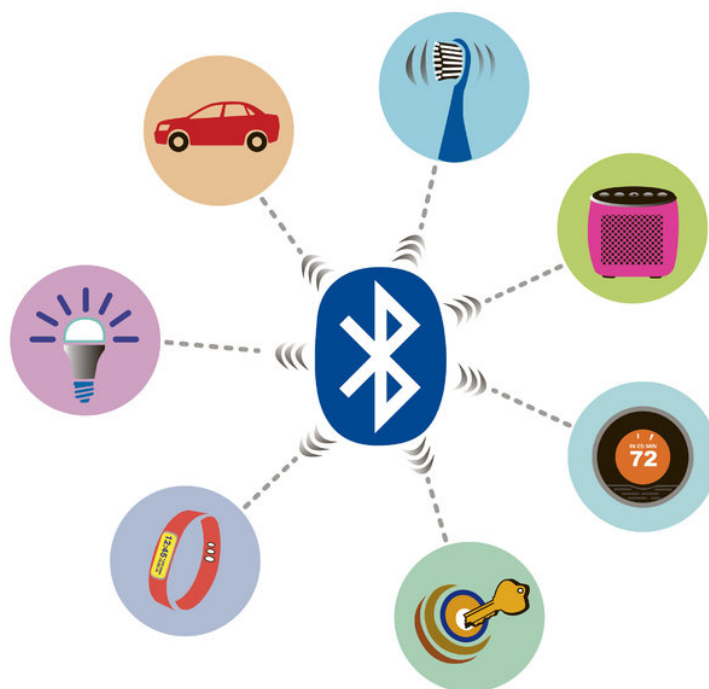
Xuất phát từ các lý do trình bày ở trên, chúng tôi đã thực hiện đề tài “Thiết kế sản phẩm móc khóa thông minh (Smart Keyring) dựa trên nền tảng công nghệ Bluetooth Low Energy (BLE)”. Mục tiêu của đề tài là:

- Tìm hiểu về công nghệ Bluetooth Low Energy
- Kế thiết bị “Móc khóa thông minh - SmartKeyring” sử dụng công nghệ

Bluetooth và kết nối với ứng dụng Android trên điện thoại.

1.2 Bluetooth Low Energy (BLE)

Khái niệm BLE xuất hiện từ phiên bản 4.0, là một bước ngoặt lớn trong sự phát triển kết nối không dây. Các mạch BLE rất nhỏ cùng với công suất tiêu thụ hiệu năng cực thấp (khoảng vài chục uA khi hoạt động), nên hầu hết các thiết bị đều có thể tích hợp công nghệ này, từ các thiết bị nhỏ bé như tai nghe, chìa khóa.. cho tới các thiết bị lớn như tủ lạnh, tivi, xe máy... Nhờ đó, các thiết bị có thể trở nên "smart".[3]



Hình 1.1 Các ứng dụng Bluetooth

1.2.1 Khái niệm và lịch sử phát triển Bluetooth

Bluetooth là công nghệ không dây cho phép các thiết bị điện, điện tử giao tiếp với nhau trong khoảng cách ngắn, bằng sóng vô tuyến qua băng tần chung ISM (Industrial, Scientific, Medical) trong dãy tần 2.40- 2.48 GHz. Đây là dãy băng tần không cần đăng ký được dành riêng để dùng cho các thiết bị không dây trong công nghiệp, khoa học, y tế.

Lịch sử phát triển: [5]

- Bluetooth 4.0 - Bluetooth Low Energy: Là sự kết hợp của các đời Bluetooth trước đó với nhau. Bluetooth 4.0 đạt tốc độ truyền tải lên đến 25Mbps, dễ dàng ghép đôi các thiết bị với nhau, hiệu năng tiêu thụ thấp. Đây là chuẩn Bluetooth được sử dụng trên hầu hết các thiết bị hiện nay.

1.2 Bluetooth Low Energy (BLE)

	Bluetooth	BLE	Wifi	Zigbee
Radio Frequency	2.4G	2.4G	2.4G	2.4G
Distance Range	10m	>60m	30m	10-100m
Air Datarate	1-3Mbps	1Mbps	54Mbps	250kbps
Application Throughput	0.7-2.1Mbps	305kbps	Depend	120kbps
Security	64bit, 128bit	128-bit	AES SSID, WEP	128-bit AES
Power consumption	Low	Very Low	High	Low
Certification Body	Bluetooth SIG	Bluetooth SIG	IEEE802.11	IEEE802.15.4
Network topology	Point-to-Point Scatternet	Point-to-Point Star	Point-to-Hub	Mesh, Ad-hoc

Bảng 1.1 Bảng so sánh các công nghệ truyền không dây

- Bluetooth 4.1 và 4.2: Là phiên bản ra đời đầu năm 2014 với nhiều cải tiến vượt bậc so với Bluetooth 4.0 như khả năng điều chống chồng chéo tín hiệu, kết nối thực sự thông minh và khả năng truyền dữ liệu độc lập mà không cần phụ thuộc vào trung tâm điều khiển. Phiên bản 4.2 được phát triển có khả năng truyền tải cao và bảo mật hơn, nhưng quan trọng hơn cả là cho phép các vi xử lý sử dụng chuẩn giao thức Ipv6 để truy cập trực tiếp vào internet.

- Bluetooth 5.0: theo dự kiến sẽ bắt đầu xuất hiện trên các thiết bị thương mại vào cuối 2016 nay hoặc đầu năm 2017 (Q1). Bluetooth 5.0 có tầm phủ sóng tăng lên gấp 4 lần so với Bluetooth 4.2 hiện nay, còn tốc độ truyền dữ liệu thì tăng lên cao nhất là 2 lần. Việc mở rộng khả năng phủ sóng của Bluetooth sẽ giúp các thiết bị Internet of Things sẽ có thể giao tiếp với nhau cũng như với trạm điều khiển một cách dễ dàng hơn, vượt qua bức tường của một căn nhà bình thường, trong khi lại tăng tốc thu thập và truyền dữ liệu. Chuẩn Bluetooth mới cũng sẽ giúp các beacon và giải pháp nhận diện địa điểm trở nên thông minh, chính xác và phản hồi nhanh hơn với sự hiện diện của người dùng.

1.2.2 Phân loại vai trò thiết bị BLE

Có 4 loại thiết bị BLE (có thể gọi là chế độ hoạt động) đó là Peripheral, Central, Observer và Broadcaster và bình thường thì một thiết bị BLE chỉ hoạt động trong một chế độ.

- **Central** là thiết bị sẽ chủ động yêu cầu kết nối đến các thiết bị BLE khác (thường là smartphone, tablet). Sau khi kết nối thì chúng ta lại gọi BLE Central là BLE Master.

- **BLE Peripheral** là thiết bị chấp nhận yêu cầu kết nối (thường là đồ vật BLE). Tương tự, sau khi kết nối thì chúng ta gọi BLE Peripheral là BLE Slave.

- **BLE Observer** là BLE Central nhưng chỉ nhận dữ liệu nhận dạng của các thiết bị xung quanh nhưng không bao giờ tạo kết nối

- **BLE Broadcaster** là BLE Peripheral chỉ phát dữ liệu nhận dạng nhưng không bao giờ chấp nhận yêu cầu kết nối từ các BLE Central.

1.2.3 Cách thức hoạt động của BLE

Theo chuẩn BLE định nghĩa thì các thiết bị BLE có 4 hoạt động cơ bản như sau: [3]

- **Advertising**: là hoạt động phát dữ liệu nhận dạng cơ bản của thiết bị BLE Peripheral ra môi trường xung quanh trước khi kết nối

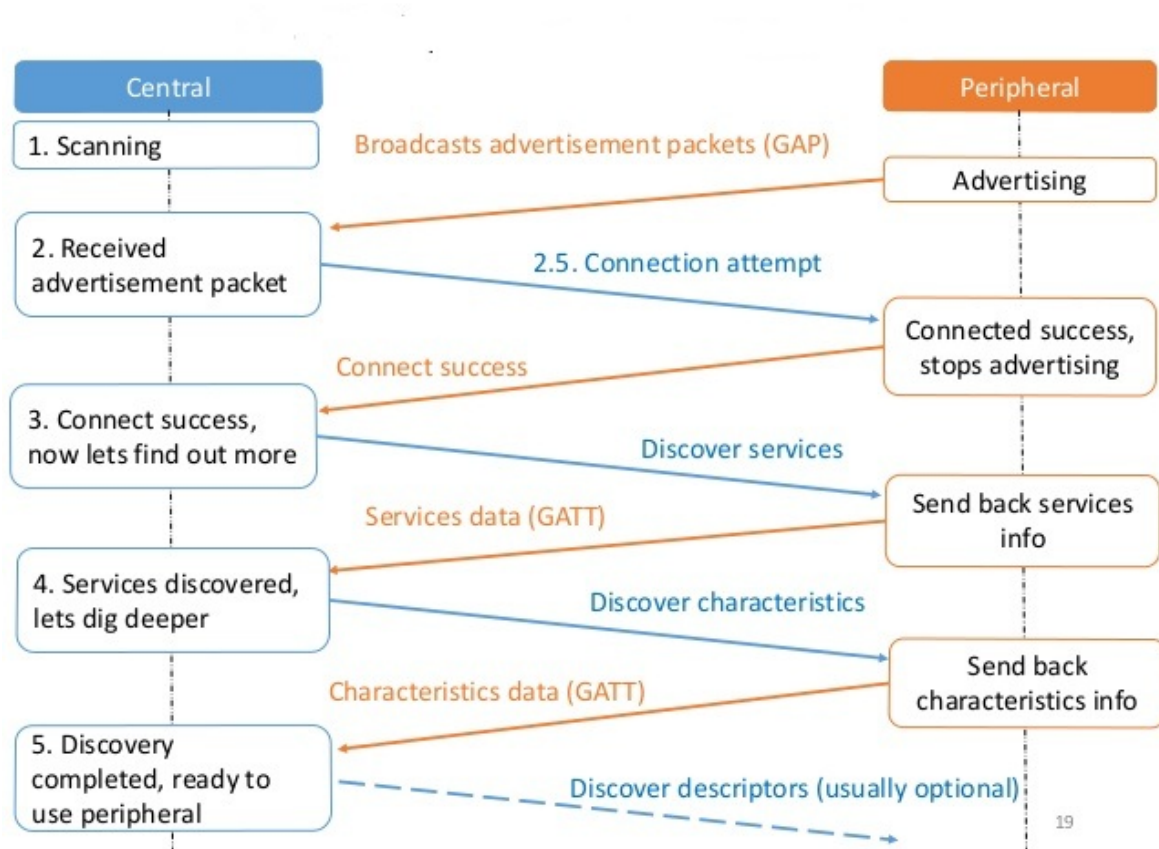
- **Scanning**: là hoạt động của thiết bị BLE Central để thu thập dữ liệu nhận dạng của nhiều thiết bị BLE Peripheral xung quanh

- **Connecting**: là hoạt động của cả thiết bị BLE Central và BLE peripheral trong đó thiết bị BLE Central có thể gửi yêu cầu thêm thông tin nhận dạng (gọi là Scan Request) và BLE Peripheral gửi theo yêu cầu (gọi là Scan Response). Sau đó BLE Central sẽ kiểm tra đầy đủ thông tin nhận dạng (từ Advertising data và từ Scan Response data) và gửi yêu cầu kết nối (gọi là Connection Request), cuối cùng thiết bị BLE Peripheral sẽ trả lời chấp nhận hay từ chối kết nối (gọi là Connection Response)

1.2 Bluetooth Low Energy (BLE)

- **Discovering:** là hoạt động của thiết bị BLE Client sau khi kết nối nhằm lấy thông tin về các loại dữ liệu mà thiết bị BLE Server có thể cung cấp. Ví dụ, thiết bị BLE Server có thể có dữ liệu về gia tốc, hoặc có dữ liệu về nhiệt độ, độ ẩm, v.v.. và thiết bị BLE Client sẽ có nhu cầu biết các loại dữ liệu nào có thể nhận từ BLE Server

Cách thức hoạt động của BLE ở hình 1.2:



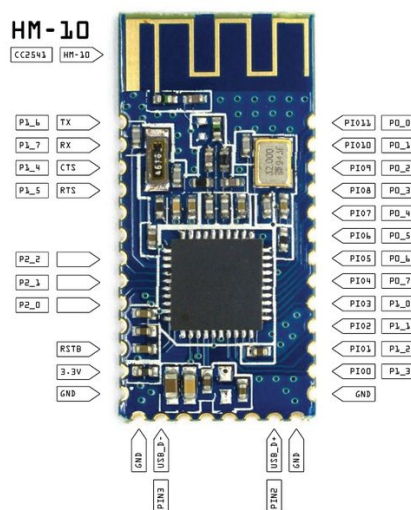
Hình 1.2 Sơ đồ hoạt động của BLE

1.2.4 Các vi điều khiển tích hợp công nghệ BLE

Các SoC tích hợp sẵn công nghệ BLE phổ biến có thể kể đến:

- Texas Instruments: CC2540/CC2541, dòng CC256x, CC26xx
- Nordic Semiconductor: nRF51822, nRF8001
- CSR CSR101x
- Cypress Semiconductor PSoC 4 BLE / PSoC BLE

Tuy nhiên ở Việt Nam tại thời điểm bắt đầu nghiên cứu và hiện thực đề tài thì 2 loại chipset phổ biến và có giá tiền phổ thông là CC2540 và CC2541 được cung cấp theo dạng Module HM-10.



Hình 1.3 Module HM-10

Chương 2

Thiết kế và hiện thực sản phẩm móc khóa thông minh - Smart Keyring

2.1 Thiết kế tính năng sản phẩm

Thiết kế tính năng sản phẩm:

Sản phẩm Smart Keyring sẽ có các tính năng cơ bản như:

- Báo hiệu khi mất kết nối: hỗ trợ việc cảnh báo người tránh bỏ quên 1 trong 2 thiết bị.
- Báo hiệu khi kích hoạt chức năng tìm kiếm: cho phép người dùng định vị thiết bị còn lại trong phạm vi kết nối.
- Hai chế độ báo hiệu bằng âm thanh hoặc ánh sáng đèn led hoặc cả 2: mục đích sử dụng trong nhiều trường hợp khác nhau như đêm tối, không gian yên tĩnh...

2.2 Các hướng tiếp cận vấn đề

Tại thời điểm tìm hiểu và hiện thực đề tài NCKH, tại Việt Nam chỉ có module HM-10 có sử dụng chip BLE CC2540/2541 được phát triển thành board mạch hoàn chỉnh nên phần này chỉ nói về hướng phát triển với board mạch này.

2.2.1 Phát triển thiết bị chỉ dựa vào SoC CC2540/2541

Về hướng này chúng ta sẽ phát triển lập trình thiết bị chỉ trên duy nhất 1 SoC CC2540/2541.

Ưu điểm:

- Có thể thu nhỏ thiết kế đến mức tối thiểu
- Viết ứng dụng ngay trên nền tảng BLE sẽ tiết kiệm tối đa năng lượng tiêu thụ.

Nhược điểm:

- Bị hạn chế về khả năng phát triển cả về phần cứng lẫn phần mềm.
- Không tìm được source code firmware cho module.
- Không có tài liệu chính thống nào hướng dẫn các bước lập trình cho vi điều

khiển CC2540/2541 được tích hợp trên module HM-10

- Nhà sản xuất không công khai thiết kế mạch của sản phẩm HM-10
- Chỉ có duy nhất 1 phần mềm hỗ trợ các gói thư viện lập trình cho CC2540/2541

là IAR Embedded Workbench for 8051 [4] hỗ trợ Texas Instrument nhưng bản quyền cho phần mềm có giá quá cao và các thư viện này sử dụng mã nguồn đóng nên không chuyển sang các phần mềm khác được.

Vì những cản trở đó, nhóm quyết định chuyển sang phương pháp tiếp cận khác đơn giản và khả thi hơn.

2.2.2 Kết hợp MCU và Module BLE HM-10

Cách tiếp cận này khá quen thuộc với đa số hệ thống và sản phẩm hiện nay bao gồm 1 vi điều khiển trung tâm: chứa toàn bộ chương trình hoạt động của sản phẩm và các thiết bị ngoại vi (sensor, các module giao tiếp rf, Bluetooth...)

Ưu điểm:

- Dễ tiếp cận, do người hiện thực có thể kiểm soát được công nghệ mình sử dụng
- Tùy biến các loại vi điều khiển sao cho thích hợp nhất đối với yêu cầu đề tài.

Các vi điều khiển riêng rẽ hiện nay rất đa dạng chủng loại và chức năng, đi kèm theo nó là hệ thống hỗ trợ cực kì tốt từ nhà sản xuất về tài liệu, môi trường lập trình, các forum trao đổi. điển hình là các thương hiệu Atmel, Microchip...

Nhược điểm:

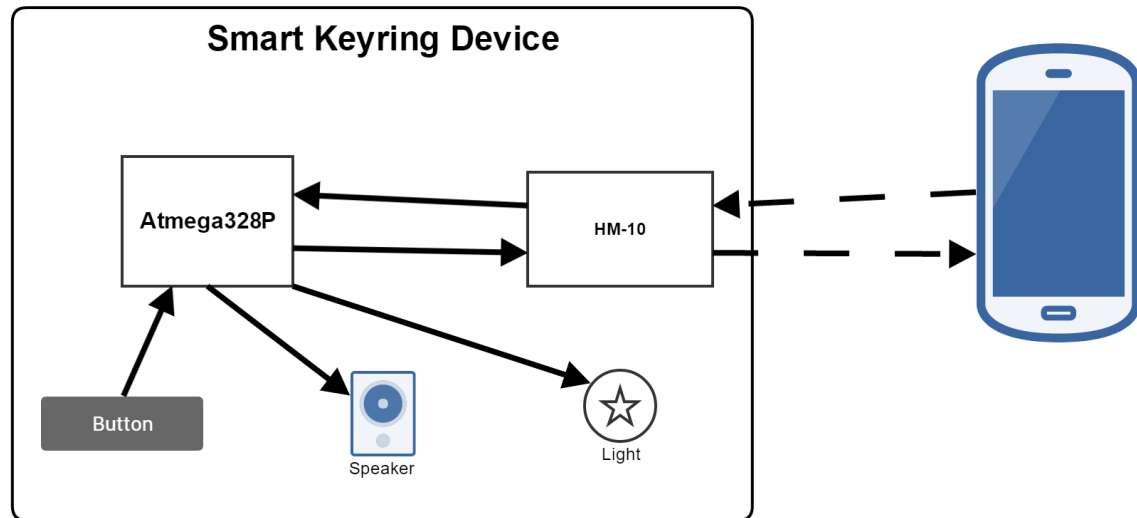
- Đối nghịch lại với ưu điểm của cách tiếp cận đầu tiên thì hướng tiếp cận này sẽ cần nhiều không gian hơn (thêm 1 vi điều khiển).
- Làm cho hệ thống mất đi tính linh động và gọn nhẹ so với tính chất sản phẩm cũng như là năng lượng tiêu thụ không được tối ưu.
- Thêm 1 vi điều khiển tương đương với việc thêm 1 nguồn tiêu thụ điện, giảm thời gian hoạt động của sản phẩm.

2.3 Sơ đồ hoạt động

2.3.1 Tổng quát

Sơ đồ hoạt động tổng quát:

Như hình 2.1, thiết bị Smart Keyring giao tiếp với thiết bị di động thông qua module BLE HM-10 và được điều khiển bởi MCU ATmega328P đảm nhận chức năng quản lý I/O như nút ấn, loa báo hiệu và đèn cũng như là truyền nhận thông điệp với module HM10.



Hình 2.1 Sơ đồ hoạt động tổng quát

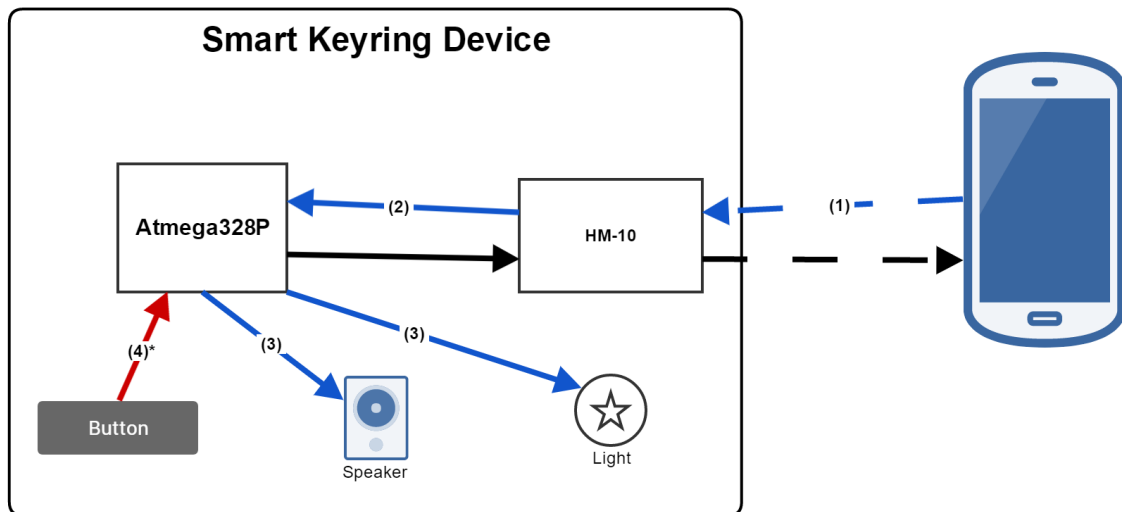
2.3.2 Nhận lệnh báo từ thiết bị di động

Chức năng tìm kiếm thiết bị được kích hoạt bởi thiết bị di động được mô tả ở hình 2.2.

Trình tự các hoạt động như sau:

- (1) Thiết bị di động gửi gói tin với nội dung yêu cầu phát tín hiệu báo tới module HM-10.
- (2) MCU ATmega328P nhận gói tin từ module HM-10 bằng giao thức UART với chế độ interrupt.
- (3) Loa và đèn báo hiệu được kích hoạt tùy theo nội dung gói tin: kích hoạt cả hai hoặc chỉ kích hoạt đèn báo hiệu
- (4*) Nút nhấn có chức năng ngắt chế độ báo hiệu khi cần thiết thông qua interrupt GPIO

Sơ đồ hoạt động trên đúng với chức năng ngắt báo hiệu thiết bị được điều khiển bởi thiết bị di động, chỉ khác tại bước (3) là ngắt loa và đèn và không có bước (4).



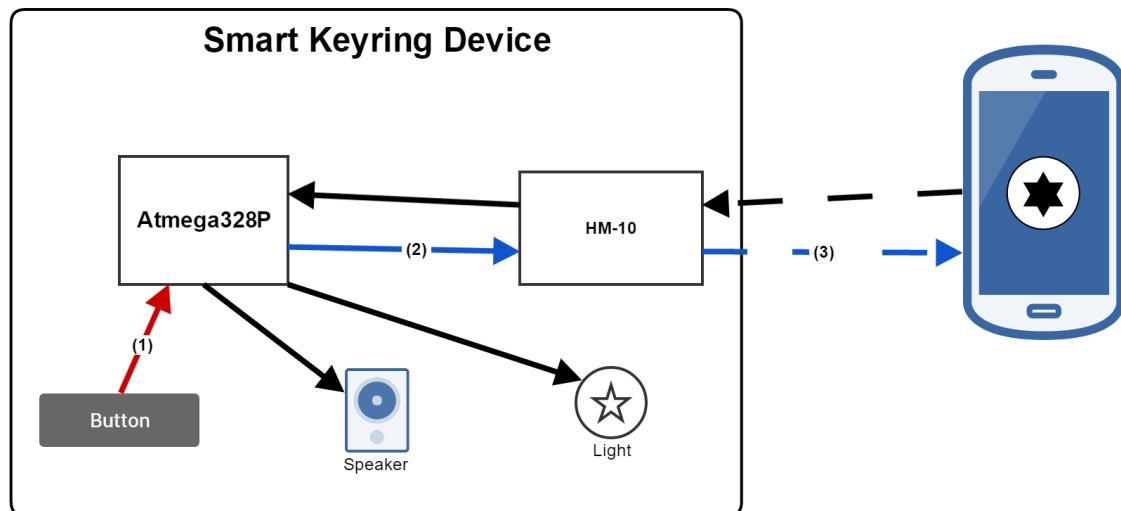
Hình 2.2 Sơ đồ hoạt động khi nhận lệnh báo từ thiết bị di động

2.3.3 Kích hoạt thiết bị di động bật chế độ báo hiệu

Chức năng kích hoạt thiết bị di động bật chế độ báo hiệu được mô tả ở hình 2.3.

Trình tự các hoạt động như sau:

- (1) MCU ATmega328P nhận tín hiệu điều khiển từ nút ấn thông qua interrupt GPIO
- (2) Module BLE HM-10 nhận gói tin điều khiển từ MCU ATmega328 thông qua UART
- (3) Thiết bị di động nhận gói tin truyền từ Module HM-10 và kích hoạt chế độ báo hiệu

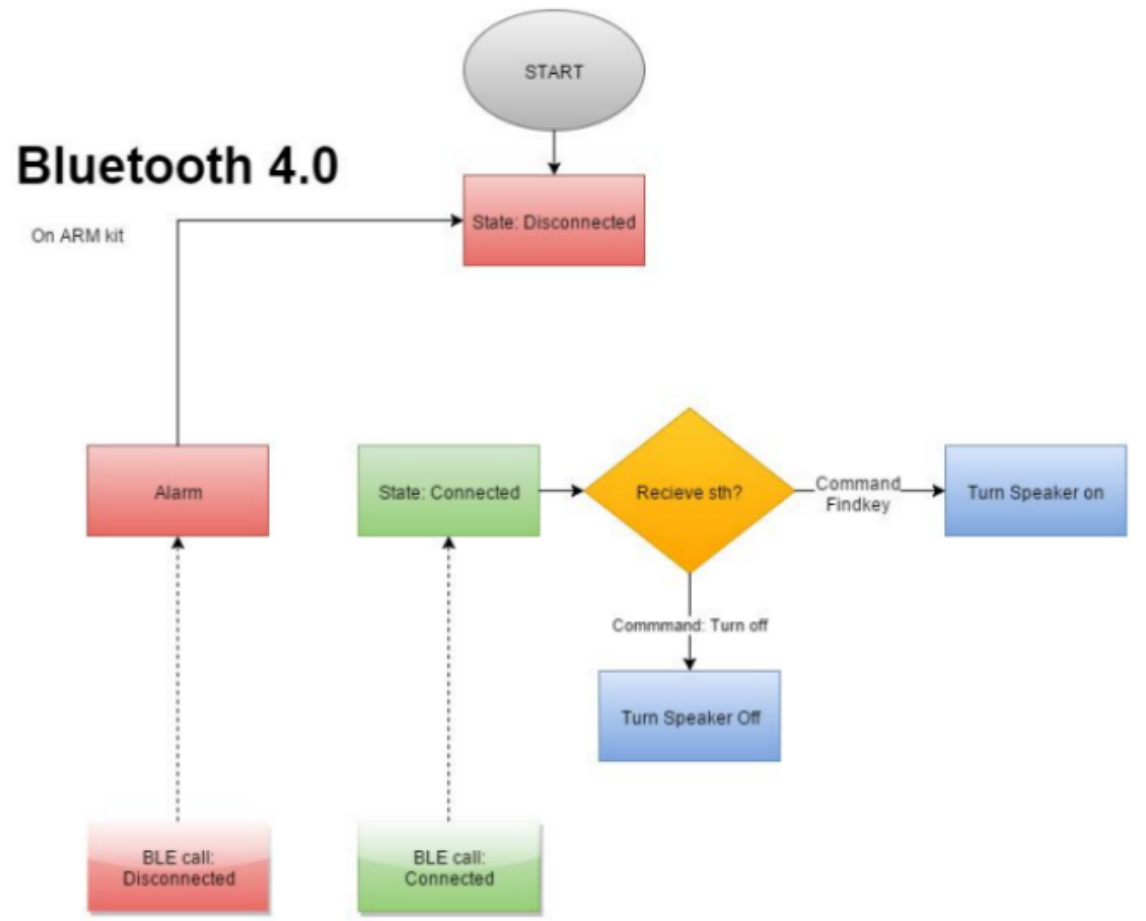


Hình 2.3 Sơ đồ kích hoạt thiết bị di động bật chế độ báo hiệu

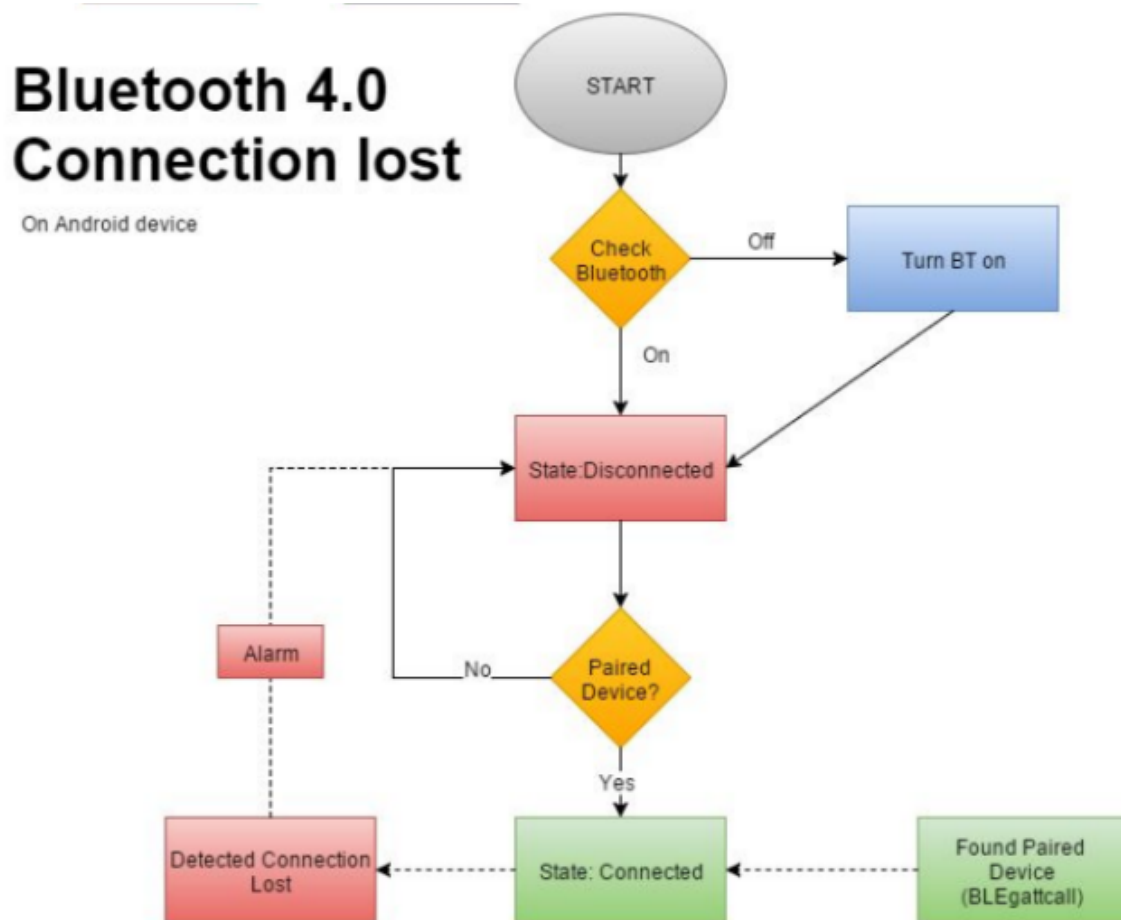
2.3.4 Sơ đồ trạng thái hoạt động khi ngắt kết nối

Dựa theo tính năng sản phẩm ở mục 2.1, sơ đồ trạng thái hoạt động được thiết kế ở hình 2.4 và 2.5

Bluetooth 4.0 (cùng với bộ thư viện và công nghệ BLE)



Hình 2.4 Sơ đồ trạng thái trên thiết bị Smart Keyring



Hình 2.5 Sơ đồ hoạt động trên thiết bị di động

2.4 Hiện thực phần cứng

2.4.1 Lựa chọn linh kiện và thiết bị phần cứng

Sản phẩm hiện thực gồm 3 phần:

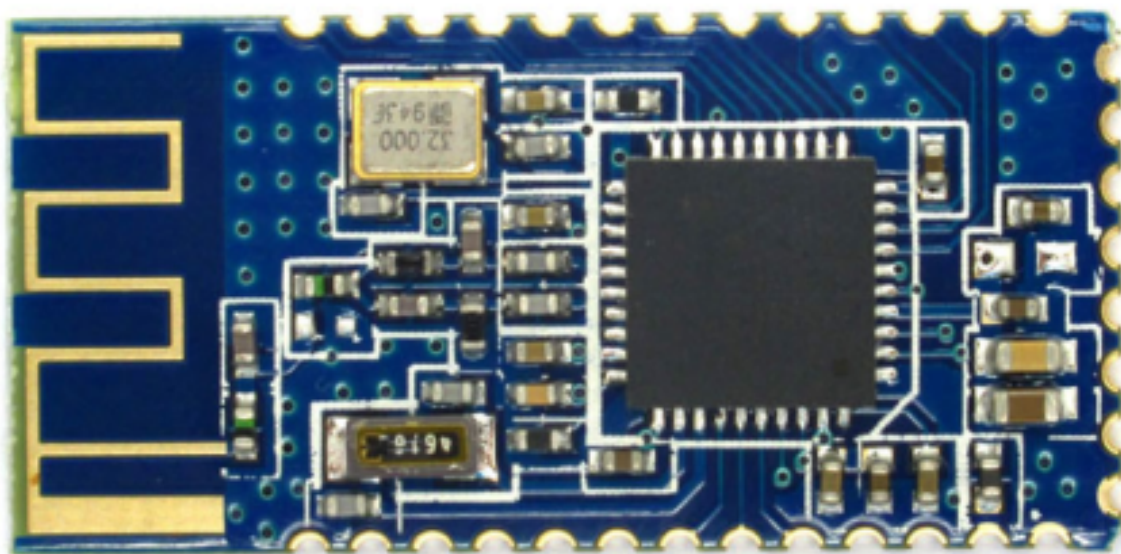
Module giao tiếp không dây:

Tất nhiên yêu cầu tiên quyết của sản phẩm là việc giao tiếp không dây giữa các thiết bị, tiếp theo là việc làm sao để tiết kiệm năng lượng trên từng phần của sản

2.4 Hiện thực phần cứng

phẩm. Thông qua tìm hiểu, nhóm đã phát hiện ra công nghệ truyền dữ liệu không dây trong khoảng cách ngắn phổ biến nhất hiện nay là công nghệ Bluetooth Low Energy.

Hiện nay, trên thị trường có rất nhiều loại mẫu mã và thiết kế cho module Bluetooth Low Energy từ nhiều hãng sản xuất khác nhau. Thời gian đầu lúc tiếp cận, nhóm đã lựa chọn mẫu phổ biến nhất trên thị trường hiện nay với giá thành tương đối hợp lí. Đó là module Bluetooth BLE HM-10 của hãng JN-Huamao Technology Company có xuất xứ từ Trung Quốc. Sau thời gian thử nghiệm và nghiên cứu với nhiều phiên bản của sản phẩm, nhóm nhận thấy module hoạt động tốt và ổn định, thông số kỹ thuật mà nhà sản xuất cung cấp gần đúng với các thông số kỹ thuật mà nhóm rút ra được trong quá trình thử nghiệm.



Hình 2.6 Module BLE HM-10

Thông số kỹ thuật của module: (do nhà sx cung cấp)

- SoC: CC2540/2541 (tùy theo đợt sản xuất)
- BT version: Bluetooth Specification V4.0 BLE.
- Tần số hoạt động: 2.4GHz ISM band.
- RF power: -23dbm, -6dbm, 0dbm, 6dbm.

2.4 Hiện thực phần cứng

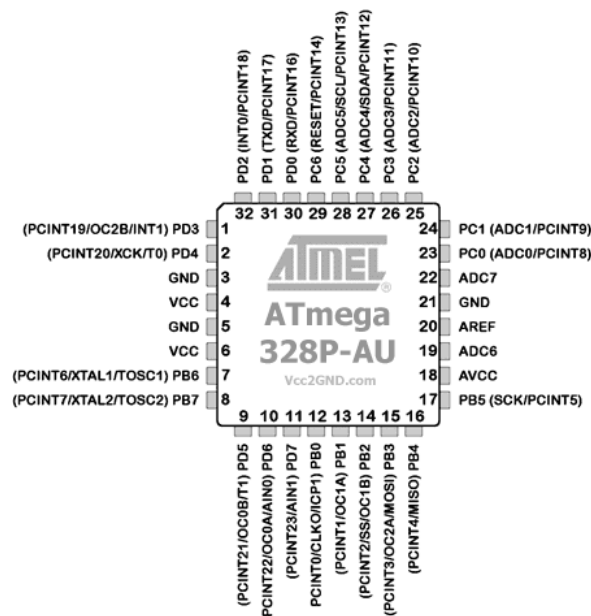
- Baudrate: hỗ trợ đến 230400, mặc định là 9600.
- Nguồn: +2.5 - 3.3VDC 50mA.
- Dòng tiêu thụ: Active mode 8.5mA, Sleep mod 50-200 uA.
- Thực nghiệm cho thấy module có thể thực hiện truyền nhận trong khoảng cách 20m – 30m(tùy vào thiết bị di động).

Vi điều khiển trung tâm:

Vi điều khiển trung tâm cũng cần thỏa mãn một số yêu cầu sau:

- Điện áp hoạt động: 3.3V hoặc ít hơn.
- Có ít nhất 1 kênh giao tiếp UART.
- Tiết kiệm năng lượng nhiều nhất có thể.
- Công cụ lập trình dễ tiếp cận và có bộ thư viện hỗ trợ tương đối đầy đủ.

Với những yêu cầu trên, qua quá trình tìm hiểu và đúc kết kinh nghiệm từ trước, nhóm đã chọn được vi điều khiển thích hợp cho mô hình. Đó là vi điều khiển ATmega328P của hãng Atmel.



Hình 2.7 MCU ATmega328P

Các thông số kỹ thuật:

Core: AVR 8-bit

Kích thước bộ nhớ FLASH: 32 KB Flash

Tần số hoạt động tối đa: 20 MHz

Các chuẩn giao tiếp: I²C, SPI, UART/USART

Số lượng cổng GPIO: 23

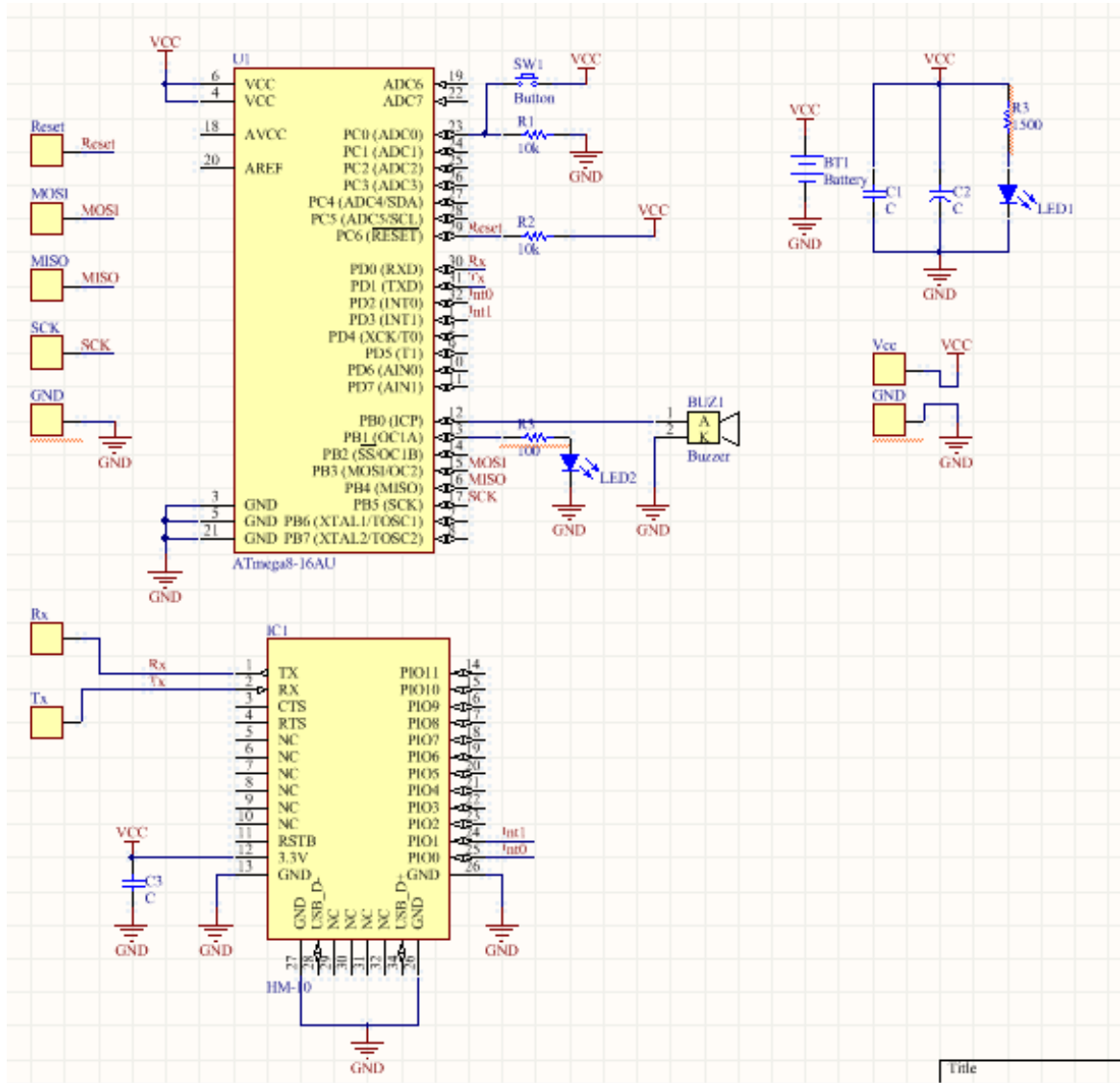
Điện áp hoạt động: 1.8V to 5.5V

Các thiết bị xuất nhập:

- LED nguồn: chức năng báo hiệu khi thiết bị đang hoạt động.
- LED báo hiệu: chức năng bật tắt bằng ánh sáng LED khi có yêu cầu từ thiết bị di động.
- Buzzer: chức năng phát âm thanh khi có yêu cầu từ thiết bị di động.
- Nút bấm: chức năng kích hoạt thiết bị Smart Keyring gửi lệnh cho thiết bị di động báo chuông.

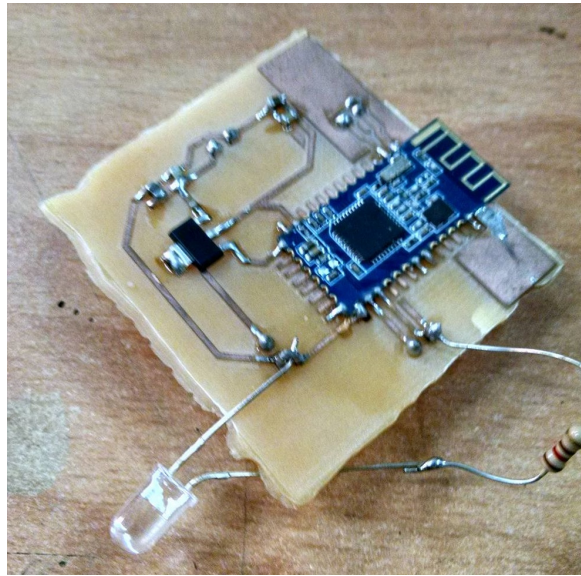
Theo như yêu cầu chức năng ở mục 2.1, schematic được thiết kế như hình 2.8

2.4 Hiện thực phần cứng



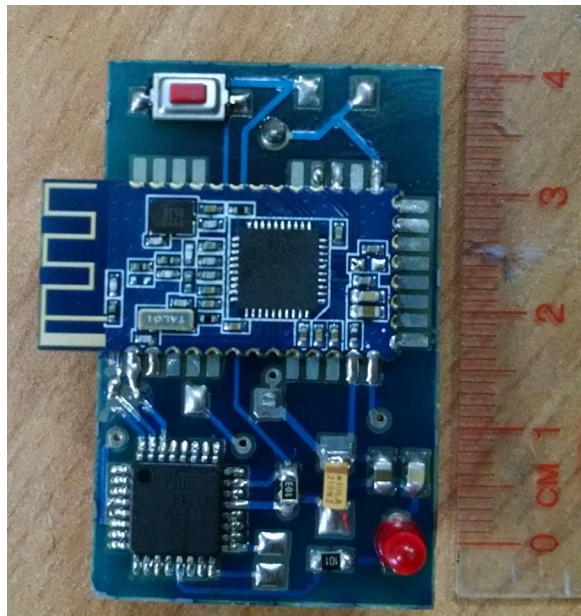
Hình 2.8 Schematic của thiết bị Smart Keyring

Mạch thiết bị thử nghiệm theo schematic như hình 2.9

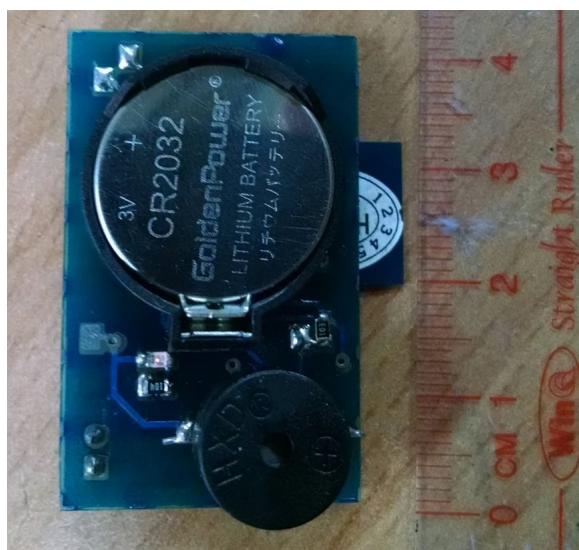


Hình 2.9 Mạch thử nghiệm

Kết quả thử nghiệm giao tiếp và các thiết bị IO hoạt động tốt nên tiến hành đặt mạch và làm mạch chính thức ở hình 2.10 và 2.11



Hình 2.10 Mặt trước mạch thiết bị



Hình 2.11 Mặt sau mạch thiết bị

2.4.2 Lập trình firmware cho thiết bị

Tận dụng tính đơn giản của sản phẩm ứng dụng, các chức năng của sản phẩm được lập trình sao cho việc xử lý các input và output được thực hiện hoàn toàn bằng interrupt. Như vậy, ta sẽ không cần hiện thực thêm chương trình chính và có thể đưa vi điều khiển vào chế độ sleep mode, cụ thể hơn là Idle mode. Từ đó giúp hệ thống tiết kiệm năng lượng hơn.

Sleep Mode	Active Clock Domains					Oscillators		Wake-up Sources						
	clkCPU	clkFLASH	clkIO	clkADC	clkASY	Main Clock Source Enabled	Timer Oscillator Enabled	INT and PCINT	TWI Address Match	Timer2	SPM/EEPROM Ready	ADC	WDT	Other I/O
Idle			Yes	Yes	Yes	Yes	Yes ⁽²⁾	Yes	Yes	Yes	Yes	Yes	Yes	Yes
ADC Noise Reduction				Yes	Yes	Yes	Yes ⁽²⁾	Yes ⁽³⁾	Yes	Yes ⁽²⁾	Yes	Yes	Yes	
Power-down								Yes ⁽³⁾	Yes				Yes	
Power-save					Yes		Yes ⁽²⁾	Yes ⁽³⁾	Yes	Yes			Yes	
Standby ⁽¹⁾						Yes		Yes ⁽³⁾	Yes				Yes	
Extended Standby					Yes ⁽²⁾	Yes	Yes ⁽²⁾	Yes ⁽³⁾	Yes	Yes			Yes	

Hình 2.12 Các chế độ sleep mode khác nhau và khả năng hoạt động của từng chế độ

Khi MCU đi vào Idle mode, CPU sẽ ngừng hoạt động nhưng các bộ USART, SPI, Timer/Counters, hệ thống Interrupt vẫn sẽ tiếp tục hoạt động. Idle mode cho phép

MCU ‘wake up’ khi xảy ra ngắt ngoài cũng như ngắt trong từ USART Receive, Timer Overflow...

Hiện thực interrupt

Như được đề cập ở trên, việc hiện thực chương trình cho ứng dụng cũng là hiện thực việc xử lý ở các interrupt, bao gồm USART Receive Interrupt và External Interrupt.

USART Receive Interrupt

Xảy ra khi nhận được dữ liệu bất kì từ UART: Dữ liệu này vi điều khiển nhận được thông qua module Bluetooth gửi từ thiết bị di động. Nếu dữ liệu nhận được là ‘R’, chương trình sẽ cho active chân kết nối với chuông. Nếu dữ liệu nhận được là ‘L’, chương trình cho active chân kết nối đến đèn. Ngoài ra, chương trình sẽ không có phản hồi nào khác. Chức năng trên được biểu diễn dưới dạng code:

```
ISR (USART_RX_vect)
{
    Uart_rev = getchar_uart(); // Nhận dữ liệu từ USART
    if(Uart_rev == 'L'){       // Nếu dữ liệu nhận được là L
        PORTB ^= 1<<PORTB1;  // Active Port nối với đèn
    }else if(Uart_rev == 'R'){ // Nếu dữ liệu nhận được là R
        PORTB ^= 1<<PORTB0;  // Active Port nối đến Buzzer
    }
}
```

External Interrupt

Xảy ra khi nhận input từ nút nhấn hoặc tín hiệu disconnect từ module Bluetooth): Mỗi khi có interrupt từ nút nhấn, chương trình sẽ cho truyền kí tự ‘R’ đến module Bluetooth, từ đó truyền đến thiết bị di động để phát lệnh đổ chuông. Khi có interrupt từ module Bluetooth, việc này xảy ra chỉ khi có 1 thiết bị đang kết nối với module Bluetooth đột nhiên mất kết nối. Khi đó, chương trình sẽ điều khiển chân kết nối với chuông để phát tín hiệu mất kết nối được thể hiện bằng đoạn code:

2.4 Hiện thực phần cứng

```
ISR(INT1_vect) // chân INT1 kết nối đến chân trạng thái của module bluetooth
{
    PORTB |= 1<<PORTB0;           //
    _delay_ms(200);                //
    PORTB &= ~(1<<PORTB0);         //
    _delay_ms(50);                 //
    PORTB |= 1<<PORTB0;           //
    _delay_ms(200);                //
    PORTB &= ~(1<<PORTB0);         //Tạo 2 tiếp bip để báo hiệu mất kết nối
}
```

Tùy chỉnh baudrate

Do tính chất hoạt động của sản phẩm đơn giản, hàm lượng dữ liệu truyền nhận giữa các thiết bị ít và không cần đòi hỏi tốc độ quá nhanh, việc hạ thấp baudrate sẽ giúp sản phẩm tiết kiệm năng lượng hơn. Thực tế, mỗi lần cho phát 1 lệnh bất kì giữa 2 thiết bị (smart keyring và tb di động) thì một bên chỉ truyền 1 kí tự ('L', 'R') để cho biết bên nhận phải phát sáng đèn hay đổ chuông. Vì lí do đó, nhóm quyết định hạ thấp baudrate đến mức nhỏ nhất mà module Bluetooth cho phép: 2400Hz Việc điều chỉnh baudrate cho module Bluetooth khá đơn giản, chỉ cần gửi lệnh theo đúng cú pháp AT+BAUD? Giá trị thay vào ? sẽ tương ứng với tần số mình cần điều chỉnh

Send	Receive	Parameter
AT+BAUD?	OK+Get:[P1]	P1: Baud rate No.
AT+BAUD[P1]	OK+Set:[P1]	0-----9600 1-----19200 2-----38400 3-----57600 4-----115200 5-----4800 6-----2400 7-----1200 8-----230400 Default: 0(9600)

Như vậy, ta chỉ cần gửi lệnh **AT+BAUD6**. Nếu module Bluetooth phản hồi lại bằng lệnh OK+SET:6 thì ta đã thiết lập thành công **Baudrate 2400** cho thiết bị

2.5 Hiện thực ứng dụng di động trên Android

Điều chỉnh tần số xung clock hoạt động của vi điều khiển. Tốc độ xử lý càng nhanh thì đi đôi với việc tiêu tốn năng lượng càng nhiều. Để hạn chế điều này, nhóm tìm cách hạ thấp tần số xung clock hoạt động của vi điều khiển đến mức thấp nhất có thể (128KHz). Và với công việc chính của vi điều khiển là giao tiếp UART ở tần số 2400, thì tần số xung clock này hoàn toàn có thể đáp ứng được theo công thức sau:

$$UBRRn = FOSC/(16BAUD) - 1$$

UBRRn: giá trị cần tính để gán vào thanh ghi

FOSC: tần số xung clock hoạt động của vi điều khiển

BAUD: Baud rate hoạt động khi truyền nhận USART

Thay các giá trị cần tính vào ta được:

$$UBRRn = 128000/(16*2400)-1 = 2.33 > 0$$

Do đó việc sử dụng tần số xung clock thấp nhất (128KHz) là hoàn toàn hợp lý, đảm bảo được việc truyền nhận UART trong khi tiết kiệm năng lượng đáng kể so với việc sử dụng tần số cao hơn.

2.5 Hiện thực ứng dụng di động trên Android

2.5.1 Các khái niệm trong lập trình BLE trên Android

Dưới đây là những khái niệm chính về BLE được sử dụng: [2]

Generic Attribute Profile (GATT) - Cấu hình thuộc tính chung — Cấu hình GATT là đặc điểm kỹ thuật chung cho việc truyền nhận các gói dữ liệu nhỏ được biết đến như là các "đặc tính" trên đường truyền BLE. Tất cả các cấu hình ứng dụng BLE đều dựa trên GATT.

Attribute Protocol (ATT) - Thuộc tính của giao thức—GATT được xây dựng bên trên lớp ATT, thường được nhắc đến là GATT/ATT. ATT được tối ưu hóa trên các thiết bị BLE và sử dụng ít dữ liệu nhất có thể. Các thuộc tính này được định

2.5 Hiện thực ứng dụng di động trên Android

danh duy nhất bởi Universally Unique Identifier (UUID), là 1 chuỗi định danh dưới chuẩn 128-bit để định danh thông tin duy nhất. Các thuộc tính được truyền bởi ATT được định dạng dưới các characteristics và services.

Characteristic - Đặc tính— Bao gồm một giá trị và 0-n các khóa để miêu tả giá trị của đặc tính. Một đặc tính có thể xem như là 1 kiểu tương tự như lớp.

Descriptor - Khóa mô tả— Khóa mô tả được xác định để miêu tả giá trị đặc tính. Thí dụ, một khóa mô tả có thể đọc được bởi con người, thuộc phạm vi giá trị của đặc tính hoặc một giá trị đo cụ thể của một giá trị đặc tính nào đó.

Service - Dịch vụ— Service là một gói tổng hợp các đặc tính. Ví dụ, ta có thể có dịch vụ gọi là "Theo dõi nhịp tim" vừa bao gồm các đặc tính như "đo nhịp tim". Danh sách các cấu hình GATT và dịch vụ có thể tìm thấy tại bluetooth.org.

2.5.2 Phát triển ứng dụng Android

Ứng dụng Android được phát triển từ nguồn ứng dụng mở android-BluetoothLeGatt[1] được cung cấp bởi Google. Ứng dụng này có chức năng demo thiết lập kết nối và nhận dữ liệu theo chuẩn BLE.

Nhóm chúng tôi trong quá trình tìm hiểu đã nhận thấy phù hợp với đề tài và có thể phát triển chức năng cảnh báo khi mất kết nối và báo hiệu để thiết bị Smart Keyring kích hoạt buzzer/led.

Ứng dụng lập trình dựa theo 4 file chính:

DeviceScanActivity.java: chức năng tìm kiếm và hiện thị thiết bị BLE

DeviceControlActivity.java: chức năng quản lý kết nối và hiện thị các thông tin về trạng thái kết nối và các service đang có.

BluetoothLeService.java: quản lý và điều khiển các service

Cơ chế hoạt động của lệnh gửi ký tự qua BLE trên Android:

```
// BluetoothLeService.java
```

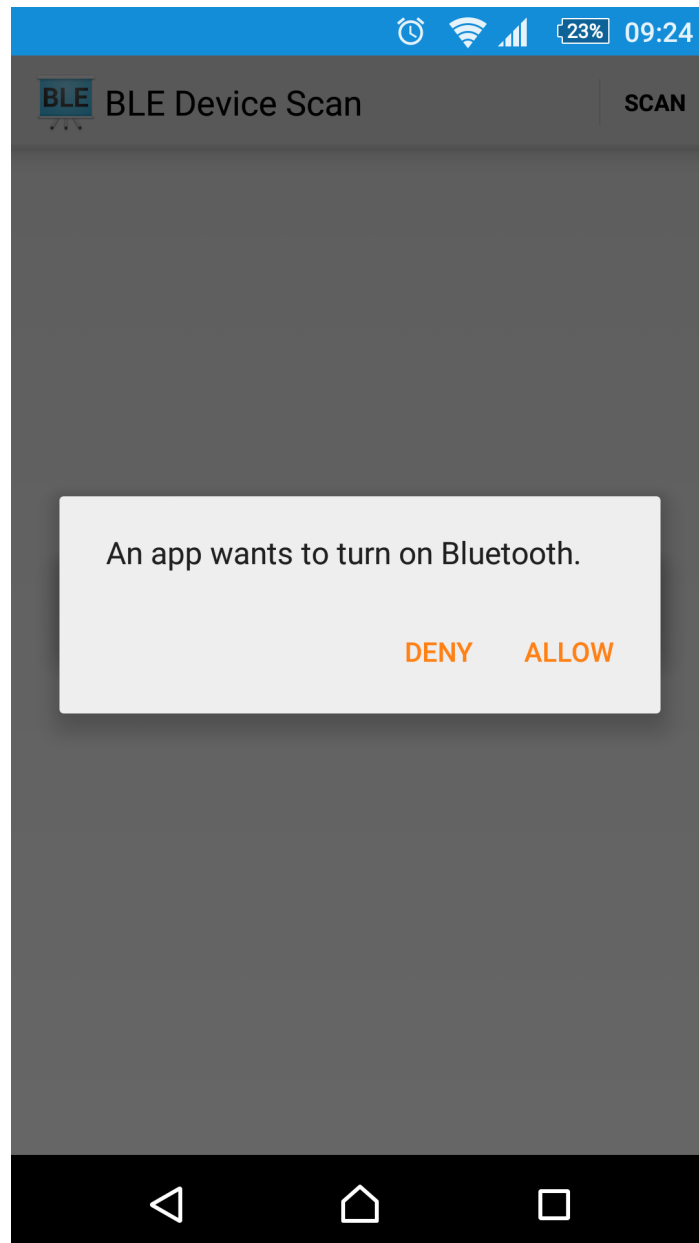
```
public void ring(){
    char sendC = 'R'; // gan ky tu can gui la 'R'
    mGattCharacteristics.setValue(String.valueOf(sendC)); // gan ky tu
        can gui cho characteristic
    mBluetoothGatt.writeCharacteristic(mGattCharacteristics); // gui ky
        tu thong qua GATT BLE
    Log.d("Ring function Call", "Service");
}
```

Xử lý khi mất kết nối: các hoạt động khi kết nối và mất kết nối được quản lý với **BroadcastReceiver()** bao gồm các giá trị **BluetoothLeService**. Khi các giá trị này thay đổi thì sẽ dẫn tới các hoạt động tương ứng:

```
// DeviceControlActivity.java
BroadcastReceiver() {
    @Override
    public void onReceive(Context context, Intent intent) {
        if (BluetoothLeService.ACTION_GATT_CONNECTED.equals(action)) {
            mConnected = true;
            updateConnectionState(R.string.connected);
            invalidateOptionsMenu();
        } else if
            (BluetoothLeService.ACTION_GATT_DISCONNECTED.equals(action)){
            mp=MediaPlayer.create(this,R.raw.noti);
            mp.start();
        }
    };
};
```

Và giao diện ứng dụng được thiết kế:

2.5 Hiện thực ứng dụng di động trên Android

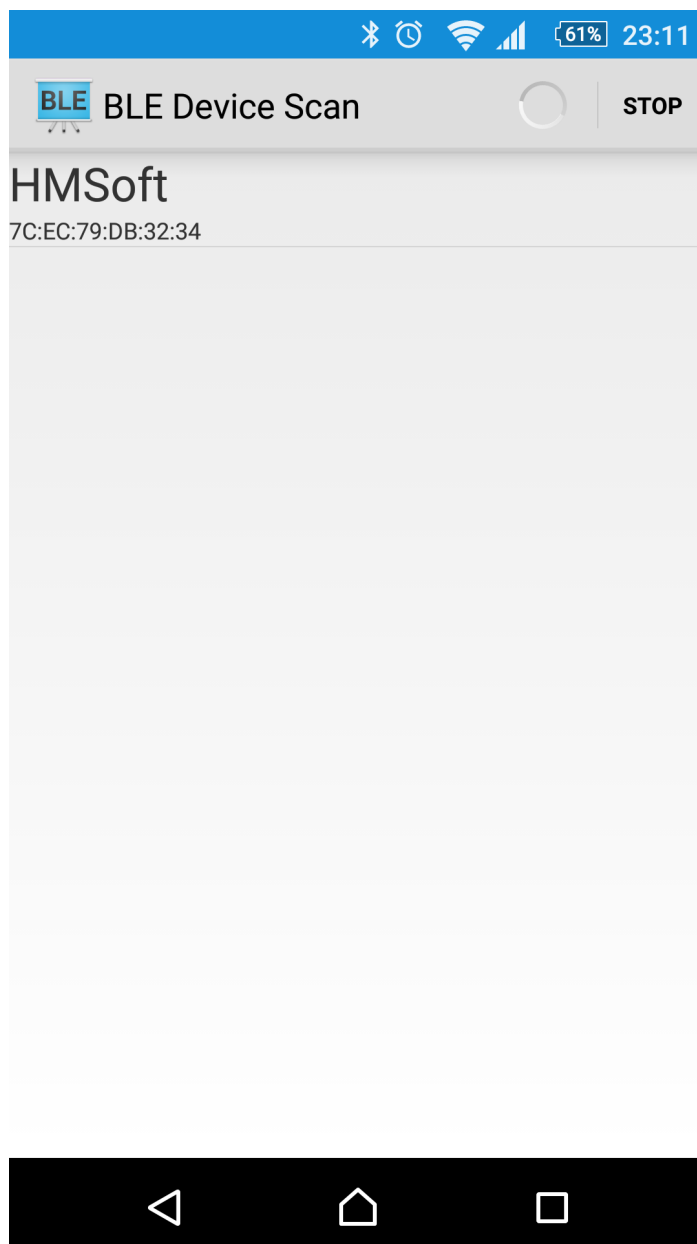


Hình 2.13 Giao diện kiểm tra BLE

Khi khởi động ứng dụng di động, chương trình sẽ kiểm tra thiết bị di động đã bật chế độ Bluetooth chưa và chọn để bắt đầu sử dụng như hình 2.13

Nếu thiết bị di động đã bật sẵn chế độ kết nối Bluetooth thì giao diện 2.13 sẽ không hiện ra.

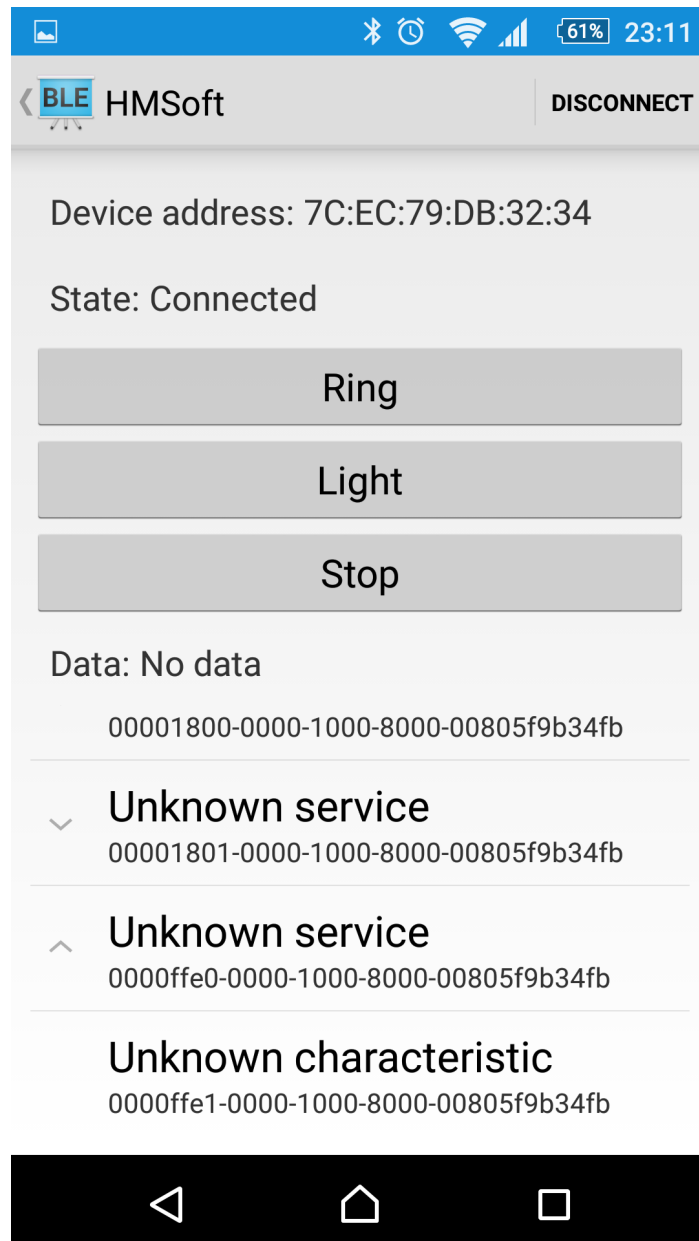
2.5 Hiện thực ứng dụng di động trên Android



Hình 2.14 Giao diện scan thiết bị BLE

Sau đó ta chọn thiết bị BLE để kết nối, ở đây thiết bị Smart Keyring sử dụng module HM-10 với thiết lập tên hiển thị mặc định là HMSoft.

2.5 Hiện thực ứng dụng di động trên Android



Hình 2.15 Giao diện chính của ứng dụng di động

Sau khi kết nối, chúng ta sẽ có những thông tin như:

- Device address: địa chỉ MAC của chip CC2540/2541.
- State: Connected hoặc Disconnected - báo trạng thái kết nối giữa ứng dụng và

thiết bị.

2.5 Hiện thực ứng dụng di động trên Android

- Các service là các characteristic (xem thêm tại 2.5.1): vì ứng dụng di động đang trong quá trình tìm hiểu và phát triển nên hiển thị các uuid để dễ theo dõi.

Ngoài ra còn có các nút chức năng:

- Ring: gửi yêu cầu bật/tắt báo hiệu bằng buzzer tới thiết bị Smart Keyring.
- Light: gửi yêu cầu bật/tắt báo hiệu bằng LED tới thiết bị Smart Keyring.
- Stop: tắt báo hiệu bằng âm thanh trên thiết bị di động sau khi mất kết nối với thiết bị Smart Keyring.

Để sử dụng các chức năng trên, ta cần chọn service **0000ffe0-0000-1000-8000-00805f9b34fb** và characteristic **0000ffe1-0000-1000-8000-00805f9b34fb** - tức là service và characteristic của chức năng đọc truyền nhận dữ liệu thông qua kết nối Bluetooth.

Source code có thể xem tại: <https://github.com/MrWhoz/ble-smartlkey>

Chương 3

Thử nghiệm và đánh giá kết quả đạt được

3.1 Thử nghiệm

3.1.1 Khoảng cách hoạt động

Phương pháp thực nghiệm: Kết nối thiết bị Smart Keyring và thiết bị di động, liên tục kiểm tra kết nối trao đổi tín hiệu giữa 2 thiết bị và tăng dần khoảng cách cho đến khi có báo hiệu mất kết nối. Thực nghiệm ở 2 trường hợp môi trường không vật cản và có vật cản.

Thử nghiệm được thực hiện với điện thoại Sony Xperia Z1:

Thực nghiệm cho thấy, phạm vi tối đa trung bình để việc truyền nhận dữ liệu còn chính xác là vào khoảng 22m đối với môi trường không vật cản, 8m đối với môi trường có vật cản(các 1 bức tường).

Các lần đo	Khoảng cách còn hoạt động
Lần 1	22m
Lần 2	22m
Lần 3	21.5m
Lần 4	23m
Lần 5	23m
Lần 6	22.5m
Lần 7	22m
Lần 8	21m
Lần 9	22m
Lần 10	22m

Bảng 3.1 Thử nghiệm khoảng cách hoạt động khi không có vật cản

Các lần đo	Khoảng cách còn hoạt động
Lần 1	8m
Lần 2	8m
Lần 3	8.5m
Lần 4	8.5m
Lần 5	8m
Lần 6	7.5m
Lần 7	8m
Lần 8	7.5m
Lần 9	8m
Lần 10	8.5m

Bảng 3.2 Thử nghiệm khoảng cách hoạt động khi có vật cản

3.1.2 Năng lượng tiêu thụ

Thông số tiêu thụ năng lượng của các linh kiện theo nhà sản xuất:

- Tiêu thụ của HM-10: 0.2mA ở sleep mode, 8mA ở active mode
- Tiêu thụ của vi điều khiển ATmega328P: 0.2mA ở active mode
- Tiêu thụ của đèn LED:
- Buzzer tín hiệu khi kích hoạt: 32mA
- Dung lượng nguồn sử dụng: Pin CR2032 dung lượng 220mAh

Thực tế năng lượng tiêu thụ:

Phương pháp thực nghiệm: kết nối thiết bị Smart Keyring với thiết bị di động, mỗi ngày kiểm tra kết nối 2 lần vào lúc 7h sáng và 5h chiều và tự đưa về chế độ Idle ở thời gian rảnh. Giữ thiết bị hoạt động liên tục với pin CR2032 cho đến khi hết pin.

Kết quả: Bắt đầu thực nghiệm vào lúc 7h sáng ngày 1/11, thiết bị không còn khả năng kết nối vào lúc 4h30 chiều ngày 18/11 mặc dù đèn nguồn vẫn còn sáng nhưng không đủ năng lượng để duy trì module BLE giữ kết nối.

3.2 Đánh giá kết quả ứng dụng thực tế

Phần này sẽ đánh giá về khả năng ứng dụng thực tế dựa trên kết quả thực nghiệm.

Về khoảng cách hoạt động tối đa và tính ứng dụng thực tế hiệu quả:

- **Môi trường không vật cản vào khoảng 22m:** hiệu quả trong việc báo mất kết nối trong các trường hợp thực tế như bỏ quên chìa khóa trên xe trong bãi giữ xe, thiết bị di động tại nơi công cộng vì khoảng cách hợp lý không quá ngắn và cũng như không quá xa.

- **Môi trường có vật cản vào khoảng 8m:** hiệu quả trong việc báo mất kết nối trong các trường hợp thực tế như để quên trong phòng, báo hiệu sớm để người dùng có thể biết ngay khi bỏ quên 1 trong 2 thiết bị khi vừa rời khỏi phòng. Và khoảng cách đủ để duy trì kết nối kích hoạt chế độ báo hiệu tìm kiếm khi thiết bị che khuất tầm nhìn giúp việc tìm kiếm nhanh chóng và hiệu quả hơn.

Về thời gian hoạt động: vì hạn chế trong việc không tối ưu thiết bị, đạt 18 ngày thay pin 1 lần là kết quả kém hiệu quả trong cuộc sống hằng ngày.

Chương 4

Tổng kết

4.1 Thành quả đạt được và khó khăn

4.1.1 Thành quả đạt được

Sau thời gian 7 tháng thực hiện đề tài NCKH, nhóm đã đạt được các thành quả:

- Tìm hiểu được về công nghệ Bluetooth Low Energy và các thiết bị phần cứng hỗ trợ.
- Kết nối Bluetooth với thiết bị di động theo chuẩn Bluetooth BLE 4.0.
- Đạt được các chức năng trên thiết bị Smart Keyring đã được đề ra.
- Khả năng hoạt động liên tục đến 430 giờ ở chế độ idle.
- Hiện thực ứng dụng thiết bị di động trên Android.

4.1.2 Khó khăn

Bên cạnh những thành quả đạt được, nhóm đã vấp nhiều khó khăn:

- Tốn nhiều thời gian cho phương pháp tiếp cận ban đầu không khả thi vì gặp trở ngại trong cả phần cứng lẫn phần mềm trong việc phát triển (đã được nhắc đến ở mục 2.2.1)

4.2 Tính ứng dụng thực tiễn của sản phẩm

- Không có đủ thời gian để thực nghiệm đầy đủ hơn.
- Không đủ thiết bị di động có chuẩn BLE cho tất cả thành viên trong nhóm làm việc hiện thực và thực nghiệm gặp nhiều hạn chế.
- Lựa thiết bị thiết bị phần cứng ở ở Việt Nam tại thời điểm tìm hiểu và phát triển đề tài NCKH.
- Ứng dụng di động trên Android chưa phát triển hoàn thiện và mất nhiều thời gian tìm hiểu lập trình trên Android cho người mới bắt đầu.

4.2 Tính ứng dụng thực tiễn của sản phẩm

Theo đánh giá về tính ứng dụng thực tiễn được dựa theo mục 3.2, ta rút ra kết luận:

- Khoảng cách hoạt động chấp nhận được.
- Các chức năng hoạt động tốt và có lợi ích trong cuộc sống hằng ngày.
- Ứng dụng di động trên Android chưa hỗ trợ chế độ chạy nền nên hơi bất tiện cho người dùng trong việc sử dụng hằng ngày.
- Thời gian hoạt động kém hiệu quả, chưa sử dụng tốt được trong cuộc sống hằng ngày.

4.3 Hướng phát triển

- Tìm hiểu các linh kiện điện tử khác hỗ trợ lập trình SoC BLE để tối ưu hóa hiệu năng hoạt động cũng như kích thước của sản phẩm.
- Thiết kế ngoại hình bên ngoài sản phẩm để ứng dụng thực tế hơn.
- Phát triển ứng dụng Android hỗ trợ chế độ chạy ngầm.
- Có thể tùy chỉnh được khoảng cách kết nối.

Tài liệu tham khảo

- [1] Google Dev. Android bluetoothlegatt sample. <https://github.com/googlesamples/android-BluetoothLeGatt/>, .
- [2] Google Dev. Bluetooth low energy - api guides. <https://developer.android.com/guide/topics/connectivity/bluetooth-le.html>, .
- [3] HTelectronic. Hướng dẫn giao tiếp với smartphone, máy tính bảng qua bluetooth low energy. <http://htelectronics.vn/huong-dan-giao-tiep-voi-smartphone-may-tinh-bang-qua-bluetooth-low-energy-ble>.
- [4] IAR Systems. Iar embedded workbench for 8051. <https://www.iar.com/iar-embedded-workbench/>.
- [5] thegioididong.com. Tìm hiểu công nghệ bluetooth. <https://www.thegioididong.com/tin-tuc/tim-hieu-cong-nghe-bluetooth-590690>.