

**ĐẠI HỌC QUỐC GIA TP.HỒ CHÍ MINH**  
**TRƯỜNG ĐẠI HỌC BÁCH KHOA**  
**KHOA ĐIỆN – ĐIỆN TỬ**  
**BỘ MÔN VIỄN THÔNG**

-----o0o-----



**BÁO CÁO ĐỀ CƯƠNG LUẬN VĂN**

**MẠCH RF CC2538 VÀ CONTIKI-NG**

**GVHD: VÕ QUẾ SƠN**

**SVTH: NGUYỄN HỮU HOÀNG**

**MSSV: 1411319**

**TP. HỒ CHÍ MINH, THÁNG 6 NĂM 2018**

## TÓM TẮT ĐỀ CƯƠNG LUẬN VĂN

Đề cương này trình bày về phát triển, thiết kế phần cứng và sử dụng hệ điều hành thời gian thực Contiki-NG trên phần cứng đã thiết kế, hướng đến việc ứng dụng trong các sản phẩm liên quan đến IoT.

## MỤC LỤC

1. GIỚI THIỆU.....	3
1.1 Tổng quan.....	3
1.2 Nội dung.....	4
2. LÝ THUYẾT.....	4
2.1 Giới thiệu Contiki-NG.....	4
2.2 IPv6.....	4
2.3 RPL.....	5
2.4 CoAP.....	5
2.5 OMA Lightweight M2M và IPSO Smart Objects.....	6
2.6 TSCH và 6TiSCH.....	7
3 THIẾT KẾT VÀ THỰC HIỆN PHẦN CỨNG.....	7
3.1 Yêu cầu thiết kế.....	7
3.2 Sơ đồ nguyên lý và kết quả thực hiện mạch.....	8
3.2 Phần cứng sử dụng.....	11
4. KIỂM TRA KHOẢNG CÁCH TRUYỀN (RANGE TEST).....	11
4.1 Antena sử dụng.....	11
4.2 Thực hiện.....	12
4.3 Kết quả thực hiện.....	13
4.4 Kết luận.....	14
5. THỰC HIỆN CONTIKI-NG TRÊN CC2538 DEVELOPMENT KIT.....	14
5.1 Build and flash programming trên cc2538.....	14
5.2 Logging.....	14
5.3 Shell.....	15
5.4 RPL Lite and RPL border router.....	16
5.5 TSCH and 6TiSCH.....	23
5.6 CoAP.....	24
6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....	26
6.1 Kết luận.....	26
6.2 Hướng phát triển.....	27

## DANH SÁCH HÌNH MINH HỌA

Hình 1: Cấu trúc CoAP.....	6
Hình 2: Cấu trúc OMA LWM2M và IPSO.....	6
Hình 3: Sơ đồ khối phần cứng.....	8
Hình 4: Sơ đồ nguyên lý.....	9
Hình 5: Hình vẽ layout.....	9
Hình 6: Hình vẽ 3D.....	10
Hình 7: Kết quả mạch thực.....	10
Hình 8: Module CC2538.....	11
Hình 9: Hình dạng Antena được sử dụng.....	11
Hình 10: Các trục quy định trên Antena.....	12
Hình 11: Sơ đồ Range Test.....	13
Hình 12: Build và nạp code cho cc2538.....	14
Hình 13: Logging Terminal.....	15
Hình 14: Shell Terminal.....	16
Hình 15: RPL Client.....	17
Hình 16: RPL Server.....	17
Hình 17: Kiểm tra cấu hình.....	19
Hình 18: Hello World Terminal.....	20
Hình 19: Kết nối border router qua Tunslip6.....	21
Hình 20: Ping Terminal.....	21
Hình 21: Kiểm tra các node trong RPL trên trình duyệt.....	22
Hình 22: Border Router từ Hello-World.....	22
Hình 23: Xem cấu hình TSCH.....	23
Hình 24: CoAP Server.....	24
Hình 25: CoAP Client.....	25
Hình 26: Chạy Border Router.....	25
Hình 27: Ping Node CoAP Server.....	26
Hình 28: Get thông tin từ CoAP Server.....	26
Hình 29: Get đoạn Text trong test/hello-world.....	26

## **DANH SÁCH BẢNG SỐ LIỆU**

Bảng 1: Thông số độ lợi các hướng của Antena.....	12
Bảng 2: Kết quả đo đặc thực tế khoảng cách truyền nhận trên CC2538.....	13

## 1. GIỚI THIỆU

### 1.1 Tổng quan

Cách mạng Công nghiệp IV, Internet of Things (IoT), những cụm từ quen thuộc trong những năm gần đây khi nhắc đến công nghệ. Kiến trúc IoT được đại diện cơ bản bởi 4 phần: Vạn vật (Things), trạm kết nối (Gateways), hạ tầng mạng và điện toán đám mây (Network and Cloud) và các lớp tạo và cung cấp dịch vụ (Services-creation and Solutions Layers).

Đề cương đề cập vào việc phát triển một phần trong Vạn vật (Things), phát triển phần cứng và ứng dụng phần mềm chạy phần cứng, sau đó sẽ định hướng cho Luận Văn Tốt Nghiệp sau này.

### 1.2 Nội dung

Đề cương này tập trung vào các phần chính sau:

- Phát triển phần cứng trên dựa vào chip RF CC2538 của hãng TI.
- Kiểm tra, thử nghiệm khoảng cách truyền của mạch phát triển.
- Tìm hiểu hệ điều hành Contiki-ng và phát triển nó trên CC2538

## 2. LÝ THUYẾT

Lý thuyết sẽ tập trung chủ yếu vào các vấn đề liên quan đến hệ điều hành Contiki-NG, hệ điều hành sẽ chạy cho phần cứng.

### 2.1 Giới thiệu Contiki-NG

Contiki-NG là một hệ điều hành cho các thiết bị có giới hạn về nguồn tài nguyên sử dụng trong Internet of Thing. Contiki-NG bao gồm các chuẩn tuân thủ theo RFC, các giao tiếp sử dụng năng lượng thấp chạy trên IPv6, cho phép kết nối Internet. Contiki-NG được bắt đầu phát triển vào năm 2017, dựa trên hệ thống Contiki được phát triển trước đó, vì thế nó được gọi là Contiki-NG (Contiki Next Generation), phiên bản với các tính năng và hoàn thiện hơn hẳn so với Contiki, cả về cấu trúc lẫn phát triển ứng dụng. Contiki-NG tập trung vào các mục tiêu sau;

- Tăng độ tin cậy và bảo mật, truyền thông giao tiếp IPv6 làm chuẩn cơ sở.
- Tập trung cho các nền tảng phần cứng IoT mới và mạnh mẽ với các lõi như ARM Cortex M3 và các lõi 32-bit MCUs khác.
- Hiện đại hóa, cải thiện cấu trúc, cấu hình, logging và platforms,..
- Cung cấp các tài liệu, các code API (Application Programming Interface), mô tả các module.
- Thực hiện xử lý phát triển nhanh hơn, với việc dễ dàng ứng dụng các tính năng mới.

Phiên bản đầu tiên là Contiki-NG 4.0, phiên bản hiện hành là Contiki-NG 4.1. Bài làm này sử dụng phiên bản Contiki-NG 4.1.

### 2.2 IPv6

Một trong những tính năng chính của Contiki-NG là thiết kế ứng dụng IPv6 Network Stack cho hệ thống mạng sử dụng năng lượng thấp và có tồn hao. Network stack bao gồm các giao thức như IPv6, TCP, UDP, DNS, RPL, CoAP, LWM2M và Websockets. Bên dưới IPv6 Stack, Contiki-NG hỗ trợ chuẩn truyền thông không dây 802.15.4 với Time Slotted Channel Hopping (TSCH).

IPv6 stack trên Contiki-NG hỗ trợ multicast. Hiện tại hỗ trợ 3 mode: 'Stateless Multicast RPL Forwarding' (SMRF), 'Enhanced Stateless Multicast RPL Forwarding' (ESMRF), Multicast Forwarding with Trickle'.

### 2.3 RPL

RPL (IPv6 routing protocol for lossy and low-power networks) là một giao thức định tuyến sử dụng IPv6 cho mạng tổn hao và sử dụng năng lượng thấp, được mô tả trong [RFC 6550](#), được thực hiện trong IPv6 network stack của Contiki-NG. RPL xây dựng và duy trì một topology là DODAG (Destination-Oriented Directed Acyclic Graph), một mạng định hướng, hướng về đích đến của dữ liệu, cập nhật mạng theo chu kỳ gọi là Trickle timer, mạng sẽ xây dựng dựa theo một OF (Objective Function) hàm chức năng nào đó như [MRHOF\(The Minimum Rank with Hysteresis Objective Function\)](#), [OFO\(Objective Function Zero\)](#).

RPL Lite là một phiên bản kế thừa của phiên bản RPL trước đó (hay còn gọi là RPL Classic), được thực hiện RPL mặc định trong Contiki-NG, tập trung vào các chức năng chủ yếu, quan trọng và ổn định, sau khi phát triển, nghiên cứu, thảo luận và trải qua nhiều phiên bản RPL Classic trước đó.

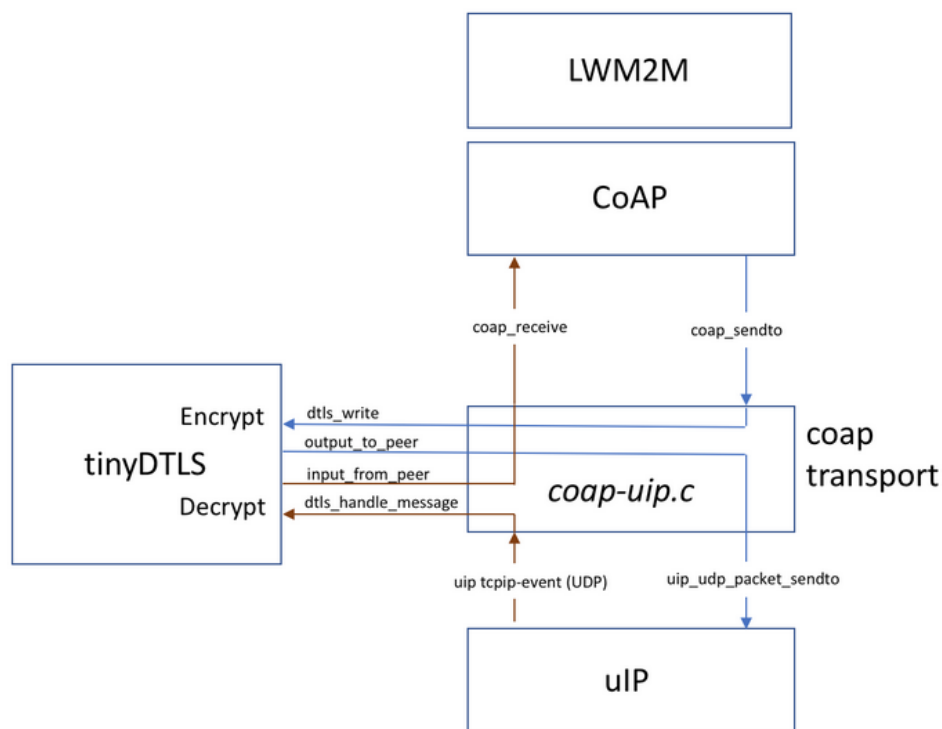
RPL lite loại bỏ tính năng Storing Mode, ủng hộ sử dụng Non-storing Mode, loại bỏ tính phức tạp của nhiều phiên bản và DODAGs trước đó. Những thay đổi này giúp RPL Lite cải thiện hiệu năng cũng như dung lượng code cho ROM. Tuy nhiên, nó cũng sẽ có sự trả giá, đó là một số tính năng, và khả năng tương tác bị hạn chế lại, như tính năng Storing Mode.

### 2.4 CoAP

CoAP là một giao thức ở lớp ứng dụng (Application-Layer protocol), tương tự như HTTP nhưng vận chuyển thông qua UDP thay vì TCP. CoAP được định nghĩa trong [RFC 7252](#).

Contiki-NG CoAP được xây dựng dựa trên Erbium thực hiện bởi Mattias Kovatsh nhưng được tái cấu trúc lại để dễ sử dụng hơn.

Bảo mật trên Contiki-NG CoAP được thực hiện thông qua TinyDTLS, tích hợp sẵn trong Contiki-NG 4.1.

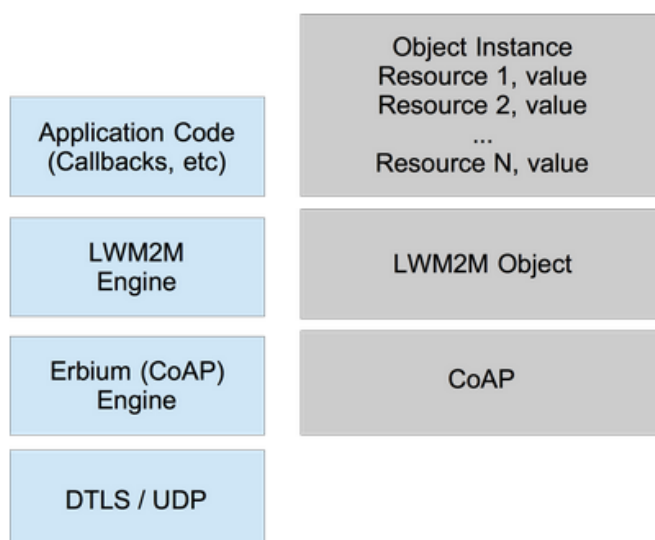


Hình 1: Cấu trúc CoAP

## 2.5 OMA Lightweight M2M và IPSO Smart Objects

OMA được định nghĩa là một chuẩn cho các thiết bị IoT hoặc M2M, được xây dựng trên CoAP (với DTLS). Nó định nghĩa bootstrap và quản lý thiết bị. Lớp trên của LWM2M là IPSO Smart Object mô tả cho các sensors và actuators.

Trong Contiki-NG phát triển OMA LWM2M trên lớp cao nhất của Erbium CoAP. Nó cung cấp bootstrap và làm việc với cả hai server Leshan và Wakaama LWM2M. Cung cấp các format data là TLV và JSON.



Hình 2: Cấu trúc OMA LWM2M và IPSO



## 2.6 TSCH và 6TiSCH

TSCH ( Time Slotted Channel Hopping) là một giao thức lớp MAC.

6TiSCH là IPv6 dựa trên TSCH được định nghĩa trong [\[RFC2119\]](#).

Contiki-NG thực hiện TSCH và 6TiSCH gọi là “minimal configuration”, để định nghĩa làm thế nào để chạy RPL+TSCH network.

Các node chạy TSCH theo cơ chế đồng bộ toàn cục tiêu tốn ít năng trong mạng mesh. Các node có thể tham gia vào mạng sau khi nghe một Enhanced Beacon (EB) từ một node khác. Đồng bộ thời gian trickle từ các PAN Coordinator xuống các leaf node (các node con) theo cấu trúc DAG trong RPL. Thời gian sẽ được cắt thành các timeslots, các timeslot sẽ được nhóm lại thành một hay nhiều slotframes. Một timeslot thông thường là 10ms, đủ dài để một node gửi một frame và nhận ACK cho nó. Một TSCH schedule chỉ định cho một node sẽ phải làm gì trong mỗi timeslot như truyền, nhận, ngủ. Mỗi timeslot trong một slotframe được định nghĩa bởi time offset của nó, channel offset của nó và theo các tính chất như liệu nó được sử dụng cho truyền, nhận, đồng bộ thời gian... .

TSCH network sử dụng nhảy kênh (channel hopping): các slot giống nhau trong lịch trình sẽ chuyển thành các tần số khác nhau tại mỗi lần lặp của frame. Kết quả là các gói giữa các nút lân cận trao đổi ở các tần số khác nhau. Trong trường hợp lỗi, sẽ truyền lại gói với một tần số khác, thường xác suất thành công sẽ tốt hơn so với truyền lại cùng một tần số.

6TiSCH bao gồm architered, interface, scheduling và security cho phép chạy IPv6 trên TSCH. 6TiSCH cho phép các node và quản lý schedule thông qua CoAP, xây dựng trên COMI (CoAP Management Interface). Nó cũng định nghĩa 6top, một lớp phụ cho phép neighbor-to-neighbor slot có thể installation/removal. 6top có thể chạy một hay nhiều scheduling functions, với định nghĩa khi nào và làm thế nào add và remove các slot ở mỗi node. 6TiSCH cũng định nghĩa một khái niệm gọi là *minimal configuration*, cung cấp các thiết lập thực hiện tung tác cơ bản, bao gồm các cấu hình cho schedule và security đơn giản dựa trên hai keys K1, K2. K1 sử dụng cho việc xác thực các beacons. K2 cho việc mã hóa, xác thực data và ACK traffic. Thêm nữa, *minimal configuration* định nghĩa làm thế nào TSCH có thể tương tác với các lớp trên nó và giao thức định tuyến RPL.

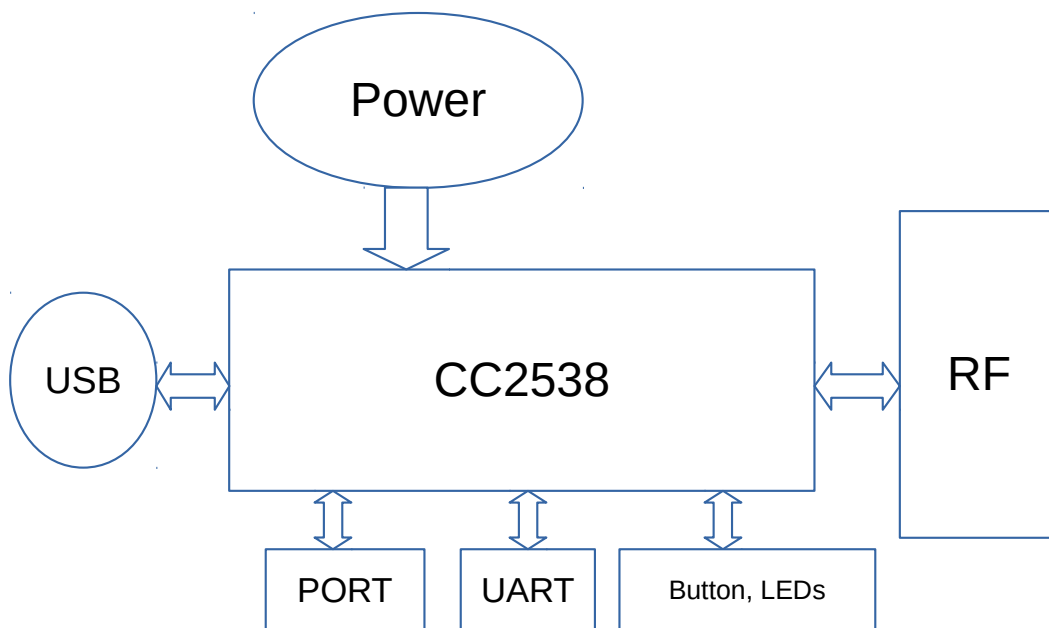
## 3 THIẾT KẾT VÀ THỰC HIỆN PHẦN CỨNG

### 3.1 Yêu cầu thiết kết

- Mạch phần cứng dựa trên chip RF CC2538 của hãng TI ( Texas Instruments).
- Nguồn cấp USB và PIN, có khả năng sạc pin.
- Nạp code thông qua serial.
- Các chân port của chip, LED, Button cho người sử dụng.
- Các chân để cho mạch.

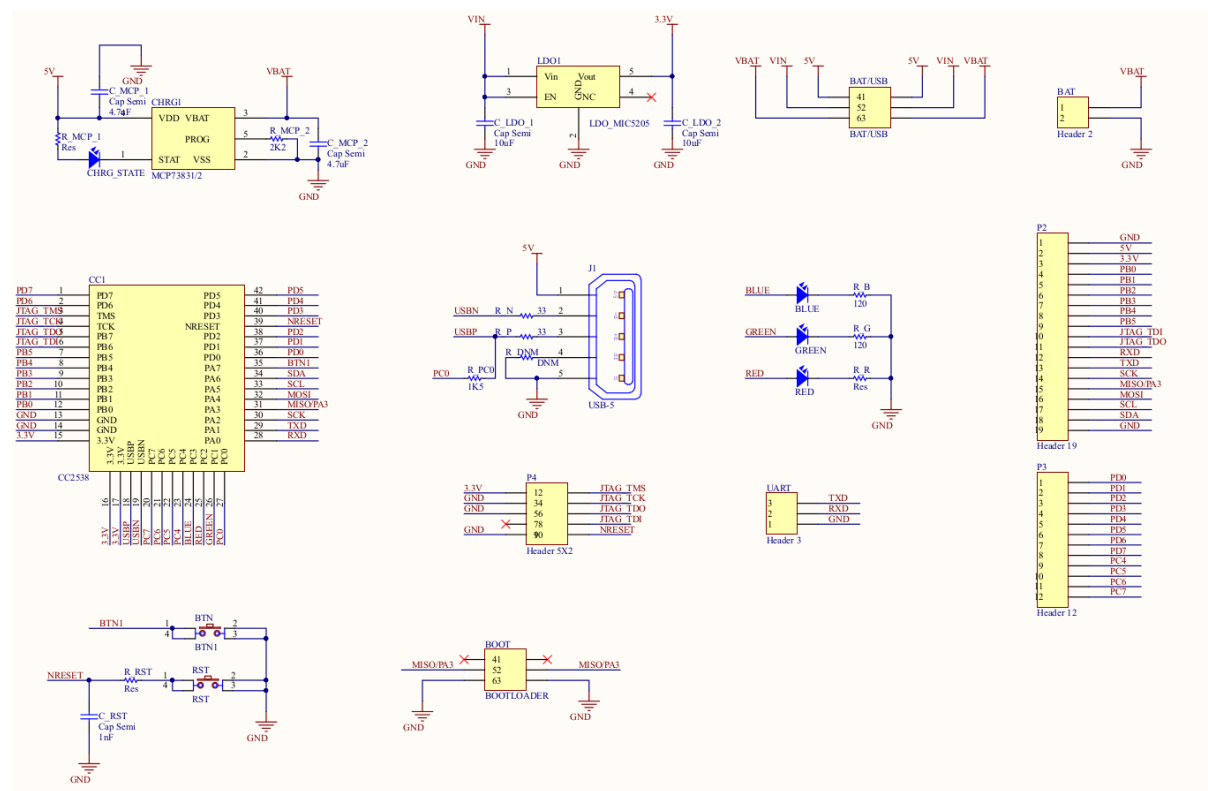
### 3.2 Sơ đồ nguyên lí và kết quả thực hiện mạch

- Sơ đồ khối



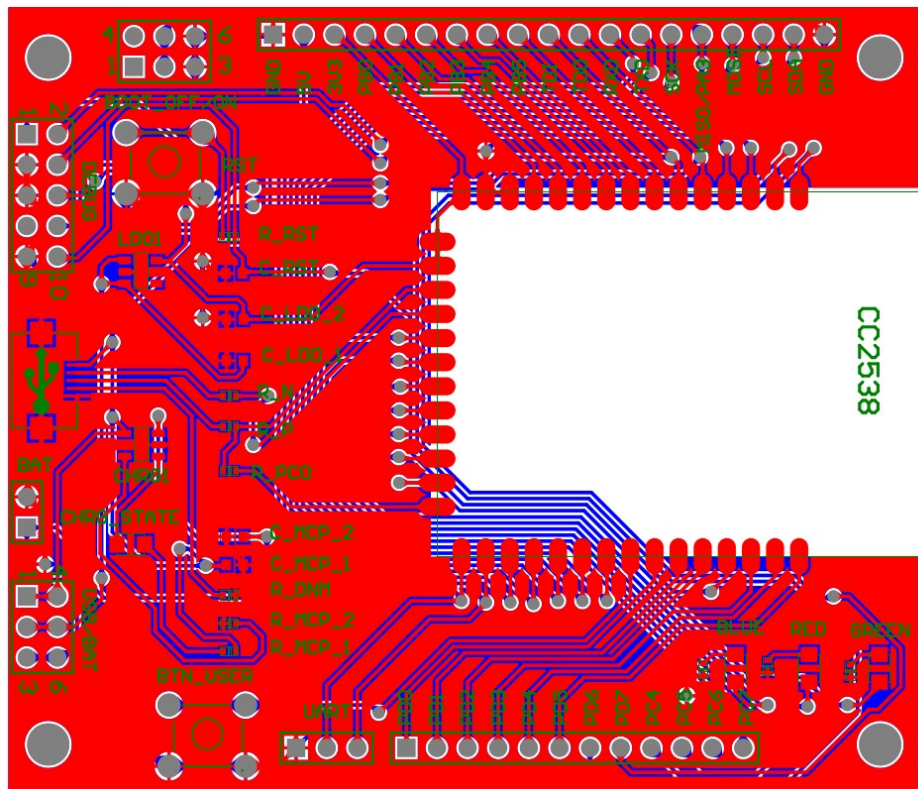
Hình 3: Sơ đồ khối phần cứng

- Sơ đồ nguyên lí



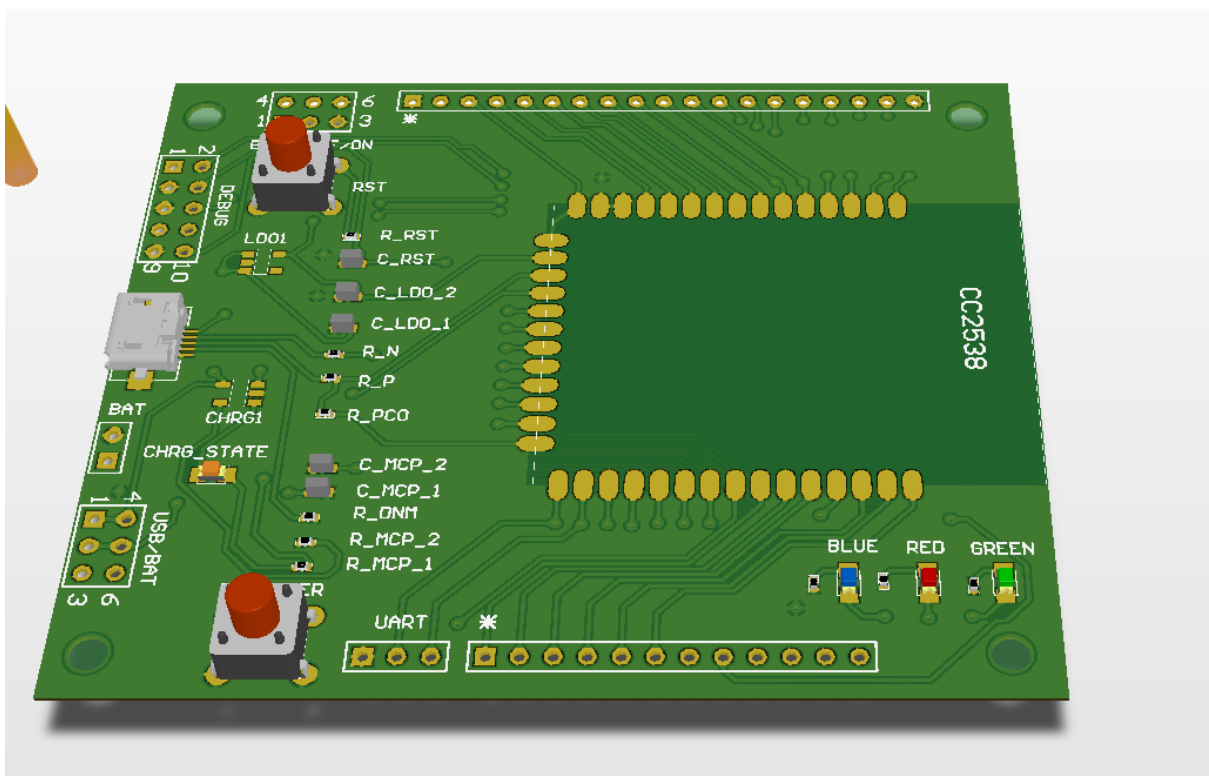
Hình 4: Sơ đồ nguyên lí

- Layout



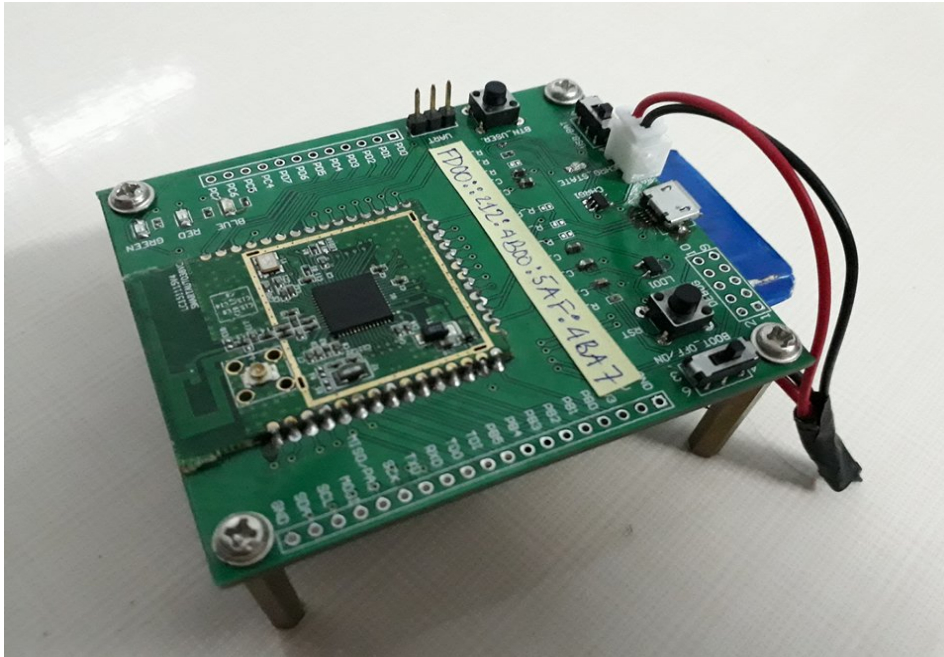
Hình 5: Hình vẽ layout

- Hình ảnh 3D



Hình 6: Hình vẽ 3D

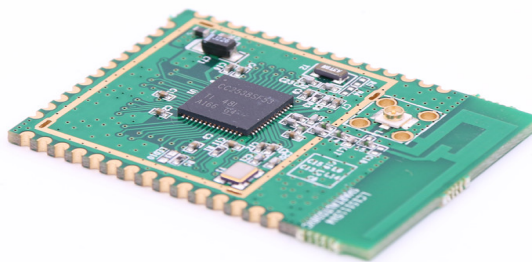
- Kết quả mạch thực



Hình 7: Kết quả mạch thực

### 3.2 Phần cứng sử dụng

Module CC2538 được sử dụng cho khai triển phần cứng:



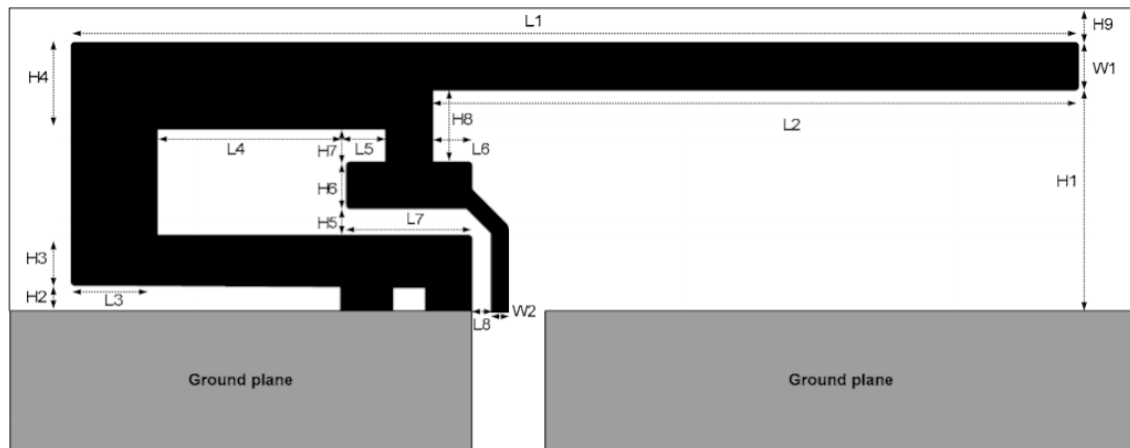
Hình 8: Module CC2538

#### Thông số kỹ thuật:

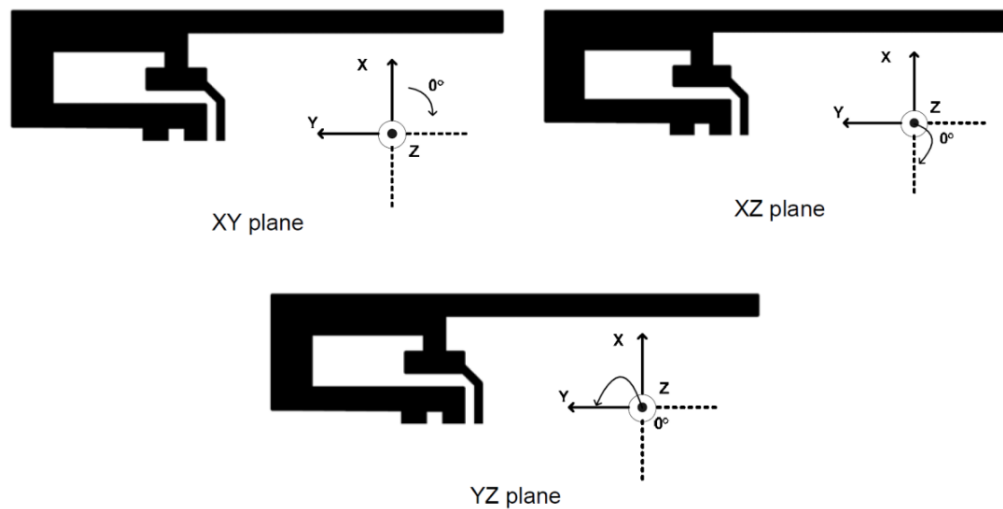
- Chip CC2538 TI, lõi ARM Cortex-M3, clock 32MHz, Flash 512KB, RAM 32KB.
- RF 2.4 GHz, 802.15.4, RSS -97dBm, 7dBm transmit power.
- Support SMA, IPX chip base, support on plate antenna

## 4. KIỂM TRA KHOẢNG CÁCH TRUYỀN (RANGE TEST)

### 4.1 Antena sử dụng



Hình 9: Hình dạng Antena được sử dụng



Hình 10: Các trục quy định trên Antena

- Với các thông số về mức phản xạ sóng của antena (Reflection) theo lý thuyết:

STT	Plane	Reflection level (dBm) Vertical Polarization	Reflection level (dBm) Horizontal Polarization
1	XY	1.0	1.1
2	XZ	3.3	-1.5
3	YZ	1.6	1.1

Bảng 1: Thông số độ lợi các hướng của Antena



Trong bài test, antenna được tích hợp sẵn trên module, bộ truyền và bộ nhận giao tiếp trong khoảng hoạt động theo hướng XZ.

## 4.2 Thực hiện



Hình 11: Sơ đồ Range Test

- Địa điểm thực hiện là ở tòa nhà C4, C5, C6 của trường Đại Học Bách Khoa TP.HCM.
- Với một node cố định, thu thập dữ liệu là vòng tròn đồng.
- Các node ở vòng tròn vàng, xanh dương, xanh lục lần lượt cách node ở vòng tròn đỏ 25m, 50m, 75m.
- Điều kiện môi trường: thời tiết không mưa, node ở vòng tròn đỏ đặt ở độ cao 2m so với mặt đất, các node còn lại ở độ cao 1m đến 1,5m. Xung quang có nhiều tòa nhà, cây cối, người qua lại.

## 4.3 Kết quả thực hiện

Thực hiện sử dụng lệnh ping6 trên node border router, ping liên tục với khoảng cách các lần ping là 1s, thực hiện gửi hơn 2000 gói, mỗi gói 64 bytes.

STT	Khoảng cách (m)	PRR (%)	Round-trip time (min/avg/max/mdev) (ms)	RSSI (dBm)
1	25	1572/2100=75	29.416 /42.489 /801.4 /20.141	- 86
2	50	1365/2018=68	29.687 /43.694 /55.977 /261.00	- 89
3	75	165/2556=7	30.644 /37.519 /202.654 /14.220	-91

Bảng 2: Kết quả đo đặc thực tế khoảng cách truyền nhận trên CC2538

#### 4.4 Kết luận

- Khoảng cách truyền nhận được dữ liệu với antenna có sẵn trên module là khá thấp, chỉ số PRR thấp.
- Lệnh Ping hoạt không ổn định, đôi lúc phải reset lại node mới hoạt động.
- Ở khoảng cách 75m, không thể nhận data từ coap server được, tỉ lệ nhỏ, vì packet không nhận được liên tục, thời gian nhận phản hồi quá lâu.
- Ở khoảng cách 25m, 50m, hoạt động ổn định hơn, có thể chạy được phương thức Coap.
- Đối với antenna có sẵn trên module này, thích hợp cho các ứng dụng truyền gần như Smart Home.
- Để có thể sử dụng vào các ứng dụng cần khoảng cách xa hơn, cần phải gắn antenna ngoài có tính đẳng hướng và độ lợi cao.

## 5. THỰC HIỆN CONTIKI-NG TRÊN CC2538 DEVELOPMENT KIT

### 5.1 Build and flash programming trên cc2538

Set up : PA3 → 0 logic level and reset chip

Command: **make TARGET=cc2538dk file\_name.upload**

```

ngghuuhoang@ngghuuhoang-Latitude-E5550: ~/contiki-ng/examples/hello-world
ngghuuhoang@ngghuuhoang-Latitude-E5550:~/contiki-ng/examples/hello-world$ make TAR
GET=cc2538dk hello-world.upload
CC      ../../arch/cpu/cc2538/./ieee-addr.c
CC      ../../arch/cpu/cc2538/cc2538.lds
CC      ../../arch/cpu/cc2538/./startup-gcc.c
CC      hello-world.c
LD      hello-world.elf
arm-none-eabi-objcopy -O binary --gap-fill 0xff hello-world.elf hello-world.bin
python ../../tools/cc2538-bsl/cc2538-bsl.py -e -w -v -a 0x00202000 hello-world.b
in
Opening port /dev/ttyUSB0, baud 500000
Reading data from hello-world.bin
Cannot auto-detect firmware filetype: Assuming .bin
Connecting to target...
CC2538 PG2.0: 512KB Flash, 32KB SRAM, CCFG at 0x0027FFD4
Primary IEEE Address: 00:12:4B:00:05:AF:4B:A7
Erasing 524288 bytes starting at address 0x00200000
Erase done
Writing 516096 bytes starting at address 0x00202000
Write 8 bytes at 0x0027FFF8F00
Write done
Verifying by comparing CRC32 calculations.
Verified (match: 0x1aac6168)
rm hello-world.o obj_cc2538dk/startup-gcc.o
ngghuuhoang@ngghuuhoang-Latitude-E5550:~/contiki-ng/examples/hello-world$

```

Hình 12: Build và nạp code cho cc2538

## 5.2 Logging

Logging là cách thức hiển thị thông tin của Contiki-NG thông qua Terminal, gồm các message sau: error code, warning, information, debug message. Đây là cách thức giúp quản lý, debug tốt code cũng như rõ ràng hơn, được cải tiến ở phiên bản Contiki-NG so với phiên bản Contiki trước đó.

Kết quả trên terminal, sử dụng Putty hay command: **cat /dev/ttyUSB0**

```

ngghuuhoang@ngghuuhoang-Latitude-E5550:~/contiki-ng/examples/hello-world$ cat /dev
/ttyUSB0
-dirty
[INFO: Main      ] - Routing: RPL Lite
[INFO: Main      ] - Net: sicslowpan
[INFO: Main      ] - MAC: CSMA
[INFO: Main      ] - 802.15.4 PANID: 0xabcd
[INFO: Main      ] - 802.15.4 Default channel: 26
[INFO: Main      ] Link-layer address 0012.4b00.05af.4ba7
[INFO: IPv6 DS   ] Init: 16 neighbors
[INFO: IPv6 DS   ] 2 default routers
[INFO: IPv6 DS   ] 3 prefixes
[INFO: IPv6 DS   ] 0 routes
[INFO: IPv6 DS   ] 3 unicast addresses
[INFO: IPv6 DS   ] 5 multicast addresses
[INFO: IPv6 DS   ] 2 anycast addresses
[INFO: IPv6 DS   ] Adding prefix fe80::length 64, flags 0, Valid lifetime 0, Pre
ffered lifetime 0
[INFO: RPL       ] initializing
[INFO: Main      ] Tentative link-local IPv6 address fe80::212:4b00:5af:4ba7
[INFO: CC2538DK  ] TI SmartRF06 + cc2538EM
Hello, world
[ERR : hello     ] some error code (42)
[WARN : hello     ] some warning
[INFO: hello     ] some information

```

Hình 13: Logging Terminal



### 5.3 Shell

Contiki-NG cho phép sử dụng shell (sử dụng với dấu nhắc) có thể tương tác được thiết bị chạy IPv6. Để sử dụng Shell, thêm vào Makefile trong project:

**MODULES += os/services/shell**

Chạy code và nạp cho cc2538, giao tiếp serial với máy tính. Theo hướng dẫn của Contiki-NG thì khi kết nối được thiết lập, chỉ cần nhấn enter thì terminal sẽ vào chế độ shell.

Lưu ý: Thực hiện với putty terminal, thì khi nhấn enter, sẽ gửi ký tự \n (0x0A) tới phần cứng, mà Conkiti-NG chạy trên board mong đợi ký tự \r (0x0D), nên khi khai triển trên putty, shell chưa chạy được. Vì thế, chưa thể tương tác shell trực tiếp, hi vọng trong bản cập nhật tiếp theo Contiki-NG sẽ khắc phục lỗi này.

Giải pháp tạm thời là tương tác shell thông qua việc gửi echo trực tiếp thông qua command line:

Để gửi lệnh kiểm tra ip của node :

**echo "ip-addr" > /dev/ttyUSB0**

node sẽ reply ip của nó lên terminal:

Thực hiện trên cc2538dk :

```
nguuhoang@nguuhoang-Latitude-E5550:~/contiki-ng/examples/hello-world$ echo "ip-addr" > /dev/ttyUSB0
nguuhoang@nguuhoang-Latitude-E5550:~/contiki-ng/examples/hello-world$ cat /dev/ttyUSB0
Node IPv6 addresses:
-- fe80::212:4b00:5af:4bb7
#0012.4b00.05af.4bb7> ip-addr
```

Hình

#### 14: Shell Terminal

Một số câu lệnh với shell:

Available commands:

'> help': Shows this help

'> reboot': Reboot the board by watchdog\_reboot()

'> ip-addr': Shows all IPv6 addresses

'> ip-nbr': Shows all IPv6 neighbors

'> log module level': Sets log level (0--4) for a given module (or "all"). For module "mac", level 4 also enables per-slot logg'> ping addr': Pings the IPv6 address 'addr'

'> rpl-set-root 0/1 [prefix]': Sets node as root (on) or not (off). A /64 prefix can be optionally specified.

'> `rpl-status`': Shows a summary of the current RPL state

'> `rpl-global-repair`': Triggers a RPL global repair

'> `rpl-local-repair`': Triggers a RPL local repair

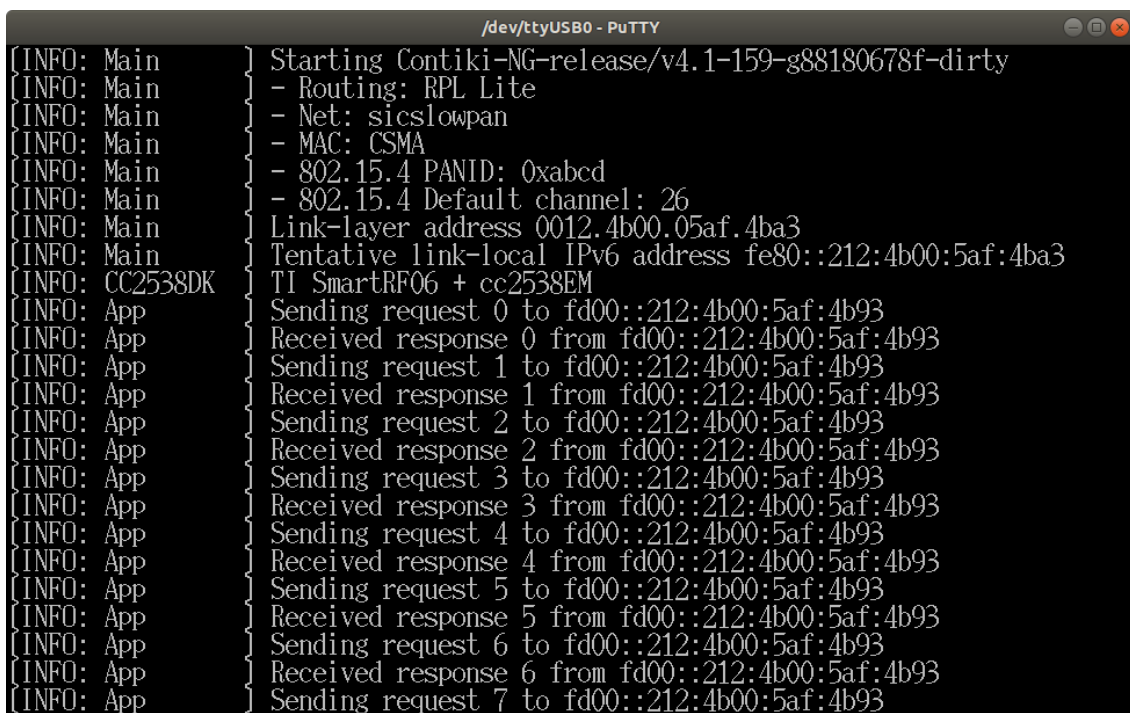
'> `routes`': Shows the route entries

## 5.4 RPL Lite and RPL border router

### Thực hiện giao tiếp UDP đơn giản với RPL Lite

Sử dụng hai node cc2538dk, một node chạy udp-server như node root, node còn lại chạy udp-client, code thực hiện trong thư mục `contiki-ng/examples/rpl-udp/`, client sẽ gửi dữ liệu về server sau mỗi khoảng thời gian nhất định, server sẽ nhận và reply lại client, thông tin được hiển thị qua UART, ở đây sử dụng Putty để theo dõi:

Tại client:



```

/dev/ttyUSB0 - PuTTY
[INFO: Main] Starting Contiki-NG-release/v4.1-159-g88180678f-dirty
[INFO: Main] - Routing: RPL Lite
[INFO: Main] - Net: sicslowpan
[INFO: Main] - MAC: CSMA
[INFO: Main] - 802.15.4 PANID: 0xabcd
[INFO: Main] - 802.15.4 Default channel: 26
[INFO: Main] Link-layer address 0012.4b00.05af.4ba3
[INFO: Main] Tentative link-local IPv6 address fe80::212:4b00:5af:4ba3
[INFO: CC2538DK] TI SmartRF06 + cc2538EM
[INFO: App] Sending request 0 to fd00::212:4b00:5af:4b93
[INFO: App] Received response 0 from fd00::212:4b00:5af:4b93
[INFO: App] Sending request 1 to fd00::212:4b00:5af:4b93
[INFO: App] Received response 1 from fd00::212:4b00:5af:4b93
[INFO: App] Sending request 2 to fd00::212:4b00:5af:4b93
[INFO: App] Received response 2 from fd00::212:4b00:5af:4b93
[INFO: App] Sending request 3 to fd00::212:4b00:5af:4b93
[INFO: App] Received response 3 from fd00::212:4b00:5af:4b93
[INFO: App] Sending request 4 to fd00::212:4b00:5af:4b93
[INFO: App] Received response 4 from fd00::212:4b00:5af:4b93
[INFO: App] Sending request 5 to fd00::212:4b00:5af:4b93
[INFO: App] Received response 5 from fd00::212:4b00:5af:4b93
[INFO: App] Sending request 6 to fd00::212:4b00:5af:4b93
[INFO: App] Received response 6 from fd00::212:4b00:5af:4b93
[INFO: App] Sending request 7 to fd00::212:4b00:5af:4b93

```

Hình 15: RPL Client

Tại server:



```

/dev/ttyUSB0 - PuTTY
[INFO: Main] Starting Contiki-NG-release/v4.1-159-g88180678f-dirty
[INFO: Main] - Routing: RPL Lite
[INFO: Main] - Net: sicslowpan
[INFO: Main] - MAC: CSMA
[INFO: Main] - 802.15.4 PANID: 0xabcd
[INFO: Main] - 802.15.4 Default channel: 26
[INFO: Main] Link-layer address 0012.4b00.05af.4b93
[INFO: Main] Tentative link-local IPv6 address fe80::212:4b00:5af:4b93
[INFO: CC2538DK] TI SmartRF06 + cc2538EM
[INFO: App] Received request 0 from fd00::212:4b00:5af:4ba3
[INFO: App] Sending response 0 to fd00::212:4b00:5af:4ba3
[INFO: App] Received request 1 from fd00::212:4b00:5af:4ba3
[INFO: App] Sending response 1 to fd00::212:4b00:5af:4ba3
[INFO: App] Received request 2 from fd00::212:4b00:5af:4ba3
[INFO: App] Sending response 2 to fd00::212:4b00:5af:4ba3
[INFO: App] Received request 3 from fd00::212:4b00:5af:4ba3
[INFO: App] Sending response 3 to fd00::212:4b00:5af:4ba3
[INFO: App] Received request 4 from fd00::212:4b00:5af:4ba3
[INFO: App] Sending response 4 to fd00::212:4b00:5af:4ba3
[INFO: App] Received request 5 from fd00::212:4b00:5af:4ba3
[INFO: App] Sending response 5 to fd00::212:4b00:5af:4ba3
[INFO: App] Received request 6 from fd00::212:4b00:5af:4ba3
[INFO: App] Sending response 6 to fd00::212:4b00:5af:4ba3
[INFO: App] Received request 7 from fd00::212:4b00:5af:4ba3
  
```

Hình 16: RPL Server

### RPL border router:

Sử dụng một node chạy chương trình rpl-border-router trong thư mục /contiki-ng/examples/rpl-border-router/ và một node chạy chương trình hello-world trong thư mục /contiki-ng/examples/hello-world

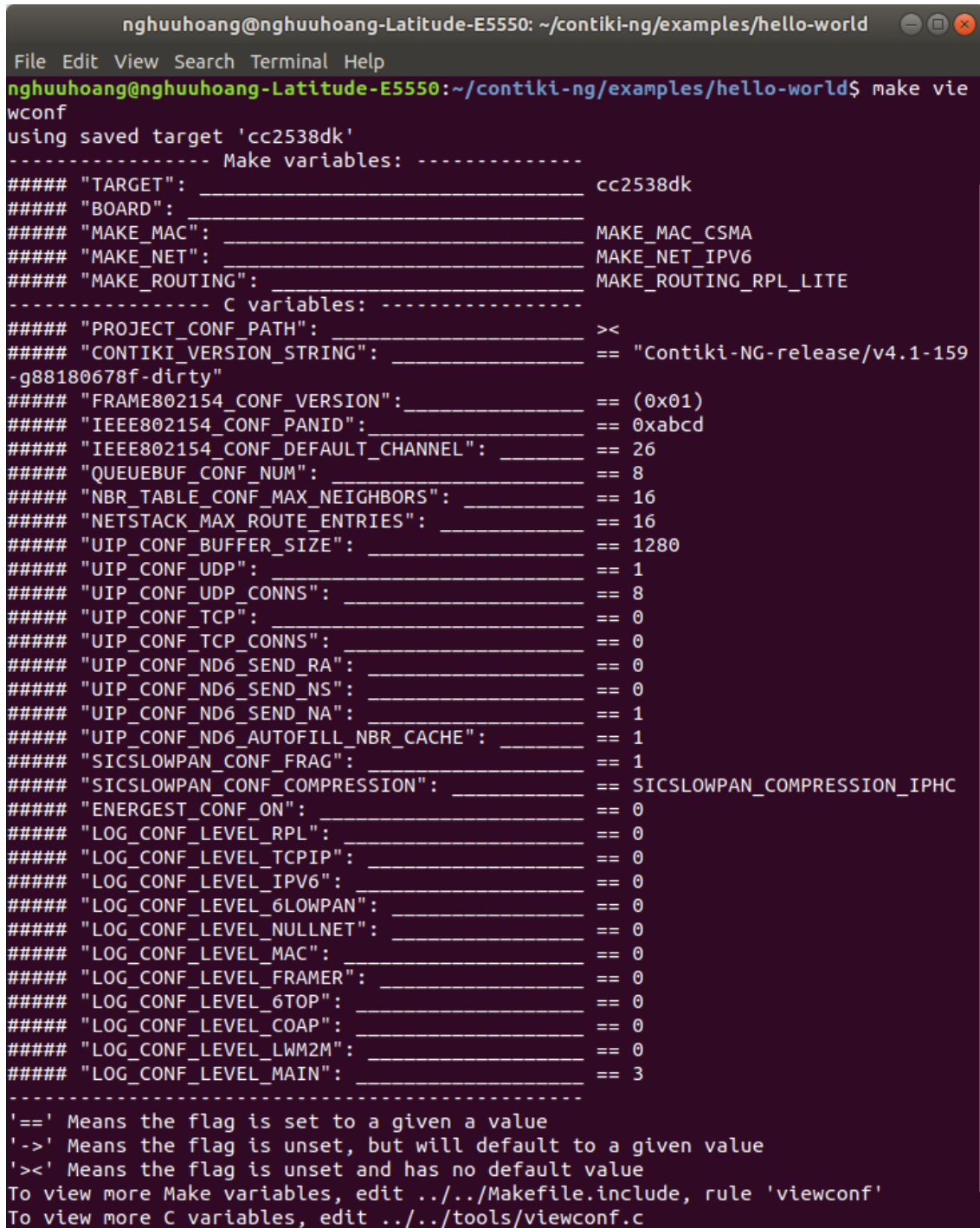
#### Node hello-world:

Build và upload code lên cc2538dk bằng lệnh sau:

**make TARGET=cc2538dk hello-world.upload**

Theo dõi thông tin cấu hình bằng lệnh:

**make viewconf**



```

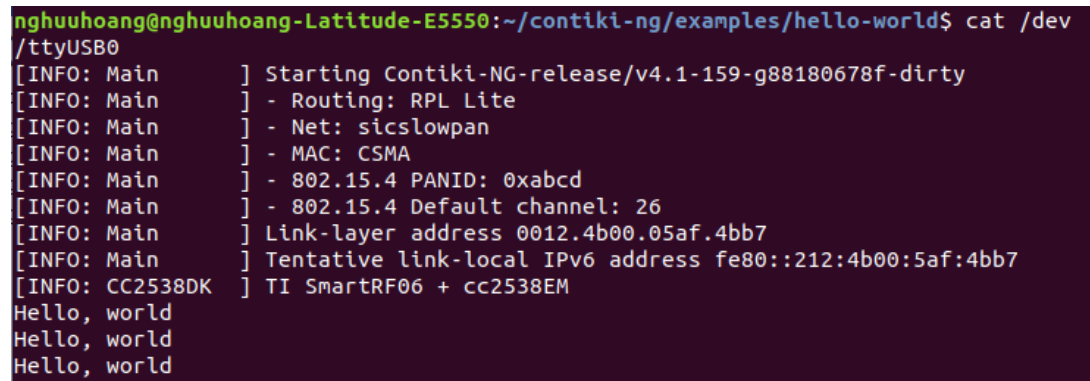
ngghuuhoang@ngghuuhoang-Latitude-E5550: ~/contiki-ng/examples/hello-world
File Edit View Search Terminal Help
ngghuuhoang@ngghuuhoang-Latitude-E5550:~/contiki-ng/examples/hello-world$ make viewconf
using saved target 'cc2538dk'
----- Make variables: -----
##### "TARGET": _____ cc2538dk
##### "BOARD": _____
##### "MAKE_MAC": _____ MAKE_MAC_CSMA
##### "MAKE_NET": _____ MAKE_NET_IPV6
##### "MAKE_ROUTING": _____ MAKE_ROUTING_RPL_LITE
----- C variables: -----
##### "PROJECT_CONF_PATH": _____ ><
##### "CONTIKI_VERSION_STRING": _____ == "Contiki-NG-release/v4.1-159-g88180678f-dirty"
##### "FRAME802154_CONF_VERSION": _____ == (0x01)
##### "IEEE802154_CONF_PANID": _____ == 0xabcd
##### "IEEE802154_CONF_DEFAULT_CHANNEL": _____ == 26
##### "QUEUEBUF_CONF_NUM": _____ == 8
##### "NBR_TABLE_CONF_MAX_NEIGHBORS": _____ == 16
##### "NETSTACK_MAX_ROUTE_ENTRIES": _____ == 16
##### "UIP_CONF_BUFFER_SIZE": _____ == 1280
##### "UIP_CONF_UDP": _____ == 1
##### "UIP_CONF_UDP_CONNS": _____ == 8
##### "UIP_CONF_TCP": _____ == 0
##### "UIP_CONF_TCP_CONNS": _____ == 0
##### "UIP_CONF_ND6_SEND_RA": _____ == 0
##### "UIP_CONF_ND6_SEND_NS": _____ == 0
##### "UIP_CONF_ND6_SEND_NA": _____ == 1
##### "UIP_CONF_ND6_AUTOFILL_NBR_CACHE": _____ == 1
##### "SICSLOWPAN_CONF_FRAG": _____ == 1
##### "SICSLOWPAN_CONF_COMPRESSION": _____ == SICSLOWPAN_COMPRESSION_IPHC
##### "ENERGEST_CONF_ON": _____ == 0
##### "LOG_CONF_LEVEL_RPL": _____ == 0
##### "LOG_CONF_LEVEL_TCPIP": _____ == 0
##### "LOG_CONF_LEVEL_IPV6": _____ == 0
##### "LOG_CONF_LEVEL_6LOWPAN": _____ == 0
##### "LOG_CONF_LEVEL_NULLNET": _____ == 0
##### "LOG_CONF_LEVEL_MAC": _____ == 0
##### "LOG_CONF_LEVEL_FRAMER": _____ == 0
##### "LOG_CONF_LEVEL_6TOP": _____ == 0
##### "LOG_CONF_LEVEL_COAP": _____ == 0
##### "LOG_CONF_LEVEL_LWM2M": _____ == 0
##### "LOG_CONF_LEVEL_MAIN": _____ == 3
-----
'==' Means the flag is set to a given a value
'->' Means the flag is unset, but will default to a given value
'><' Means the flag is unset and has no default value
To view more Make variables, edit ../../Makefile.include, rule 'viewconf'
To view more C variables, edit ../../tools/viewconf.c

```

Hình 17: Kiểm tra cấu hình

Theo cấu hình trên thì chương trình hello-world đang chạy RPL-Lite, Mac chạy CSMA, lớp Network chạy IPv6, PANID là 0xabcd, kênh truyền RF là 26, ...

Giao tiếp UART:



```
nguuhoang@nguuhoang-Latitude-E5550:~/contiki-ng/examples/hello-world$ cat /dev
/ttyUSB0
[INFO: Main      ] Starting Contiki-NG-release/v4.1-159-g88180678f-dirty
[INFO: Main      ] - Routing: RPL Lite
[INFO: Main      ] - Net: sicslowpan
[INFO: Main      ] - MAC: CSMA
[INFO: Main      ] - 802.15.4 PANID: 0xabcd
[INFO: Main      ] - 802.15.4 Default channel: 26
[INFO: Main      ] Link-layer address 0012.4b00.05af.4bb7
[INFO: Main      ] Tentative link-local IPv6 address fe80::212:4b00:5af:4bb7
[INFO: CC2538DK  ] TI SmartRF06 + cc2538EM
Hello, world
Hello, world
Hello, world
```

Hình 18: Hello World Terminal

Địa chỉ IPv6 là: **fd00::212:4b00:5af:4bb7**

#### Node RPL Border-router

Build và upload code lên cc2538dk bằng lệnh sau:

**make TARGET=cc2538dk rpl-border-router.upload**

Sử dụng tunslip cho việc kết nối border-router với network ( network ở đây là local host) thông qua USB-Uart.

Sử dụng lệnh: **make TARGET=cc2538dk connect-router**

```

nguuhoang@nguuhoang-Latitude-E5550: ~/contiki-ng/examples/rpl-border-router
File Edit View Search Terminal Help
nguuhoang@nguuhoang-Latitude-E5550:~/contiki-ng/examples/rpl-border-router$ make TARGET=cc2538dk connect-router
sudo ../../tools/tunslip6 fd00::1/64
[sudo] password for nguuhoang:
*****SLIP started on ``/dev/ttyUSB0''
opened tun device ``/dev/tun0''
ifconfig tun0 inet `hostname` mtu 1500 up
ifconfig tun0 add fd00::1/64
ifconfig tun0 add fe80::0:0:0:1/64
ifconfig tun0

tun0: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST> mtu 1500
    inet 127.0.1.1 netmask 255.255.255.255 destination 127.0.1.1
    inet6 fe80::1 prefixlen 64 scopeid 0x20<link>
    inet6 fe80::3631:b07d:f5c9:6de8 prefixlen 64 scopeid 0x20<link>
    inet6 fd00::1 prefixlen 64 scopeid 0x0<global>
    unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00 txqueuelen 500
(UNSPEC)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

[INFO: Main      ] Starting Contiki-NG-release/v4.1-159-g88180678f-dirty
[INFO: Main      ] - Routing: RPL Lite
[INFO: Main      ] - Net: sicslowpan
[INFO: Main      ] - MAC: CSMA
[INFO: Main      ] - 802.15.4 PANID: 0xabcd
[INFO: Main      ] - 802.15.4 Default channel: 26
[INFO: Main      ] Link-layer address 0012.4b00.05af.4ba7
[INFO: Main      ] Tentative link-local IPv6 address fe80::212:4b00:5af:4ba7
[INFO: CC2538DK  ] TI SmartRF06 + cc2538EM
[INFO: RPL BR    ] Contiki-NG Border Router started
[INFO: BR        ] RPL-Border router started
*** Address:fd00::1 => fd00:0000:0000:0000
[INFO: BR        ] Waiting for prefix
[INFO: BR        ] Server IPv6 addresses:
[INFO: BR        ] fd00::212:4b00:5af:4ba7
[INFO: BR        ] fe80::212:4b00:5af:4ba7

```

Hình 19: Kết nối border router qua Tunslip6

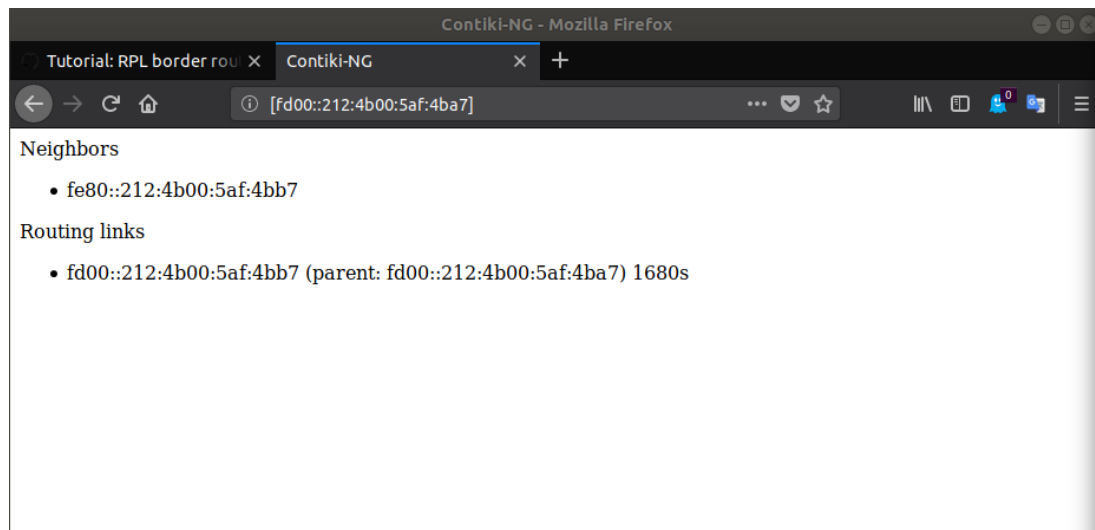
Địa chỉ IPv6 là: **fd00::212:4b00:5af:4ba7**

Ping node chạy hello-world trên terminal với địa chỉ : **fd00::212:4b00:5af:4bb7**

```
nguuhoang@nguuhoang-Latitude-E5550:~$ ping6 fd00::212:4b00:5af:4bb7
PING fd00::212:4b00:5af:4bb7(fd00::212:4b00:5af:4bb7) 56 data bytes
64 bytes from fd00::212:4b00:5af:4bb7: icmp_seq=1 ttl=63 time=55.0 ms
64 bytes from fd00::212:4b00:5af:4bb7: icmp_seq=2 ttl=63 time=53.7 ms
64 bytes from fd00::212:4b00:5af:4bb7: icmp_seq=3 ttl=63 time=114 ms
64 bytes from fd00::212:4b00:5af:4bb7: icmp_seq=4 ttl=63 time=89.8 ms
64 bytes from fd00::212:4b00:5af:4bb7: icmp_seq=5 ttl=63 time=97.3 ms
64 bytes from fd00::212:4b00:5af:4bb7: icmp_seq=6 ttl=63 time=64.9 ms
64 bytes from fd00::212:4b00:5af:4bb7: icmp_seq=7 ttl=63 time=71.8 ms
64 bytes from fd00::212:4b00:5af:4bb7: icmp_seq=8 ttl=63 time=63.6 ms
64 bytes from fd00::212:4b00:5af:4bb7: icmp_seq=9 ttl=63 time=45.8 ms
64 bytes from fd00::212:4b00:5af:4bb7: icmp_seq=10 ttl=63 time=82.7 ms
^C
--- fd00::212:4b00:5af:4bb7 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9011ms
rtt min/avg/max/mdev = 45.843/74.011/114.977/20.761 ms
```

Hình 20: Ping Terminal

Trên trình duyệt nhập địa chỉ border-router vào url:



Hình 21: Kiểm tra các node trong RPL trên trình duyệt

Một điểm nâng cấp trong Contiki-NG là có thể chạy rpl-border-router ở file bất kì với ở đó có file Makefile thêm vào:

**MODULES += os/services/rpl-border-router**

Thực hiện sau trên hello-world:



```

ngghuuhoang@ngghuuhoang-Latitude-E5550:~/contiki-ng/examples/hello-world$ make TAR
GET=cc2538dk connect-router
sudo ../../tools/tunslip6 fd00::1/64
[sudo] password for ngghuuhoang:
*****SLIP started on ``/dev/ttyUSB0''
opened tun device ``/dev/tun0''
ifconfig tun0 inet 'hostname' mtu 1500 up
ifconfig tun0 add fd00::1/64
ifconfig tun0 add fe80::0:0:0:0:1/64
ifconfig tun0

tun0: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST> mtu 1500
    inet 127.0.1.1 netmask 255.255.255.255 destination 127.0.1.1
    inet6 fe80::1 prefixlen 64 scopeid 0x20<link>
    inet6 fe80::3745:1b4:2470:1ad9 prefixlen 64 scopeid 0x20<link>
    inet6 fd00::1 prefixlen 64 scopeid 0x0<global>
    unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00 txqueuelen 500
(UNSPEC)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

[INFO: BR          ] Waiting for prefix
*** Address:fd00::1 => fd00:0000:0000:0000
Hello, world
[INFO: BR          ] Waiting for prefix
[INFO: BR          ] Server IPv6 addresses:
[INFO: BR          ]   fd00::212:4b00:5af:4bb7
[INFO: BR          ]   fe80::212:4b00:5af:4bb7
Hello, world
Hello, world
Hello, world
Hello, world
Hello, world
Hello, world
Hello, world
Hello, world

```

Hình 22: Border Router từ Hello-World

## 5.5 TSCH and 6TiSCH

Các tính năng TSCH và 6TiSCH trên Contiki-NG:

- Standard TSCH joining procedure with Enhanced Beacons with the following Information Elements:
  - TSCH synchronization (join priority and ASN)
  - TSCH slotframe and link (basic schedule)
  - TSCH timeslot (timeslot timing template)
  - TSCH channel hopping sequence (hopping sequence template)
- Standard TSCH link selection and slot operation (10ms slots by default)
- Standard TSCH synchronization, including with ACK/NACK time correction Information Element
- Standard TSCH queues and CSMA-CA mechanism
- Standard TSCH and 6TiSCH security
- Standard 6TiSCH TSCH-RPL interaction (6TiSCH Minimal Configuration and Minimal Schedule)
- A scheduling API to add/remove slotframes and links
- A system for logging from TSCH timeslot operation interrupt, with postponed printout



- Orchestra: an autonomous scheduler for TSCH+RPL networks
- A drift compensation mechanism

Sử dụng TSCH trên cc2538dk

Để cho phép TSCH, thêm dòng lệnh sau vào Makefile trong project:

```
MAKE_MAC = MAKE_MAC_TSCH
```

Build và kiểm tra cấu hình bằng lệnh:

```
make TARGET=cc2538dk hello-world.upload
```

```
make viewconf
```

```
##### "TSCH_CONF_DEFAULT_HOPPING_SEQUENCE": ____ -> (uint8_t[]){ 15, 25, 26, 20 }
##### "TSCH_CONF_JOIN_HOPPING_SEQUENCE": ____ -> (uint8_t[]){ 15, 25, 26, 20 }
##### "TSCH_CONF_EB_PERIOD": ____ -> (16 * 128)
##### "TSCH_CONF_MAX_EB_PERIOD": ____ -> (16 * 128)
##### "TSCH_CONF_DEFAULT_TIMESLOT_LENGTH": ____ == 10000
##### "TSCH_SCHEDULE_CONF_DEFAULT_LENGTH": ____ -> 7
```

Hình 23: Xem cấu hình TSCH

Mặc định IPv6 và RPL sẽ được chạy, nên 6TiSCH cũng sẽ được setup, nó sẽ giúp cho việc tương tác giữa TSCH và IPv6, với độ dài slotframe là

```
TSCH_SCHEDULE_CONF_DEFAULT_LENGTH = 7
```

## 5.6 CoAP

Thực hiện CoAP trên CC2538, sử dụng 2 node, một node coap-server, một node coap-client, client thực hiện gửi yêu cầu cho server thực hiện đảo LED trên board sau mỗi 10 giây. CoAP server sẽ gọi file res\_toggle trong thư mục resource ở contiki-ng/examples/coap/coap-example-server. CoAP client cần có địa chỉ của CoAP server, trong trường hợp này là địa chỉ link-local với prefix là fe80.

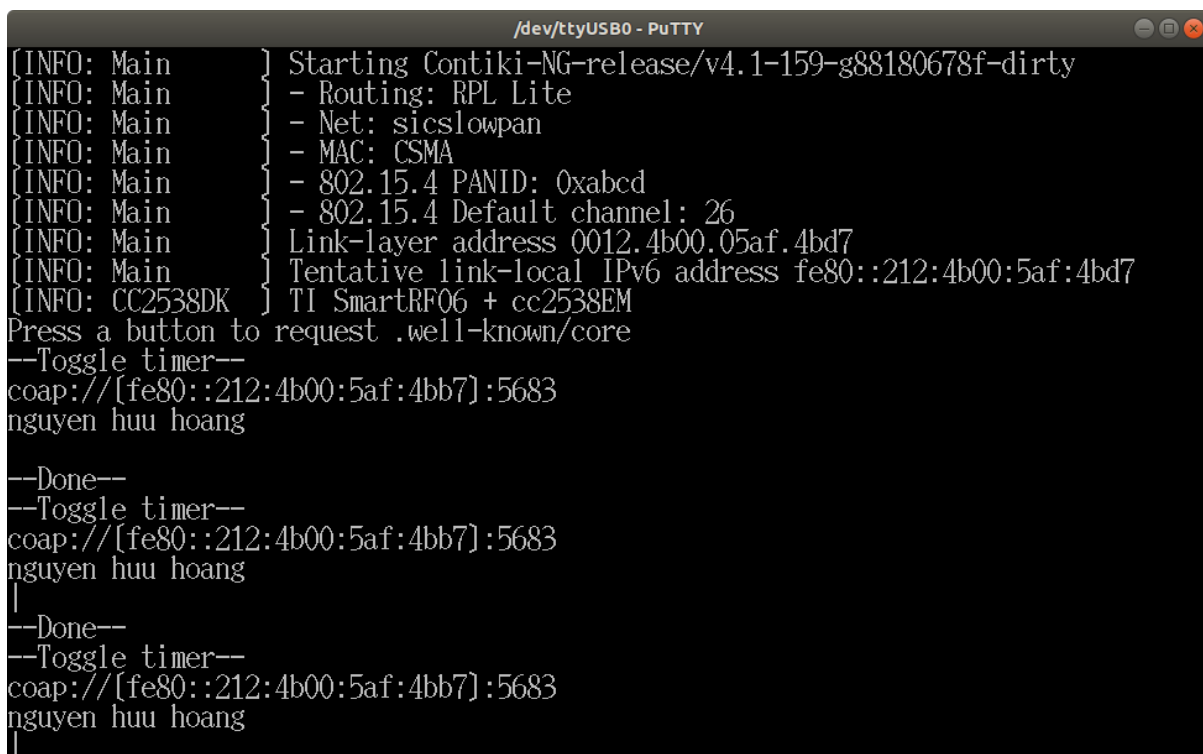
Tại server:



```
/dev/ttyUSB0 - PuTTY
[INFO: Main ] Starting Contiki-NG-release/v4.1-159-g88180678f-dirty
[INFO: Main ] - Routing: RPL Lite
[INFO: Main ] - Net: sicslowpan
[INFO: Main ] - MAC: CSMA
[INFO: Main ] - 802.15.4 PANID: 0xabcd
[INFO: Main ] - 802.15.4 Default channel: 26
[INFO: Main ] Link-layer address 0012.4b00.05af.4bb7
[INFO: Main ] Tentative link-local IPv6 address fe80::212:4b00:5af:4bb7
[INFO: CC2538DK ] TI SmartRF06 + cc2538EM
[INFO: App ] Starting Erbium Example Server
```

Hình 24: CoAP Server

Tại Client:



```
/dev/ttyUSB0 - PuTTY
[INFO: Main ] Starting Contiki-NG-release/v4.1-159-g88180678f-dirty
[INFO: Main ] - Routing: RPL Lite
[INFO: Main ] - Net: sicslowpan
[INFO: Main ] - MAC: CSMA
[INFO: Main ] - 802.15.4 PANID: 0xabcd
[INFO: Main ] - 802.15.4 Default channel: 26
[INFO: Main ] Link-layer address 0012.4b00.05af.4bd7
[INFO: Main ] Tentative link-local IPv6 address fe80::212:4b00:5af:4bd7
[INFO: CC2538DK ] TI SmartRF06 + cc2538EM
Press a button to request .well-known/core
--Toggle timer--
coap://[fe80::212:4b00:5af:4bb7]:5683
nguyen huu hoang
--Done--
--Toggle timer--
coap://[fe80::212:4b00:5af:4bb7]:5683
nguyen huu hoang
|
--Done--
--Toggle timer--
coap://[fe80::212:4b00:5af:4bb7]:5683
nguyen huu hoang
|
```

Hình 25: CoAP Client

Chạy border-router và CoAP server:

Tại border-router:

```
[INFO: Main      ] Starting Contiki-NG-release/v4.1-159-g88180678f-dirty
[INFO: Main      ] - Routing: RPL Lite
[INFO: Main      ] - Net: sicslowpan
[INFO: Main      ] - MAC: CSMA
[INFO: Main      ] - 802.15.4 PANID: 0xabcd
[INFO: Main      ] - 802.15.4 Default channel: 26
[INFO: Main      ] Link-layer address 0012.4b00.05af.4bd7
[INFO: Main      ] Tentative link-local IPv6 address fe80::212:4b00:5af:4bd7
[INFO: CC2538DK  ] TI SmartRF06 + cc2538EM
[INFO: RPL BR    ] Contiki-NG Border Router started
[INFO: BR        ] RPL-Border router started
*** Address:fd00::1 => fd00:0000:0000:0000
[INFO: BR        ] Waiting for prefix
[INFO: BR        ] Server IPv6 addresses:
[INFO: BR        ]     fd00::212:4b00:5af:4bd7
[INFO: BR        ]     fe80::212:4b00:5af:4bd7
```

Hình 26: Chạy Border Router

Ping node CoAP Server:

```
nguuhoang@nguuhoang-Latitude-E5550:~/contiki-ng/examples/rpl-border-router$ ping6 fd00
::212:4b00:5af:4bb7
PING fd00::212:4b00:5af:4bb7(fd00::212:4b00:5af:4bb7) 56 data bytes
64 bytes from fd00::212:4b00:5af:4bb7: icmp_seq=17 ttl=63 time=89.1 ms
64 bytes from fd00::212:4b00:5af:4bb7: icmp_seq=18 ttl=63 time=103 ms
64 bytes from fd00::212:4b00:5af:4bb7: icmp_seq=19 ttl=63 time=71.0 ms
64 bytes from fd00::212:4b00:5af:4bb7: icmp_seq=20 ttl=63 time=62.0 ms
64 bytes from fd00::212:4b00:5af:4bb7: icmp_seq=21 ttl=63 time=43.6 ms
64 bytes from fd00::212:4b00:5af:4bb7: icmp_seq=22 ttl=63 time=67.0 ms
64 bytes from fd00::212:4b00:5af:4bb7: icmp_seq=23 ttl=63 time=97.4 ms
64 bytes from fd00::212:4b00:5af:4bb7: icmp_seq=24 ttl=63 time=64.8 ms
64 bytes from fd00::212:4b00:5af:4bb7: icmp_seq=25 ttl=63 time=86.0 ms
64 bytes from fd00::212:4b00:5af:4bb7: icmp_seq=26 ttl=63 time=38.8 ms
64 bytes from fd00::212:4b00:5af:4bb7: icmp_seq=27 ttl=63 time=67.8 ms
64 bytes from fd00::212:4b00:5af:4bb7: icmp_seq=28 ttl=63 time=71.7 ms
64 bytes from fd00::212:4b00:5af:4bb7: icmp_seq=29 ttl=63 time=111 ms
64 bytes from fd00::212:4b00:5af:4bb7: icmp_seq=30 ttl=63 time=71.0 ms
64 bytes from fd00::212:4b00:5af:4bb7: icmp_seq=31 ttl=63 time=68.7 ms
64 bytes from fd00::212:4b00:5af:4bb7: icmp_seq=32 ttl=63 time=91.0 ms
64 bytes from fd00::212:4b00:5af:4bb7: icmp_seq=33 ttl=63 time=81.7 ms
64 bytes from fd00::212:4b00:5af:4bb7: icmp_seq=34 ttl=63 time=45.7 ms
64 bytes from fd00::212:4b00:5af:4bb7: icmp_seq=35 ttl=63 time=100 ms
^C
--- fd00::212:4b00:5af:4bb7 ping statistics ---
35 packets transmitted, 19 received, 45% packet loss, time 34395ms
rtt min/avg/max/mdev = 38.821/75.498/111.755/19.989 ms
```

Hình 27: Ping Node CoAP Server

Yêu cầu thông tin của CoAP server với địa chỉ của nó:

```
nguuhoang@nguuhoang-Latitude-E5550:~/contiki-ng/examples/rpl-border-router$ coap get c
oap://[fd00::212:4b00:5af:4bb7]/.well-known/core
(2.05) </.well-known/core>;ct=40,</test/hello>;title="Hello world: ?len=0..";rt="Text",
</debug/mirror>;title="Returns your decoded message";rt="Debug",</test/chunks>;title="Bl
ockwise demo";rt="Data",</test/separate>;title="Separate demo",</test/push>;title="Perio
dic demo";obs,</sensors/button>;title="Event demo";obs,</test/sub>;title="Sub-resource d
emo",</test/b1sepb2>;title="Block1 + Separate + Block2 demo",</actuators/toggle>;title="
Red LED";rt="Control"
```

Hình 28: Get thông tin từ CoAP Server

Get đoạn text hello:

```
nguuhoang@nguuhoang-Latitude-E5550:~/contiki-ng/examples/rpl-border-router$ coap get c
oap://[fd00::212:4b00:5af:4bb7]:5683/test/hello
(2.05) Hello World!
```

Hình 29: Get đoạn Text trong test/hello-world

## 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

### 6.1 Kết luận

#### Phần cứng:

##### Ưu điểm:

- Đáp ứng được yêu cầu mạch ban đầu, nạp code ổn, sạc Pin ổn, RF ổn, các chức năng của chip hoạt động bình thường.
- Tương đối đẹp.

##### Nhược điểm:

- Micro USB khá nhỏ, dễ gãy khi rút ra vào nhiều.
- Có một số thành phần không cần thiết.
- Thời gian để sạc PIN là khá lâu, khoảng một ngày cho viên Pin 1300 mAh.
- Khoảng cách giao tiếp RF còn hạn chế đối với Antena trên mạch.

#### Phần mềm:

##### Ưu điểm:

- Đã chạy được trên phần cứng dựa trên chip cc2538 được thiết kế.
- Contiki-NG là một phiên bản tốt, có nhiều cải tiến cả nội dung lẫn hình thức, khắc phục được sự phân mảnh của phiên bản cũ Contiki 3.0 về trước.
- Khả năng khai triển ứng dụng dựa trên Contiki-NG là nhanh chóng.
- Phát triển cho nhiều nền tảng phần cứng mới.

##### Nhược điểm:

- Do Contiki-NG 4.1 mới được phát hành và đang được phát triển nên còn có một số lỗi như shell đã trình bày.
- Các nền tảng phần cứng cũ Contiki-NG không còn hỗ trợ.
- Khâu cài đặt thiết lập cho Contiki-NG còn mất nhiều thời gian.
- Đề cương chưa có mô phỏng Cooja cho hệ thống.

## 6.2 Hướng phát triển

### Phần cứng:

- Phát triển phần cứng phiên bản tiếp theo, khắc phục và thêm các tính năng cần thiết, rút gọn các thành phần không cần thiết.
- Phát triển phiên bản phần cứng tiếp theo cho Luận văn tốt nghiệp sau này.

### Phần mềm:

- Đề tài cần phát triển đề tài tới các ứng dụng cụ thể cho Luận văn sau này.
- Đưa ra được hệ thống tổng quát, từ thu thập dữ liệu tới giao diện người dùng cụ thể.
- Mô phỏng mạng để đánh giá mạng thực quan, nhiều thông số.

## 7. TÀI LIỆU THAM KHẢO

- [1] "Contiki-NG Wiki".[Online].<https://github.com/contiki-ng/contiki-ng/wiki>, May 2018.
- [2] Dr. -Ing Vo Que Son, 2017, "Slide Wireless Sensor Networks" - Department of Telecommunication Engineering, Faculty of electrical & Electronics engineering.
- [3] D. Simon."TSCH and 6TiSCH for Contiki: Challenges, Design and Evaluation",IEEE DCOSS'17, 2017.
- [4] "Data sheet SoC CC2538 TI".[Online]. <http://www.ti.com/product/cc2538>.
- [5]"2.4-GHz Inverted F Antenna",[Document], Texas Instruments, April 2007 – Revised February 2017.