

LAB 16: SPI GIAO TIẾP ETHERNET ENC424J600

1. Mục tiêu – yêu cầu

Trong bài thực hành này ta sẽ giúp người học đạt được những kiến thức sau:

- Hiểu cách cấu hình và sử dụng bộ SPI của SG8V1 .
- Hiểu chuẩn giao tiếp Ethernet.
- Biết cách giao tiếp với chip Ethernet ENC424J600
- Biết cách truyền nhận dữ liệu qua mạng Ethernet

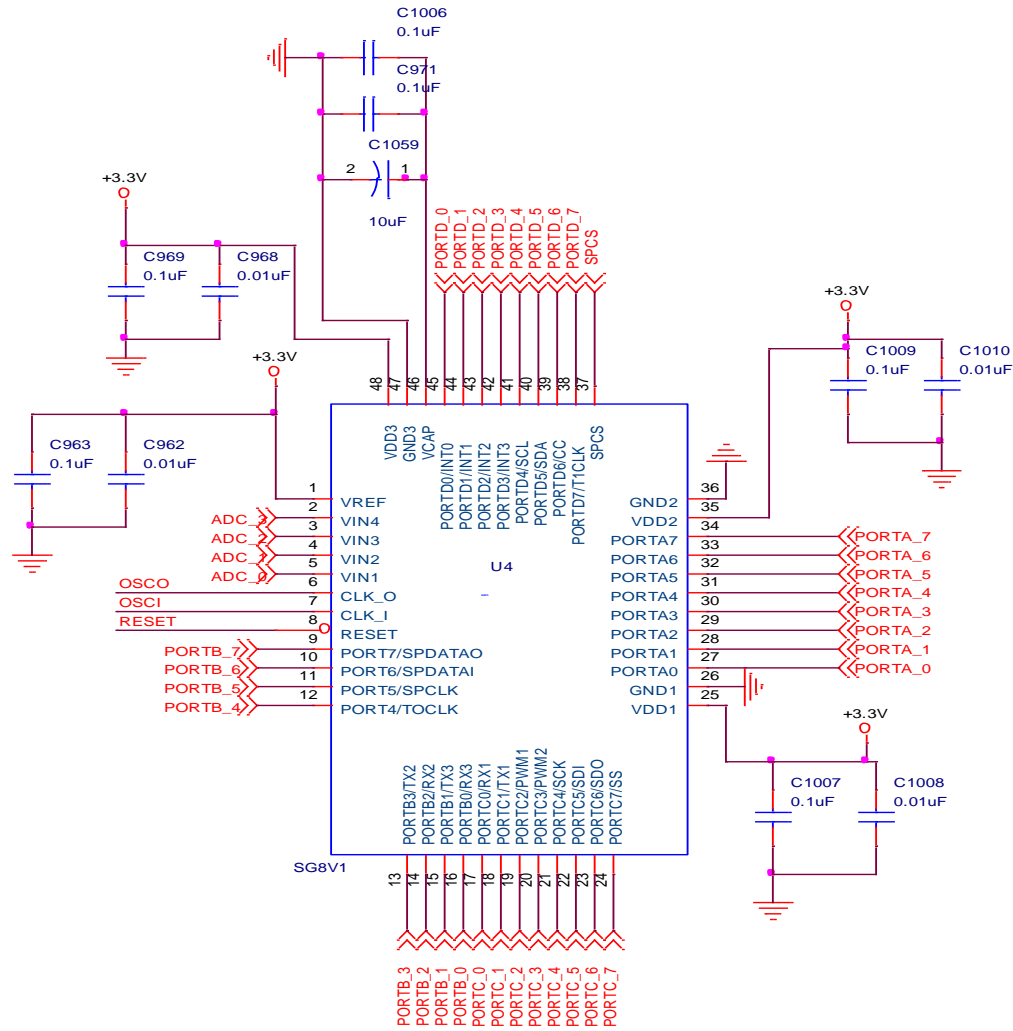
2. Tổng quan

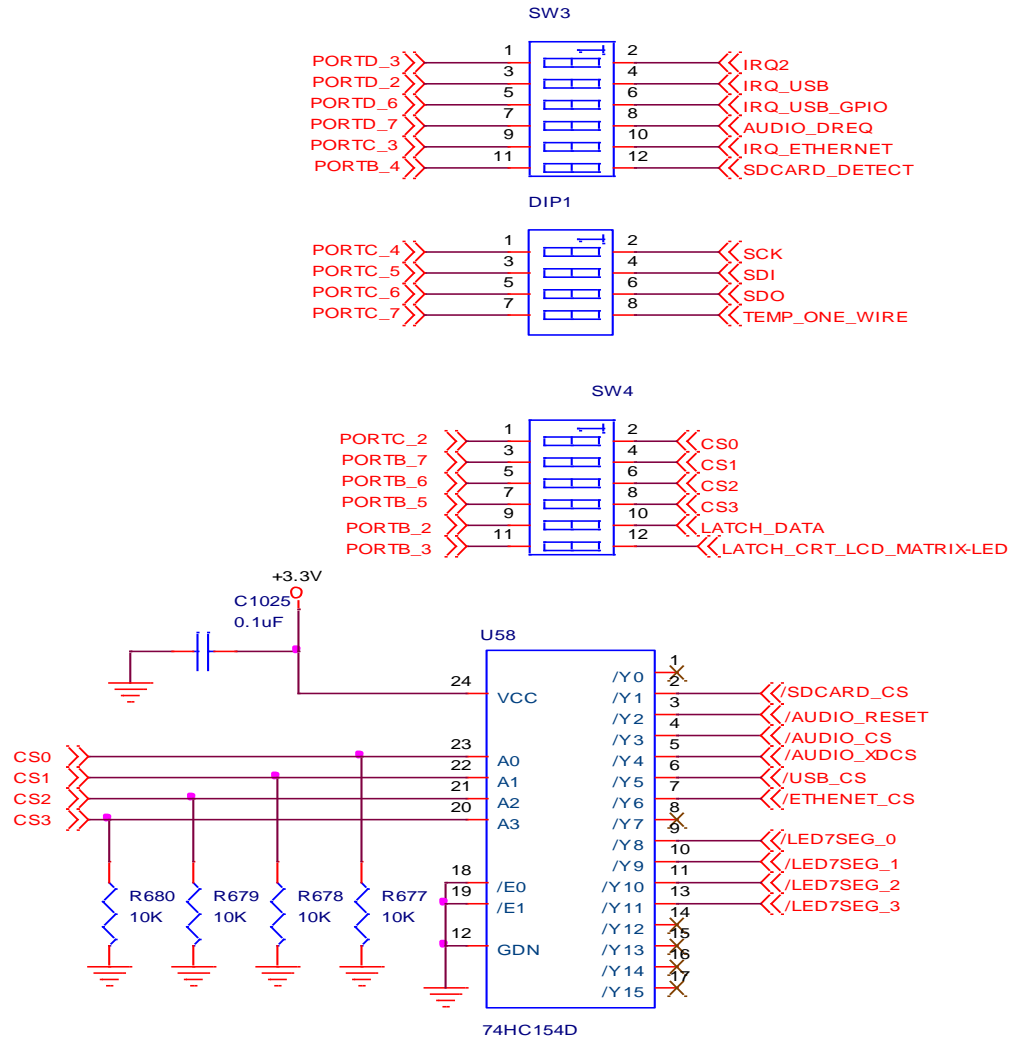
Giao tiếp giữa SG8V1 và IC ENC424J600 thông qua chuẩn SPI gồm có các tín hiệu sau:

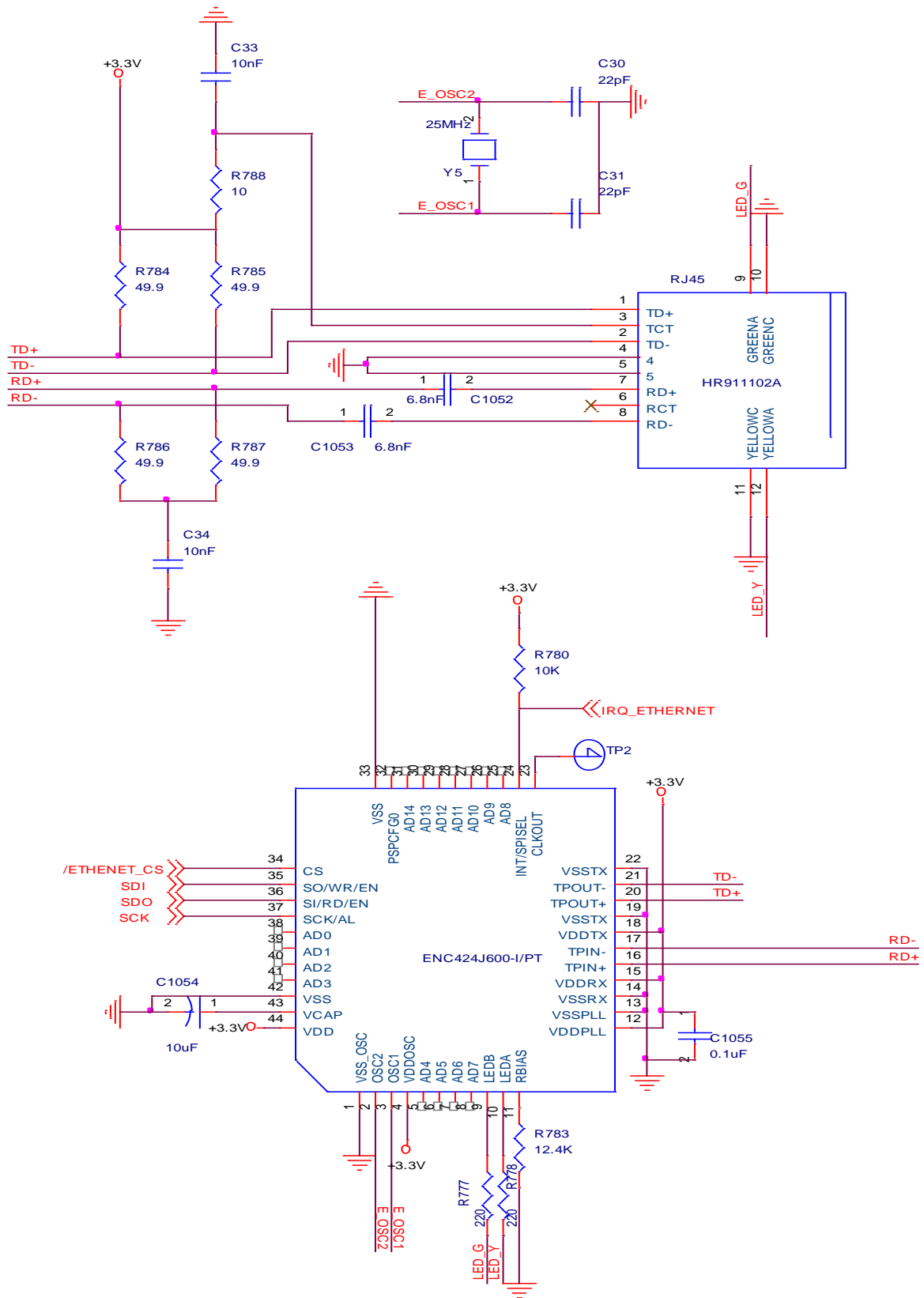
- Ngõ ra nối tiếp (MOSI) – PortC[6] /SDO
- Ngõ vào nối tiếp (MISO) – PortC[5] /SDI
- Clock nối tiếp (SCK) – PortC[4] /SCK
- Lựa chọn SPI slave (SS) – GPIOx

Phần mô tả về SPI của SG8V1 vui lòng xem lại trong lab13 hoặc tài liệu SG8V1_HuongDanSuDung.pdf

3. Phần cứng







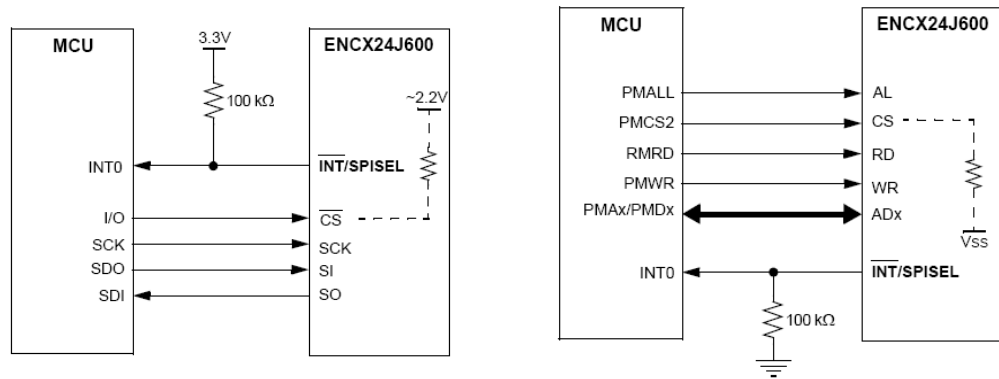
Net	ENC424J600	SG8V1
SDI	MISO	PORTC5 (DIP1 ON)

SCK	SCK	PORTC4 (DIP1 ON)
SDO	MOSI	PORTC6 (DIP1 ON)
USB_CS	SS#	PORTD[4:5] = [0,1] PORTC[2:3] = [1,0]
IRQ_ETHERNET	INT/SPISEL	PORTB5 (SW3 ON)

3.1 Giao tiếp với ENC424J600

- ENC424J600 có chức năng bộ điều khiển Ethernet chuẩn công nghiệp, tốc độ 10/100 Mbps.

- ENC424J600 kết nối với vi điều khiển sử dụng giao diện nối tiếp SPI hoặc song song 8/16bit (PSP). Lựa chọn được kiểu giao tiếp thông qua chân INT/SPISEL.



Giao diện SPI

Giao diện PSP

Hình 1. Giao tiếp SPI giữa ENC424J600 với vi điều khiển

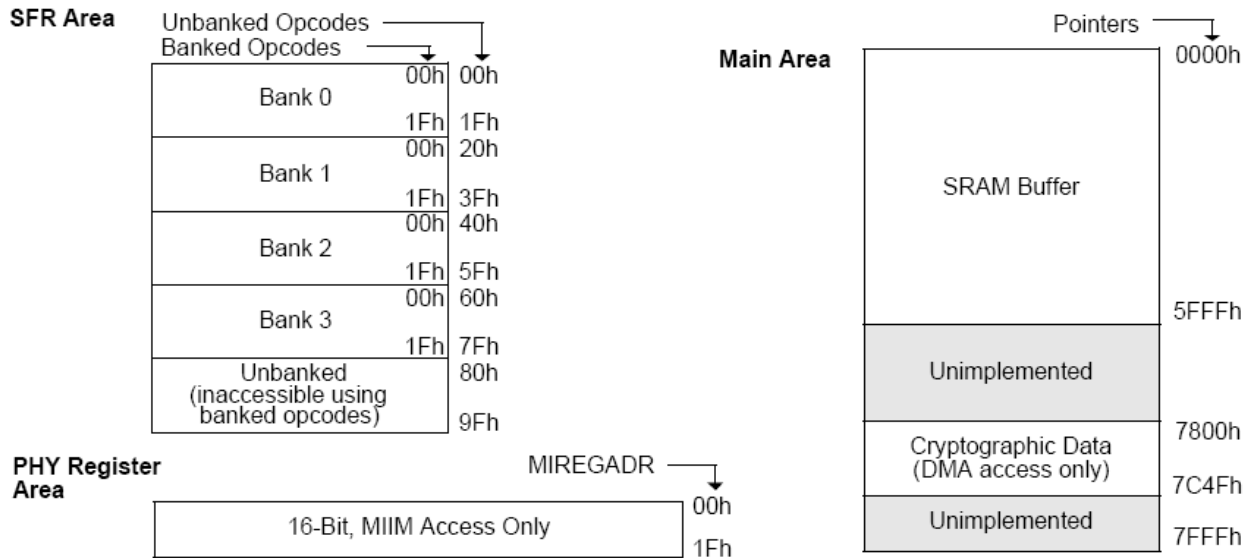
❖ Truy xuất thành phần của ENC424J600

Khí giao tiếp SPI ENC424J600 chia thành 3 không gian địa chỉ thành ghi:

- Vùng thành ghi có chức năng đặc biệt SFR: có địa chỉ tuyếntính, chiều dài 160 byte, có thể truy xuất trực tiếp
- Vùng bộ nhớ chính: có địa chỉ tuyếntính, kích thước 32Kbyte. 24Kbyte đầu tiên có chức năng như

SRAM. Có thể truy xuất gián tiếp bằng cách sử dụng các thành ghi SFR

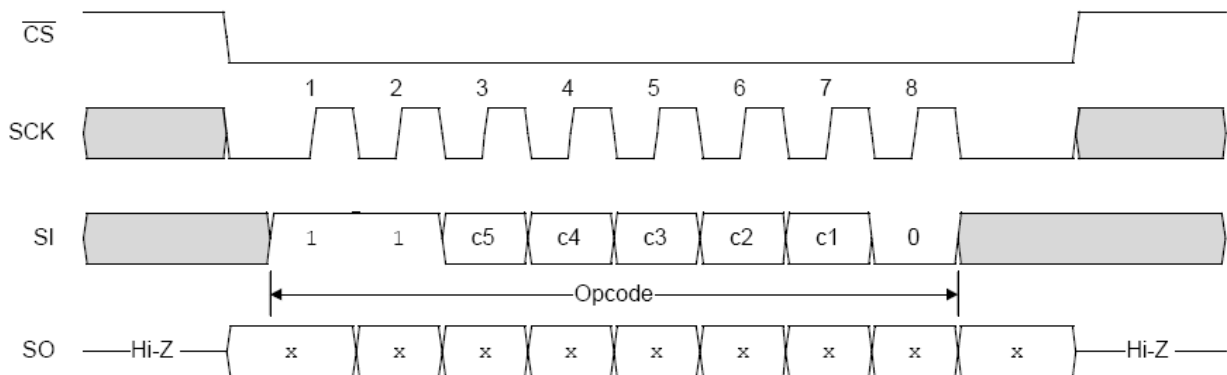
- Vùng thanh ghi PHY: tuyến tính, định địa chỉ word. Có kích thước 32 word. Vùng này chỉ có thể truy xuất thông qua giao diện MIIM



Hình 2. Phân chia vùng bộ nhớ ENC424J600

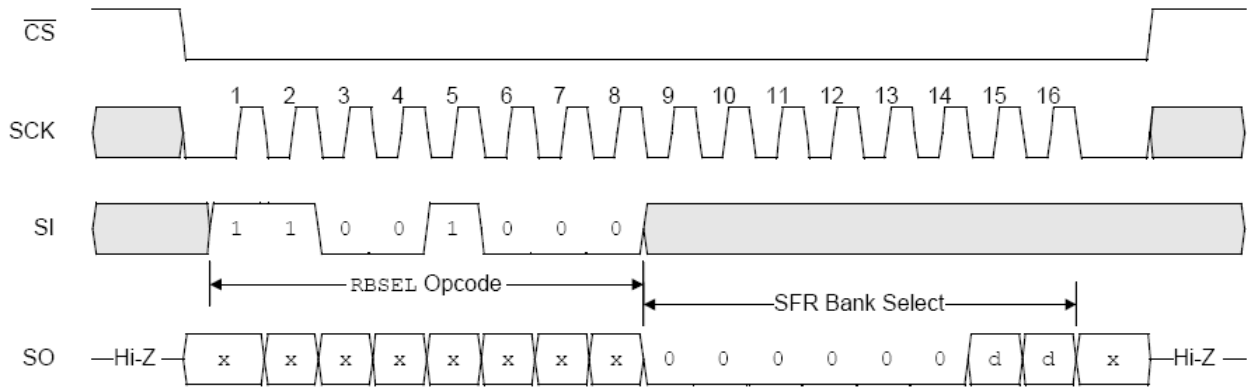
ENC424J600 chia thành 4 loại mã lệnh:

- Một Byte:** mã lệnh trực tiếp; được thiết kế cho các hoạt động như ghi nhệm vụ của các thanh ghi có chức năng đặc biệt-SFR, không có dữ liệu trả về



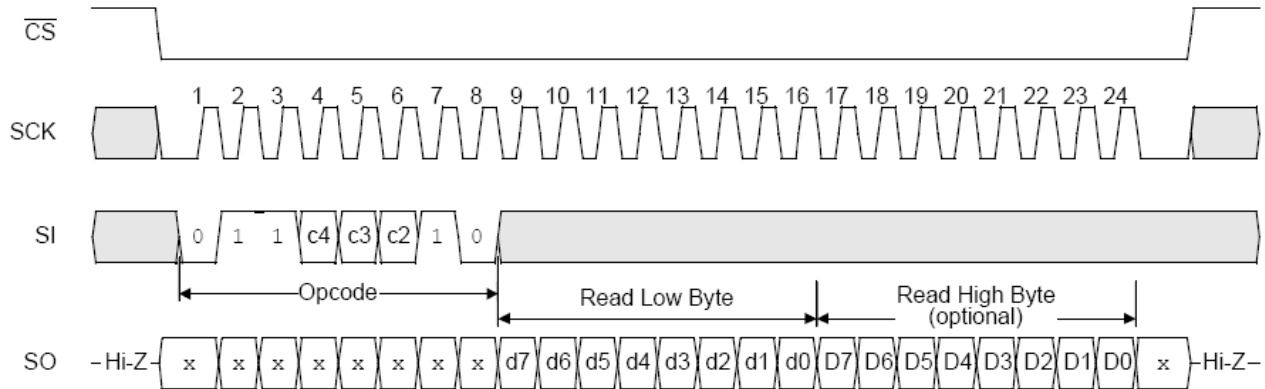
Hình 3. Timing lệnh 1 byte

- Hai-Byte:** mã lệnh trực tiếp; được thiết kế cho các hoạt động của các thanh ghi có chức năng đặc biệt-SFR với byte dữ liệu được trả về

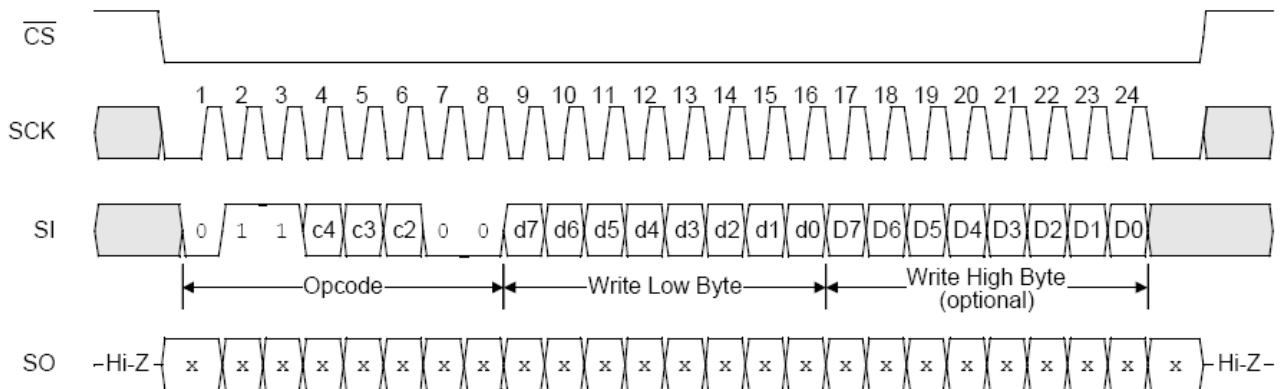


Hình4. Timing lệnh 2 byte

- Ba-Byte: mã lệnh với độ sâu có chiều dài word (2 byte); thực hiện hoạt động đọc và ghi, được thiết kế cho việc trả về dữ liệu là con trỏ với chiều dài word



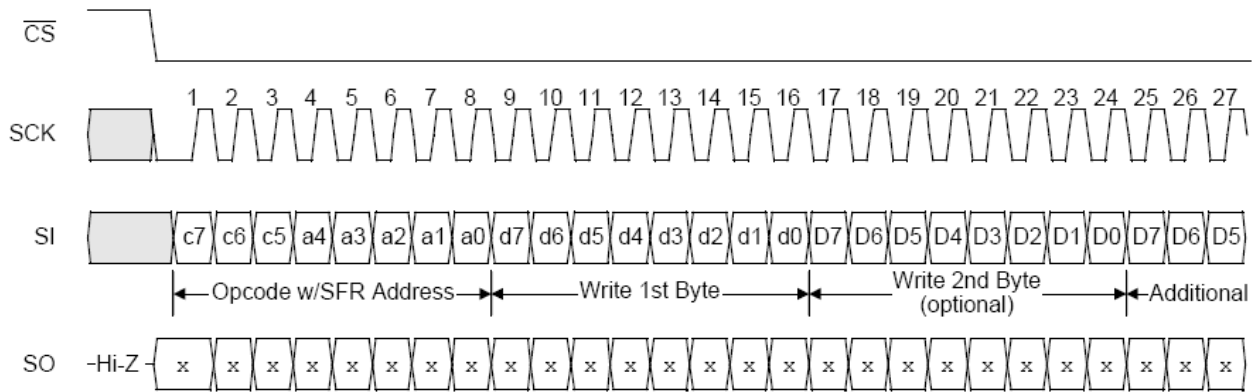
Hình5. Timing đọc lệnh 3 byte



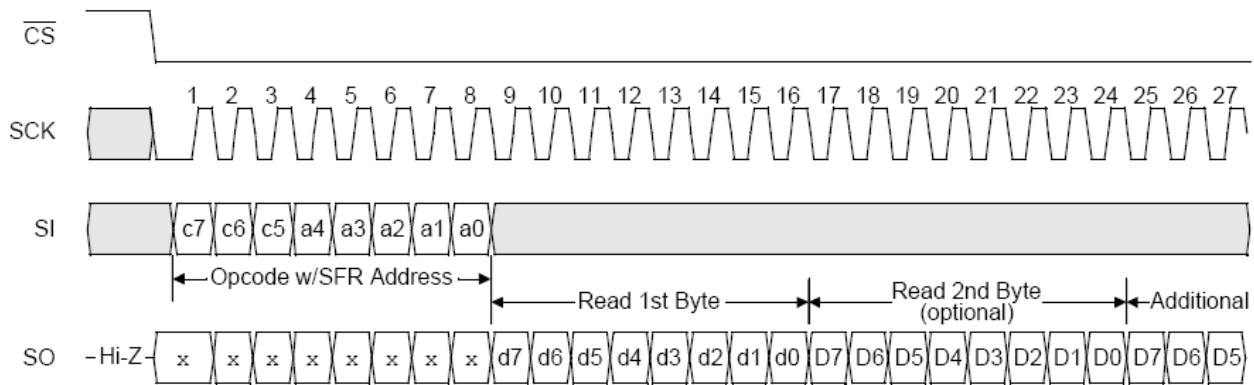
Hình6. Timing ghi lệnh 3 byte

- N-Byte: mã lệnh với đối số là một hoặc nhiều byte; thực hiện việc đọc và ghi được thiết kế cho việc truy xuất không gian bộ nhớ chung, có một hoặc nhiều byte dữ liệu được trả về.

Write Operation



Read Operation



Hình7. Timing đọc/ghi lệnh N byte

4. Phần mềm

4.1 Cách làm thủ việ liên quan

Cách làm SPI liên quan vui lòng tham khảo lab13 hoặc tài liệu

Huong_dan_su_dung_trinh_bien_dich_SG8v1_va_IDE_iFast.pdf

4.2 Cách làm thủ việ truy xuất ENC424J600

❖ Ghi vào thanh ghi

static void enc424j600WriteReg(u16 address, u16 data)

Thông số:

address: địa chỉ thanh ghi

data: giá trị thanh ghi

❖ Đọc thanh ghi

u16 enc424j600ReadReg(u16 address)

Thông số:

address: địa chỉ thanh ghi

Giá trị trả về: giá trị thanh ghi cần đọc

❖ Đọc vào thanh ghi PHY

u16 enc424j600ReadPHYReg(u08 address)

Thông số:

address: địa chỉ thanh ghi

Giá trị trả về: giá trị thanh ghi cần đọc

❖ Ghi vào thanh ghi PHY

void enc424j600WritePHYReg(u08 address, u16 Data)

Thông số:

address: địa chỉ thanh ghi

data: giá trị thanh ghi

❖ Đọc nhiều byte

static void enc424j600ReadN(u08 op, u08* data, u16 dataLen)

Thông số:

op: mã lệnh N byte

data: con trỏ chứa giá trị thanh ghi trả về

dataLen: tổng số byte cần đọc

❖ Ghi nhiều byte

static void enc424j600WriteN(u08 op, u08* data, u16 dataLen)

Thông số:

op: mã lệnh N byte

data: con trỏ chứa giá trị dữ liệu cần ghi

dataLen: tổng số byte cần ghi

❖ Set bit thanh ghi

static void enc424j600BFSReg(u16 address, u16 bitMask)

Thông số:

address: địa chỉ thanh ghi

bitMask: mặt nạ bit

- ❖ Clear bit thanh ghi

static void enc424j600BFCReg(u16 address, u16 bitMask)

Thông số:

address: địa chỉ thanh ghi

bitMask: mặt nạ bit

- ❖ Ghi vào bộ nhớ SRAM

void enc424j600WriteMemoryWindow(u08 window, u08 *data, u16 length)

Thông số:

window: địa chỉ vùng nhớ cần ghi (RX hoặc General Purpose buffer)

data: con trỏ đến bộ đệm cần ghi

length: chiều dài dữ liệu

- ❖ Đọc bộ nhớ SRAM

void enc424j600WriteMemoryWindow(u08 window, u08 *data, u16 length)

Thông số:

window: địa chỉ vùng nhớ cần đọc (RX hoặc General Purpose buffer)

data: con trỏ đến bộ đệm lưu dữ liệu đọc về

length: chiều dài dữ liệu

- ❖ Nhận gói dữ liệu qua mạng

u16 enc424j600PacketReceive(u16 len, u08* packet)

len: chiều dài gói dữ liệu

packet: con trỏ đến bộ đệm lưu dữ liệu đọc về

- ❖ Gửi gói dữ liệu qua mạng

void enc424j600PacketSend(u16 len, u08* packet)

len: chiều dài gói dữ liệu

packet: con trỏ đến bộ đệm chứa dữ liệu cần gửi

- ❖ Reset ENC424J600

static void enc424j600SendSystemReset(void)

- ❖ Khởi tạo ENC424J600

void enc424j600Init(void)

4.3. Yêu cầu

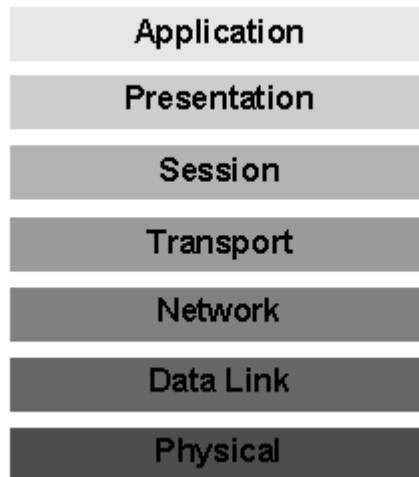
Thực hiện viết chương trình:

- Khởi tạo hoạt động của ENC424j600 và các tiến trình của giao thức ethernet/internet.
- Viết chương trình thực hiện việc lấy IP động cho Kit.
- Viết chương trình thực hiện việc PING tới trang chủ Google mỗi khi phím KEY17 được nhấn.
- Viết chương trình thực hiện việc gửi dữ liệu qua giao thức TCP/IP từ 1 Client đến Server mỗi khi phím KEY18 được nhấn.

4.4. Các khái niệm cơ bản về ethernet/internet

4.4.1. Mô hình OSI

ISO (the International Standards Organization) là một tổ chức được thành lập năm 1971 với mục đích thiết lập các tiêu chuẩn quốc tế. Một trong các chuẩn ISO bao hàm mọi mặt của truyền thông mạng là mô hình OSI - Open Systems Interconnection model (Mô hình liên kết giữa các hệ thống mở). Đây là mô hình cho phép bất cứ hai hệ thống nào (cho dù khác nhau) có thể truyền thông với nhau mà không cần quan tâm đến kiến trúc bên dưới của chúng. Mô hình OSI gồm 7 tầng:



Hình 8 Bảy tầng trong mô hình OSI

Trong 7 tầng của OSI có 3 tầng dưới được thực hiện trên kênh truyền, định nghĩa cách thức thiết lập đầu cuối trên thiết bị phần cứng cho kết nối, 4 tầng

trên được thực hiện trên host, phục vụ cho việc định nghĩa các chuẩn chung phát triển trên ứng dụng, giao tiếp người dùng.

Định nghĩa các tầng của OSI:

Tầng 1 : Physical Layer (tầng vật lý) Điều khiển việc truyền tải các bit trên đường truyền vật lý. Tầng vật lý định nghĩa các tín hiệu điện, trạng thái đường truyền, phương pháp mã hóa dữ liệu, các loại đầu nối được sử dụng ...

Tầng 2 : Data-link Layer (tầng liên kết dữ liệu) Đảm bảo việc truyền các frame (Khung dữ liệu) giữa hai máy tính có đường truyền vật lý nối trực tiếp với nhau. Tầng liên kết dữ liệu hỗ trợ cơ chế phát hiện và xử lý lỗi dữ liệu.

Tầng 3 : Network Layer (tầng mạng) Đảm bảo cho việc truyền packet (gói dữ liệu) giữa hai máy tính bất kỳ trong mạng máy tính (có thể có hoặc không có kết nối đường truyền vật lý trực tiếp). Hay nói cách khác, tầng mạng có nhiệm vụ tìm đường đi cho dữ liệu truyền thông giữa hai máy bất kỳ.

Tầng 4 : Transport Layer (tầng vận chuyển) Tầng vận chuyển làm nhiệm vụ phân nhỏ các gói tin có kích thước lớn khi gửi và tập hợp lại khi nhận, đảm bảo tính toàn vẹn cho dữ liệu (không lỗi, không mất, không lặp, đúng thứ tự).

Tầng 5 : Session Layer (tầng phiên) Quản lý các phiên làm việc giữa các người sử dụng. Tầng phiên cung cấp cơ chế nhận biết tên và chức năng về bảo mật thông tin khi truyền qua mạng máy tính.

Tầng 6 : Presentation Layer (tầng trình bày) Cung cấp định dạng dữ liệu cho ứng dụng. Trong quá trình truyền dữ liệu, trên tầng trình bày bên máy gửi có nhiệm vụ dịch dữ liệu từ định dạng riêng sang định dạng chung và quá trình ngược lại trên tầng trình bày bên máy nhận. Tầng trình bày đảm bảo cho các máy tính có định dạng dữ liệu khác nhau vẫn có thể truyền thông tin với nhau bình thường.

Tầng 7 : Application Layer (tầng ứng dụng) Cung cấp các ứng dụng truy xuất đến các dịch vụ mạng như Web Browser, Mail User Agent ... hoặc các chương trình làm server cung cấp các dịch vụ mạng như Web Server, FTP Server, Mail server ...

4.4.2. Mô hình TCP/IP

Mô hình TCP/IP là mô hình có trước so với mô hình OSI, mục đích mô hình OSI để tham chiếu làm rõ ràng cách thức trao đổi thông tin giữa các máy tính

với nhau, hay nói cách khác là mô hình OSI mang tính chất dùng cho học tập nghiên cứu nhiều hơn là đưa vào triển khai thực tế. Mô hình TCP/IP gồm 4 tầng, so với mô hình OSI có sự tương ứng giữa các tầng như sau:

OSI model (7 layers)	TCP/IP model (4 layers)	Internet protocol suite
Application Presentation	Application	HTTP, FTP, TFTP, NFS, RPC, XDR, SMTP, POP, IMAP, MIME, SNMP, DNS, RIP, OSPF, BGP, TELNET, Rlogin
Session		
Transport	Transport	TCP, UDP
Network	Internet	IP, ICMP, IGMP, ARP, RARP
Data link Physical	Network access	Ethernet, token ring, FDDI, PPP, X.25, frame relay, ATM

Tầng 1 : Network Access Layer (tầng truy cập) là việc kết hợp giữa Physical Layer (Layer 1) and the Data Link layer (Layer 2) của mô hình OSI.

Chức năng của tầng này sử dụng để truyền các bit 0 hoặc 1 qua môi trường vật lý, cung cấp các trình điều khiển để tương tác với các thiết bị phần cứng ví dụ như Token Ring, Ethernet, FDDI...

Cấu trúc Ethernet Frame:

Preamble	Start of Frame Delimiter	Destination Address	Source Address	Length/Type	Data	FCS Frame Check Sequence
7 bytes	1 bytes	6 bytes	6 bytes	2 bytes	64-1500 bytes	4 bytes

Preamble là chuỗi 56 bit để đồng bộ hóa và phát hiện tín hiệu

Start of Frame Delimiter (1 byte): báo hiệu bắt đầu khung.

Source Address (6 byte): chứa địa chỉ MAC của thiết bị truyền.

Destination Address (6 byte): chứa địa chỉ MAC của thiết bị nhận.

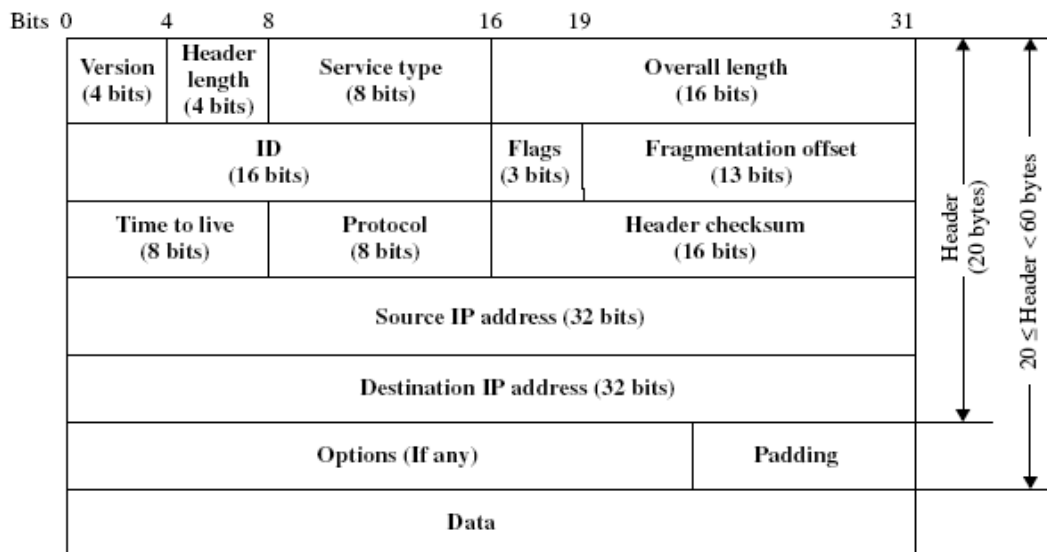
Type/Length (2 byte): chứa tổng số byte trong trường dữ liệu.

Data: là dữ liệu thật sự được truyền và nhận từ Internet Layer, tối đa 1500 byte

FCS: chứa giá trị Cyclic Redundancy Check (CRC) dùng để kiểm tra lỗi.

Tầng 2 : Internet Layer (tầng mạng) Nhiệm vụ của tầng mạng trên mô hình TCP/IP là giải quyết vấn đề dẫn các gói tin đi qua các mạng để đến đúng đích mong muốn. Vì vậy tại tầng này bao gồm các thủ tục cần thiết giữa các hosts và gateways để di chuyển các gói giữa các mạng khác nhau. Một gateway kết nối hai mạng, và sử dụng kết nối mạng bao gồm IP (Internet Protocol), ICMP (Internet Control Message Protocol)....

Cấu trúc IP(IPv4) Datagram:



- Version: Trường 4 bit chứa phiên bản IP được dùng (IPv4, IPv6)
- Header length: chiều dài của IPv4 trong 32 bit
- Type of service: trường 8 bit cung cấp chất lượng dịch vụ mong muốn (Quality Of Service) như: độ ưu tiên, độ trễ, thông lượng, độ tin cậy.
- Overall length: để chỉ chiều dài (byte) gồm chiều dài IP header và dữ liệu đi kèm.
- Source IP Address: gồm 32 bit địa chỉ của thiết bị mà gửi Ipv4 Datagram.
- Destination IP Address: gồm 32 bit địa chỉ của thiết bị mà nhận Ipv4 Datagram.
- Data: chứa đựng dữ liệu được phát ra bởi Transport layer (TCP hay UDP)

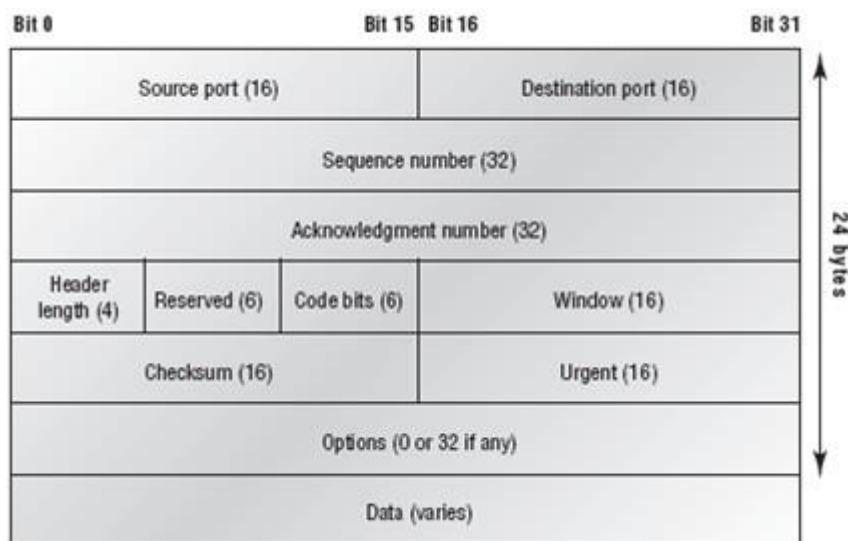
Tầng 3 : Transport Layer (tầng vận chuyển) cũng giống với tầng vận chuyển của mô hình OSI, làm nhiệm vụ phân nhỏ các gói tin có kích thước lớn khi gửi và tập hợp lại khi nhận, đảm bảo tính toàn vẹn cho dữ liệu (không lỗi, không mất, không lặp, đúng thứ tự).

Transport layer là một giao thức mà cung cấp một kết nối logic giữa các thực thể cấp cao bao gồm việc điều khiển lỗi và điều khiển luồng. Gồm 2 giao thức: Transmission Control Protocol (TCP) và User Datagram Protocol (UDP)

Giao thức TCP:

TCP là một giao thức Connection-Oriented nên để truyền được dữ liệu thì trước đó nó phải thiết lập kết nối rồi duy trì kết nối và sau khi hết dữ liệu cần gửi nó phải giải phóng kết nối. Trong quá trình truyền dữ liệu có sử dụng cơ chế điều khiển luồng (flow control) và điều khiển lỗi.

Cấu trúc của TCP Segment



- Source port (16bits) : Số hiệu cổng của trạm nguồn
- Destination port (16bits) : Số hiệu cổng của trạm đích
- Sequence Number (32bits): Số hiệu của byte đầu tiên của segment trừ khi bit SYN được thiết lập. Nếu bit SYN được thiết lập thì nó là số hiệu tuần tự khởi đầu (ISN)
- Acknowledgment Number (32bits) : Số hiệu của segment tiếp theo mà trạm nguồn đang chờ nhận được và nó có ý nghĩa báo nhận tốt
- Header length (4bits) : Số lượng word (32bits) trong TCP header. Nó có tác dụng chỉ ra vị trí bắt đầu của vùng data.

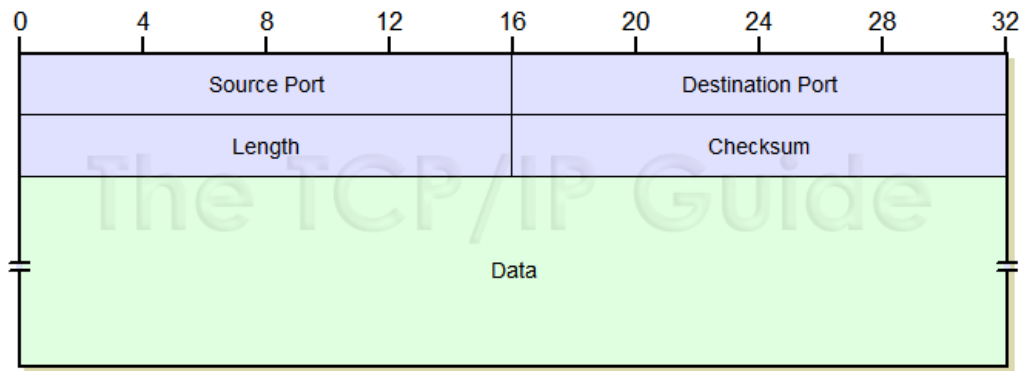
- Reserved (6bits) : dành để sử dụng sau này
- Code bits hay các bits điều khiển (6bits) theo thứ tự từ trái sang phải như sau :
 - URG : vùng con trỏ khẩn (Urgent Pointer) có hiệu lực
 - ACK : vùng báo nhận (ACK number) có hiệu lực
 - PSH : chức năng PUSH
 - RST : khởi động lại liên kết
 - SYN : đồng bộ hoá các số hiệu tuần tự (sequence number)
 - FIN : không còn dữ liệu từ trạm nguồn
- Window (16bits) : cấp phát credit để kiểm soát luồng dữ liệu (cơ chế cửa sổ). Đây chính là số lượng các byte dữ liệu, bắt đầu từ byte được chỉ ra trong vùng ACK number, mà trạm nguồn đã sẵn sàng nhận
- Check sum (16bits) : mã kiểm soát lỗi (theo phương pháp CRC)
- Urgent Poiter (16bits) : con trỏ này trỏ tới số hiệu tuần tự của byte đi theo sau dữ liệu khẩn, cho phép bên nhận biết được độ dài của dữ liệu khẩn, chỉ có hiệu lực khi bit URG được thiết lập.
- Options (độ dài thay đổi) : khai báo các options của TCP
- Data : phần dữ liệu của TCP segment.

Giao thức UDP:

UDP là giao thức kết nối không định hướng (connectionless) nên khác với TCP, UDP không có quá trình thiết lập và giải phóng kết nối, không hỗ trợ chức năng báo nhận (acknowledgement), không sắp xếp tuần tự các đơn vị dữ liệu (packet) đến và có thể dẫn đến tình trạng mất hoặc trùng dữ liệu (packet) mà không hề có thông báo lỗi.

Do có ít chức năng phức tạp nên UDP có tốc độ truyền và nhận nhanh hơn TCP trong các mạng có độ tin cậy cao như LAN. UDP thường dùng cho các ứng dụng đơn giản.

Cấu trúc của UDP Segment



- Source port (16bits) : Số hiệu cổng của trạm nguồn
- Destrination port (16bits) : Số hiệu cổng của trạm đích
- Length (16 bits) : chiều dài toàn bộ UDP diagram, gồm cả header và data.
- Checksum (16 bits) : mã kiểm soát lỗi.
- Data : phần dữ liệu của UDP segment.

Cả hai UDP và TCP sẽ lấy dữ liệu từ lớp Application và thêm Header vào nó khi truyền dữ liệu. Khi nhận dữ liệu, Header sẽ bị gỡ trước khi gửi dữ liệu tới cổng thích hợp. Trong Header này có một vài thông tin điều khiển. liên quan tới số cổng nguồn, số cổng tới đích, chuỗi số (để hệ thống sắp xếp lại dữ liệu và hệ thống Acknowledge sử dụng trong TCP) và Checksum (dùng để tính toán xem dữ liệu đến đích có bị lỗi hay không).

Tầng 4 : Application Layer (tầng ứng dụng): Tầng ứng dụng của mô hình TCP/IP bao gồm cả các giao thức hoạt động như tầng trình diễn và phiên trong mô hình OSI. Việc này thường do người lập trình viên thực hiện.

Tầng này bao gồm các giao thức cấp cao mà chúng được sử dụng để cung cấp các giao diện với người sử dụng hoặc các ứng dụng. Một số giao thức quan trọng như File Transfer Protocol (FTP), HyperText Transfer Protocol (HTTP), Simple Network Management Protocol (SNMP), Domain Naming Service (DNS), Simple Mail Transport Protocol (SMTP), Post Office Protocol (POP). Internet Mail Access Protocol (IMAP), Internet Control Message Protocol (ICMP)....

4.4.3. DHCP (dynamic host configuration protocol): Giao thức cấu hình địa chỉ động.

DHCP là dịch vụ trên nền giao thức TCP/IP (nằm ở tầng ứng dụng) nhằm đơn giản hóa vai trò quản trị của việc cấu hình địa chỉ IP của mạng client. Microsoft Windows 2003 server cung cấp dịch vụ DHCP, hỗ trợ máy tính thực hiện chức năng như DHCP server để cấu hình và cấp phát các thông số cấu hình TCP/IP đến các máy client trong hệ thống mạng của bạn.

DHCP sử dụng UDP là giao thức ở tầng vận chuyển (Transport layer) trong mô hình TCP/IP, vì thế không có độ tin cậy.

Cấu trúc 1 gói dữ liệu DHCP:

MAC header	IP header	UDP header	DHCP header	Data
-------------------	------------------	-------------------	--------------------	-------------

Cấu trúc DHCP header:

0	8	16	24 32
Opcode	Hardware type	HW addr length	Hop count
Transaction Identifier (XID)			
Number of second		Flags	
Client IP address (CIAddr)			
Your IP address (YIAddr)			
Server IP address (SIAddr)			
Gateway IP address (GIAddr)			
Client hardware address (CHAddr) (16 bytes)			
Server name (Sname) (64 bytes)			
Boot filename (128 bytes)			
Option (Variable size)			

Trong đó:

- Opcode: loại hình hoạt động của DHCP. Nếu gói yêu cầu: 1, nếu gói trả lời: 2.

Message	Description	Code??
Discover	A broadcast from a client to recognise server.	
Offer	A response from a server with a proposal of parameters.	Op code 2 - BOOTREPLY
Request	A client broadcasts to a preferred server, declining all others.	Op code 1 - BOOTREQUEST
Ack	The server assigns an IP address.	
Nak	The server rejects the clients request.	
Decline	The client found a problem with the assigned address.	
Release	The client returns the assigned IP address before the lease expires.	

- Hardware type

Code	Definition
1	Ethernet (10Mb)
2	Experimental Ethernet (3Mb)
3	Amateur Radio AX.25
4	Proteon ProNET Token Ring
5	Chaos
6	IEEE 802 Networks
7	ARCNET
8	Hyperchannel
9	Lanstar
10	Autonet Short Address
11	LocalTalk
12	LocalNet (IBM PCNet or SYTEK LocalNET)
13	Ultra link
14	SMDS
15	Frame Relay
16	Asynchronous Transmission Mode (ATM)
17	HDLC
18	Fibre Channel
19	Asynchronous Transmission Mode (ATM)
20	Serial Line

- HW addr length: Chiều dài địa chỉ phần cứng. Nếu mạng Ethernet hoặc mạng khác theo tiêu chuẩn IEEE 802 thì giá trị này là 6 bytes.
- Hop count: được cài đặt là 0 trước khi gửi request, qua mỗi DHCP relay thì giá trị này tăng lên 1.
- XID: là số ngẫu nhiên 32 bit được phát ra bởi client, cho phép phù hợp giữa client và server để không có sự nhầm lẫn giữa các gói tin.
- Number of second: Thời gian trôi mà DHCP client bắt đầu gửi DHCP request.
- Flags: bit đầu tiên được sử dụng để xác định các gói tin DHCP reply được gửi ở chế độ unicast hay broadcast. Các bit còn lại được dự trữ.
- CIAddr: địa chỉ IP của DHCP client.
- YIAddr: địa chỉ IP mà DHCP server gán cho client.

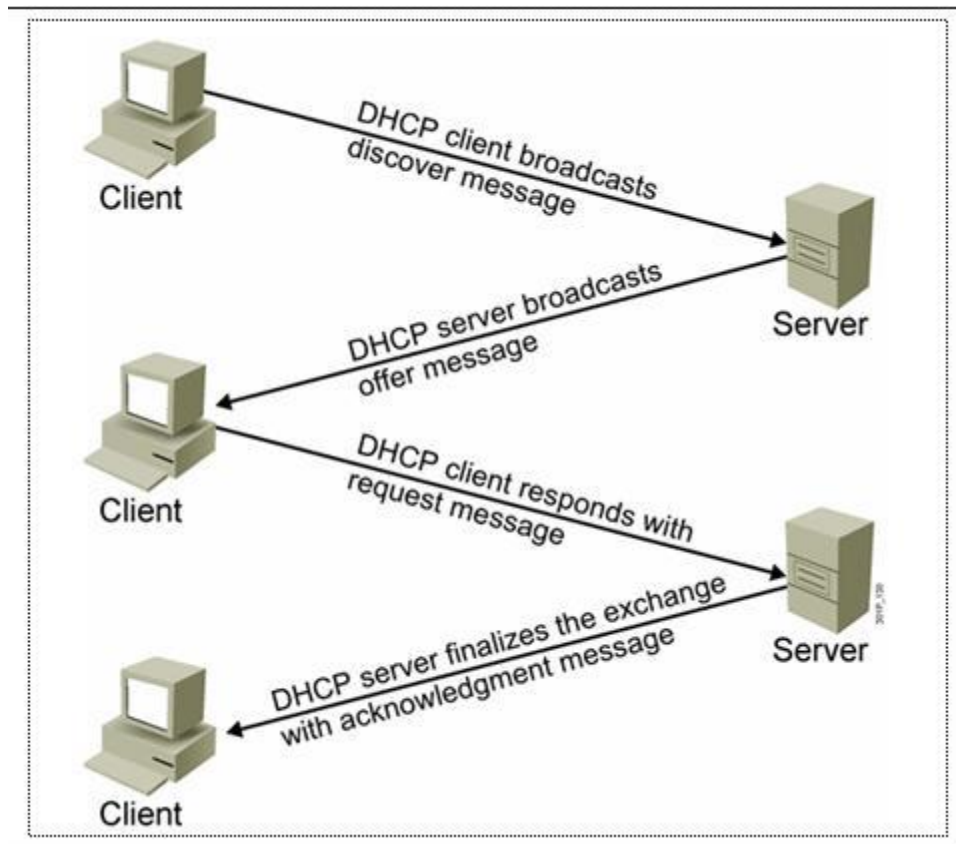
- SIAddr: địa chỉ IP của DHCP server.
- GIAddr: địa chỉ IP của DHCP relay đầu tiên mà DHCP client gửi gói request qua.
- CHAddr: địa chỉ phần cứng của DHCP client.
- SName: Tên của DHCP server.
- Boot filename: đường dẫn và tên file cấu hình boot mà DHCP server chỉ định cho DHCP client.
- Option: có chiều dài thay đổi, bao gồm các loại gói tin, thời gian hợp đồng, địa chỉ IP của DNS server, địa chỉ IP của WINS server.

Phương thức hoạt động của DHCP

Dịch vụ DHCP hoạt động theo mô hình Client / Server. Theo đó quá trình tương tác giữa DHCP client và server sẽ diễn ra theo các bước sau.

- Là giao thức cấu hình tự động địa chỉ IP
 - Được xác định trên mô hình client-server
 - +DHCP Server cấp địa chỉ mạng, phân phối thông số cấu hình tương ứng xuống cho client
 - +”Client” : chỉ các host yêu cầu các địa chỉ IP từ DHCP Server
 - DHCP hỗ trợ 3 cơ chế cấp địa chỉ IP
 - +Cấp tự động: DHCP gán 1 địa chỉ IP thường trực → 1 Client
 - +Cấp động: DHCP gán địa chỉ IP cho 1 khoảng thời gian hữu hạn nào đó
 - +Cấp thủ công: 1 địa chỉ IP được gán bởi người quản trị. DHCP chỉ để đưa địa chỉ này đến Client
- Cơ chế cấp động là cơ chế duy nhất được sử dụng để cấp lại địa chỉ mà không còn được sử dụng nữa trên client cho 1 máy client khác

Cách hoạt động DHCP



B1: Khi máy Client khởi động, máy sẽ gửi broadcast gói tin DHCP DISCOVER yêu cầu một server phục vụ mình. Gói tin này cũng chứa địa chỉ MAC của client.

Nếu client không liên lạc được với DHCP server thì sau 4 lần truy vấn không thành công nó sẽ tự động phát sinh ra 1 địa chỉ IP riêng cho chính mình nằm trong dãy 169.254.0.0 đến 169.254.255.255 dùng để liên lạc tạm thời. Và client vẫn duy trì việc phát tín hiệu boardcast sau mỗi 5 phút để xin cấp IP từ DHCP server.

B2: Các máy server trên mạng khi nhận được yêu cầu đó, nếu còn khả năng cung cấp địa chỉ IP đều gửi lại cho máy client một gói tin DHCP OFFER đề nghị cho thuê một địa chỉ IP trong khoảng thời gian nhất định, kèm theo là một Subnet Mask và địa chỉ của server. Server sẽ không cấp phát địa chỉ IP vừa đề nghị cho client thuê trong suốt thời gian thương thuyết.

B3: Máy client sẽ lựa chọn một trong những lời đề nghị (DHCP OFFER) và gửi broadcast lại gói tin DHCP REQUEST và chấp nhận lời đề nghị đó. Điều này cho phép các lời đề nghị không được chấp nhận sẽ được các server rút lại và dùng để cấp phát cho các client khác.

B4: Máy server được client chấp nhận sẽ gửi ngược lại một gói tin DHCP ACK như một lời xác nhận cho biết địa chỉ IP đó, Subnet Mask đó và thời hạn cho sử dụng đó

sẽ chính thức được áp dụng. Ngoài ra server còn gửi kèm những thông tin bổ xung như địa chỉ gateway mặc định, địa chỉ DNS server....

Nếu phát hiện ra lỗi, DHCP client gửi bản tin DHCP DECLINE tới server và bắt đầu lại quá trình cấu hình.

Nếu DHCP client nhận được bản tin DHCP NACK, quá trình cấu hình cũng sẽ bắt đầu lại.

4.4.4. ARP:

Address resolution protocol (ARP) là giao thức được dùng để ánh xạ địa chỉ mạng IP thành địa chỉ vật lý (MAC) tương ứng của thiết bị tại tầng liên kết dữ liệu (data link) trong mô hình OSI.

Cấu trúc của ARP:

MAC header	ARP header	Data
-------------------	-------------------	-------------

Cấu trúc ARP header:

Hardware type (16 bits)	
Protocol type (16 bits)	
Hardware address length (8 bits)	Protocol address length (8 bits)
Opcode (16 bits)	
Source hardware address (variable)	
Source protocol address (variable)	
Destination hardware address (variable)	
Destination protocol address (variable)	

Trong đó:

- Hardware type: Xác định loại giao diện phần cứng.

Value	Description
-------	-------------

1	Ethernet
2	Experimental Ethernet
3	X.25
4	Proteon ProNET (Token Ring)
5	Chaos
6	IEEE802.X
7	ARC network

- Protocol type: Xác định loại giao thức được dùng bởi thiết bị gửi. IPv4 là 2048 (0x0800 Hexa)
- Hardware address length: Chiều dài địa chỉ MAC. Địa chỉ MAC Ethernet là 6.
- Protocol address length: chiều dài địa chỉ IP. IPv4 là 4.
- Opcode: request là 1, reply là 2.
- Source hardware address: địa chỉ lớp MAC của thiết bị gửi.
- Source protocol address: địa chỉ IP của thiết bị gửi.
- Destination hardware address: với gói tin là request thì trường này là rỗng, với gói tin là reply trường này là địa chỉ MAC của thiết bị nhận.
- Destination protocol address: địa chỉ IP của thiết bị nhận.

Quy trình thực hiện ARP:

B1: Khi một thiết bị nguồn muốn giao tiếp với một thiết bị khác, nó sẽ kiểm tra bảng ánh xạ ARP của nó để tìm địa chỉ MAC của thiết bị đích. Nếu có, thì nó sẽ dùng địa chỉ MAC này để thực hiện quá trình giao tiếp.

B2: Nếu bảng ARP không có địa chỉ MAC của thiết bị đích, thiết bị nguồn sẽ phát 1 gói ARP request. Nó đặt địa chỉ MAC của nó lên source hardware address và địa chỉ IP của nó lên source protocol address. Địa chỉ IP của thiết bị đích được đặt lên destination protocol address và vị trí destination hardware address trong ARP header để trống.

B3: Thiết bị nguồn phát quảng bá (broadcast) gói tin ARP request đến mạng nội bộ.

- TYPE (8bit): là một số nguyên 8 bit để xác định thông điệp.

Type	Description ICMP Message Types
0	Echo Reply (Ping Reply, used with Type 8, Ping Request)
3	Destination Unreachable
4	Source Quench
5	Redirect
8	Echo Request (Ping Request, used with Type 0, Ping Reply)
9	Router Advertisement (Used with Type 9)
10	Router Solicitation (Used with Type 10)
11	Time Exceeded
12	Parameter Problem
13	Timestamp Request (Used with Type 14)
14	Timestamp Reply (Used with Type 13)
15	Information Request (obsolete) (Used with Type 16)
16	Information Reply (obsolete) (Used with Type 15)
17	Address Mask Request (Used with Type 18)
18	Address Mask Reply (Used with Type 17)

- CODE (8bit): cung cấp thêm thông tin về kiểu thông điệp.

- CHECKSUM(16bit) : mã kiểm soát lỗi, nhưng ICMP checksum chỉ tính đến thông điệp ICMP.

4.4.5.2. Lệnh PING:

PING: dùng để xác định một mạng được kết nối đến thiết bị mạng khác.

Lệnh PING thuộc nhóm thông điệp truy vấn của ICMP. Đó là ICMP echo gồm 2 loại: echo request và echo reply.

Cấu trúc echo request và echo reply:

0	8	16	24
			32

Type	Code	ICMP header checksum
Identifier		Sequence number

- + Type=0 => echo request, code=0
- + Type=8 => echo reply, code=0

Ngoài ra còn có 2 trường (size là 16bit/field) là ID và sequence Number dùng để nhận biết giữa các cặp reply/ request.

Nguyên tắc hoạt động:

Khi thiết bị A muốn liên lạc được với thiết bị B, thiết bị A bắt đầu quá trình kiểm tra bằng việc gửi gói tin ICMPv4 Echo request đến thiết bị B. Khi thiết bị B nhận được echo, nó sẽ gửi lại cho thiết bị A gói Echo reply. Và khi thiết bị A nhận được gói tin echo reply thì nó biết rằng đã giao tiếp thành công với thiết bị B.

Cú pháp: ping ip/host [-t][-a][-l][-n]

Trong đó:

- + Ip: địa chỉ IP của máy cần kiểm tra.
- + Host: tên của máy cần kiểm tra.
- + [-t]: thực hiện lệnh liên tục
- + [-a]: nhận địa chỉ IP từ tên máy.
- + [-l]: số gói tin gửi đi.
- + [-n]: kích thước gói tin gửi đi.

Vd: Khi ping đến địa chỉ google thì ta có thể thực hiện lệnh sau:

ping google.com.vn

1 trong 3 gói tin sau thường nhận được khi sử dụng lệnh ping:

- Reply from 173.194.127.127: bytes=32 time<1ms TTL=128
- Request timed out
- Destination host unreachable

Thông báo: **Reply form 173.194.127.127: bytes=32 time=29ms TTL=55**

Khi nhận được thông báo có dạng như trên thì có nghĩa là lệnh ping đã thực hiện thành công và hệ thống không có lỗi:

Địa chỉ IP sau từ **“Reply from”** cho biết máy nào đang gửi thông điệp trả lời.

Bytes=32 là kích thước gói tin ICMP được gửi đi.

Time=29ms: thời gian của quá trình hồi đáp là 29 ms

TTL=55 là giá trị “time to live” (thời gian sống) của gói tin ICMP. Hết thời gian này thì gói tin sẽ bị hủy.

Thông báo **request timed out** hoặc **destination host unreachable**: không kết nối được máy đích.

4.5. Giới thiệu về gói TCP/IP

Tên file	Nội dung
spi.c (spi.h)	Các hàm khởi tạo giao tiếp SPI và đọc ghi qua SPI
enc424j600.c (enc424j600.h)	Các hàm driver để truy xuất chip ethernet ENC424J600
Tick.c (Tick.h)	Các hàm tạo thời gian để sử dụng cho toàn bộ quá trình hoạt động giao tiếp ethernet/internet
IP.c (IP.h)	Thư viện các hàm cho lớp địa chỉ IP
UDP.c (UDP.h)	Thư viện các hàm cho giao tiếp UDP

TCP.c (TCP.h)	Thư viện các hàm cho giao tiếp TCP
ARP.c (ARP.h)	Thư viện các hàm cho giao thức ARP
DHCP.c (DHCP.h)	Thư viện các hàm cho ứng dụng DHCP
PingDemo.c (PingDemo.h)	Các hàm để demo lệnh PING
GenericTCPClient.c (GenericTCPClient.h)	Các hàm để demo quá trình gửi dữ liệu từ client đến server qua giao thức TCP/IP
define.c (define.h)	Các hàm hỗ trợ cho toàn bộ hệ thống
main.c	Chương trình chính của hệ thống