|  |
| --- |
| **TÀI LIỆU THỰC HÀNH TRUYỀN SỐ LIỆU** |
| THS. NGUYỄN LƯƠNG THANH TÙNG |
|  |

# BÀI 1: GIỚI THIỆU PHÒNG THỰC HÀNH VÀ PHẦN MỀM HỌC TẬP

## Mục đích yêu cầu

* **Nắm vững kỹ thuật an toàn trong tất cả các thao tác thực hành**
* **Nắm vững an toàn thực hành và nội quy**
* **Nắm vững các thiết bị sẽ được thực hành và các phần mềm**

## Học cụ

* **Máy Tính**
* **300Mbps Wireless N Router TP- Link TL – WR845N**
* **Router Vigor**
* **Đầu nối DB9**
* **Jack Tivi**
* **Đầu nối RJ45**
* **Dây cable UTP**
* **Kềm bấm mạng**
* **Kềm bấm dây**
* **Máy test cable mạng**
* **Mỏ hàn, chì**

## III. Nội Dung Thí Nghiệm – Thực hành

### Giới thiệu phòng thực hành

* 1. **Nội quy**
  2. **Các thiết bị đang có**
  3. **Các nguyên tắc an toàn điện khi sử dụng thiết bị**
  4. **Cách sử dụng và bảo quản thiết bị**
  5. **Vệ sinh phòng thực hành**

### Giới thiệu nội dung môn học

* 1. **Nội dung về tổng quan các bài thí nghiệm**
  2. **Yêu cầu sinh viên chuẩn bị cho các bài học**

### Giới thiệu và hướng dẫn sử dụng nhanh 5 phần mềm học tập:

* 1. **Visual Basic**
  2. **Matlab**
  3. **WireShark**
  4. **FTTP**
  5. **Arduino**

# BÀI 2: HÀN CÁP SERIAL, BẤM CÁP TIVI, BẤM CÁP MẠNG

## I. Mục đích yêu cầu

* **Nắm vững kỹ thuật an toàn trong tất cả các thao tác thực hành**
* **Tạo ra các sợi dây cable nối như Cable Serial, Cable Tivi, Cable UTP**
* **Biết cách kết nối và kiểm tra thiết bị cable có hoạt động tốt hay không**

## II. Học cụ

* **Máy Tính**
* **Đầu nối DB9**
* **Jack Tivi**
* **Đầu nối RJ45**
* **Dây cable UTP**
* **Kềm bấm mạng**
* **Kềm bấm dây**
* **Máy test cable mạng**
* **Mỏ hàn**

## III. Nội Dung Thí Nghiệm – Thực hành

### A. LÝ THUYẾT:

**1. Cáp serial**

Cáp Serial là cáp có hai đầu nối DB9 hoặc song song hoặc chéo nhau. Cáp serial mở rộng bởi một đầu là cổng USB hay cổng máy in nhằm dễ dàng kết nối những thiết bị không có tích hợp sẵn cổng nối tiếp bên ngoài.

|  |  |  |
| --- | --- | --- |
|  |  |  |

Cáp này được dùng trong truyền thông nối tiếp giữa máy tính với máy tính hoặc hai modem, hoặc máy tính với modem, hoặc máy tính với KIT vi xử lý.

**2. Cáp TIVI**

Thông thường cáp tivi là cáp đồng trục, sử dụng đầu nối BNC để kết nối. Cáp đồng trục có một lõi đồng dẫn dữ liệu ngay chính giữa cáp rồi đến phần nhựa mềm cách điện (chất liệu không cháy), rồi đến lớp phủ kim loại bạc chống nhiễu xuyên kênh và cuối cùng là vỏ bọc cáp bên ngoài.

|  |  |  |
| --- | --- | --- |
|  |  |  |

**3. Cáp Mạng UTP**

Cable mạng UTP gồm 8 dây – chia làm 4 cặp xoắn với nhau. Một cặp gồm một dây có nhuộm màu và dây còn lại có màu trắng sọc mà sọc màu giống với màu của sợi dây quấn chung với nó.

Cấu trúc xoắn từng cặp của cable UTP là để chống nhiễu nhờ vào việc triệt nhiễu vi sai trên một cặp dây.

|  |  |  |  |
| --- | --- | --- | --- |
|  | |  |  |
| **1** | **TD+** |  | |
| **2** | **TD-** |
| **3** | **RD+** |
| **4** | **NC** |
| **5** | **NC** |
| **6** | **RD-** |
| **7** | **NC** |
| **8** | **NC** |
| ***\_Chuẩn bấm cáp mạng 568A và B***  Khi kết nối hai máy tính trực tiếp với nhau hoặc là các modem với nhau thì phải kết nối chéo, một đầu bấm chuẩn A còn đầu kia bấm chuẩn B  Khi kết nối từ một máy tính đến Router hay modem, Switch, Hub thì ta bấm cáp thẳng- cả hai đầu là chuẩn A hoặc cả hai đầu là chuẩn B.   |  |  |  | | --- | --- | --- | | **Số thứ tự chân** | **Chuẩn 568A** | **Chuẩn 568B** | | 1 | Trắng – Xanh lá | Trắng – Cam | | 2 | Xanh lá | Cam | | 3 | Trắng cam | Trắng – Xanh lá | | 4 | Xanh dương | Xanh dương | | 5 | Trắng – Xanh dương | Trắng Xanh dương | | 6 | Cam | Xanh lá | | 7 | Trắng – Nâu | Trắng - Nâu | | 8 | Nâu | Nâu | | | | |

### B. THỰC HÀNH:

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | Sinh viên hàn một sợi dây gồm hai đầu DB9 để kết nối hai máy tính.   |  | | --- | | * Dùng HyperTerminal để truyền dữ liệu giữa hai máy tính | | * Thực hiện kết nối chủ tớ trên 2 PC dựa trên công cụ Communications của window theo sự hướng dẫn của giáo viên. | |
| 2 | Sinh viên bấm đầu nối cáp tivi |
| 3 | Sinh viên bấm một sợi dây mạng theo chuẩn cáp chéo   |  | | --- | | * Dùng thiết bị thử để kiểm tra kết quả bấm cáp | | * Kết nối hai máy tính bởi sợi dây mới bấm | | * Thực hiện truyền nhận dữ liệu trên hai máy tính. | |
| 4 | Sinh viên bấm một sợi dây mạng theo chuẩn cáp thẳng |
|  | |  | | --- | | * Dùng thiết bị thử để kiểm tra kết quả bấm cáp | | * Kết nối máy tính với modem ADSL bởi sợi dây mới bấm | | * Thực hiện ping từ máy tính tới modem theo hướng dẫn của GV | |
| 5 | Trả lời câu hỏi của GV trong buổi thực hành |
| 6 | Bài báo cáo thí nghiệm |

# BÀI 3: LẬP TRÌNH GIAO TIẾP HAI MÁY TÍNH THÔNG QUA CỔNG NỐI TIẾP

## I.Mục đích yêu cầu

* **Nắm vững kỹ thuật an toàn trong tất cả các thao tác thực hành**
* **Lập trình được phần mềm giao tiếp máy tính thông qua cổng nối tiếp**
* **Thực hiện kiểm tra kết quả đạt được và báo cáo kinh nghiệm làm việc**

## II.Học cụ

* **Máy Tính**
* **Đầu nối DB9**
* **Kềm bấm dây**
* **Mỏ hàn**

## III.Nội Dung Thí Nghiệm – Thực hành

### A.LÝ THUYẾT:

* + 1. **Chuẩn truyền thông RS232**

Phần lý thuyết được giảng dạy trong môn học truyền số liệu, sinh viên có thể xem lại. Ngoài ra sinh viên chú ý nắm vững một số đặc tính phần cứng để cấu hình như sau:

* + - 1. **Đầu nối DB9**

|  |  |  |
| --- | --- | --- |
|  |  |  |

Hình 1: Đầu nối DB9 và cáp nối cổng nối tiếp.

\_Đặc tính điện:

|  |
| --- |
|  |

Hình 2: đặc tính điện mô tả mức logic của chuẩn truyền thông RS232 ở máy tính.

Hướng cải tiến là tăng điện áp truyền để tăng khoảng cách truyền:

***\_Đối với đường tín hiệu dữ liệu:***

* Mức logic 1: điện áp từ .
* Mức logic 0: điện áp từ +.

***\_Đối với đường tín hiệu điều khiển:***

* Mức logic 1 (True): điện áp từ .
* Mức logic 0 (False): điện áp từ .

***\_Đầu nối DB9 gồm các tín hiệu như sau:***

|  |  |  |
| --- | --- | --- |
| Pin 1 | CD | Modem báo cho máy tính biết đã thu được sóng mang từ đường dây – Carrier Detect |
| Pin 2 | RD | Tín hiệu nhận |
| Pin 3 | TD | Tín hiệu truyền |
| Pin 4 | DTR | Báo cho biết máy tính đã sẵn sàng – Data terminal ready |
| Pin 5 | GND | Đất – dây nối mass chung. |
| Pin 6 | DSR | Báo cho biết modem đã sẵn sàng – Data set ready |
| Pin 7 | RTS | Máy tính yêu cầu để truyền dữ liệu – Request to send |
| Pin 8 | CTS | Modem trả lời cho yêu cầu truyền dữ liệu của máy tính, cho biết đường truyền đã sẵn sàng để truyền dữ liệu – Clear to send |
| Pin 9 | RI | Modem tách được tín hiệu gọi từ đường dây – Ring indicate |

***\_Nghi thức truyền dẫn (Protocol Transmission):***

Khi đường truyền rảnh gọi là Idle (Idle state), tín hiệu trên đường truyền là mức 1. Khi có một thay đổi mức đường truyền từ mức 1 về mức 0 – Start bit, sẽ bắt đầu cho một khung truyền nối tiếp. Phần kế tiếp bit Start là khung dữ liệu thực sự mà chúng ta quan tâm. Tùy theo người cấu hình cho phần cứng là truyền khung dữ liệu bao nhiêu bit. Thông thường dữ liệu là 8bit. Kế tiếp dữ liệu là một bit Parity (P). Bit P có thể có hoặc có thể không cần cài đặt. Kết thúc khung truyền là 1, 1.5 hoặc 2 bit Stop. Phía thu tự nó đồng bộ khung truyền theo phía phát (cùng tốc độ nhận và truyền) từ khi nó nhận được bit Start. Và sau khi nhận hết 8 bit dữ liệu, bit Parity(P), bit Stop thì nó kết thúc một khung truyền và chờ một cờ Start khác từ phía phát để bắt đầu cho việc nhận dữ liệu từ khung truyền mới.

* + 1. **Lập trình Visual Basic**

***\_Cách khởi động chương trình Visual Basic:***

|  |  |
| --- | --- |
| Mở chương trình | Start/program/Microsoft Visual Studio 6.0/ Visual Basic 6.0 |
| Tạo Project mới | File / New Project |
| ***\_Cửa sổ chương trình khi tạo Project mới:*** | |
|  | |
| Vùng thiết kế | Tất cả đối tượng nằm trong Form |
| Projects Explorer | Hiển thị tất cả các Form và Module của chương trình |
| Property Windown | Hiển thị và cho phép cấu hình thuộc tính của đối tượng |
| Thanh công cụ | Tạo các đối tượng cho Project |
| ***\_Các thuộc tính cơ bản của đối tượng trong VB6*** | |
| Name | Tên của đối tượng, các đối tượng khác nhau phải có tên khác nhau |
| Caption | Hiển thị nội dung của đối tượng trên cửa sổ thiết kế |
| Tabindex | Thứ tự chuyển đến khi nhấn phím Tab |
| Font | Chọn font hiển thị cho đối tượng |
| Back Colour, Fore Colour | Chọn màu hiển thị cho đối tượng |
| Value | Giá trị của đối tượng (dùng cho Check Box và Option để xác định trạng thái được chọn hay không chọn) |
| Tetx | Nội dung chứa trong một Text Box |
| Multiline | Cho phép hiển thị nhiều dòng trên một TextBox hay không |
| Enable | Cho phép đối tượng được hoạt động hay không |
| Duration | Xác định thời gian timer tràn |
| …. | …. |
| ***\_Cá sự kiện cơ bản trên Visual Basic*** | |
| Form\_Load | Các sự kiện xãy ra khi mở Form |
| Form\_Unload | Các sự kiện xãy ra khi đóng Form |
| Click | Các sự kiện khi nhấn chuột trái lên đối tượng |
| Timer | Các sự kiện xãy ra khi Timer bị tràn |
| …… | ….. |
| ***\_Các lệnh cơ bản trong Visual Basic*** | |
| IF – THEN – ELSE | IF(Condition 1) THEN  Nhóm Lệnh 1  ELSEIF  Nhóm Lệnh 2  …..  ELSEIF  Nhóm Lệnh n -1  ELSE  Nhóm Lệnh n  END IF |
| SELECT CASE | SELECT CASE (Biểu thức điều kiện)  CASE(điều kiện 1)  Nhóm lệnh 1  CASE(điều kiện 2)  Nhóm lệnh 2  ….  CASE(điều kiện n)  Nhóm lệnh n  END SELECT |
| FOR …NEXT | FOR Biến\_Đếm = init\_value to end\_value step step\_jump  Nhóm Lệnh  Next Biến\_Đếm |
| DO….LOOP WHILE | DO  Nhóm Lệnh  LOOP WHILE (Điều kiện) |
| DO WHILE… LOOP | DO WHILE (Điều kiện)  Nhóm Lệnh  LOOP |

* + 1. **Đối tượng MSCOMM trong Visual Basic**

Việc truyền thông nối tiếp trên Windows được thực hiện thông qua một ActiveX có sẵn là MSComm Control. ActiveX này được lưu trong file MSCOMM32.OCX. Quá trình này có hai khả năng thực hiện trao đổi thông tin:

|  |  |
| --- | --- |
| Điều khiển sự kiện | Truyền thông điều khiển sự kiện là phương pháp tốt nhất trong quá trình điều khiển trao đổi thông tin. Quá trình điều khiển thông qua sự kiện OnComm |
| Hỏi Vòng | Quá trình điều khiển bằng cách hỏi vòng được thực hiện thông qua việc kiểm tra các giá trị của thuộc tính CommEvent sau một chu kỳ nào đó để xem có một sự kiện nào xãy ra hay không. Thông thường phương pháp hỏi vòng chỉ sử dụng cho những chương trình nhỏ |

***\_Cách sử dụng đối tượng MSCOMM:***

|  |  |
| --- | --- |
| Tạo đối tượng | Project / Component/ Microsoft Comm Control 6.0 |
| ***\_Các thuộc tính MSCOMM*** | |
| Comport | Số thứ tự cổng truyền thông |
| Input | Nhận ký tự từ bộ đệm nhận của cổng nối tiếp |
| Output | Xuất ký tự ra bộ đệm ra của cổng nối tiếp |
| PortOpen | Mở (True), Đóng (False) cổng nối tiếp |
| Settings | Xác định cấu hình cổng nối tiếp  MsComm1.Settings = “tốc\_độ\_baud, Parity, Số\_bit\_dữ\_liệu, số\_bit\_Stop”  MsComm1.CommPort = Port\_Number  MsComm1.PortOpen = TRUE/FALSE |
| ***\_Các thuộc tính nhận dữ liệu*** | |
| Input | Nhận một chuổi ký tự và xóa khỏi bộ đệm. Cú pháp:  InputString = MsComm1.Input  \*\*\*Thuộc tính này kết hợp với InputLen để xác định số ký tự đọc vào. Nếu InputLen = 0 thì sẽ đọc toàn bộ dữ liệu có trong bộ đệm |
| InBufferCount | Số ký tự có trong bộ đệm nhân. Cú pháp:  Count = MsComm1.InBufferCount  \*\*\* Thuộc tính này dùng để xóa bộ đệm nhận bằng cách:  MsComm1.InBufferCount = 0 |
| InBufferSize | Đặt và xác định kích thước bộ đệm nhận – tính bằng byte  MsComm1.InBufferSize = byte\_numbers  \*\*\* Giá trị mặc định là 1024 bytes. Kích thước bộ đệm phải lớn để tránh trường hợp xãy ra mất dữ liệu |
| ***\_Các thuộc tính xuất dữ liệu*** | |
| Output | OutputString = MsComm1.Output |
| OutBufferCount | Count = MsComm1.OutBufferCount |
| OutBufferSize | MsComm1.OutBufferSize = byte\_numbers |
| Sthreshold | Số byte trong bộ đệm truyền làm phát sinh sự kiện OnComm. Nếu số byte này bằng 0 thì không tạo sự kiện Oncomm. Cú pháp:  MsComm1.Sthreshold = NumChar |
| HandShaking | Chọn giao thức bắt tay khi thực hiện truyền dữ liệu. Cú pháp:  MsComm1.HandShaking = Protocol  Các giao thức truyền (Protocol) bao gồm |
| CommEvent | Trả lại các lỗi truyền thông hay sự kiện xãy ra tại cổng nối tiếp. Các sự kiện này được mô tả theo bảng sau: |
| ***\_Sự kiện OnComm*** | |
| Sự kiện OnComm xãy ra khi bất cứ khi nào giá trị thuộc tính CommEvent thay đổi. Các thuộc tính Rthreshold và Sthrreshold đồng thời bằng 0 sẽ cấm xãy ra sự kiện OnComm. Thông thường Sthreshold = 0 và Rthreshold = 1.  Để thực hiện và kiểm tra một chương trình truyền thông đơn giản, sinh viên có thể loop\_back bằng cách nối TXD và RXD của Comport để kiểm tra. | |

### B.THỰC HÀNH:

|  |
| --- |
| ***\_1:*** Tạo giao diện truyền thông hai máy tính ***như sau:*** |
| ***\_2: Thêm đối tượng MSCOMM như sau:***  Lấy đối tượng MSCOMM: Component(Ctr – T): Microsoft Communication 6  Name = Mscomm1  Rthreshold = 1  Comport = 1 (hay 2 tùy theo máy tính) |
| ***\_Chương trình tham khảo cho Form***  Chú ý chương trình chỉ có tính tham khảo, sinh viên phải thêm hoặc bớt đi những gì không cần thiết. Sinh viên phải viết thêm một số lệnh để ngăn chặn việc báo lỗi khi một sự kiện OnComm xãy ra. Để làm việc này sinh viên phải đọc kỹ phần lý thuyết đã có trình ở phần lý thuyết:   |  |  | | --- | --- | | Private Sub cmdClear\_Click()  txtReceive.Text = ""  End Sub  Private Sub cmdExit\_Click()  Unload Me  End Sub | Private Sub cmdOpen\_Click()  MSComm1.PortOpen = Not MSComm1.PortOpen  If MSComm1.PortOpen Then  cmdOpen.Caption = "Close Port"  Else  cmdOpen.Caption = "Open Port"  End If  End Sub | | Private Sub cmdSend\_Click()  MSComm1.Output = txtSend.Text  End Sub  Private Sub Form\_Unload(Cancel As Integer)  If MSComm1.PortOpen Then MSComm1.PortOpen = False  End Sub | Private Sub MSComm1\_OnComm()  Dim t As String  Select Case MSComm1.CommEvent  Case comEvReceive  t = MSComm1.Input  txtReceive.Text = txtReceive.Text + t  End Select  End Sub | |
| ***\_***Trả lời câu hỏi (Vấn đáp) sau buổi thực hành***:***   |  |  | | --- | --- | | Câu 1 | Kết nối hai máy tính như thế nào? | | Câu 2 | Kiểm tra kết quả kết nối hai máy từ chương trình như thế nào? | | Câu 3 | Giải thích sự kiện nhận của hai máy? | | Câu 4 | Nếu muốn kiểm tra chẵn lẻ thì cấu hình cổng nối tiếp như thế nào? | | Câu 5 | Trình bày hoạt động của đầu nối DB9? | | Câu 6 | Trình bày cách cấu hình cổng nối tiếp trên máy tính? | | Câu 7 | Giải thích hoạt động truyền của cổng nối tiếp? | | Câu 8 | Có thể thay đổi cấu hình cổng nối tiếp so với cài đặt trên máy tính không? | |
| ***\_***Viết bài thu hoạch nộp cho GVHD thực hành |

# BÀI 4: LẬP TRÌNH GIAO TIẾP GIỮA KIT ESP32 VỚI MÁY TÍNH

## I. Mục đích yêu cầu

* **Nắm vững kỹ thuật an toàn trong tất cả các thao tác thực hành**
* **Lập trình được phần mềm giao tiếp máy tính thông qua cổng nối tiếp**
* **Lập trình giao tiếp cổng nối cho kit ESP32**
* **Thực hiện kiểm tra kết quả đạt được và báo cáo kinh nghiệm làm việc**

## II. Học cụ

* **Máy Tính**
* **KIT thực hành ESP32**
* **Đầu nối DB9**
* **Kềm bấm dây**
* **Mỏ hàn**
* **Dây nối đơn đực – cái (bus 1 đực cái)**

## III. Nội Dung Thí Nghiệm – Thực hành

### A.LÝ THUYẾT:

Về lý thuyết cổng nối tiếp và cách lập trình code visual basic cho đối tượng cổng nối tiếp đã được nói ở bài 2, Sau đây là phần lý thuyết liên quan đến bài 3:

**1. Giới thiệu KIT vi xử lý thực hiện giao tiếp với máy tính**

Kit RF thu phát Wifi BLE ESP32 NodeMCU LuaNode32 được phát triển trên nền module trung tâm là ESP32 với công nghệ Wifi, BLE và nhân ARM SoC tích hợp mới nhất hiện nay, kit có thiết kế phần cứng, firmware và cách sử dụng tương tự Kit NodeMCU ESP8266, với ưu điểm là cách sử dụng dễ dàng, ra chân đầy đủ, tích hợp mạch nạp và giao tiếp UART CP2102, Kit Wifi BLE ESP32 NodeMCU LuaNode32 là sự lựa chọn hàng đầu trong các nghiên cứu, ứng dụng về điều khiển, Wifi, BLE và IoT.

|  |
| --- |
|  |

**2. Chương trình giao tiếp qua cổng nối tiếp với KIT.**

Gồm hai chương trình: Chương trình viết code Visual Basic trên máy tính và chương trình code C(trên arduino) cho ESP32.

### B.THỰC HÀNH:

|  |  |
| --- | --- |
| 1 | \_Lập giao diện VB ***như sau:*** |
| 2 | ***\_Code cho giao diện như sau:***  ***Dim lamp\_value As Byte***  ***-------------------------------------------------------***  ***Private Sub Check1\_Click(Index As Integer)***  ***Dim i As Integer***  ***lamp\_value = 0***  ***For i = 0 To 7***  ***If Check1(i).Value = 1 Then***  ***lamp\_value = lamp\_value + 2 ^ i***  ***Else***  ***lamp\_value = lamp\_value***  ***End If***  ***Next i***  ***'Text1.Text = Str(lamp\_value)***  ***For i = 0 To 7***  ***If Check1(i).Value = 1 Then***  ***Shape1(i).FillColor = RGB(255, 0, 0)***  ***Else***  ***Shape1(i).FillColor = RGB(255, 255, 255)***  ***End If***  ***Next i***  ***'Send lamp\_value to serial' output buffer to control lamps on kit 8051***  ***If MSComm1.PortOpen = True Then***  ***MSComm1.Output = Chr(lamp\_value)***  ***End If***  ***End Sub***  ***------------------------------------------------------***  ***Private Sub cmd\_Connect\_Click()***  ***'Setting for the Mscomm1: serial port***  ***If (Combo1.Text <> "" And Combo2.Text <> "" And Combo3.Text <> "" And Combo4.Text <> "" And Combo5.Text <> "") Then***  ***MSComm1.CommPort = Val(Combo1.List(0))***  ***Dim baurate As String***  ***Dim parity As String***  ***'Dim stopbits As String***  ***'Dim numofbit As String***  ***baurate = Combo2.Text***  ***Select Case (Combo3.Text)***  ***Case ("None")***  ***parity = "n"***  ***Case ("Odd")***  ***parity = "o"***  ***Case ("Even")***  ***parity = "e"***  ***End Select***  ***MSComm1.Settings = baurate & parity & Combo5.Text & Combo4.Text***  ***MSComm1.InBufferSize = 512***  ***MSComm1.PortOpen = True***  ***lbl\_serial\_status.Caption = "SERIAL PORT WAS ACTIVE HAD:" & MSComm1.Settings***  ***MSComm1.RThreshold = 1***  ***'MSComm1.InputMode = comInputModeBinary***  ***cmd\_Connect.Enabled = False***  ***cmd\_Unconnect.Enabled = True***  ***Else***  ***lbl\_serial\_status.Caption = "YOU HAVE NOT CONFIG SERIAL PORT SUCESSFULLY"***  ***MsgBox ("Let Configure the Serial port by choose one of options of these combo box")***  ***End If***  ***End Sub***  ***---------------------------------------------------***  ***Private Sub cmd\_Exit\_Click()***  ***If MSComm1.PortOpen = True Then***  ***MSComm1.PortOpen = False***  ***End If***  ***Unload Me***  ***End Sub***  ***Private Sub cmd\_Unconnect\_Click()***  ***MSComm1.PortOpen = False***  ***cmd\_Connect.Enabled = True***  ***cmd\_Unconnect.Enabled = False***  ***lbl\_serial\_status.Caption = “THE SERIAL NOW WAS CLOSING”***  ***End Sub***  ***------------------------------------------***  ***Private Sub Form\_Load()***  ***Dim i As Integer***  *'commbo1 is used to choice port number*  ***For i = 1 To 10***  ***Combo1.List(i - 1) = Str(i)***  ***Next i***  *'combo2 is used to select the baurate*  ***For i = 1 To 8***  ***Combo2.List(i - 1) = Str(2 ^ (i - 1) \* 1200)***  ***Next i***  *'combo3 is used to select type of parity*  ***Combo3.List(0) = "None"***  ***Combo3.List(1) = "Odd"***  ***Combo3.List(2) = "Even"***  *'combo4 is used to select number of stop bits*  ***For i = 1 To 3***  ***Combo4.List(i - 1) = Str(0.5 + 0.5 \* i)***  ***Next i***  *'combo5 is used to select the number of data bits:*  ***For i = 0 To 3***  ***Combo5.List(i) = i + 5***  ***Next i***  ***cmd\_Connect.Enabled = True***  ***cmd\_Unconnect.Enabled = False***  ***For i = 0 To 7***  ***Shape1(i).FillStyle = 0***  ***Shape1(i).Shape = 3***  ***Shape1(i).FillColor = RGB(255, 255, 255)***  ***Next i***  ***For i = 0 To 7***  ***Shape2(i).FillStyle = 0***  ***Shape2(i).Shape = 3***  ***Shape2(i).FillColor = RGB(255, 255, 255)***  ***Next i***  ***Me.Width = 12765***  ***Me.Height = 8700***  ***End Sub***  ***------------------------------------------------***  ***Private Sub MSComm1\_OnComm()***  ***Dim value\_of\_lamp\_received As Integer***  ***Dim compare As Integer***  ***Select Case MSComm1.CommEvent***  ***Case comEvReceive***  ***value\_of\_lamp\_received = Asc(MSComm1.Input)***  ***End Select***  ***For i = 0 To 7***  ***compare = value\_of\_lamp\_received And (2 ^ i)***  ***If compare > 0 Then***  ***Shape2(i).FillColor = RGB(255, 0, 0)***  ***Else***  ***Shape2(i).FillColor = RGB(255, 255, 255)***  ***End If***  ***Next i***  ***If MSComm1.InBufferCount > 1 Then***  ***MSComm1.Input = ""***  ***End If***  ***End Sub*** |
| 3 | \_Code cho ESP32 ***như sau:***  #include<stdio.h>  #include <reg8252.h>  //sbit button1 = P2^1;  //sbit button2 = P2^2;  sbit button3 = P3^3;  unsigned char DuLieuNhan;  unsigned char DuLieuPhat;  void Ngat\_nhan\_DuLieu(void) interrupt 4  {  RI = 0;  DuLieuNhan = SBUF;  DuLieuNhan = DuLieuNhan^255;  P2 = DuLieuNhan;  }  // will be used if P3.2 or P3.3 is connect to the button  void Send\_Data\_To\_PC(unsigned char S\_data)  {  int i;  putchar(S\_data);  for(i = 2000;i--;i<=1)  ;  TI = 0;  }  void Init\_Serial(void)  {  SCON = 0x52; //setup serial port control  TMOD = 0x20; //hardware (9600 Buard @11.05592Mhz)  TH1 = 0xFD; //load the constant 0xFD to config baudrate = 9600  TR1 = 1; //timer1 ON  IE = 0x92; //cho phep ngat noi tiep va vector ngat toan cuc  }  void main(void)  {  Init\_Serial();  DuLieuPhat = P1;    Send\_Data\_To\_PC(DuLieuPhat);  } |
| 4 | ***\_Mô hình mô phỏng trên Protues:*** |
| C.**Câu hỏi bài thực hành** | |
| 1 | ***Sinh viên hãy thực hành 3 phần:***   * ***Code và giao diện Visual Basic*** * ***Mô hình mô phỏng trên Protues*** * ***Projects viết bằng C nạp cho vi điều khiển*** |
| 2 | ***Giải thích cơ chế truyền một byte từ máy tính xuống Kit*** ESP32 |
| 3 | ***Giải thích cơ chế truyền một byte từ Kit*** ESP32 ***lên máy tính*** |
| 4 | ***Giải thích cơ chế nhận của Kit*** ESP32 ***và máy tính.*** |
| Báo cáo bài thực hành | |
| 1 | ***Sinh viên hãy báo cáo kết quả bài thực hành trên.*** |

# BÀI 5: LẬP TRÌNH GIAO TIẾP CỔNG LAN THÔNG QUA WINSOCK

## I. Mục đích yêu cầu

* **Nắm vững kỹ thuật an toàn trong tất cả các thao tác thực hành**
* **Lập trình được phần mềm giao tiếp máy tính thông qua cổng RJ45**
* **Thực hiện kiểm tra kết quả đạt được và báo cáo kinh nghiệm làm việc**

## II. Học cụ

* **Máy Tính**
* **Đầu nối RJ45**
* **Dây Cable UTP**
* **Kềm bấm cable mạng**
* **Kềm bấm dây**
* **Máy test cable mạng**

## III. Nội Dung Thí Nghiệm – Thực hành

### A. LÝ THUYẾT:

**1. Giới thiệu Winsock**

VB6 cho ta Winsock Control để dùng cho việc giúp một program VB6 nói chuyện với một program khác trên mạng TCP/IP. Ta có thể dùng Winsock Control trong một program để làm Winsock Server hay Winsock Client. Sự khácbiệt nầy rất nhỏ, mặc dầu ta phải lưu ý để phân biệt sự khác nhau của hai trường hợp. Giả sử ta dùngWinsock Control làm Server trong một VB6 program để chạy trên một computer và dùng Winsock Control làm Client trong một VB6 program để chạy trên một computer khác trên mạng TCP/IP. Ðể cho hai programs nói chuyện (communicate) trước hết ta cần phải connect (nối) chúng lại với nhau. Ta cho Winsock Server Listen (lắng nghe) qua một LocalPort (một cổng có mang một con số, thí dụ như 101). Kế đó ta cho Winsock Client Connect (móc nối) qua LocalPort đó ở địa chỉ TCP của Computer nơi ta chạy Winsock Server program. Sở dỉ ta cần phải nói rõ LocalPort số mấy là vì Server Computer có

thể Listen qua nhiều LocalPorts cùng một lúc để nhiều Clients có thể Connect đến cùng một Computer TCP address.

**2. Cách sử dụng Winsock trên VB**

|  |  |
| --- | --- |
| Lấy Winsock | Component/microsoft winsock |

### B. THỰC HÀNH:

|  |  |
| --- | --- |
| 1 | \_Tạo giao diện trên máy chủ: |
| 2 | \_Tạo giao diện trên máy Client: |
| 3 | \_Chương trình trên máy chủ:  |  |  | | --- | --- | | ***A\_ Listen port***  Private Sub cmdlisten\_Click()  Winsock1.Close  Winsock1.LocalPort = Val(txtlocalport.Text)  Winsock1.Listen  End Sub  ***C\_Thủ tục thoát chương trình****:*  Private Sub Command1\_Click()  Winsock1.Close  Unload Me  End Sub | ***E\_Thủ tục Timer 1***  Private Sub Timer1\_Timer()  i = i + 1  If i > 9 Then  i = 0  If Winsock1.State <> 7 Then  lblstate.Caption = "SERVER IS NOT CONECTED CLIENT"  cmdlisten.Value = 1  Else  lblstate.Caption = "SERVER IS CONNECTED CLIENT"  End If  End If  End Sub | | ***F\_Thủ tục connection Request:***  Private Sub Winsock1\_ConnectionRequest(ByVal requestID As Long)  Winsock1.Close  Winsock1.Accept requestID  End Sub  ***D\_Thủ tục Formload:***  Private Sub Form\_Load()  Me.WindowState = 2  Me.Caption = "CONTROL OVER INTERNET - PC SERVER"  Shape1.FillStyle = 0  Shape2.FillStyle = 0  Shape3.FillStyle = 0  Shape4.FillStyle = 0  Shape5.FillStyle = 0  Shape6.FillStyle = 0  Shape7.FillStyle = 0  Shape8.FillStyle = 0  Shape1.FillColor = RGB(255, 255, 255)  Shape2.FillColor = RGB(255, 255, 255)  Shape3.FillColor = RGB(255, 255, 255)  Shape4.FillColor = RGB(255, 255, 255)  Shape5.FillColor = RGB(255, 255, 255)  Shape6.FillColor = RGB(255, 255, 255)  Shape7.FillColor = RGB(255, 255, 255)  Shape8.FillColor = RGB(255, 255, 255)  Winsock1.Protocol = sckTCPProtocol  Winsock1.LocalPort = 3333  End Sub | ***B\_Send Net***  Private Sub cmdsendnet\_Click()  Dim datasend As Byte  datasend = 0  If Check1.Value = 1 Then  datasend = datasend Or 1  End If  If Check2.Value = 1 Then  datasend = datasend Or 2  End If  If Check3.Value = 1 Then  datasend = datasend Or 4  End If  If Check4.Value = 1 Then  datasend = datasend Or 8  End If  If Check5.Value = 1 Then  datasend = datasend Or 16  End If  If Check6.Value = 1 Then  datasend = datasend Or 32  End If  If Check7.Value = 1 Then  datasend = datasend Or 64  End If  If Check8.Value = 1 Then  datasend = datasend Or 128  End If  txtsendcode.Text = datasend  If Winsock1.State = 7 Then  Winsock1.SendData txtsendcode.Text  Else  MsgBox ("you have not connect to client PC")  End If  End Sub | | ***G. Thủ tục nhận dữ liệu***  Private Sub Winsock1\_DataArrival(ByVal bytesTotal As Long)  Dim s As String  Dim datarev As Byte  Winsock1.GetData s  txtrev.Text = s  datarev = Val(txtrev.Text)  If (datarev < 256) Then    If datarev And 1 Then  Shape1.FillColor = RGB(255, 0, 0)  Else  Shape1.FillColor = RGB(255, 255, 255)  End If    If datarev And 2 Then  Shape2.FillColor = RGB(0, 255, 0)  Else  Shape2.FillColor = RGB(255, 255, 255)  End If    If datarev And 4 Then  Shape3.FillColor = RGB(0, 0, 255)  Else  Shape3.FillColor = RGB(255, 255, 255)  End If    If datarev And 8 Then  Shape4.FillColor = RGB(193, 46, 209)  Else  Shape4.FillColor = RGB(255, 255, 255)  End If    If datarev And 16 Then  Shape5.FillColor = RGB(193, 160, 62)  Else  Shape5.FillColor = RGB(255, 255, 255)  End If    If datarev And 32 Then  Shape6.FillColor = RGB(223, 228, 27)  Else  Shape6.FillColor = RGB(255, 255, 255)  End If    If datarev And 64 Then  Shape7.FillColor = RGB(43, 183, 213)  Else  Shape7.FillColor = RGB(255, 255, 255)  End If    If datarev And 128 Then  Shape8.FillColor = RGB(252, 3, 215)  Else  Shape8.FillColor = RGB(255, 255, 255)  End If  End If  End Sub | | |
| 4 | \_Chương trình trên máy Client ***A\_Connect***  Private Sub cmdconnect\_Click()  Winsock1.Connect txtipadd.Text, Val(txtlocalport.Text)  lblstate.Caption = Winsock1.State  End Sub  ***B\_SendData***  Private Sub cmdsenddata\_Click()  Dim a As Byte  a = 0  If clamp1.Value = 1 Then  a = a Or 1  End If  If clamp2.Value = 1 Then  a = a Or 2  End If  If clamp3.Value = 1 Then  a = a Or 4  End If  If clamp4.Value = 1 Then  a = a Or 8  End If  If clamp5.Value = 1 Then  a = a Or 16  End If  If clamp6.Value = 1 Then  a = a Or 32  End If  If clamp7.Value = 1 Then  a = a Or 64  End If  If clamp8.Value = 1 Then  a = a Or 128  End If  txt\_tranfer.Text = Str(a)  Winsock1.SendData txt\_tranfer.Text  End Sub  ***C\_Thủ tục sự kiện đến***  Private Sub Winsock1\_DataArrival(ByVal bytesTotal As Long)  Dim s As String  Dim datarev As Byte  Winsock1.GetData s  txtreceive.Text = s  datarev = Val(txtreceive.Text)  If (datarev < 256) Then  If datarev And 1 Then  slamp1.FillColor = RGB(255, 0, 0)  Else  slamp1.FillColor = RGB(255, 255, 255)  End If  If datarev And 2 Then  slamp2.FillColor = RGB(0, 255, 0)  Else  slamp2.FillColor = RGB(255, 255, 255)  End If  If datarev And 4 Then  Slamp3.FillColor = RGB(0, 0, 255)  Else  Slamp3.FillColor = RGB(255, 255, 255)  End If  If datarev And 8 Then  slamp4.FillColor = RGB(193, 46, 209)  Else  slamp4.FillColor = RGB(255, 255, 255)  End If  If datarev And 16 Then  slamp5.FillColor = RGB(193, 160, 62)  Else  slamp5.FillColor = RGB(255, 255, 255)  End If  If datarev And 32 Then  slamp6.FillColor = RGB(223, 228, 27)  Else  slamp6.FillColor = RGB(255, 255, 255)  End If  If datarev And 64 Then  slamp7.FillColor = RGB(43, 183, 213)  Else  slamp7.FillColor = RGB(255, 255, 255)  End If  If datarev And 128 Then  slamp8.FillColor = RGB(252, 3, 215)  Else  slamp8.FillColor = RGB(255, 255, 255)  End If  End If  End Sub  ***D.Thủ tục Form load:***  Private Sub Form\_Load()  Winsock1.Protocol = sckTCPProtocol  Me.WindowState = 2  slamp1.FillStyle = 0  slamp2.FillStyle = 0  Slamp3.FillStyle = 0  slamp4.FillStyle = 0  slamp5.FillStyle = 0  slamp6.FillStyle = 0  slamp7.FillStyle = 0  slamp8.FillStyle = 0  slamp1.FillColor = RGB(255, 255, 255)  slamp2.FillColor = RGB(255, 255, 255)  Slamp3.FillColor = RGB(255, 255, 255)  slamp4.FillColor = RGB(255, 255, 255)  slamp5.FillColor = RGB(255, 255, 255)  slamp6.FillColor = RGB(255, 255, 255)  slamp7.FillColor = RGB(255, 255, 255)  slamp8.FillColor = RGB(255, 255, 255)  Me.Caption = "CONTROL OVER INTERNET - CLIENT PC"  End Sub  ***E. Thủ tục thoát chương trình:***  Private Sub cmdexitprogram\_Click()  Winsock1.Close  Unload Me  End Sub |
| 5 | \_Câu hỏi thực hành  |  |  | | --- | --- | | 1 | Winsock là gì? Hoạt động của Winsock như thế nào? | | 2 | Hãy thao tác việc truyền gởi dữ liệu hai máy tính với nhau? | | 3 | Giải thích cơ chế lắng nghe của Winsock? | | 4 | Giải thích thủ tục Timer 1? | |
| 6 | \_Báo cáo thực hành |

# BÀI 6: XÂY DỰNG MÔ HÌNH MẠNG LAN – WAN

## I. Mục đích yêu cầu

* **Nắm vững kỹ thuật an toàn trong tất cả các thao tác thực hành**
* **Lập trình được phần mềm giao tiếp máy tính thông qua cổng RJ45**
* **Thực hiện kiểm tra kết quả đạt được và báo cáo kinh nghiệm làm việc**

## II. Học cụ

* **Máy Tính**
* **Router Vigor**
* **300Mbps Wireless N Router TP- Link TL – WR845N**
* **Đầu nối RJ45**
* **Kềm bấm dây**
* **Kềm bấm cable mạng**
* **Máy test cable mạng**
* **Dây cable UTP**

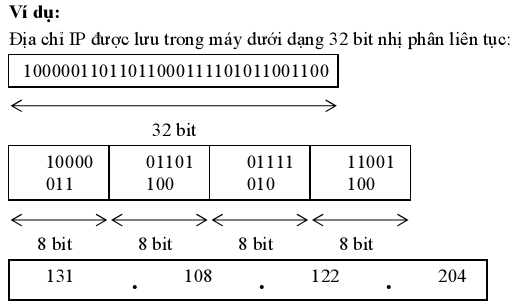
## III. Nội Dung Thí Nghiệm – Thực hành

### A. CƠ SỞ LÝ THUYẾT

* 1. Địa chỉ IPV4

Để hai hệ thống có thể giao tiếp được với nhau qua môi trường mạng, chúng phải được định danh duy nhất để có thể xác định được vị trí của chúng trong hệ thống mạng. Trong môi trường TCP/IP mỗi hệ thống phải được gán ít nhất một số định danh gọi là địa chỉ IP, thông qua địa chỉ IP mà mỗi máy có thể định vị và giao tiếp với các máy khác.

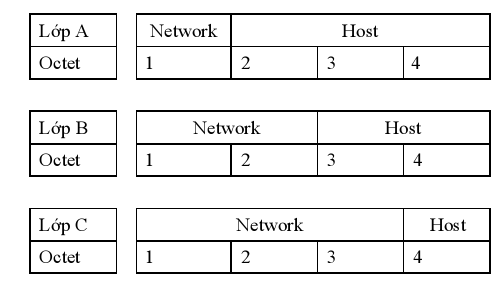
Địa chỉ IPV4 gồm có 32 bits và được lưu trong mỗi máy dưới dạng chuổi số nhị phân 32 bits 0 và 1. Tuy nhiên để con người dễ sử dụng và thao tác, địa chỉ IP được chia thành 4 nhóm, mỗi nhóm 8 bits và thường được viết dưới dạng 4 số thập phân và được ngăn cách bởi dấu chấm “.”. Ví dụ: 192.168.1.18. Mỗi nhóm nhị phân 8 bits được gọi là 1 octet.



Địa chỉ IP chia làm 2 phần:

* + - NetWork: Luôn nằm ở đầu giúp xác định mạng mà hệ thống kết nối tới
    - Host: Nằm ở cuối, xác định một hệ thống cụ thể trong hệ thống mạng đó.

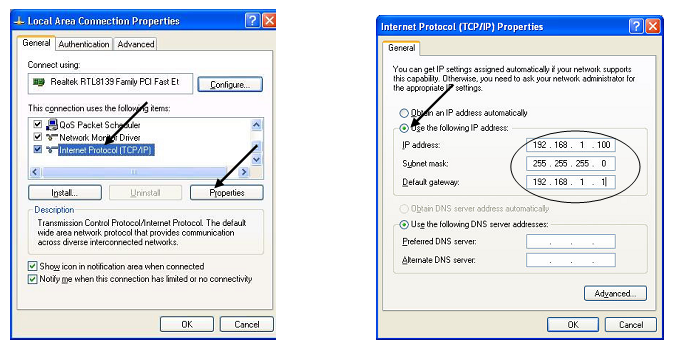
Địa chỉ IPV4 được chia làm 5 lớp A, B, C, D, E theo bảng dưới dây:



Dựa vào Octet đầu tiên của địa chỉ IP mà xác định thuộc lớp mạng nào.

* Lớp A: 1 🡪 126 (0 và 127 không được sử dụng)
* Lớp B: 128 🡪 191
* Lớp C: 192 🡪 223
  1. Cách gián địa chỉ IP cho một máy tính

Start/ Settings/ Network Connections, trong cửa sổ mới double click vào biểu tượng Local Area Connections, Chọn Internet Protocol (TCP/IP) rồi bấm vào nút **Properties** . Ở cửa sổ mới, click chọn “Use the following IP address” rồi gõ địa chỉ IP , Subnetmask và Default Gateway (Default Gateway : có thể được hiểu có thể hiểu là địa chỉ IP của thiết bị kết nối với phần mạng hiện tại với các mạng khác, thông thường là địa chỉ của cổng Router nối với phần mạng hiện tại. Nếu mạng Lan không kết nối với mạng khác thì để trống phần này.



### B. THỰC HÀNH

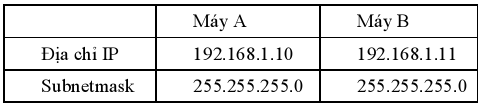
#### Xây dựng mô hình mạng pear to pear

Sinh viên dùng dây cáp mạng thích hơp để kết nối mô hình dưới đây:



Để kết nối PC đến PC ta dùng loại cáp nào? ……………………………..

Gán địa chỉ máy A và máy B theo bảng địa chỉ sau:



Tại sao địa chỉ network của hai máy A và B phải giống nhau? ……………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………….

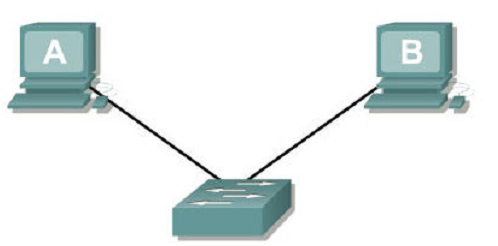
Tại sao phần Host của hai máy A và B phải khác nhau? .......................................................................................................................................................................................................................................................................................................................................................................................................................................

Từ PC A thực hiện lệnh: ping 192.168.1.11 ở command prompt của window. Kết quả ping? ………………………………………………………………………………………………………………………………………………………………………………………….

Từ PC B thực hiện lệnh: ping 192.168.1.10 ở command prompt của window. Kết quả ping? ………………………………………………………………………………………………………………………………………………………………………………………….

#### Xây dựng mô hình mạng Switch Base.

Sinh viên dùng loại dây mạng thích hợp đã bấm từ Bài thực hành số 1 để kết nối theo mô hình dưới đây:



Để kết nối PC với Switch ta dùng loại cable nào? ……………………………………..

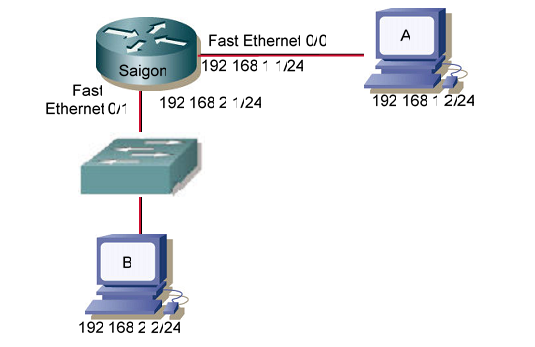
So với mô hình Pear to Pear thì mô hình Switch Base có ưu điểm và khuyết điểm gì ? ………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

Vẫn dùng địa chỉ IP như ở phần I sinh viên thực hiện lệnh ping từ PC A đến PC B và ngược lại. Ghi lại kết quả

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

#### Xây dựng mạng Router Based

Sinh viên dùng loại dây thích hợp để kết nối mô hình dưới đây.



Sinh viên hãy cho biết loại dây mà sinh viên dùng đối với từng loại kết nối

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

Sinh viên thực hiện lệnh ping từ PC A sang PC B và ngược lại? Kết quả như thế nào?

Đánh giá kết quả đó?

……………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

#### Thực hiện mô hình LAN trong phòng thí nghiệm

Phòng thí nghiệm Truyên số liệu có 10 PC. Một Modem ADSL và 2 router.

Sinh viên thực hiện kết nối một mạng LAN gồm 10 PC trên.

GVHD chia sinh viên thành nhiều nhóm, mỗi nhóm 4 người để thực hiện mô hình LAN 4 máy (phù hợp với Router có ngõ kết nối LAN hoặc là phù hợp với modem ADSL có 4 ngõ ra người dùng.

Sinh viên trả lời những câu hỏi ở phần V theo yêu cầu của GVHD thí nghiệm.

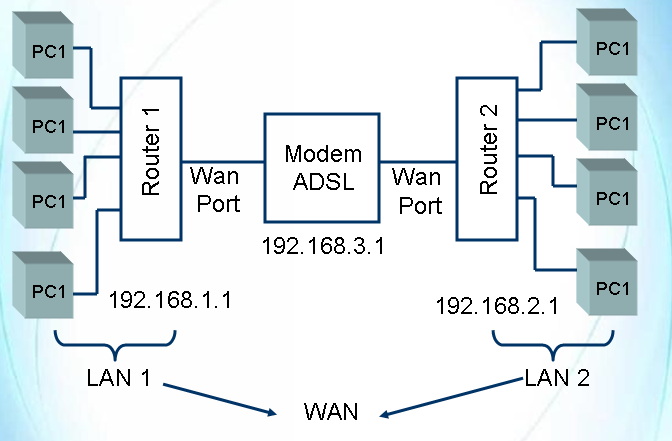
#### Thực hiện mô hình Wan trong phòng thí nghiệm

Lưu ý:

Modem ADSL có một ngõ vào (đường DSL) và 4 port ngõ ra.

Router : có một đường WAN và 4 ngõ ra.

Sinh viên có thể thực hiện mô hình WAN gồm 2 mạng LAN như sau:



Sinh viên thực hiện cài đặt địa chỉ IP cho hệ thống WAN trên theo sự chỉ dẫn của GVHD.

Sinh viên thực hành và trả lời các câu hỏi sau:

* + - 1. Sự khác biết cơ bản của hệ thống LAN1 và LAN2? ………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………
      2. Thực hiện gởi dữ liệu trong mạng nội. Cho nhận xét …………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

* + - 1. Thực hiện kết nối một máy trong hệ thống LAN1 và một máy trong hệ thống LAN2?

Kết quả có thành công hay không? Trong trường hợp nào thì hai mạng LAN1 và LAN2 có thể kết nối với nhau?

…………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

d. DNS là gì? Sinh viên thử đặt DNS cho máy tính mà mình đang thực hành với DNS dưới đây và cho biết sự khác nhau khi có đặt DNS và không đặt DNS cho hệ thống.

DNS1: 203.162.4.190 và DNS2: 203.162.4.191.

…………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

#### Báo cáo kết quả thực hành

Mỗi sinh viên làm một báo cáo kết quả thực hành – có thể là phần trả lời câu hỏi trong phần thực hành hoặc là yêu cầu thêm của GVHD thí nghiệm.

# BÀI 7: PHÂN TÍCH CÁC PROTOCOL THÔNG DỤNG CỦA TCP/IP

## I. Mục đích yêu cầu

* Nắm vững kỹ thuật an toàn trong tất cả các thao tác thực hành
* Sinh viên làm quen và phân tích được các chuẩn protocol thông dụng của TCP/IP
  + ARP
  + DHCP
  + ICMP
  + TELNET
  + Phân tích quá trình thiết lập và kết thúc một kết nối TCP
* Thực hành phân tích protocol bằng chương trình WireShark

## II. Học cụ

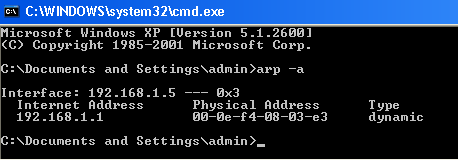
* **Máy Tính**
* **Router Vigor**
* **300Mbps Wireless N Router TP- Link TL – WR845N**
* **Đầu nối RJ45**
* **Kềm bấm dây**
* **Kềm bấm cable mạng**
* **Máy test cable mạng**
* **Dây cable UTP**

## III. Nội Dung Thí Nghiệm – Thực hành

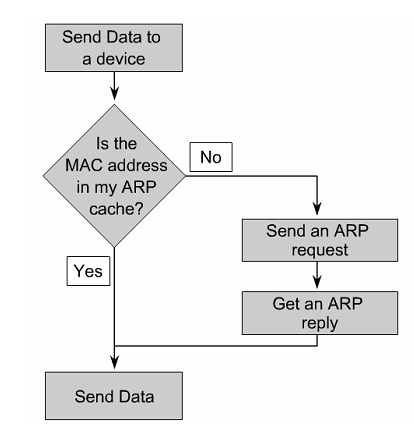
### A. CƠ SỞ LÝ THUYẾT

* 1. ARP – Address Resolution Protocol

Để các máy có thể trao đổi dữ liệu được với nhau thì phía gửi phải biết được thông tin về địa chỉ IP và địa chỉ MAC của máy nhận. Trong khi địa chỉ IP có thể được thông qua một số phương pháp như DNS hay hệ thống tên thiết bị (devices names) thì địa chỉ MAC gần như là chưa được biết trước. TCP/IP định nghĩa một giao thức để thực hiện việc tìm địa chỉ MAC với địa chỉ IP đã biết đó là ARP. Nói cách khác là ARP cho phép ánh xạ một địa chỉ IP với một địa chỉ MAC tương ứng, thông tin này sau đó được lưu vào trong một cơ sở dữ liệu là bảng ARP (lưu trong RAM) để dùng sau này. Ta có thể xem bảng ARP trong hệ điều hành Windows bằng lệnh arp –a trong cmd.exe của windows.



Hoạt động ARP có thể được tóm tắt như sau: khi một máy A muốn gửi dữ liệu đến máy B(đã biết địa chỉ IP), nó sẽ tra địa chỉ IP này trong bảng ARP để tìm địa chỉ MAC. Nếu trong bảng ARP vẫn chưa có địa chỉ này thì máy gửi sẽ gửi một gói ARP request với địa chỉ IP nguồn và đích tương ứng của máy A và máy B, địa chỉ MAC nguồn là của máy A và địa chỉ MAC đích là địa chỉ MAC quảng bá (ff- ff- ff- ff- ff – ff). Do địa chỉ MAC của máy đích là địa chỉ quảng bá nên tất cả các thiết bị mạng trên phần mạng đó sẽ nhận gói ARP request này, các máy mở gói và kiểm tra địa chỉ IP đích, máy B kiểm tra thấy địa chỉ IP đích là địa chỉ IP của nó nên B sẽ gửi lại một gói ARP reply. Tất cả các máy nhận được ARP request mà địa chỉ IP của chính nó không giống với địa chỉ IP đích trong gói ARP reques lập tức hủy gói ARP request này. Trước khi gửi gói ARP reply máy B sẽ trích lại địa chỉ IP và MAC của máy A lưu vào trong bảng ARP của nó. B trả lại gói ARP reply cho máy A bao gồm địa chỉ nguồn là IP và MAC của nó, còn địa chi đích là địa chỉ IP và MAC của máy A.

Lưu đồ về gói ARP:

Trên đây chỉ trình bày cách thức quá

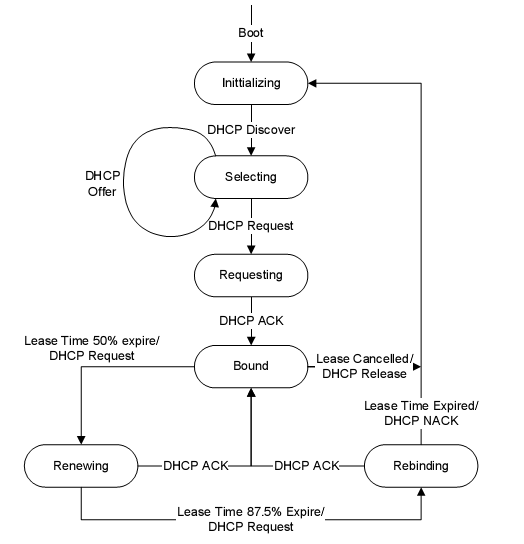
trình ARP giữa hai máy trong cùng một   
mạng, sinh viên tự tìm hiểu thêm về quá   
quá trình ARP giữa hai máy khác mạng   
đó là Proxy ARP, gratuitous ARP.

* 1. DHCP – Dynamic Host Configuration Protocol

Để một máy có thể chuyển gói trên mạng TCP/IP thì nó phải được định danh bằng một địa chỉ IP. Có hai cách cung cấp địa chỉ IP cho một máy:

* + - Gán địa chỉ IP tĩnh: người quản trị mạng tự gán địa chỉ IP cho máy
    - Gán địa chỉ IP động: địa chỉ được tự động cung cấp cho các máy thông qua các giao thức như RARP, BOOTP, DHCP….

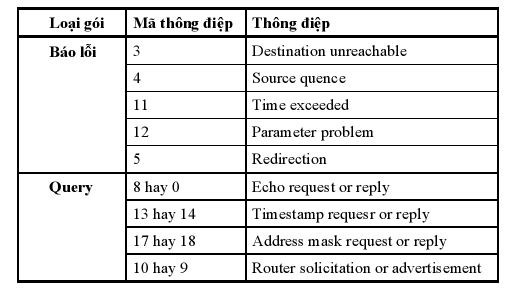
Trong bài này chúng ta sẽ tìm hiểu giao thức DHCP. DHCP hoạt động theo mô hình Server/Client. Hoạt động của DHCP có thể được tóm tắt theo bảng sau:



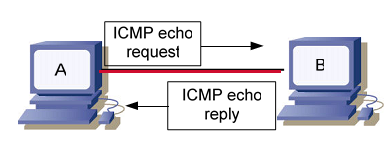
* + - * Trạng thái Initializing: khi một client DHCP được khởi động, nó sẽ vào trạng thái Initializing, client này sẽ gửi một gói DHCP discovery dưới dạng một gói quảng bá (Broadcast).
      * Trạng thái Selecting: sau khi gửi gói DHCP discovery, client sẽ vào trạng thái Selecting và chờ DHCP server trả lời bằng một gói DHCP offer chứa địa chỉ IP, Subnet Mask, và một số thông số khác để gán cho client. Tổng quát trong một mạng thì có thể có nhiều DHCP server, do đó client có thể nhận được nhiều gói DHCP offer trả lời cho gói DHCP discovery của nó. Client sẽ chọn một gói DHCP đến sớm nhất rồi trả lời lại cho DHCP một gói DHCP request và chuyển sang trạng thái Requesting. Nếu không nhận được DHCP Offer thì client sẽ gởi DHCP Discovery thêm 4 lần nữa cách nhau 2 giây, nếu vẫn không có trả lời thì client sẽ ngừng 5 phút rồi lặp lại quá trình trên.
      * Trạng thái Requesting: ở trạng thái này Client đợi gói DHCP Ack từ DHCP server để biết nó có được phép sử dụng địa chỉ IP mà nó được cung cấp hay không.
      * Trạng thái Bound: khi nhận được gói DHCP Ack từ Server thì Client sẽ bắt đầu sử dụng địa chỉ mà DHCP Server cho nó để thực hiện quá trình truyền nhận dữ liệu. Khi ở trạng thái Bound thì Client có thể hủy, không sử dụng địa chỉ IP mà nó được cung cấp bằng cách nó gởi một gói DHCP Release đến DHCP Server và trở về trạng thái Initializing.
      * Trạng thái Renew: một địa chỉ IP được gán bởi DHCP server luôn có một thời gian sống là Lease Time. Sau khi vượt qua 50% thời gian sống thì Client sẽ rơi vào trạng thái Renewing để làm mới thời gian sống, và gửi một gói DHCP Request. Và sau đó nếu Client nhận được gói DHCP Ack thì có nghĩa Server cho phép Client tiếp tục sử dụng địa chỉ IP này. Đến đây Client lại rơi vào trạng thái Bound và làm mới thời gian sống. Nếu như Client không nhận được DHCP Ack sau 87,5% thời gian sống thì nó sẽ chuyển sang trạng thái Rebinding.
      * Trạng thái Rebinding: Client sẽ ở trạng thái Rebinding nếu Client nhận được gói DHCP Nack hoặc là thời gian sống địa chỉ của nó đã hết.

* 1. ICMP – Internet control Message Protocol

Trong mô hình TCP/IP thì IP cung cấp phương pháp truyền thông không đáng tín cậy, không kết nối (connectionless), nó được thiết kế để tận dụng tối đa tài nguyên mạng. Tuy nhiên IP không có cơ chế sữa lỗi hay báo lỗi. Chuyện gì sẽ xãy ra nếu có sự cố, chẳng hạn router loại bỏ gói khi nó không tìm thấy đường đến đích? ICMP được thiết kế để hoàn tất hai nhiệm vụ báo lỗi và query. Để phục vụ nhiệm vụ này ICMP có hai loại gói: gói báo lỗi và gói query. Mỗi loại có nhiều thông điệp mang ý nghĩa khác nhau.



Trong phạm vi thí nghiệm bài này chúng ta chỉ xét đến các thông điệp echo request và echo reply của ICMP. Thông điệp echo request và ech reply được thiết kế để phát hiện và chuẩn đoán lỗi. Hoạt động của cặp thông điệp này rất đơn giản: người dùng hoặc người quản trị gửi một thông điệp echo request từ một hệ thống, hệ thống sau khi nhận được echo request sẽ gởi lại một thông điệp echo reply cho hệ thống gửi. cặp thông điệp này có thể cho biết được hai hệ thống đang liên lạc được với nhau ở lớp 3 hay không, đồng thời cũng cho biết các thiết bị trung gian (router, switch) đã nhận, xử lý và chuyển được thông điệp IP. Nếu vì một lý do nào đó mà máy đích không nhận được thông điệp request thì tại thiết bị cuối cùng nhận được thông điệp echo request sẽ phúc đáp lại bằng một thông điệp báo lỗi để cho biết đó là lỗi gì.

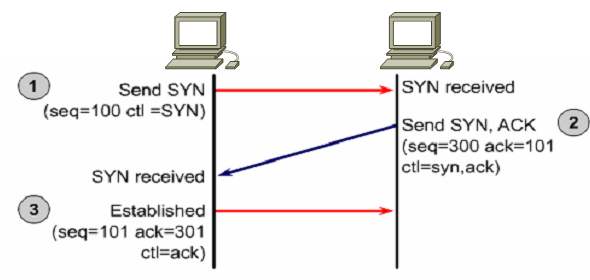


* 1. Quá trình thiết lập và giải toả một kết nối TCP

TCP là một giao thức lớp 4 (lớp Transport). Có chức năng đảm bảo sự vận chuyển đáng tin cậy của dữ liệu qua môi trường mạng, ngoài ra TCP còn được thiết kế với chức năng kiểm soát luồng và kiểm soát lỗi. Chi tiết TCP được đề cập chi tiết trong các giáo trình truyền số liệu và hệ thống viễn thông. Ở đây chỉ tóm tắt quá trình thiết lập kết nối và giải toả một TCP.

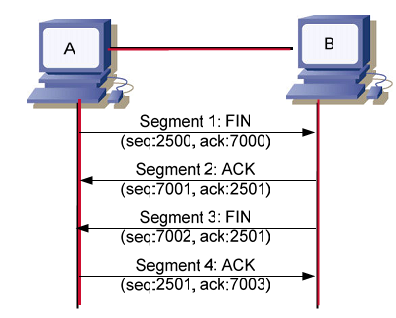
**Quá trình thiết lập một kết nối TCP**: còn gọi là một quá trình bắt tay 3 chiều (three way handshake), được tiến hành trước khi dữ liệu có thể được chuyển giữa các thiết bị nhằm đồng bộ các thông số của kết nối. Quá trình này gồm 3 bước như sau:

* Bước 1: Client khởi tạo kết nối server bằng cách gởi một gói TCP vời cờ SYN được bật, thông báo cho Server biết số thứ tự x của gói nhằm đồng bộ về thông số với server
* Bước 2: Server nhận được gói này và lưu lại số thứ tự x, và trả lời lại bằng một gói có thứ tự x + 1, trong đó chứa số thứ tự y của nó với cờ SYN và ACK được bật. Việc trả lời bằng gói có thứ tự là x + 1 nhằm mục đích thông báo cho client biết được máy nhận đã nhận được tất cả dữ liệu cho đến số thứ tự là x và mong chờ gói có số thứ tự là x+ 1.
* Bước 3: sau khi nhận được gói này, Client phúc đáp lại một gói TCP có cờ ACK được bật và có thứ tự là y + 1 . Sau buớc này thì dữ liệu có thể truyền giữa Client và Server.



**Quá trình giải toả một kết nối TCP:** bao gồm 4 bước (four way handshake) được tóm tắt như sau:

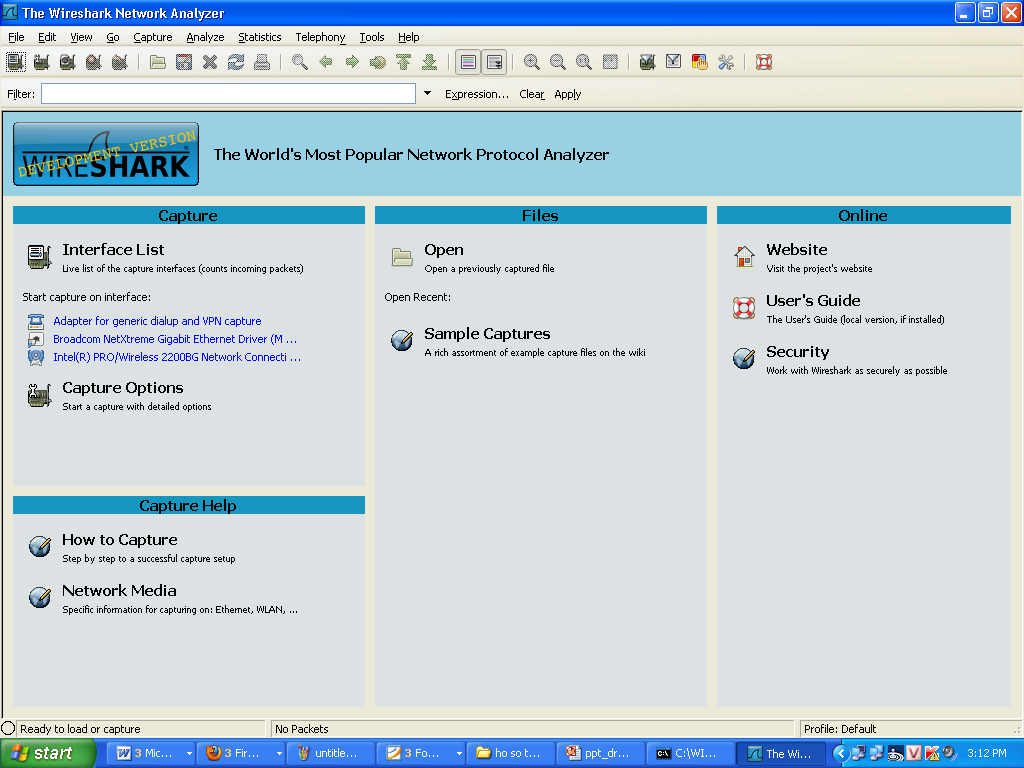
* Bước 1: khi client muốn kết thúc một kết nối TCP, sẽ gởi gói TCP với cờ FIN được bật nhằm thông báo cho server một giải toả kết nối.
* Bước 2: Server sẽ thông báo cho Client một gói TCP với cờ ACK được bật nhằm xác nhận đã nhận được gói trước đó của Client.
* Bước 3: Server tiếp tục gởi một gói TCP với cờ FIN được bật nhằm thông báo cho Client biết về việc giải toả kết nối TCP.
* Bước 4: Client trả lời Server bằng một gói cờ ACK được bật để xác nhận là đã nhận được gói FIN của Server. Sau gói này cả Client và Server giải toả kết nối TCP.



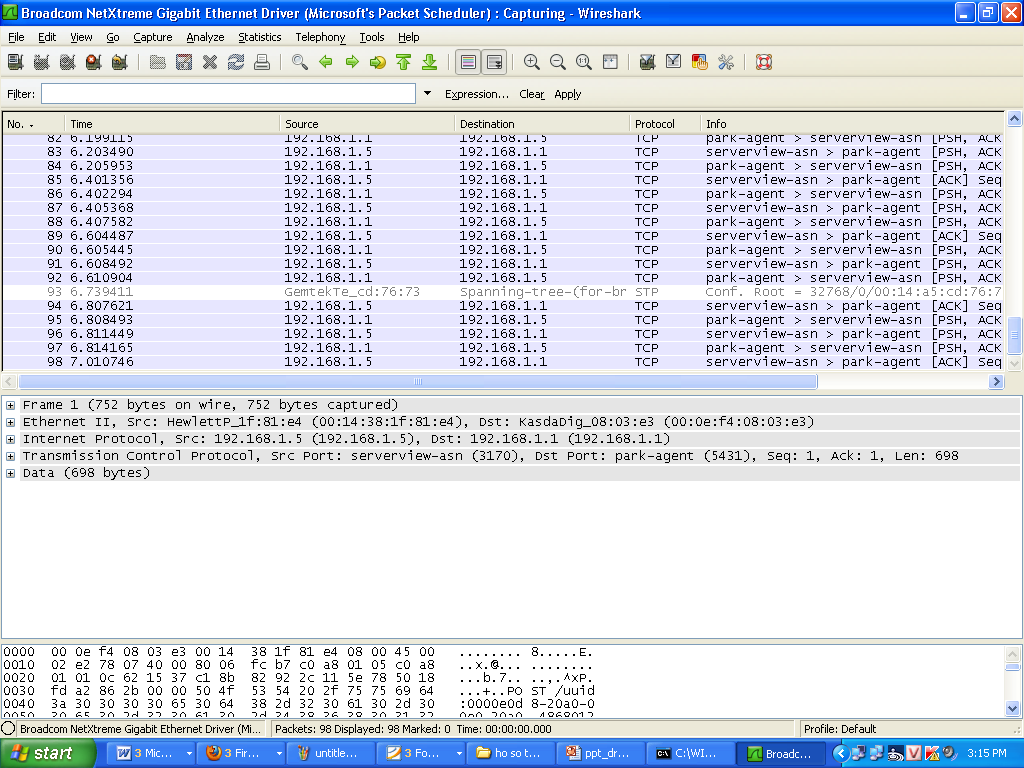
* 1. Dùng chương trình WireShark để phân tích giao thức mạng

WireShake là một chương trình phân tích các giao thức mạng được cung miễn phí tại địa chỉ <http://www.wireshark.org/>

Sau khi cài đặt chạy chương trình wireshark, giao diện chương trình như sau:



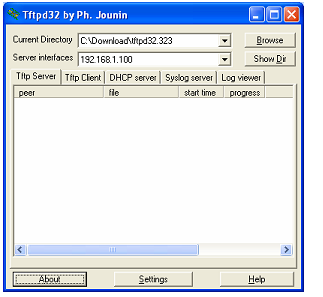
Để bắt đầu bắt gói để phân tích, từ menu Capture chọn Interface – chọn card mạng mà máy chúng ta đang chạy , rồi chọn “start” để bắt đầu bắt gói.



* 1. Sử dụng chương trình TFTPD32 làm DHCP server.

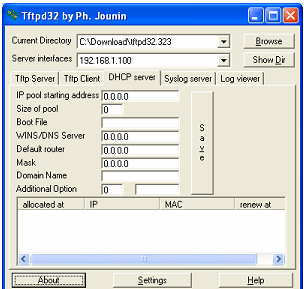
Chương trình TFTPD32 là một phần mềm cho phép xây dựng TFTP server, TFTP Client, DHCP Server, Syslog Server. TFTPD32 được cung cấp miễn phí tại địa chỉ: <http://tftpd32.jounin.net/>

Trong bài thí nghiệm này dùng TFTPD32 để xây dựng một DHCP Server. Sau khi cài đặt, khởi động chương trình với giao diện chính như sau:



Ta chọn Tab DHCP từ giao diện chương trình. Ý nghĩa của các trường chủ yếu như sau:

* IP pool starting address: địa chỉ IP bắt đầu để cung cấp cho các Client trong mạng.
* Size of pool: số lượng địa chỉ IP cung cấp cho các máy.
* WINS/DNS server: địa chỉ WINS hay DNS server cung cấp cho các Client.
* Default router: địa chỉ của “default gate way”.
* Mask: subnet mark cung cấp cho các Client.



### B. THỰC HÀNH

#### Dùng WireShark để phân tích quá trình ARP và ICMP

1. mô hình kết nối



1. chạy chương trình WireShark bắt gói cho hai máy.

Từ dấu nhắc DOS thực hiện xoá bảng ARP bằng lệnh: “arp –d”.

Kiểm tra lại bảng ARP của hai máy bằng lệnh “arp –a”. xem bảng arp của hai máy đã trống hay chưa.

Thực hiện lệnh ping từ máy A sang máy B. Có thành công hay không? ………..

Nếu không thành công thì nhờ GVHD thí nghiệm giúp đở.

Sau khi thực hiện thành công lệnh ping thì dừng quá trình bắt máy trên cả hai máy. Xem lại bảng ARP trên cả hai máy. Ghi lại kết quả này.

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

Xem địa chỉ IP và địa chỉ MAC của hai máy bằng lệnh : “ipconfig /all” . Rồi xem sự tương quan của chúng với các thông số địa chỉ IP và MAC trong gói ARP.

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

Phân tích gói ARP request và ARP reply và điền vào bảng sau:

Gói ARP request:

Layer 2 dest Address …………………. Layer source Address……………………

Layer 2 type for encapsulated data ………………………………………………...

Hardware type……………………….Layer 3 protocol type………………………

Hardware Addr Length…………….. Layer 3 Addr Length………………………

ARP operation code and name…………………………………………………….

Sender hardware address…………………………………………………………..

Sender IP address………………………………………………………………….

Target Hardware address…………………………………………………………..

Target IP address…………………………………………………………………..

Gói ARP reply:

Layer 2 dest Address …………………. Layer source Address……………………

Layer 2 type for encapsulated data ………………………………………………...

Hardware type……………………….Layer 3 protocol type………………………

Hardware Addr Length…………….. Layer 3 Addr Length………………………

ARP operation code and name…………………………………………………….

Sender hardware address…………………………………………………………..

Sender IP address………………………………………………………………….

Target Hardware address…………………………………………………………..

Target IP address…………………………………………………………………..

Phân tích quá trình gửi và nhận gói giữa hai máy tính thông qua các gói bắt được ………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

Phân tích các trường lớp 2 và lớp 3 của gói ICMP echo request và echo reply. Dữ liệu trong 2 gói echo request và echo reply là gì? Có giống nhau hay không? Mục đích của dữ liệu này là gì?

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

#### Dùng Wireshark để phân tích quá trình DHCP

#### Mô hình kết nối

Máy A đóng vai trò là DHCP server. Máy B đóng vai trò là DHCP client. Máy B sẽ được cấu hình các địa chỉ IP, DNS từ máy A.



Chạy chương trình TFTD32 trên máy A, điền các thông số sau vào Tab DHCP server rồi bấm Save. (Những thông nào không đề cập, sinh viên tự để trống).

* IP pool starting address: 192.168.1.2
* Size of pool: 100
* Mark: 255.255.255.0
* Domain name: TNTSL

Chạy chương trình Wireshark bắt đầu bắt máy cho cả hai máy. Từ dấu nhắc Dos trên máy B , cho máy B thực hiện lấy địa chỉ IP động bằng lệnh Ipconfig /renew. Sau khi máy B lấy xong địa chỉ IP từ DHCP server thì ngừng chương trình bắt gói.

Trên máy B thực hiện lệnh: ipconfig /all. Ghi lại kết quả của lệnh này:

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

Địa chỉ và các tham số khác nhận được trên máy B có giống với được cài đặt trên máy A không ……………………..

Phân tích quá trình DHCP dựa trên các gói bắt được?

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

#### Phân tích qus tình thiết lập và gở bỏ một kết nối TCP

#### Sinh viên thực hiện mô hình kết nối sau:



Trên máy A kích hoạt chức năng telnet. Trong cửa sổ Cmd.exe gõ lệnh: services.msc. Trong cửa sổ Services mới hiện ra, click phải vào dòng Telnet, chọn Properties, ở Tab General, chọn Starup type là Manual, rồi bấm vào nút Start. Chờ đợi quá trình kích hoạt telnet thành công.

Chạy chương trình Wireshark bắt đầu bắt gói cho cả hai máy.

Từ máy B thực hiện telnet tới máy A bằng lệnh: telnet 192.168.1.1 . Sau khi telnet thành công, sinh viên gõ một lệnh bất kỳ ở dấu nhắc DOS trong của sổ Telnet. (sinh viên có thể gõ lệnh help để được hướng dẫn). Sau đó thoát khỏi telnet bằng lệnh **exit** và dừng quá trình bắt gói.

Chọn vào một gói của kết nối Telnet mà Wireshark bắt được, chọn menu Statics>Flow Graph, thì có một cửa sổ mới hiện ra, sửa phần **Choose flow type** thành **TCP type** . Sinh viên trả lời các câu hỏi sau:

Dựa vào các gói Wireshark bắt được, phân tích quá trình thiết lập kết nối của TCP (ở đây là Telnet).

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

Dựa vào các gói bắt được của Wireshark, phân tích quá trình gửi dữ liệu của kết nối TCP (ở đây là Telnet)

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

Dựa vào các gói bắt được của Wireshark, phân tích quá trình giải toả kết nối của TCP (ở đây là Telnet)

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

Chọn vào một gói của kết nối Telnet, chọn menu **Analyzer>Follow TCP Stream** (Follow TCP Stream là chức năng dựng thông tin trao đổi của kết nối TCP dựa vào dữ liệu nhận được trong các gói).

Hãy nhận xét thông tin mới nhận được từu **Follow TCP Stream** so với thông tin nhận được từ kết nối thật….

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

Sinh viên hãy rút ra hoạt động chuyển dữ liệu của Telnet , tại sao Telnet được gọi là một “Terminal Emulator”.

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………Báo cáo kết quả thực hành

# BÀI 8: ĐIỀU CHẾ SỐ

## I. Mục đích yêu cầu

* Nắm vững kỹ thuật an toàn trong tất cả các thao tác thực hành
* Sinh viên điều chế được các loại điều chế số ASK, FSK, BPSK, QPSK

## II. Học cụ

* **Máy Tính (đã cài đặt phần mềm Matlab).**

## III. Nội Dung Thí Nghiệm – Thực hành

### A. Cơ sở Lý thuyết

Sinh viên sẽ được khảo sát và thực hiện quá trình điều chế số trên phần mềm Matlab.

Các loại tín hiệu mà sinh viên phải điều chế đó là:

* + ASK
  + FSK
  + BPSK
  + QBSK

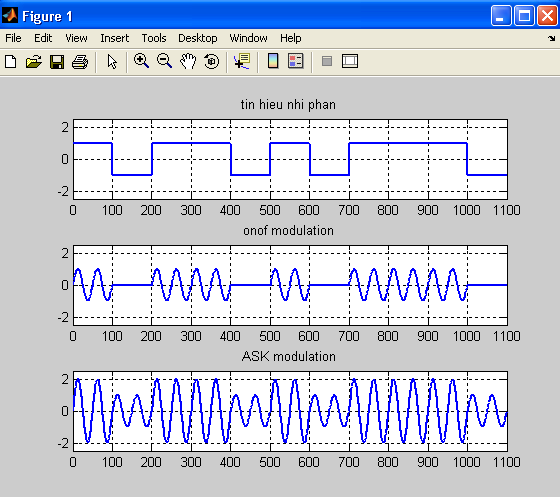
Về phần lý thuyết, không trình bày ở đây vì trong giáo trình truyền số liệu đã nói rất kỹ. Ở đây chúng ta chỉ khảo sát việc sử dụng phần mềm Matlab để tạo ra chúng.

### B. Thực hành

Phần source code matlab hổ trở cho sinh viên sẽ được GVHD cho sinh viên trước bài thí nghiệm một buổi. Vì source code hơi dài nên không trình bày ở đây.

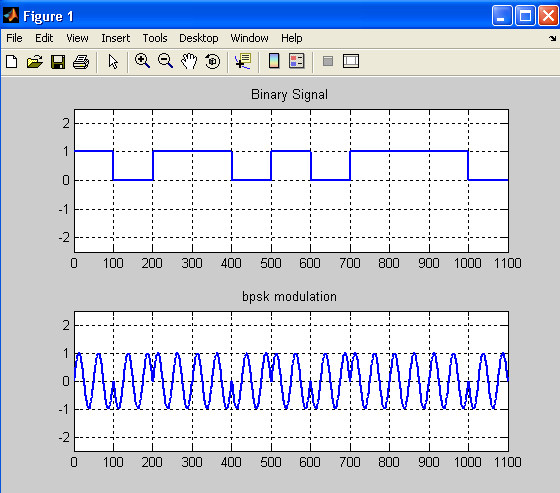
Điều chế ASK.

Sinh viên thực hiện điều chế số ASK dưới sự hướng dẫn của GV để có được kết quả sau:



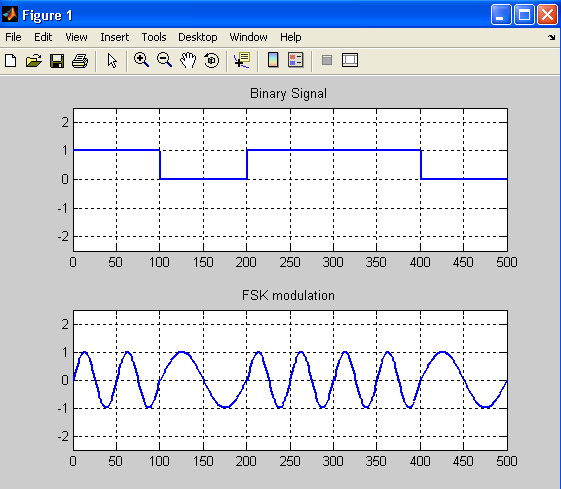
#### Điều chế BPSK

Sinh viên thực hiện điều chế số BPSK dưới sự hướng dẫn của GV để có được kết quả sau:



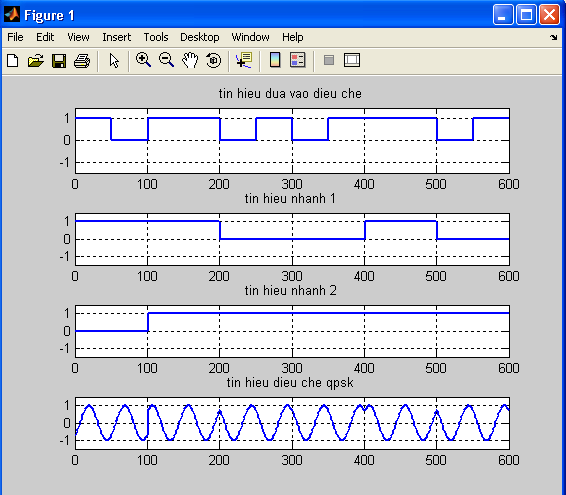
#### Điều chế FSK

Sinh viên thực hiện điều chế số FSK dưới sự hướng dẫn của GV để có được kết quả sau:



#### Điều chế QPSK

Sinh viên thực hiện điều chế số BPSK dưới sự hướng dẫn của GV để có được kết quả sau:



# BÀI 9: TRUYỀN NHẬN DỮ LIỆU GIỮA KIT ESP32 VỚI CLOUDING, ĐIỆN THOẠI THÔNG QUA WIFI

## I. Mục đích yêu cầu

* Nắm vững kỹ thuật an toàn trong tất cả các thao tác thực hành
* Lập trình giao tiếp KIT ESP32 theo chuẩn WiFi để gửi dữ liệu lên clouding
* Lập trình giao diện điện thoại để thu thập giám sát dữ liệu trên clouding.

## II. Học cụ

* **Máy Tính**
* **Kit ESP32**
* **300Mbps Wireless N Router TP- Link TL – WR845N**
* **Kềm bấm dây**
* **Bus 1 (đực cái).**

## III. Nội Dung Thí Nghiệm – Thực hành

### A. Cơ sở Lý Thuyết

1. Giới thiệu về KitKit RF thu phát Wifi BLE ESP32 NodeMCU

Kit RF thu phát Wifi BLE ESP32 NodeMCU LuaNode32 được phát triển trên nền module trung tâm là ESP32 với công nghệ Wifi, BLE và nhân ARM SoC tích hợp mới nhất hiện nay, kit có thiết kế phần cứng, firmware và cách sử dụng tương tự Kit NodeMCU ESP8266, với ưu điểm là cách sử dụng dễ dàng, ra chân đầy đủ, tích hợp mạch nạp và giao tiếp UART CP2102, Kit Wifi BLE ESP32 NodeMCU LuaNode32 là sự lựa chọn hàng đầu trong các nghiên cứu, ứng dụng về Wifi, BLE và IoT.

Thông số kỹ thuật như sau:

* Module trung tâm: [Wifi BLE Soc ESP32.](http://hshop.vn/products/mach-thu-phat-wifi-ble-soc-esp32-v1)
* Nguồn sử dụng: 5VDC từ cổng Micro USB.
* Tích hợp mạch nạp và giao tiếp UART CP2102
* Ra chân đầy đủ module ESP32, chuẩn chân cắm 2.54mm
* Tích hợp Led Status, nút BOOT và ENABLE.
* Kích thước: 28.33x51.45mm

1. Giới thiệu Lưu Trử trên Clouding của Google

Tài liệu về Clouding của Google tham khảo tại đây: <http://vn.apps-gcp.com/gioi-thieu-ve-google-cloud-platform-va-cac-dich-vu/>

1. Code Arduino mẫu cho việc truyền nhận dữ liệu từ google sheet

|  |
| --- |
| #include <ESP32WiFi.h>  #include "HTTPSRedirect.h"  const char\* ssid = "";  const char\* password = "";  // The ID below comes from Google Sheets.  // Towards the bottom of this page, it will explain how this can be obtained  const char \*GScriptId = "AKfycbzjj1Aeydood19No-S4TDuUP-VvmAxb-HkDvMuDB\_982OMhFmA";  // Push data on this interval  //const int dataPostDelay = 900000; // 15 minutes = 15 \* 60 \* 1000  const int dataPostDelay = 60000;  const char\* host = "script.google.com";  const char\* googleRedirHost = "script.googleusercontent.com";  const int httpsPort = 443;  HTTPSRedirect client(httpsPort);  // Prepare the url (without the varying data)  String url = String("/macros/s/") + GScriptId + "/exec?";  const char\* fingerprint = "";// "F0 5C 74 77 3F 6B 25 D7 3B 66 4D 43 2F 7E BC 5B E9 28 86 AD";  // We will take analog input from A0 pin  const int AnalogIn = A0;  long t\_pos\_gg=0;  void setup() {  Serial.begin(115200);  Serial.println("Connecting to wifi: ");  Serial.println(ssid);  Serial.flush();  WiFi.begin(ssid, password);  while (WiFi.status() != WL\_CONNECTED) {  delay(500);  Serial.print(".");  }  Serial.println(" IP address: ");  Serial.println(WiFi.localIP());    Serial.print(String("Connecting to "));  Serial.println(host);  bool flag = false;  for (int i=0; i<5; i++){  int retval = client.connect(host, httpsPort);  if (retval == 1) {  flag = true;  break;  }  else  Serial.println("Connection failed. Retrying...");  }  // Connection Status, 1 = Connected, 0 is not.  Serial.println("Connection Status: " + String(client.connected()));  Serial.flush();    if (!flag){  Serial.print("Could not connect to server: ");  Serial.println(host);  Serial.println("Exiting...");  Serial.flush();  return;  }  // Data will still be pushed even certification don't match.  if (client.verify(fingerprint, host)) {  Serial.println("Certificate match.");  } else {  Serial.println("Certificate mis-match");  }  }  // This is the main method where data gets pushed to the Google sheet  void postData(String tag, float value){  if (!client.connected()){  Serial.println("Connecting to client again...");  client.connect(host, httpsPort);  }  String urlFinal = url + "tag=" + tag + "&value=" + String(value);  client.printRedir(urlFinal, host, googleRedirHost);  }  // Continue pushing data at a given interval  void loop() {    // Read analog value, in this case a soil moisture  int data = 1023 - analogRead(AnalogIn);  // Post these information  if (millis()-t\_pos\_gg>60000)  {  postData("SoilMoisture", data);  t\_pos\_gg = millis();  }  //delay (dataPostDelay); //60 giay toi post len google  //ESP.restart();  } |

1. **Code html giao diện website trên điện thoại**

|  |
| --- |
| <html>  <head>  <script>  function InitHTML(){  document.getElementById("table\_displayGGS").style.visibility = "hidden";    }  //init load html don't allow the table display  function OpenGGS\_Func(){  document.getElementById("table\_displayGGS").style.visibility = "";  }  function CloseGGS\_Func(){  document.getElementById("table\_displayGGS").style.visibility = "hidden";    }    </script>  </head>  <body onload ="InitHTML()">  <button id = "viewGG" onclick = "OpenGGS\_Func()" >View Google Sheet</button>  <p>  <table id = "table\_displayGGS">  <iframe id = "displayGG" src = "https://docs.google.com/spreadsheets/d/1SdqfW-ye9HTBM0-OTkDl9L314qx-vrYZZ-odvtrqm14/edit#gid=1107747444" height = "400px" width = "600px"></iframe>  <br></br>  <button id = "CloseGG" onclick = "CloseGGS\_Func()" >Close Google Sheet</button>  </table>  </p>  </body>  </html> |

### B. Thực Hành

#### Viết chương trình Arduino, cứ 10s gửi tín hiệu chân analog a0 lên google Sheet

#### Viết chuơng trình giao diện html và xuất bản thành app trên android để điện thoại theo dõi thông số a0.

#### Báo cáo kết quả thực hành

**MỤC LỤC:**

[BÀI 1: GIỚI THIỆU PHÒNG THỰC HÀNH VÀ PHẦN MỀM HỌC TẬP 1](#_Toc8931966)

[I. Mục đích yêu cầu 1](#_Toc8931967)

[II. Học cụ 1](#_Toc8931968)

[III. Nội Dung Thí Nghiệm – Thực hành 1](#_Toc8931969)

[Giới thiệu phòng thực hành 1](#_Toc8931970)

[Giới thiệu nội dung môn học 1](#_Toc8931971)

[Giới thiệu và hướng dẫn sử dụng nhanh 5 phần mềm học tập: 1](#_Toc8931972)

[BÀI 2: HÀN CÁP SERIAL, BẤM CÁP TIVI, BẤM CÁP MẠNG 2](#_Toc8931973)

[I. Mục đích yêu cầu 2](#_Toc8931974)

[II. Học cụ 2](#_Toc8931975)

[III. Nội Dung Thí Nghiệm – Thực hành 2](#_Toc8931976)

[A. LÝ THUYẾT: 2](#_Toc8931977)

[B. THỰC HÀNH: 4](#_Toc8931978)

[BÀI 3: LẬP TRÌNH GIAO TIẾP HAI MÁY TÍNH THÔNG QUA CỔNG NỐI TIẾP 5](#_Toc8931979)

[I.Mục đích yêu cầu 5](#_Toc8931980)

[II.Học cụ 5](#_Toc8931981)

[III.Nội Dung Thí Nghiệm – Thực hành 5](#_Toc8931982)

[A.LÝ THUYẾT: 5](#_Toc8931983)

[B.THỰC HÀNH: 11](#_Toc8931984)

[BÀI 4: LẬP TRÌNH GIAO TIẾP GIỮA KIT ESP32 VỚI MÁY TÍNH 13](#_Toc8931985)

[I. Mục đích yêu cầu 13](#_Toc8931986)

[II. Học cụ 13](#_Toc8931987)

[III. Nội Dung Thí Nghiệm – Thực hành 13](#_Toc8931988)

[A.LÝ THUYẾT: 13](#_Toc8931989)

[B.THỰC HÀNH: 14](#_Toc8931990)

[\_Lập giao diện VB 14](#_Toc8931991)

[\_Code cho ESP32 18](#_Toc8931992)

[C.*Câu hỏi bài thực hành* 20](#_Toc8931993)

[Báo cáo bài thực hành 20](#_Toc8931994)

[BÀI 5: LẬP TRÌNH GIAO TIẾP CỔNG LAN THÔNG QUA WINSOCK 21](#_Toc8931995)

[I. Mục đích yêu cầu 21](#_Toc8931996)

[II. Học cụ 21](#_Toc8931997)

[III. Nội Dung Thí Nghiệm – Thực hành 21](#_Toc8931998)

[A. LÝ THUYẾT: 21](#_Toc8931999)

[B. THỰC HÀNH: 22](#_Toc8932000)

[\_Tạo giao diện trên máy chủ: 22](#_Toc8932001)

[\_Tạo giao diện trên máy Client: 23](#_Toc8932002)

[\_Chương trình trên máy chủ: 23](#_Toc8932003)

[\_Chương trình trên máy Client 26](#_Toc8932004)

[\_Câu hỏi thực hành 29](#_Toc8932005)

[\_Báo cáo thực hành 29](#_Toc8932006)

[BÀI 6: XÂY DỰNG MÔ HÌNH MẠNG LAN – WAN 30](#_Toc8932007)

[I. Mục đích yêu cầu 30](#_Toc8932008)

[II. Học cụ 30](#_Toc8932009)

[III. Nội Dung Thí Nghiệm – Thực hành 30](#_Toc8932010)

[A. CƠ SỞ LÝ THUYẾT 30](#_Toc8932011)

[B. THỰC HÀNH 32](#_Toc8932012)

[Xây dựng mô hình mạng pear to pear 32](#_Toc8932013)

[Xây dựng mô hình mạng Switch Base. 33](#_Toc8932014)

[Xây dựng mạng Router Based 34](#_Toc8932015)

[Thực hiện mô hình LAN trong phòng thí nghiệm 35](#_Toc8932016)

[Thực hiện mô hình Wan trong phòng thí nghiệm 36](#_Toc8932017)

[Báo cáo kết quả thực hành 37](#_Toc8932018)

[BÀI 7: PHÂN TÍCH CÁC PROTOCOL THÔNG DỤNG CỦA TCP/IP 38](#_Toc8932019)

[I. Mục đích yêu cầu 38](#_Toc8932020)

[II. Học cụ 38](#_Toc8932021)

[III. Nội Dung Thí Nghiệm – Thực hành 38](#_Toc8932022)

[A. CƠ SỞ LÝ THUYẾT 38](#_Toc8932023)

[B. THỰC HÀNH 49](#_Toc8932024)

[Dùng WireShark để phân tích quá trình ARP và ICMP 49](#_Toc8932025)

[Dùng Wireshark để phân tích quá trình DHCP 52](#_Toc8932026)

[Mô hình kết nối 52](#_Toc8932027)

[Phân tích qus tình thiết lập và gở bỏ một kết nối TCP 53](#_Toc8932028)

[Sinh viên thực hiện mô hình kết nối sau: 53](#_Toc8932029)

[BÀI 8: ĐIỀU CHẾ SỐ 56](#_Toc8932030)

[I. Mục đích yêu cầu 56](#_Toc8932031)

[II. Học cụ 56](#_Toc8932032)

[III. Nội Dung Thí Nghiệm – Thực hành 56](#_Toc8932033)

[A. Cơ sở Lý thuyết 56](#_Toc8932034)

[B. Thực hành 56](#_Toc8932035)

[Điều chế ASK 56](#_Toc8932036)

[Điều chế BPSK 57](#_Toc8932037)

[Điều chế FSK 58](#_Toc8932038)

[Điều chế QPSK 59](#_Toc8932039)

[BÀI 9: TRUYỀN NHẬN DỮ LIỆU GIỮA KIT ESP32 VỚI CLOUDING, ĐIỆN THOẠI THÔNG QUA WIFI 60](#_Toc8932040)

[I. Mục đích yêu cầu 60](#_Toc8932041)

[II. Học cụ 60](#_Toc8932042)

[III. Nội Dung Thí Nghiệm – Thực hành 60](#_Toc8932043)

[A. Cơ sở Lý Thuyết 60](#_Toc8932044)

[B. Thực Hành 64](#_Toc8932045)

[Viết chương trình Arduino, cứ 10s gửi tín hiệu chân analog a0 lên google Sheet 64](#_Toc8932046)

[Viết chuơng trình giao diện html và xuất bản thành app trên android để điện thoại theo dõi thông số a0. 64](#_Toc8932047)

[Báo cáo kết quả thực hành 64](#_Toc8932048)