

eBLOCKS[®]

LEARN·DESIGN·BUILD

Embedded Internet Communications



EB921-80-07

MATRIX

www.matrixmultimedia.com

EB921

TCP/IP

Solution

Course notes

Contents

1	Getting started.....	6
1.1	Required hardware	6
1.2	Test routine.....	6
1.3	Required software.....	7
1.4	Additional software	7
1.5	Documentation	7
1.6	Examples and exercises	7
1.7	Disclaimer	7
2	Introduction.....	8
2.1	Using this course	8
2.2	Who this course is aimed at.....	8
3	TCP/IP basic concepts	9
3.1	What is TCP/IP?.....	9
3.2	OSI layers	9
3.3	TCP_IP component modes	10
3.4	Frames and datagram's - Messages within messages.....	10
3.4.1	The Ethernet physical layer	11
3.4.2	Onto the network with IP.....	11
3.4.3	Delivering data with TCP	12
3.4.4	Applications and content	12
3.5	Other Protocols.....	12
3.5.1	ARP – Address Resolution Protocol.....	12
3.5.2	ICMP – Internet Control Message Protocol.....	12
3.5.3	UDP – User Datagram Protocol.....	12
4	Hardware and software	13
4.1	Base systems	13
4.1.1	E-Blocks backplane	13
4.1.2	FACET board	13
4.2	Port connections	14
4.3	Clock selection	14
4.4	Practical limitations for the microcontroller.....	15
4.5	The internet board	15
4.6	Internet board setup	15
4.7	Flowcode and the TCP_IP component.....	15
4.8	Basic direct connection setup	16
4.9	Dual Ethernet board system.....	17
4.10	Network monitoring software.....	17
4.10.1	Debugging with Network traffic analysis tools.....	18
4.11	Packet injectors.....	18
4.11.1	Debugging with Packet injectors	19
4.12	Server configuration	19
5	The TCP_IP component.....	20
5.1	TCP_IP properties	20
5.2	TCP_IP component macros	20
6	Ethernet layer	21
6.1	Overview	21
7	Address Resolution Protocol.....	23
7.1	What's it used for?	23
7.1.1	ARP in action: finding the MAC address for a message	23
7.2	Ethernet frame: The outer wrapper.....	24
7.3	The ARP datagram	24
8	Implementing Ethernet mode in Flowcode	26
8.1.1	Initializing.....	26
8.1.2	Creating a MAC socket	26
8.1.3	Sending data	26
8.1.4	Reading in the data	27
8.1.5	Matching data	27
9	Exercise 1: ARP scanner.....	28

10	Worked Example: Implementing ARP in Flowcode.....	29
10.1	Network traffic analysis	29
10.2	The basic structure.....	29
10.3	Initializing	30
10.4	Sending the ARP request.....	31
10.4.1	ARP datagram.....	31
10.4.2	Sending the message.....	32
10.5	Getting a response	32
10.6	Timeout	33
10.7	Displaying the response.....	33
10.8	Finishing off the program.....	33
10.9	Example program	34
	IP layer	35
11	Using the IP layer: ICMP and Ping	37
11.1	The ICMP datagram	37
11.2	ICMP Ping data	38
12	Implementing IP mode in Flowcode.....	39
12.1.1	Where are the Ethernet and IP headers?	39
12.1.2	Setting up the IP connection	39
12.1.3	Sending the datagram.....	39
12.1.4	Receiving a response.....	39
12.2	The Windows Ping program.....	39
13	Exercise 2: Ping program.....	41
14	Worked example: Ping.....	42
14.1.1	Ping overview	42
14.1.2	Setting up the IP connection	42
14.1.3	The ICMP header values.....	43
14.1.4	The checksum	43
14.1.5	The data	44
14.1.6	Sending the ICMP datagram.....	44
14.1.7	Receiving a response.....	44
14.1.8	Ping reply.....	45
14.1.9	Ping request	45
14.1.10	Expanding the Ping program	45
14.1.11	Example program.....	45
15	UDP	46
15.1	Sockets	46
15.2	Predefined responses and reserved ports	46
15.3	Ports and firewalls	46
15.4	The UDP datagram.....	47
15.5	Handling UDP data	47
16	Implementing UDP mode in Flowcode.....	48
16.1	Port bytes.....	48
16.2	Setting up a UDP connection	48
16.3	Sending data	48
16.4	Receiving data	49
16.5	Using UDP	50
17	Exercise 3: Time and date using UDP mode.....	51
18	Notes for Exercise 3: Time and date using UDP mode.....	52
18.1	Firewall warning.....	52
18.2	General notes.....	52
18.3	Program flowchart.....	52
18.4	Example program	53
19	TCP.....	54
19.1	Connections.....	54
19.2	Acknowledgements	54
19.3	Fragmentation.....	55
19.4	TCP state diagram	55
19.4.1	Passive open.....	56
19.4.2	Active open.....	56

19.4.3	Closing the connection.....	56
19.4.4	Responding to a close request.....	56
20	Implementing TCP mode in Flowcode	57
20.1	Initializing a TCP connection.....	57
20.1.1	Active connection with TCP_Connect.....	57
20.1.2	Passive connections with TCP_Listen	57
20.1.3	Sending data.....	58
20.1.4	Checking for and receiving data	58
20.1.5	Closing a connection.....	58
20.1.6	<CRLF>, quotes and other problem characters	58
20.2	Communication sequences	58
21	Exercise 4: Sending an HTML page using HTTP.....	60
21.1	Instructions	60
22	Notes on Exercise 4: Sending an HTML page using HTTP	61
22.1	Prerequisites	61
22.2	Useful tools	61
22.3	Sample HTML.....	61
22.4	Program flowchart.....	62
22.4.1	Creating the connection	62
22.4.2	Checking for the web page request	62
22.4.3	Sending the data.....	63
22.4.4	Closing the connection.....	63
22.5	Spicing up the page	63
22.6	Suggestions for further work	63
22.6.1	Variable markers.....	63
22.6.2	Multiple pages.....	63
22.6.3	Handling errors with multiple pages.....	63
22.6.4	Images and other files	63
22.7	Example program	64
23	Exercise 5: Receiving HTML	65
23.1	Instructions	65
24	Notes on Exercise 5: Receiving HTML	66
24.1	Prerequisites	66
24.2	Useful tools	66
24.3	Receiving the HTML.....	66
24.4	Program overview	66
24.5	Requesting the HTML page	67
24.6	Further work.....	68
24.7	Example program	68
25	Exercise 6: Sending an SMTP email message	69
25.1	Instructions	69
26	Notes for Exercise 6: Sending an SMTP email message	70
26.1	SMTP email.....	70
26.2	Key transmission messages	70
26.3	SMTP Acknowledgment codes	70
26.4	Sending an Email message step by step	71
26.5	Sample email message to sent:	71
26.6	Implementing the SMTP program in Flowcode	72
26.6.1	Hardware considerations.....	72
26.6.2	<CRLF>, quotes and other problem characters	72
26.7	Example program	73
27	Advanced Exercise 1: Custom messaging using UDP.....	74
28	Notes for Advanced Exercise 1: Custom messaging using UDP	75
28.1	Hardware setup.....	75
28.2	Notes on the programs.....	75
28.3	Program overview	76
28.4	Further work.....	76
29	Advanced Exercise 2: Firewall application design	77
29.1	Hardware considerations	77
29.2	Security Criteria	77

29.3	Program design.....	77
30	Further work	79
30.1	Fragmentation.....	79
30.2	Header options.....	79
30.3	Error checking	79
30.4	Message data	79
30.5	Other protocols	79
31	Web links and resources	80
32	Glossary	80
32.1	Acronyms	80
32.2	Glossary.....	80

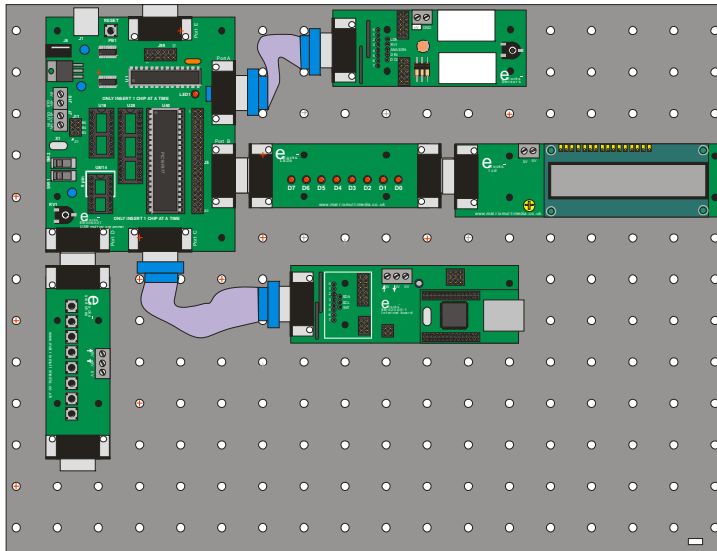
1 Getting started

The following information is designed to aid you in getting the E-blocks internet solution up and running.

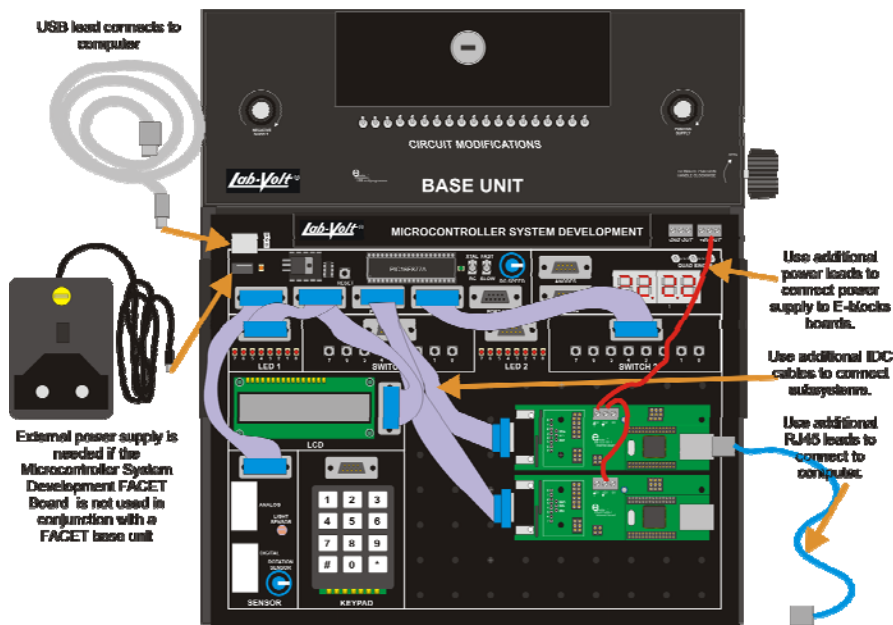
1.1 Required hardware

This document is designed for use with either of the following hardware systems:

- EB643 E-blocks internet solution



- FACET board with additional EB-023 IINTERNET BOARD(s)



Install and setup instructions for the E-blocks internet solution can be found on the datasheets supplied on the E-Blocks support section of the Matrix website.

<http://www.matrixmultimedia.com/eblocks/index.php>

1.2 Test routine

A test routine for the internet board and internet solution is supplied on the internet solution CD in the Test_file folder. Details on testing the internet solution can be found in the internet board datasheet.

Additional test routines are available for the other E-blocks on the E-blocks support section of the Matrix website: <http://www.matrixmultimedia.com/eblocks/index.php>

1.3 Required software

The example programs supplied on the TCP/IP Solutions CD require Flowcode V4.1 or higher to be installed on the host PC. A compatible version of Flowcode is supplied with the solution. Please refer to the contents of the Flowcode CD for instructions on installation and operation.

1.4 Additional software

Additional software to aid in learning TCP/IP is included in the Additional_Software folder, and its subfolders, on the FC487 CD supplied as part of the E-block internet solution.

The additional software includes copies of:

- Ethernet Network traffic analyzer
- Excalibur packet injector

Install instructions and details of the additional software will be found in the Additional_Software folder.

1.5 Documentation

Documentation for the various E-blocks and other items supplied as part of the internet solution is provided on the E-blocks members' area of the Matrix Multimedia web site:

<http://www.matrixmultimedia.com/eblocks/index.php>

1.6 Examples and exercises

The TCP/IP Solution CD contains a folder of example and exercise programs that can be used in conjunction with this document

1.7 Disclaimer

The information contained here is correct at time of going to press (Jan 2007), but may be superseded or changed at a later date. If this information is superseded a revised Getting started document will be issued with the E-blocks internet solution.

2 Introduction

The TCP/IP Course is designed to introduce you to the concepts required to understand the communication protocols generally referred to as TCP/IP, including the Ethernet, TCP, IP, UDP and other protocols.

This course is carried out using Flowcode, which is shipped with the TCP/IP solution. Flowcode uses the TCP/IP component which has a number of properties and macros to allow the student to build up the system they require. This allows students to learn about TCP/IP without getting bogged down in the problems of programming in C or a lower level language.

2.1 Using this course

This course goes through the various layers of TCP/IP starting with the lowest physical communication layer and moving up to application data communications. The course covers the various TCP/IP layers and protocols and the Flowcode TCP_IP component modes used to implement them.

A number of exercises are included along with notes on how implement the exercise in Flowcode. For the first few protocols the notes are relatively extensive, forming a kind of walkthrough of the exercise. For later exercises the notes become more geared towards explaining specifics and how the protocol differs from those previously covered. By that stage the student should have sufficient understanding of the basics involved to complete the task without extensive notes. Example solutions are provided for the exercises for demonstration, discussion and use as a starting point for further programming.

2.2 Who this course is aimed at

This course is aimed at two main groups: Electronics technicians seeking to use TCP/IP communications and Network technicians seeking to understand TCP/IP data structure.

- Electronics technicians will generally have some experience with Microcontrollers, and will be mostly concerned with implementing TCP/IP communications from a practical point of view – i.e. how to create programs and send data.
The general goal of the Electronics technician will be to send messages.
- Network technicians and Computer will be more concerned with gaining an understanding of the communications process itself i.e. datagram's and frameworks, and how this relates to the data viewable with network analyzer tools.
The general goal of the Network technician will be to understand and debug messages on the network, and to troubleshoot communications errors.

Both groups are catered for in equal measure, with both the theory and the practical aspects being covered. It will be up to the teacher involved to place the emphasis during teaching.

3 TCP/IP basic concepts

3.1 What is TCP/IP?

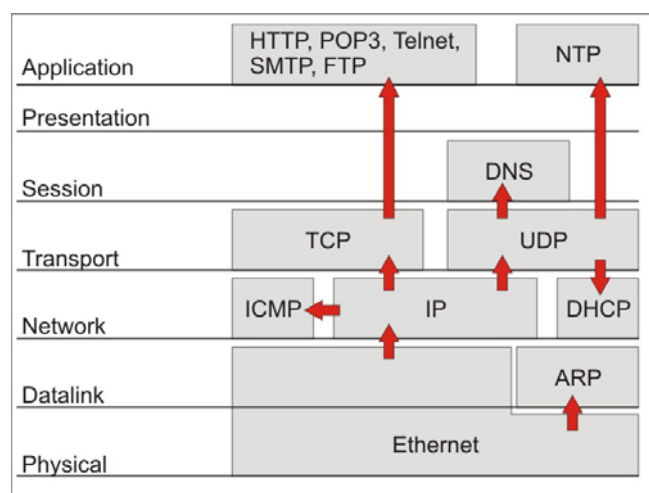
TCP stands for **T**ransmission **C**ontrol **P**rotocol. And IP stands for **I**nternet **P**rotocol. Some other common acronyms that we will be using throughout the course include: **M**AC – **M**edia **A**ccess **C**ontrol, **U**DP – **U**ser **D**atagram **P**rotocol, **I**CMP – **I**nternet **C**ontrol **M**essage **P**rotocol and **A**RP – **A**ddress **R**esolution **P**rotocol. Other acronyms will be introduced as they occur, and can also be found in the glossary at the end.

We can refer to TCP/IP in both a general and a specific way. The general term TCP/IP is used here to refer to the messaging system that transfers data between computers via the internet. The end result can be Email, web pages, or a data file. The process of transferring the message will involve a number of layered communication protocols. One of these can be TCP. One of them could be IP. And there are others – Ethernet, ICMP, UDP etc. This is where the specific definition of TCP/IP comes in, or rather the specific definition of TCP and the specific definition of IP. TCP and IP are two of the more important protocols used in internet communications, hence the general term TCP/IP to describe the whole process.

Note: Here we will use *TCP/IP* to refer to the general system, and *TCP* and *IP* to refer to the specific protocols.

3.2 OSI layers

TCP/IP is a multi-layered system with higher layers becoming increasingly abstracted from the serial data communications of the first Physical Layer. At the Physical and Datalink layers the prime consideration is shunting the data to and fro between specific nodes – usually a PC and a hub or router. The Network level introduces systems to specify the end receiver thus allowing the network to decide where data goes and how it gets there. Built on top of these the higher level protocols, such as TCP and UDP, are more concerned with getting the data to the computer applications that use the data. Even higher up the level structure are the applications that can either handle the data for us, or start the procedure of sending or requesting a message for us.



In the higher level layers more and more of the data transmission process becomes automated. For example users do not need to know how the routers will find the correct recipient for an IP data transmission – they only need to know that the routers can. Provide the correct IP address and the data will get there whether it's across the room, or across the world.

In the higher level layers more and more consideration is given to error detection and transmission sequencing, and it is likely that the receiving system will expect a two-way dialogue with data being passed in a specific sequence, and with error codes being returned at specific times. Checking for these codes and dealing with any errors means a corresponding increase in data processing.

The Application layer protocols are the parts that we normally associate TCP/IP with – web browsers and email programs etc. Many would think of them as TCP/IP, but in truth they just manipulate and display the raw data sent by TCP/IP. What you see in an email program is not SMTP (Simple Mail Transfer Protocol), but a visual representation of the data sent via TCP/IP in the SMTP format.

3.3 TCP_IP component modes

The TCP_IP component has four modes that can be used as entry points into the OSI model.

- MAC mode
Brings the user in at the bottom physical level.
- IP mode
Brings the user in at the Network level.
Automatically handles the Physical layer procedures.
- UDP mode
Brings the user in at the Transport level using the UDP protocol.
Useful for applications such as DNS or DHCP that are built on UDP.
- TCP mode
Brings the user in at the Transport level using TCP datagram's.
Useful as a starting point for applications such as SMTP or HTTP that are built on TCP.

The modes automate the levels below them, so implementing the TCP mode for instance will handle the Network (IP) and Physical (MAC) layers automatically. Headers are also handled automatically in all but MAC mode allowing you to concentrate more on the data processes.

3.4 Frames and datagram's - Messages within messages

The most basic message consists of a stream of serial data with a header section giving information about the data. As all messages sent in TCP/IP need this base level, a mechanism is needed to adapt this to serve the many purposes and formats needed for the different protocol layers. The mechanism chosen was to put the message meant for a higher protocol layers inside the data section of a lower layer. A TCP message is put into the data section of an IP message which is put inside the data section of an Ethernet section. As the message gets passed up the layers these outer wrappings can be discarded until the message finally gets to the level that is meant to handle it.