

The Internet of Things

Lecture 14

Microprocessor-based Systems (E155)

Prof. Josh Brake

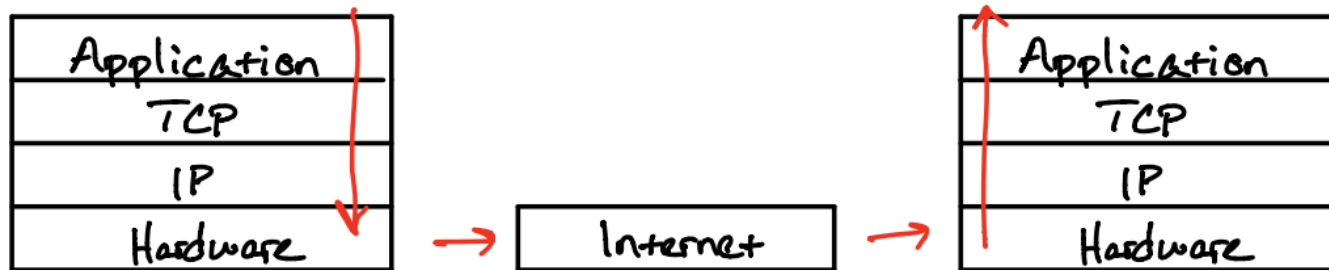


Outline

- General Internet Architecture
 - Protocol layers
 - Browsing the Web
 - HTTP - Commands and Format
 - HTML - Hypertext Markup Language
- ESP8266
 - Overview
 - Lab 7 Webserver Code #Demo
 - Basic workflow for whole system

Protocol Layers

IP - Internet Protocol Address



TCP: Transmission Control Prot.
IP: Internet Protocol
Hardware: Network card, modem, etc.

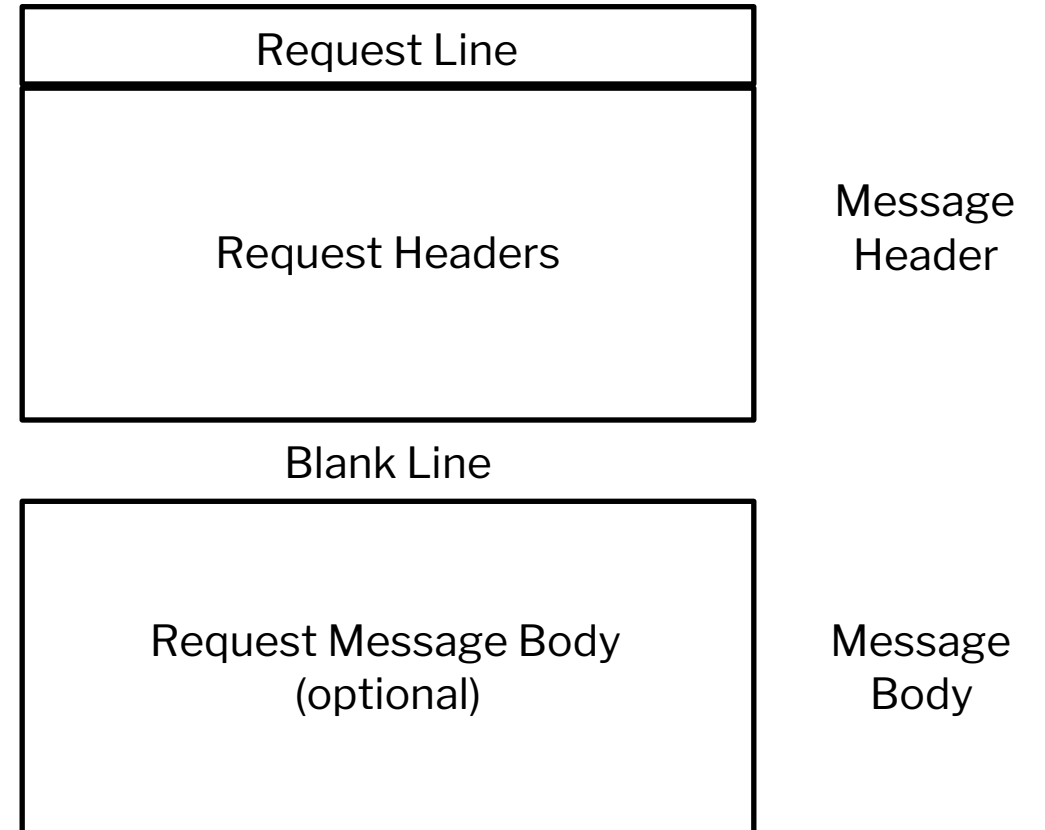
- Worldwide web is a service on the Internet
- Uses Hypertext Transfer Protocol (HTTP)
 - What layer is this protocol at?
- URL: Uniform Resource Locator
 - URL format:
`<protocol>://<hostname>:<port>/<path_and_filename>`

Browsing the Web

- What happens when you type in a URL?
 - Finds IP for domain if necessary (Using Dynamic Nameserver (DNS))
 - Connects to server, send HTTP request
 - Server receives request, searches for desired page.
 - If it exists, sends it.
 - If not, sends 404 "Page Not Found" error code.
 - Web browser gets page, closes connection
 - Parses webpage sending HTTP requests as necessary to get all the elements

HTTP: Commands and Format

- GET
 - Most common
 - Used to request a resource
 - Format
 - GET / HTTP/1.1 Host: Accept



HTML – Hypertext Markup Language

- Simple text format to specify webpage formatting
- Elements
 - DOCTYPE statement
 - HTML tag
 - Head
 - Body
- Tags look like `<tag>...</tag>`
 - Common tags: `html`, `head`, `body`, `p`, `h x={1,2,3}`, `title`

Activity: Simple HTML Page

- Open text editor (e.g., VSCode)
- Save document as .html
- Example webpage

demo_webpage.html

```
<!DOCTYPE html>
<head>
  <title>My First Webpage</title>
</head>
<body>
  <h1>E155 Demo</h1>
  <p>Put text here!</p>
</body>
```

Other HTML elements

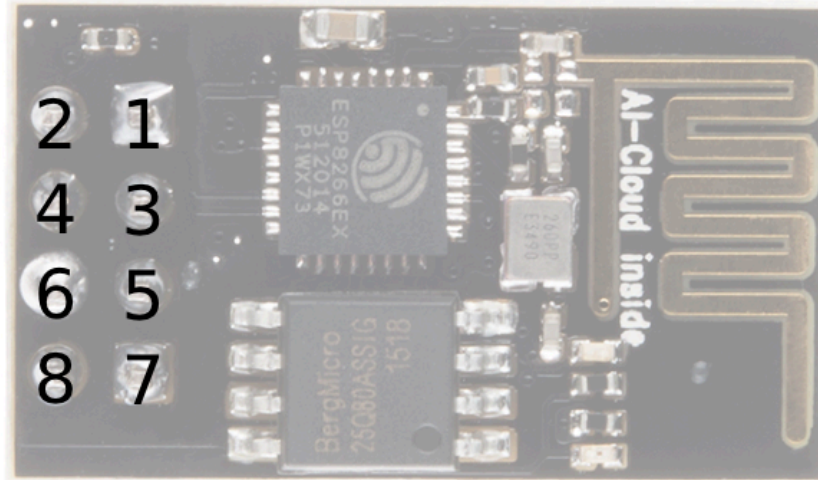
- Other HTML elements
 - Form
 - Attributes
 - type - submit
 - action - where to send form data
 - value - text on button
 - Add form to webpage

```
<form action="action_key">  
  <input type="submit" value="Send GET request">  
</form>
```


ESP8266

Overview

D7	GPIO1	TX	2- TXO
		Chip Enable	4- CHPD
		Reset	6- RST
		3.3V	8- 3V
		GND	1- GND
	D2/SDA	GPIO2	3- GPIO2
	D0	GPIO0	5- GPIO0
D8	GPIO3	RX	7- RXI



PCB Antenna



Note: 3.3V!

Overview

Power

VCC-3.0-3.6V

Standby ~ 0.9uA

Running ~60-215mA,

Average ~ 80mA

Wifi Features

802.11 b/g/n

2.4GHz

WPA/WPA2

Wifi Direct

+20dBm output power (802.11b)

I/O Features

Integrated TCP/IP

Integrated TR switch, LNA,
balun

Memory/Speed Features

80MHz

64KB instruction RAM

96KB data RAM

64K boot ROM

1MB* Flash Memory

Basic Connection

VCC - 3.3V

GND - GND

TX - RX on Arduino or FTDI

RX - TX on ARduino or FTDI

Chip Enable - 3.3V

Default Baud Rate

11520* 8N1

LEDs

Red: Power

Blue: TX

*milage may vary on different
version of the board

AT Command List

- Set = AT+<x>=<. . . > - Sets the value
- Inquiry = AT+<x>? - See what the value is set at
- Test = AT+<x>=? - See the possible options
- Execute = AT+<x> - Execute a command

Must make sure to end each command with carriage return and new line (\r\n)

See documentation for more details!

Deliverables

- Project proposal
 - 2-page proposal
 - What are you going to build?
 - Be clear about deliverables, no stretch goals or wiggle room.
 - Include a budget for supplies
- Status report
 - 4-page report (plus appendices)
 - Joint team checkoff over Zoom
- Problem Presentations
- Project Demonstration

Demo: Webserver Code

```
// main.c
// Josh Brake
// jbrake@hmc.edu
// 9/30/20

#include "STM32F401RE.h"
#include "main.h"
#include <string.h>

#define ESP_USART_ID USART1_ID
#define TERM_USART_ID USART2_ID
#define DELAY_TIM TIM2
#define CMD_DELAY_MS 20

/*
| Initialize the ESP and print out IP address to terminal
*/
void initESP8266(USART_TypeDef * ESP_USART, USART_TypeDef * TERM_USART){ ...

/*
| Send command to ESP and echo to the terminal.
*/
void serveWebpage(uint8_t str []) { ...

/*
| Map USART1 IRQ handler to our custom ISR
*/
void USART1_IRQHandler(){ ...

int main(void) { ...
```

Wireshark

Capturing from Wi-Fi: en0

http && ip.addr==192.168.1.40

No.	Time	Source	Destination	Protocol	Length	Info
435613	8501.487211	192.168.1.40	192.168.1.124	HTTP	73	Continuation
435611	8501.417374	192.168.1.40	192.168.1.124	HTTP	126	Continuation
435609	8501.342990	192.168.1.40	192.168.1.124	HTTP	124	Continuation
435603	8501.271980	192.168.1.40	192.168.1.124	HTTP	90	Continuation
435601	8501.206040	192.168.1.40	192.168.1.124	HTTP	72	Continuation
435599	8501.140933	192.168.1.40	192.168.1.124	HTTP	83	Continuation
435595	8501.013972	192.168.1.124	192.168.1.40	HTTP	452	GET /REQ=OFF? HTTP/1.1
435552	8498.421552	192.168.1.40	192.168.1.124	HTTP	73	Continuation
435550	8498.352296	192.168.1.40	192.168.1.124	HTTP	126	Continuation
435548	8498.283938	192.168.1.40	192.168.1.124	HTTP	124	Continuation
435545	8498.206892	192.168.1.40	192.168.1.124	HTTP	90	Continuation
435543	8498.140880	192.168.1.40	192.168.1.124	HTTP	72	Continuation
435541	8498.078352	192.168.1.40	192.168.1.124	HTTP	83	Continuation

Host: 192.168.1.40\r\n
Connection: keep-alive\r\n
Upgrade-Insecure-Requests: 1\r\n
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\r\n
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/14.0 Safari/605.1.15\r\n
Referer: http://192.168.1.40/REQ=ON?\r\n

0050 73 74 3a 20 31 39 32 2e 31 36 38 2e 31 2e 34 30 st: 192. 168.1.40
0060 0d 0a 43 6f 6e 6e 65 63 74 69 6f 6e 3a 20 6b 65 ..Connec tion: ke
0070 65 70 2d 61 6c 69 76 65 0d 0a 55 70 67 72 61 64 ep-alive ..Upgrad
0080 65 2d 49 6e 73 65 63 75 72 65 2d 52 65 71 75 65 e-Insecu re-Reque
0090 73 74 73 3a 20 31 0d 0a 41 63 63 65 70 74 3a 20 sts: 1
00a0 74 65 78 74 2f 68 74 6d 6c 2c 61 70 70 6c 69 63 text/h tml,applic
00b0 61 74 69 6f 6e 2f 78 68 74 6d 6c 2b 78 6d 6c 2c ation/xh tml+xml,
00c0 61 70 70 6c 69 63 61 74 69 6f 6e 2f 78 6d 6c 3b applicat ion/xml;
00d0 71 3d 30 2e 39 2c 2a 2f 2a 3b 71 3d 30 2e 38 0d q=0.9,*/ *;q=0.8
00e0 0a 55 73 65 72 2d 41 67 65 6e 74 3a 20 4d 6f 7a ..User-Ag ent: Moz
00f0 69 6c 6c 61 2f 35 2e 30 20 28 4d 61 63 69 6e 74 illa/5.0 (Macint
0100 6f 73 68 3b 20 49 6e 74 65 6c 20 4d 61 63 20 4f osh; Int el Mac O
0110 53 20 58 20 31 30 5f 31 35 5f 37 29 20 41 70 70 S X 10_1 5_7) App
0120 6c 65 57 65 62 4b 69 74 2f 36 30 35 2e 31 2e 31 leWebKit /605.1.1
0130 35 20 28 4b 48 54 4d 4c 2c 20 6c 69 6b 65 20 47 5 (KHTML , like G
0140 65 63 6b 6f 29 20 56 65 72 73 69 6f 6e 2f 31 34 ecko) Ve rsion/14
0150 2e 30 20 53 61 66 61 72 69 2f 36 30 35 2e 31 2e .0 Safar i/605.1.
0160 31 35 0d 0a 52 65 66 65 72 65 72 3a 20 68 74 74 15
0170 70 3a 2f 2f 31 39 32 2e 31 36 38 2e 31 2e 34 30 p://192. 168.1.40
0180 2f 52 45 51 3d 4f 4e 3f 0d 0a 41 63 63 65 70 74 /REQ=ON? ..Accept
0190 2d 4c 61 6e 67 75 61 67 65 3a 20 65 6e 2d 75 73 -Languag e: en-us
01a0 0d 0a 41 63 63 65 70 74 2d 45 6e 63 6f 64 69 6e ..Accept -Encodin
01b0 67 3a 20 67 7a 69 70 2c 20 64 65 66 6c 61 74 65 g: gzip, deflate
01c0 0d 0a 0d 0a

HTTP Referer (http.referer), 38 bytes

Packets: 484207 · Displayed: 887 (0.2%)

Profile: Default

Up Next

- Monday: Digital Signal Processing
- Wednesday: How to Pick an MCU and Board Bringup
- Lab 7: The Internet of Things

Lecture Feedback

- What is the most important thing you learned in class today?
- What point was most unclear from lecture today?

<https://forms.gle/Ay6MkpZ6x3xsW2Eb8>

