Lab 4– Internet of Things

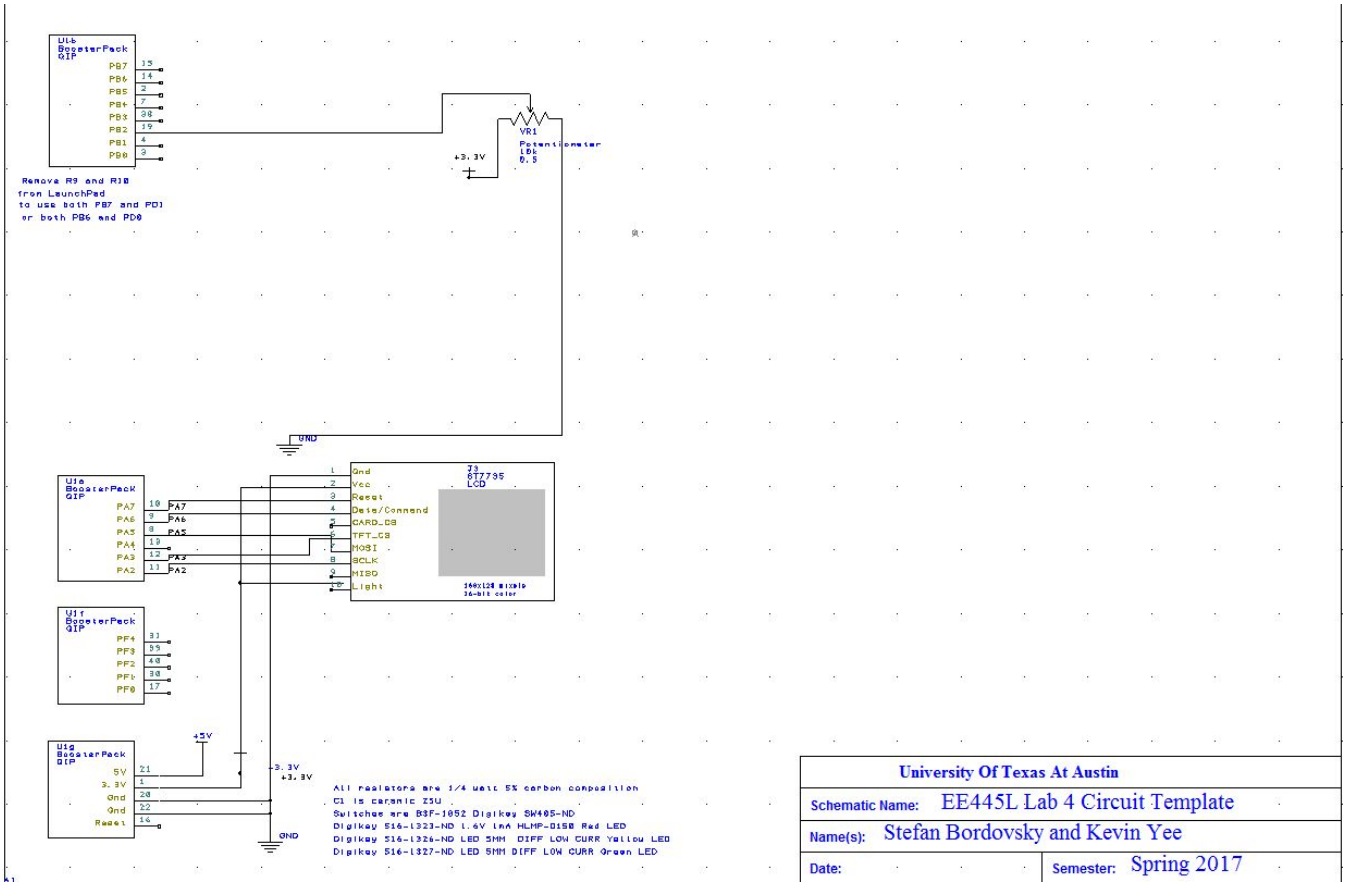
Report

Stefan Bordovsky and Kevin Yee

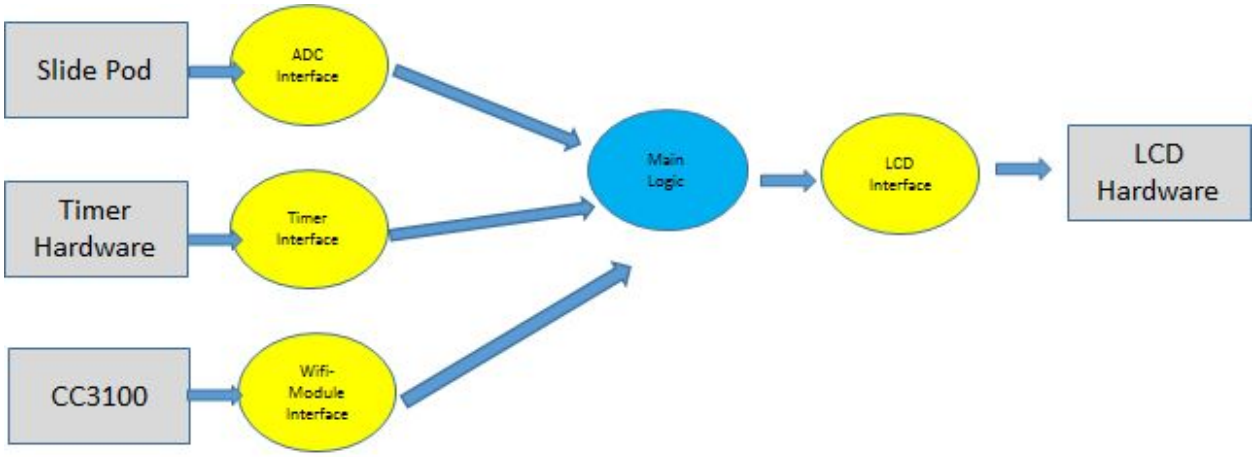
A) Objectives (1/2 page maximum)

The purpose of this lab was to implement a system that connected to the internet via an IEEE 802.11 - Wifi module. The CC3100 was used to receive weather data using TCP Packets from openweathermap.org. Furthermore, ADC samples were taken and used to send TCP Packets onto an internet server. Lastly, the lab provided an opportunity for the students to create their own web server using Google App Engine.

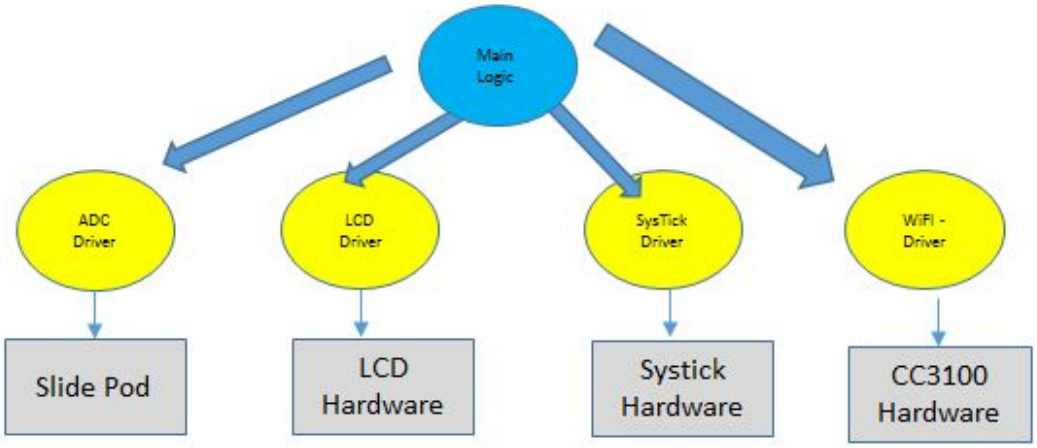
B) Hardware Design



C) Software Design (a hardcopy software printout is due at the time of demonstration)



Call Graph



D) Measurement Data

Percentage of lost packets. Basically how reliable is the system (assuming you have a connection to the AP)

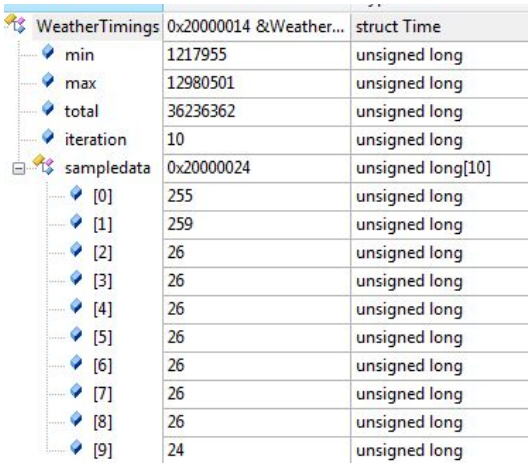
Minimum, maximum, and average times from 10 transmissions to openweathermap.org

Minimum, maximum, and average times from 10 transmissions to your server

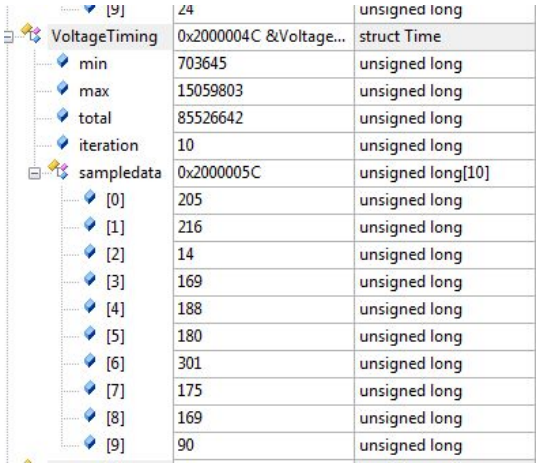
0% Packet Lost

Figure 1. Showing all data outputted to LCD Screen

Figure 2. Showing debugging buffers for time needed to upload and download TCP



WeatherTimings	0x20000014 &Weather...	struct Time
min	1217955	unsigned long
max	12980501	unsigned long
total	36236362	unsigned long
iteration	10	unsigned long
sampledata	0x20000024	unsigned long[10]
[0]	255	unsigned long
[1]	259	unsigned long
[2]	26	unsigned long
[3]	26	unsigned long
[4]	26	unsigned long
[5]	26	unsigned long
[6]	26	unsigned long
[7]	26	unsigned long
[8]	26	unsigned long
[9]	24	unsigned long



VoltageTiming	0x2000004C &Voltage...	struct Time
min	703645	unsigned long
max	15059803	unsigned long
total	85526642	unsigned long
iteration	10	unsigned long
sampledata	0x2000005C	unsigned long[10]
[0]	205	unsigned long
[1]	216	unsigned long
[2]	14	unsigned long
[3]	169	unsigned long
[4]	188	unsigned long
[5]	180	unsigned long
[6]	301	unsigned long
[7]	175	unsigned long
[8]	169	unsigned long
[9]	90	unsigned long

Table 1. Showing data for all ten runs of TCP Upload and Download

Run Number	Time to Download Weather (ms)	Time to Upload to Server (ms)
Run 1	255	205
Run 2	259	216
Run 3	26	14

Run 4	26	169
Run 5	26	188
Run 6	26	180
Run 7	26	301
Run 8	26	175
Run 9	26	169
Run 10	24	90

Table 2. Showing Min, Max, and Average Upload and Download Times

	Time to Download Weather (ms)	Time to Upload to Server (ms)
Min	24	14
Max	259	301
Average	72	171

E) Analysis and Discussion (1 page maximum)

1) In the client server paradigm, explain the sequence of internet communications sent from client to server and from server to client as the client saves data on the server. Assume the client already is connected to the wifi AP and the client knows the IP address of the server.

In the client-server paradigm, a server creates a connection socket and waits for a client to connect to the connection socket. The client then creates a client socket and sends a request to the waiting server. When free, the server will accept the client request using a server socket. The server socket will create a thread to serve the client which has requested it. This thread will process any request data coming from the client, and if necessary will cause the server to create and send a response to the client any number of times. In the context of saving data, the client will send a request to the server with data and a request to save the data on the server. After the client socket connects with a server socket, the server can then process this request and store the requested data.

2) What is the purpose of the DNS?

DNS, the Domain Name System, is a sort of translation scheme between website domain names and the IP addresses of the machines which host those websites. It is a layer of abstraction that creates a more user-friendly means of accessing different domains (e.g. being able to type google.com into one's address

bar instead of 8.8.8.8). This also means that even if the underlying IP address for a domain changes, a user may still access that domain with the same domain name they are used to--a good use of information hiding that takes the burden of IP address knowledge off the average user of the Internet.

3) What is the difference between UDP and TCP communication? More specifically when should we use UDP and when should we use TCP?

UDP and TCP are both Internet communication protocols, however they share distinctions in transmission, speed, and reliability. UDP consists of sending information asynchronously in packets. As a result of this asynchronous transmission method, data on the receiving end could end up in a different order than how it was sent, or might not arrive at all. This means UDP suffers from a certain amount of unreliability, but what it lacks in reliability it gains in speed. Due to its speed, UDP is an ideal protocol for such communication needs as streaming services and video game servers, where speed is a must and losing packets is okay. TCP, on the other hand, is a synchronous, connection-based method of data transmission. It requires a “handshake” between client and server sockets, which guarantees a stable connection and mitigates data loss. The overhead which allows for this reliability comes at the cost of speed, so TCP is a slower protocol than UDP.