Ryan Comer – rnc525
Connor White – bcw824

Lab Report

Lab 4 – Internet of Things

Objectives

The objective of this lab was to build an embedded system that would act like a 'thing' in the context of the Internet of Things. A 'thing' is something that has a microprocessor, sensors for inputs, a potential output, and can communicate with other things through the internet. The CC3100 booster pack for the TM4C123 was used to communicate with the internet.

There were 2 primary things that we had to accomplish: pull weather data down from openweathermap.org, and push data on to an EE445L server. The weather data was obtained through a socket connection, and the resulting JSON was parsed to find the current temperature for the desired city (Austin in this case). Pushing data onto the server was similar, and the on-board ADC was sampled so the data could be sent through a web socket to the server. This whole process of gathering inputs from sensors, doing optional computations on the data locally, and sending the data off onto a server is what the Internet of Things is all about.

Hardware Design

The only hardware for this lab, other than attaching the CC3100 to the TM4C123, was the input to the on-board ADC. We decided to use a slide potentiometer to represent a sensor with variable resistance that could detect useful environmental information. The ADC samples the voltage across the potentiometer before sending the data off onto the EE445L server. The following figure shows the PCB Artist diagram for the circuit. The PCB Artist file will also be submitted along with this report.

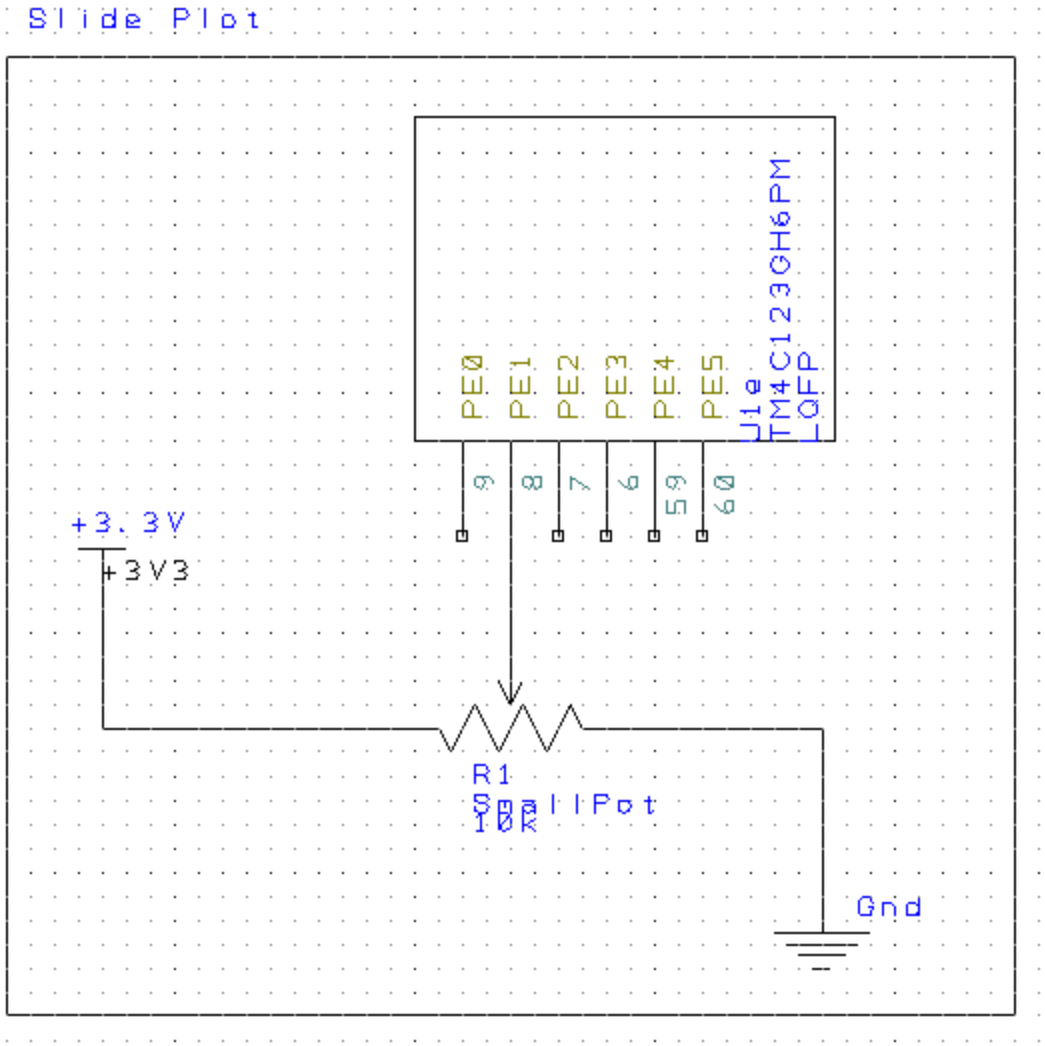

Figure 1: Slide Pot Circuit

Software

The Keil project that represents the software for this lab will be submitted along with this report

Measurement Data

The measurement data for this lab involved measuring statistics about a connection over the internet. We were to measure the time it took for 10 different requests sent to both openweathermap.org and the EE445L server. We also needed to find the average, min, and max times for both of those cases. The following tables show the data we collected

Request #	Time (ms)
1	77
2	306
3	307
4	195

5	307
6	305
7	311
8	303
9	304
10	310

Table 1: Request Times for Openweathermap.org

Request #	Time (ms)
1	171
2	307
3	307
4	307
5	307
6	306
7	305
8	309
9	306
10	74

Table 2: Request Times for EE445L Server

The following list shows the calculates we made for the measurement data

Openweathermap.org:

- **Min –** 77 ms
- **Max –** 311 ms
- **Average –** 272 ms

EE445L Server:

- **Min –** 74 ms
- **Max –** 309 ms
- **Average –** 269 ms

There wasn't any noticeable difference between the data for openweathermap.org and the EE445L server.

We also had to measure the percentage of lost packets for our system to find the system reliability. Because our system used TCP as the connection protocol, there were no packets lost in the communication. This is due to the fact that TCP ensures all packets are sent before the transmission is done: you either get them all or you don't get any at all. It is possible to break the connection before the transfer is done to miss out on some packets, but at that point you are missing packets due to connection issues and not due to the communication protocol that you are using.

Analysis and Discussion

This section contains answers to the questions described in the lab manual.

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.
The client connects to the server and then sends a request. The server then reads the request, processes it, and sends a response back to the client.

2) What is the purpose of the DNS?
The DNS translates domain names into IP addresses so that the computer can understand where you want to go.

3) What is the difference between UDP and TCP communication? More specifically when should we use UDP and when should we use TCP?
TCP is a connection oriented protocol while UDP is connectionless. TCP is thus better for information that requires high reliability such as file transfers, whereas UPD is better for cases where losing a piece of information is tolerable, such as streaming music or video.