Lab 4 Report

Objectives:

* Implement a “smart object” that connects to a phone application (Blynk) via a cloud based server using the ESP8266 WiFi module connected to the TM4C123. (Fig 4.1)
* Remotely control the alarm clock developed in Lab 3 via the “Blynk Virtual Pin” interface.
  + - Replace the manual switches from Lab 3 with Virtual Pin (VP) Switches.
    - In addition to replacing the manual switches, use the “Step V” widget to control the clock
* **Extra Credit**: Remotely read any sensor and via the “Blynk Virtual Pin” interface. Display the data on the Blynk App. Reading a sensor and displaying its value is an extra credit component.
* Remotely read the Hour/Minutes/Seconds via the “Blynk Virtual Pin” interface. Display the data on the Blynk App.

Required Hardware:

Component Hardware Type

ESP8266 Wifi Module

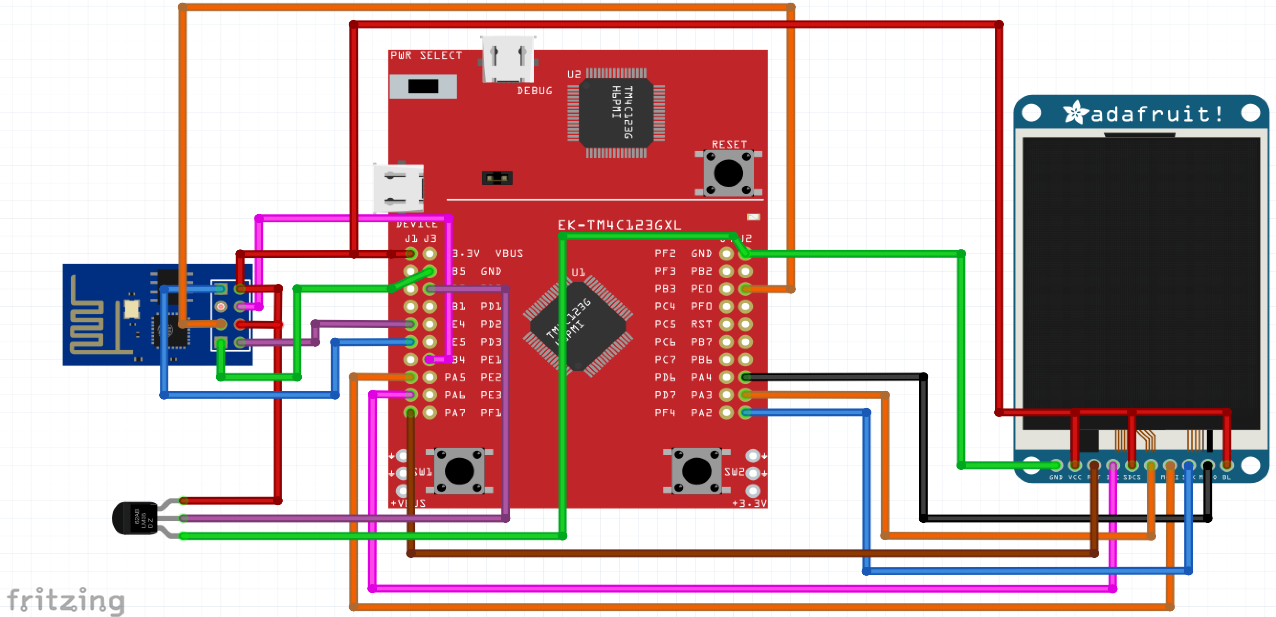
ST7735 LCD Module

EK-TM4C123GXL Tiva LaunchPad

Sparkfun-TMP36 Temperature Sensor (optional)

LM2937-3.3 3.3V linear regulator

Hardware Schematic:





Above is the schematic for ESP8266 connections to LaunchPad

Analysis:

The EP8266 drew 0.072 amps while running in idle and not sending or receiving data packets. When actively sending or receiving data packets the ESP8266 drew 0.085 amps. Powered using a 3.3V supply, the minimum and maximum power consumption was therefore 0.2376 watts and 0.2805 watts respectively. Below on the left is the current draw in idle and on the right is the current draw when transmitting or receiving data.



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 sends a request to the server with a randomized port number requesting a socket connection. The server creates a socket and sends a reply to the client. The client then sends the data to the created socket and the data is received by the server and saved. A response is sent by the server to the client and then the socket is closed, ending the connection.

1. What is the purpose of the DNS?

DNS, or Domain Name Service, is used to map a symbolic name such as “google.com” to the actual IP address of the network/machine connection. It allows for people to use an identifier rather than memorizing the 32-bit IP address of the network/machine they are trying to connect to.

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

UDP ports are designed to be a “best effort” for of communication and use socket APIs, where the goal is speed and not necessarily the reliability of the data transfer. The data is sent without any assurance it was sent or received correctly. TCP ports are designed to be a “guaranteed” reliable method for communication rather than a speedy method. Examples of TCP communication application layer protocols are HTTP, SMTP, and FTP where accuracy is key, while UDP is used in multimedia applications like VoIP for higher transfer speeds.