829H1 Real-Time Embedded Systems Final Report

Candidate No: 105936

5th January 2019

Contents

1. Abstract

The IoT space is full of applications and devices to make life easier this project has also been inspired by this trend. The main section of this report is how to connect the freedom K64F[1] to an external API and display text on the application shield. It looks into the process of displaying current weather information on the application shields display.

2. Introduction

This report focuses on using the FRDM-K64F[1] as a connected device to display weather information on the application shields display. Developing for real time systems relies on making sure that there is no delay or wait for the program in terms of using interrupts correctly and making sure the program runs quickly.

3. Exercises

4. Project

4.1. Getting the location

The first part of the program was to get the board to get it's own location allowing the board to query the weather for the correct location. To do this I first needed to get the current public IP address of the board using the code EthernetInterface.getIPAddress() unfortunately this only returned the internal IP address. Therefore, I had to use a public API to get the public IP address for this I used `https://api.ipify.org`[2]. To use this I would send off a http request to the address and then read the received text back. To send the http request I had to use the `https://os.mbed.com/teams/sandbox/code/mbed-http/` library simply importing this library didn't work therefore I had to fork the http examples repository found at `https://os.mbed.com/teams/sandbox/code/http-example/` and modify it for my purposes.

4.2. Displaying on the LCD

This was fairly simple to implement compared to the http library all I needed to do was import the library from `https://os.mbed.com/users/chris/code/C12832/` and use the example from `https://os.mbed.com/users/chris/code/app-shield-LCD/file/f8ef5e45e488/main.cpp/` as guidance on how to use the library.

4.3. Re-factoring the code

After implementing the function to get the IP address and the display it to the screen I realised I would either have to change the layout of the code so that I could support the display of all of the information or just display one piece of information. I decided that the best way to go was to support displaying all of the information this meant that I would need to create a separate class which would manage the getting and displaying of the data.

5. Analysis

6. Conclusion

Appendices

A. Program Extracts

A.1. Get IP address via HTTP code

```
char* get_ip_address()
{

    HttpRequest* request = new HttpRequest(network, HTTP_GET, "http://api.
   ipify.org/");
    request->set_header("Content-Type", "application/json");
    HttpResponse* response = request->send(body, strlen(body));
    // if response is NULL, check response->get_error()

    pc.printf("status is %d - %s\n", response->get_status_code(), response
   ->get_status_message());
    char* ipAddress = response->get_body_as_string().c_str();
    pc.printf("body is:\n%s\n", ipAddress);

    delete request; // also clears out the response
    return ipAddress;
}
```

References

[1] N. Products. (Mar. 2014). Frdm-k64f: Freedom development platform for kinetis® k64, k63, and k24 mcus, [Online]. Available: https://www.nxp.com/products/processors-and-microcontrollers/arm-based-processors-and-mcus/kinetis-cortex-m-mcus/k-seriesperformancem4/k2x-usb/freedom-development-platform-for-kinetis-k64-k63-and-k24-mcus:FRDM-K64F.

[2] R. Degges. (). Ipify, [Online]. Available: https://www.ipify.org/.