HTTPBaseStation instructions - Amr Mohamed

The HTTPBaseStation program can be used to connect to wifi and make network requests over said wifi to valid devices. These network requests can be made in the form of get, put, post, and any other protocols supported by the http client library. Eventually the plan is to merge the HTTPBaseStation and the Bluetooth to Bangle Communications program from Bryan Ponce in order to create a program capable of communication with a Bangle.JS and a remote Web Server created by Brook. This program would be running on an esp32 microcontroller known as the base station which would act as somewhat of a middle man between this communication. This esp32 would most likely also handle some error handling and could even temporarily store updates for the Bangle in its storage system depending on how the system is implemented. This would make the base station a necessary component in the communication of the Bangle with the internet.

In order to use the HTTPBaseStation.ino program one must start by opening the arduino program and connecting their esp32(specifically the M5stick.C plus) to the Arduino Ide, this is as simple as selecting it from the board dropdown.

Next the user must check if they have the M5StickCPlus Library from M5Stack installed. It is available from the arduino library manager and can also be downloaded from the following reference link: https://github.com/m5stack/M5StickC-Plus. This library was also used as a source for some of the material in the file and has some good material that could be used in future applications.

Next one must verify the information in the setup loop and input their Wifi Name and password on the wifiMulti.addAP line.

```
void setup() {
    M5.begin();
    M5.Lcd.setRotation(3);
    wifiMulti.addAP("Wifi", "Pass");

    M5.Lcd.print("\nConnecting Wifi...\n");

}
```

Finally the user needs to alter the address in the http.begin connection line in the main loop in order to connect to the correct web page. Here the Usercan also change their http request and manipulate the data received. In the example the root page of the UGA Smartwatch web server is used.

```
void loop() {
    M5.Lcd.setCursor(0, 0);
    if ((wifiMulti.run() ==
         WL_CONNECTED)) {  //
        M5.Lcd.print("[HTTP] begin...\n");
        http.begin(
            "https://vn.ugavel.com/ugasmartwatch/");
        M5.Lcd.print("[HTTP] Request...\n");
        int httpCode = http.GET();
        if (httpCode >
            0) {
            Serial.printf("[HTTP] Request code: %d\n", httpCode);
            M5.Lcd.print("Please see Serial.");
            if (httpCode ==
                HTTP_CODE_OK) {
                String payload = http.getString();
                Serial.println(payload);

            }
        } else {
            M5.Lcd.printf("[HTTP] Request failed, error: %s\n",
                          http.errorToString(httpCode).c_str());
        }
        http.end();
    } else {
        M5.Lcd.print("connect failed");
    }
    delay(5000);
    M5.Lcd.fillRect(0, 0, 160, 80, BLACK);
}
```

Now the program is ready to run. It will begin by initializing a wifi connection. After which the M5 will begin the http connection to the address specified. Once the connection is successfully established The M5 can send the appropriate HTTP request. Currently the program is set to get the server status from the root webpage and display in the serial monitor. However, the full functions of the program can be altered by the user at will. The program will continue to send

this request until it ends or there is an error at which point it will start to try to establish connection until successful.

In case of any errors One should first attempt to troubleshoot their connection. Most problems arise from wifi or network connection issues. The next step could be to troubleshoot the board being used and ensure that it is correctly powered on and connected. Finally there could also be some sort of issue in the configuration of the library or the actual program itself so retracing steps and making sure everything is correctly configured is always a good idea. There are also error print statements that can help a user pinpoint what the problem might be. With that being said, the program is quite reliable and can be used extensively without running into too many issues.

If there are any questions or concerns please contact me at aam08331@uga.edu or Dr. Johnson could also be contacted at kjohnsen@uga.edu.