#include <Arduino.h>

#include <ESP8266WiFi.h>

#include <ESP8266WiFiMulti.h>

#include <WebSocketsClient.h>

#include <Hash.h>

ESP8266WiFiMulti WiFiMulti;

WebSocketsClient switches;

WebSocketsClient charts;

#define USE\_SERIAL Serial

String rawTxt="{\"TruePower\":0,\"ApparentPower\":0,\"Frequency\":0,\"PowerFactor\":0,\"Vrms\":0,\"Irms\":0}";

long unsigned int thisTime;

void switchesEvent(WStype\_t type, uint8\_t \* payload, size\_t length) {

switch(type) {

case WStype\_DISCONNECTED:

USE\_SERIAL.printf("[WSc] Switches Disconnected!\n");

break;

case WStype\_CONNECTED: {

switches.sendTXT("Connected to Switches");

}

break;

case WStype\_TEXT:

USE\_SERIAL.println((char\*)payload);

break;

case WStype\_BIN:

USE\_SERIAL.printf("[WSc] get binary length: %u\n", length);

hexdump(payload, length);

break;

}

}

void chartsEvent(WStype\_t type, uint8\_t \* payload, size\_t length) {

switch(type) {

case WStype\_DISCONNECTED:

USE\_SERIAL.printf("[WSc] Charts Disconnected!\n");

break;

case WStype\_CONNECTED: {

charts.sendTXT("Connected to Charts");

}

break;

case WStype\_TEXT:

charts.sendTXT(rawTxt);

break;

case WStype\_BIN:

USE\_SERIAL.printf("[WSc] get binary length: %u\n", length);

hexdump(payload, length);

break;

}

}

void setup() {

USE\_SERIAL.begin(38400);

//Serial.setDebugOutput(true);

//USE\_SERIAL.setDebugOutput(true);

USE\_SERIAL.println();

USE\_SERIAL.println();

USE\_SERIAL.println();

for(uint8\_t t = 4; t > 0; t--) {

USE\_SERIAL.printf("[SETUP] BOOT WAIT %d...\n", t);

USE\_SERIAL.flush();

delay(1000);

}

WiFiMulti.addAP("FLIP IO", "capstone2017");

//WiFi.disconnect();

while(WiFiMulti.run() != WL\_CONNECTED) {

delay(100);

}

// server address, port and URL

switches.begin("192.168.43.49", 80, "/switches/");

charts.begin("192.168.43.49", 80, "/charts/");

// event handler

switches.onEvent(switchesEvent);

charts.onEvent(chartsEvent);

// try ever 5000 again if connection has failed

switches.setReconnectInterval(5000);

charts.setReconnectInterval(5000);

thisTime=millis();

}

void loop() {

switches.loop();

if (USE\_SERIAL.available())

{

char recieved = USE\_SERIAL.read();

if (recieved == '\n')

{

charts.loop();

rawTxt = "";

}

else rawTxt += recieved;

}

}