#include <WebServer.h>

#include <WiFi.h>

#include <esp32cam.h>

const char\* WIFI\_SSID = "Laptop Của Quỷ";

const char\* WIFI\_PASS = "lamgicomatkhau";

WebServer server(80);

static auto loRes = esp32cam::Resolution::find(320, 240);

static auto midRes = esp32cam::Resolution::find(350, 530);

static auto hiRes = esp32cam::Resolution::find(800, 600);

void serveJpg()

{

  auto frame = esp32cam::capture();

  if (frame == nullptr) {

    Serial.println("CAPTURE FAIL");

    server.send(503, "", "");

    return;

  }

  Serial.printf("CAPTURE OK %dx%d %dbn", frame->getWidth(), frame->getHeight(),

                static\_cast<int>(frame->size()));

  server.setContentLength(frame->size());

  server.send(200, "image/jpeg");

  WiFiClient client = server.client();

  frame->writeTo(client);

}

void handleJpgLo()

{

  if (!esp32cam::Camera.changeResolution(loRes)) {

    Serial.println("SET-LO-RES FAIL");

  }

  serveJpg();

}

void handleJpgHi()

{

  if (!esp32cam::Camera.changeResolution(hiRes)) {

    Serial.println("SET-HI-RES FAIL");

  }

  serveJpg();

}

void handleJpgMid()

{

  if (!esp32cam::Camera.changeResolution(midRes)) {

    Serial.println("SET-MID-RES FAIL");

  }

  serveJpg();

}

void  setup(){

  Serial.begin(115200);

  Serial.println();

  {

    using namespace esp32cam;

    Config cfg;

    cfg.setPins(pins::AiThinker);

    cfg.setResolution(hiRes);

    cfg.setBufferCount(2);

    cfg.setJpeg(80);

    bool ok = Camera.begin(cfg);

    Serial.println(ok ? "CAMERA OK" : "CAMERA FAIL");

  }

  WiFi.persistent(false);

  WiFi.mode(WIFI\_STA);

  WiFi.begin(WIFI\_SSID, WIFI\_PASS);

  while (WiFi.status() != WL\_CONNECTED) {

    delay(500);

  }

  Serial.print("http://");

  Serial.println(WiFi.localIP());

  Serial.println("  /cam-lo.jpg");

  Serial.println("  /cam-hi.jpg");

  Serial.println("  /cam-mid.jpg");

  server.on("/cam-lo.jpg", handleJpgLo);

  server.on("/cam-hi.jpg", handleJpgHi);

  server.on("/cam-mid.jpg", handleJpgMid);

  server.begin();

}

void loop()

{

  server.handleClient();

}