#include <WiFi.h>

// WiFi credentials

const char\* ssid1 = "Deva1";

const char\* password1 = "Vittala@26";

const char\* ssid2 = "Deva";

const char\* password2 = "Vittala@26";

void WiFiEvent(WiFiEvent\_t event) {}

// WiFi failover variables

int currentWiFi = 0;

unsigned long wifiRetryTimer = 0;

unsigned long bootTime = 0;

const unsigned long WIFI\_RETRY\_DELAY = 30000;

const unsigned long BOOT\_SCAN\_DELAY = 30000; // 30 seconds after boot

int wifiRetryCount = 0;

const int MAX\_WIFI\_RETRIES = 3;

bool bootScanDone = false;

void scanWiFiNetworks() {

Serial.println("Scanning for WiFi networks...");

int n = WiFi.scanNetworks();

if (n == 0) {

Serial.println("No networks found");

} else {

Serial.printf("%d networks found:\n", n);

for (int i = 0; i < n; ++i) {

Serial.printf("%d: %s (%d dBm) %s\n", i + 1, WiFi.SSID(i).c\_str(),

WiFi.RSSI(i), (WiFi.encryptionType(i) == WIFI\_AUTH\_OPEN) ? "Open" : "Encrypted");

}

}

WiFi.scanDelete();

}

void connectWifi() {

WiFi.disconnect(true);

delay(1000);

WiFi.onEvent(WiFiEvent);

if (currentWiFi == 0) {

Serial.println("Connecting to " + String(ssid1));

WiFi.begin(ssid1, password1);

} else {

Serial.println("Connecting to " + String(ssid2));

WiFi.begin(ssid2, password2);

}

wifiRetryTimer = millis();

wifiRetryCount++;

}

void handleWiFiFailover() {

// Boot scan after 30 seconds

if (!bootScanDone && millis() - bootTime > BOOT\_SCAN\_DELAY) {

scanWiFiNetworks();

bootScanDone = true;

}

// Check if we need to retry or switch networks

if (WiFi.status() != WL\_CONNECTED &&

millis() - wifiRetryTimer > WIFI\_RETRY\_DELAY) {

if (wifiRetryCount >= MAX\_WIFI\_RETRIES) {

currentWiFi = (currentWiFi == 0) ? 1 : 0;

wifiRetryCount = 0;

Serial.println("Switching to alternate WiFi network");

}

connectWifi();

}

}

void setup() {

Serial.begin(115200);

bootTime = millis();

connectWifi(); // Start initial WiFi connection

}

void loop() {

handleWiFiFailover();

delay(1000); // Check every second

}