//definitions

#define STATUS\_LED 12 // Green LED

#define AMBER\_LED 14 // Amber LED

#define WIFI\_STA\_TIMEOUT 20000 // try to connect to wifi for x seconds

const char\* CUST\_SSID = ""; // SSID of the network you want ESP to connect to

const char\* PASS = ""; // pass of the network you want ESP to connect to

const char\* \_ssid = "wi-pj-g2";

const char\* \_password = "srmedw32";

const char\* \_GMailServer = "";

const char\* \_mailUser = "";

const char\* \_mailPassword = "";

//IPAddress cloudServerIP(34,83,30,196); // IP address

//IPAddress cloudServerIP(10,1,1,10);

// REST client does not take IPAddress so converting to const char

//String cloudServerIPstr = String(cloudServerIP[0])+'.'+String(cloudServerIP[1])+'.'+String(cloudServerIP[2])+'.'+String(cloudServerIP[3]);

//char const\* RESTServer = cloudServerIPstr.c\_str();

//unsigned int RESTPort = 80; // port on REST server

// Constants

const char\* FIRMWARE\_VER = " GarageOS version 1.0.2";

const char\* EXTERNAL\_IP\_ADDRESS = "http://34.83.30.196:80/device";

// BLE scan structures for namespace/UUID filter

const int noOfNamespaces = 10; // no of namesaces or UUIDs to scan for

struct scanFilter {

uint8\_t uuid[16];

uint8\_t uidNamespace[10];

};

static scanFilter scanOnly[noOfNamespaces];

// Function declarations

char\* nrf\_send\_recv(char\*, boolean return\_resp = false);

void esp\_hb\_ticker\_trigger();

void post\_to\_cloud();

void http\_ota\_update();