

DOKUMENTÁCIA PROJEKTU

Mikroprocesorové a vstavané systémy

Ovládanie RGB LED

15.12.2022

Šimon Kadnár xkadna00

1 Úvod

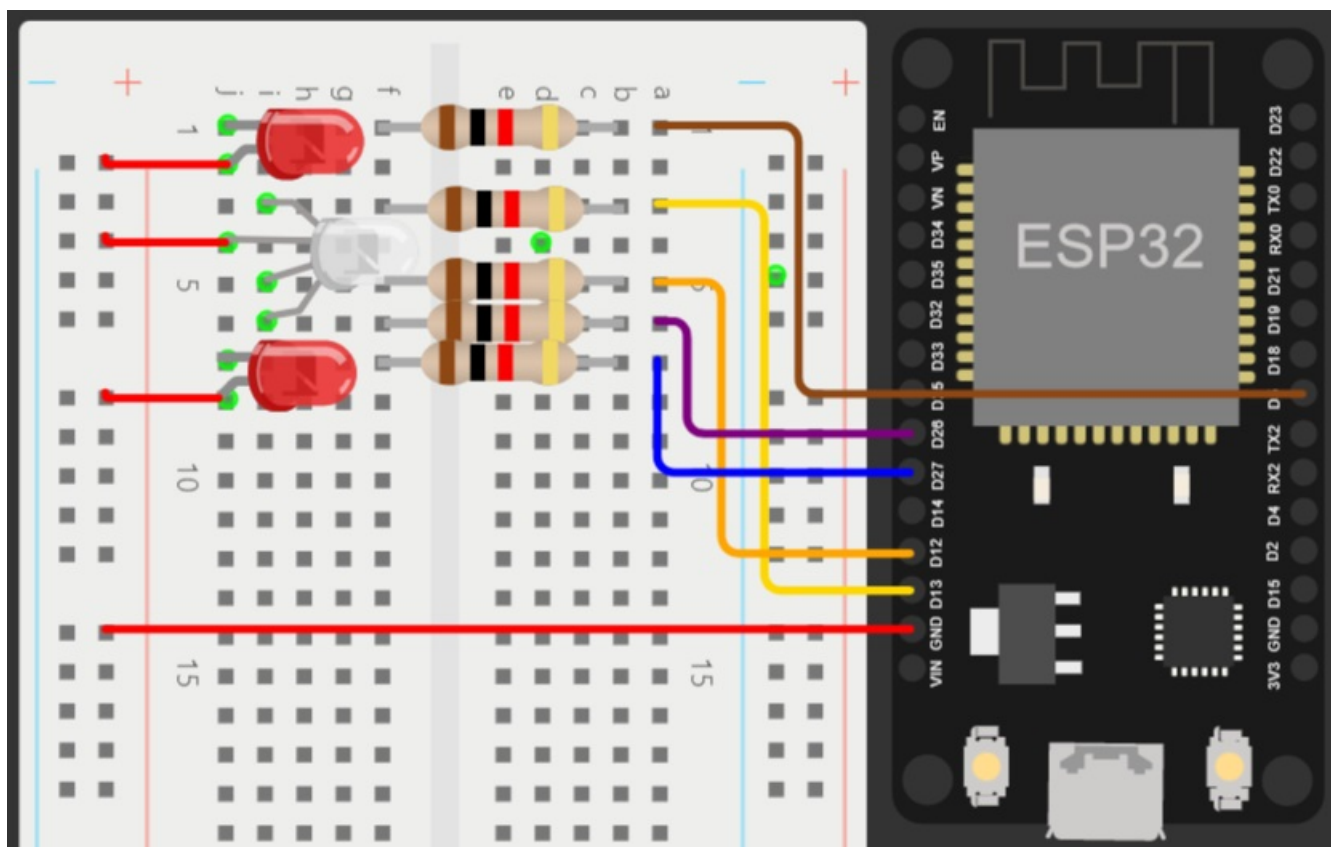
Zadaním projektu bolo implementovať vstavanú aplikáciu na doske ESP32, ktorá bude schopná ovládať skupinu LED diód a jednu RGB led pomocou WIFI rozhrania. Výsledná aplikácia obsahuje niekoľko rôznych animácií svetiel a možnosť upraviť ich rýchlosť. Aplikácia bola implementovaná v jazyku C++ cez prostredie PlatromIO na platforme Arduino.

2 Zapojenie hardware

Pri riešení tohto projektu boli použité porty(GPIO): 2, 5, 13, 12, 14, 27.

2.1 Využitie portov

- Port 2 - značí príjem požiadavku
- Port 5 - ovláda ľavú diódu
- Port 13 - nastavuje červenú farbu na RGB led
- Port 12 - nastavuje zelenú farbu na RGB led
- Port 14 - nastavuje modrú farbu na RGB led
- Port 27 - ovláda pravú diódu



3 Popis riešenia

Na začiatku sa vo funkcii *setup()* nastaví výstupné piny, vytvorí sa proces *animation_mod()*, ktorý ovláda výstupe piny pomocou volania funkcie pre konkrétnu animáciu. Následne sa vytvorí wifi sieť(*rgb_esp32*) a založí sa server na ktorom sa nachádza ovládacia stránka. Potom nasleduje funkcia *loop()* v ktorej ak je k dispozícii nový klient tak sa mu vygeneruje stránka s aktuálnym nastavením. V prípade, že dôjde k stlačeniu jedného z možných nastavení animácie funkcia obsahuje sadu podmienok ktorá detekuje zmenu url a podľa toho buď zmení rýchlosť v premennej *speed* (zavolaním funkcie *get_speed()*) alebo nastaví do premennej *mod* hodnotu odpovedajúcej animácií a vynuluje riadiace premenné *cnt1* a *cnt2*.

3.1 Globálne premenné

- *cnt1*, *cnt2* – riadia priebeh animácie,
- *mod* – určuje typ animácie,
- *speed* – určuje rýchlosť animácie
- *header* – obsahuje odoslané informácie od klienta

3.2 knižnice

- *stdio.h* - práca so stringami
- *WiFi.h* – slúži pre vytvorenie vlastnej wifi a vytvorenie servera
- *ESPmDNS.h* – slúži pre prekrytie ip adresy názvom

3.3 *animation_mod()*

Na začiatku sa nachádza nekonečný cyklus v ktorom sa nachádza sada podmienok kde sa najskôr určí, ktorá animácia má byť spustená podľa premennej *mod*. Následne sa volá funkcia s danou animáciou (funkcie: *animation1()*, *animation2()*,...). Každá z nich obsahuje sadu podmienok, ktoré riadia priebeh animácie pomocou premenných *cnt*, *cnt1* a *cnt2*, podľa ich inkrementácie. Na konci sa nachádza časovač ktorý obsahuje hodnotu *speed* podľa ktorej sa určuje akou rýchlosťou má bežať vybraná animácia.

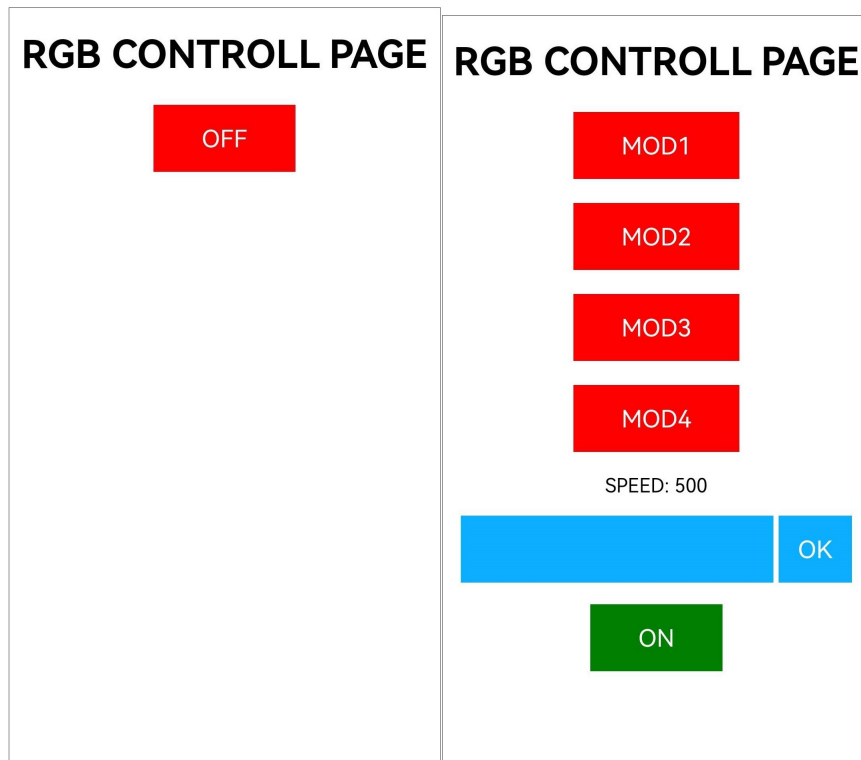
3.4 *get_speed()*

Funkcia pracuje s premenou *header*, ktorú prejde v cykle a nájde požadovanú rýchlosť, ktorú uloží do premennej *speed*, čím dôjde k zmene rýchlosti vybranej animácie.

4 Ovládanie

Po spustení aplikácie je potrebné sa pripojiť na wifi sieť *rgb_esp32*, ktorú vysiela zariadenie. Následne je potrebné sa pripojiť na ovládaciu stránku cez preferovaný prehliadač zadaním *http://192.168.4.1* alebo *http://esp32*. Po pripojení na stránku sa zobrazí jedno tlačítko *OFF* ktoré po stlačení sprístupní ďalšie 4 tlačítka ktoré ovládajú typ animácie a jedno textové pole do ktorého je možné zadať ľubovoľnú rýchlosť vybranej animácie. Po stlačení tlačidla sa na doske rozsvieti na malú chvíľu, červené svetlo, ktoré signalizuje prijatie požiadavku zo siete. V prípade, že chceme vypnúť animácie stačí stlačiť spodné tlačítko *ON*.

Demonštračné video: [Demo](#)



5 Záver

Výsledná aplikácia obsahuje 4 tlačidlá slúžiace pre nastavenie typu animácie, 1 textové pole pre zadávanie ľubovoľnej rýchlosti a jedno tlačidlo pre vypnutie animácie. Možné problémy sú s prekrytím ip adresy pomocou protokolu mDNS, ktorá nefunguje na niektorých mobilných zariadeniach a je teda zapotreby sa pripojiť na ovládaciu stránku pomocou ip adresy.

5.1 Hodnotiaci kľúč

E = 1

F = 4,5

Q = 2

P = 1

D = 4

$(0,25 + 0,75 \cdot F/5) \cdot (E + F + Q + P + D) = 11,5$

6 Literatura

<https://randomnerdtutorials.com/esp32-web-server-arduino-ide/>