

Variantá termínu - 01. Bezdrátový textový zobrazovač

René Rešetár

8. mája 2022

1 Úvod

V tejto variante bolo za úlohu vytvoriť za pomoci ESP-WROOM-32 (na ochrannom obale Wemos D1 R32) a TFT LCD displeja bezdrôtový textový zobrazovač. Budeme ho implementovať pomocou vytvorenia jednoduchého Soft-Access (SA) serveru a wifi siete na ESP čipe. Na server budeme zasielať požadovaný text v URL. Projekt je vypracovaný pre esp-idf. Keďže bluetooth pripojenie ste mi poslali skoro vypracované skúsil som to vypracovať pomocou Wifi.

2 Použité komponenty

2.1 ESP-WROOM-32

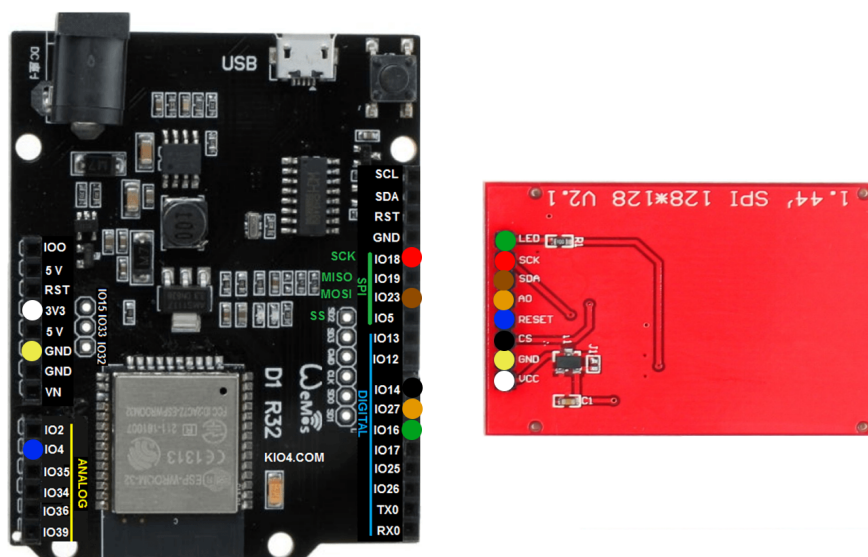
Je to výkonný modul, ktorý sa zameriava na širokú aplikáciu. Tento mikrokontrolér bude slúžiť k riadeniu celého obvodu, čiže na zachytávanie requestov a následný prepis obsahu displeja. K jeho programovaniu budeme využívať *esp-idf* od firmy *Espressif*.

2.2 TFT SPI LCD 128*128

Je to maličký displej, ktorý dokáže zobrazit' až 262 144 a bude nám slúžiť ako výstupná periféria nášho projektu pre zobrazenie textu.

3 Zapojenie

- LCD — ESP-32
- LED — GPIO16
- SCK — GPIO18
- SDA — GPIO23
- AO — GPIO27
- RST — GPIO04
- CS — GPIO14
- GND — GND
- VCC — 3,3V



Obr. 1: Zapojenie displeja

4 Implementácia

- **main.c** implementuje vytvorenie a riadenie SA serveru
- **display_driver.h/.c** ovládanie displeja
- **font.c** nastavenie fontu

Najprv zadefinujeme konfiguráciu k prístupu na Wifi sieť.

```
#define EXAMPLE_ESP_WIFI_SSID      "NAV"
#define EXAMPLE_ESP_WIFI_PASS      "NAV_password"
#define EXAMPLE_ESP_WIFI_CHANNEL  1
#define EXAMPLE_MAX_STA_CONN      4
```

Obr. 2: Zapojenie displeja

V hlavnej funkcii inicializujeme displej a vypíšeme na neho uvítaciu správu. Taktiež inicializujeme pamäť a SA server. Pri pripojení zariadenia na Wifi sa priradí IP adresa a udalosť **IP_EVENT_AP_STAIPASSIGNED** preda kontrolu *connect_handler*.

```

void app_main(void)
{
    //inicializace displeje
    lcd_init();
    //vypis uvitaci zpravy
    char *msg = "Aplikace\nspustena,\ncekam na\nzpravu z web serveru";
    for (int i = 0; i < strlen(msg); i++) {
        lcd_write_char(msg[i], true, COLOR_GREEN);
    }

    static httpd_handle_t server = NULL;

    esp_err_t ret = nvs_flash_init();
    if (ret == ESP_ERR_NVS_NO_FREE_PAGES || ret == ESP_ERR_NVS_NEW_VERSION_FOUND) {
        ESP_ERROR_CHECK(nvs_flash_erase());
        ret = nvs_flash_init();
    }
    ESP_ERROR_CHECK(ret);

    ESP_LOGI(TAG, "ESP_WIFI_MODE_AP");
    wifi_init_softap();

    ESP_ERROR_CHECK(esp_netif_init());
    ESP_ERROR_CHECK(esp_event_handler_register(IP_EVENT, IP_EVENT_AP_STAIPASSIGNED, &connect_handler, &server));
}

```

Obr. 3: Zapojenie displeja

connect_handler zavolá *start_webserver*, ktorý spustí server s defaultnou konfiguráciou. Taktiež zaregistruje udalosť **HTTPD_404_NOT_FOUND** pre *random_handler* na spracovanie.

```

static httpd_handle_t start_webserver(void)
{
    httpd_handle_t server = NULL;
    httpd_config_t config = HTTPD_DEFAULT_CONFIG();
    config.lru_purge_enable = true;

    // Start the httpd server
    ESP_LOGI(TAG, "Starting server on port: '%d'", config.server_port);
    if (httpd_start(&server, &config) == ESP_OK) {
        // Set URI handlers
        ESP_LOGI(TAG, "Registering URI handlers");

        httpd_register_err_handler(server, HTTPD_404_NOT_FOUND, &random_handler);
        return server;
    }

    ESP_LOGI(TAG, "Error starting server!");
    return NULL;
}

```

Obr. 4: Zapojenie displeja

`random_handler` dostane kontrolu vždy, keď užívateľ zadá neznámu (teda neregistrovanú) URI. Túto URI vezme, upraví pomocou `replace_char` a následne vypíše na displej.

```
char* replace_char(char* str, char find, char replace){
    char *current_pos = strchr(str,find);
    while (current_pos) {
        *current_pos = replace;
        current_pos = strchr(current_pos,find);
    }
    return str;
}

esp_err_t random_handler(httpd_req_t *req){
    lcd_clear();
    char* spaces = replace_char(req->uri, '_', ' ');
    char* out = replace_char(spaces, '&', '\n');
    for (int i = 1; i < strlen(out); i++) {
        lcd_write_char(out[i], true, COLOR_GREEN);
    }
    return ESP_OK;
}
```

Obr. 5: Zapojenie displeja

Pre implementáciu som využil Vami zaslaný projekt a ESP knižnice a príklady z */examples*.

5 Prístup a použitie

Po spustení je potrebné počkať na inicializovanie Wifi siete. Potom sa cez telefón alebo počítač treba pripojiť s údajmi spomenutými vyššie. Po úspešnom prihlásení na Wifi otvoríme prehliadač a zadáme IP adresu 192.168.4.1 kde beží vytvorený SA server. Vo webovom prehliadači zadáme k IP správu pre výpis ako súčasť URI následovne:

- **192.168.4.1/Message** vypíše "Message"
- **192.168.4.1/Toto_je_správa** vypíše "Toto je správa", teda "_" sa nahradí za medzeru
- **192.168.4.1/Toto_je&správa** vypíše "Toto je(nový riadok)správa", teda "&" sa nahradí za nový riadok

Vypisovanie zabezpečuje `display_driver.c/.h` a `font.h`, ktorý som od Vás obdržal ako funkčný príklad. Kvôli nedostatku času z mojej strany je výpis implementovaný iba staticky a teda pri dlhšej správe bude vidieť iba jej koniec.

Demo link: <https://nextcloud.fit.vutbr.cz/s/ZWoTZcBmFkcBdXA> s krátkym demonštračným videom.