

LED Matrix hodiny ovládané mikrokontrolerem ESP8266

Ročníková práce z předmětu PSS

Martin Morávek

C3c

Obsah

Zadání práce	3
Anotace	4
Úvod	5
Ekonomická rozvaha.....	6
Vývoj, realizace	7
Nasazení	10
GIT	16
Závěr.....	17

Zadání práce

V mém projektu jsem dělal hodiny, které na LED matici s rozlišením 64x32 pixelů s roztečí 2,5mm zobrazují přesný čas.

Anotace

Ročníková práce je založena na realizaci digitálních hodin na RGB LED matici s rozlišením 64x32 pixelů s roztečí 2,5 mm a scanem 1/16 za pomoci mikrokontroleru ESP8266. Čas je synchronizován pomocí třídy NTP (Network Time Protocol) a zobrazován ho ve formátu hh:mm:ss s 24hodinovým formátem.

Úvod

V tomto projektu jsem vytvořil jednoduché digitální hodiny postavených za pomoci mikrokontroleru ESP8266 NodeMCU a maticového RGB LED displeje o rozlišení 64×32 pixelů. Displej typu P2.5 s řízením 1/16 zobrazuje přesný čas ve formátu hh:mm:ss ve 24hodinovém režimu. ESP8266 se automaticky připojuje k Wi-Fi síti a synchronizuje aktuální čas přes internet za pomoci NTP (Network Time Protocol). Zohledňuje přitom časové pásmo, např. středoevropský letní čas (CEST).

Hodiny jsou navrženy s důrazem na čitelnost – číslice jsou vycentrovány na displeji a mezi jednotlivými znaky je nastaveno přesné rozestoupení pro co nejlepší čitelnost čísel. Dvojtečky mezi hodinami a minutami blikají v intervalu 500 ms, což připomíná efekt přirozeného chodu digitálních hodin. Při startu zařízení je na displeji zobrazená IP adresa zařízení v lokální síti sloužící k přístupu na integrovaný webový server, který běží přímo na ESP8266. Webové rozhraní nabízí intuitivní HTML stránku, kde může uživatel:

- nastavit čas budíku
- upravit textovou zprávu, která se zobrazí na displeji jako běžící text
- sledovat aktuální čas synchronizovaný pomocí NTP.

Budík je jednoduchý – na displeji ukáže vypsanou upravitelnou zprávu, jakmile dosáhne nastaveného času. Při zapínání programu a připojování k internetu zůstává displej prázdný, dokud mikrokontroler nesynchronizuje čas, čímž se předejde zobrazení problikávající obrazovky.

Celý projekt propojuje programování mikrokontrolerů, práci s LED maticemi a tvorbu jednoduchých webových stránek v HTML. Díky své přehlednosti a názornosti je vhodný jak pro domácí použití, tak i jako vzorová práce pro ukázání všestrannosti zařízení z oblasti IoT.

Ekonomická rozvaha

Na rozdíl od velkých společností má můj projekt výhodu, že nesdílí a neprodává uživatelské informace.

Vývoj, realizace

Použité technologie:

- ESP8266 (NodeMCU) – řídící mikrokontroler s podporou Wi-Fi, naprogramovaný v jazyce C++ v Arduino IDE.
- **64×32 RGB LED matice (typ P2.5)** – displej pro zobrazování času a zpráv; používá tzv. **1/16 skenování**, což znamená, že řádky jsou zobrazovány postupně v rychlém sledu.
- **Knihovna PxMatrix** – ovladač pro LED matici, který umožňuje výstup grafiky přes GPIO piny ESP8266. Podporuje různé typy matic a režimy řízení.
- **Knihovny WiFi, NTPClient a ESP8266WebServer** – zajišťují připojení k Wi-Fi, synchronizaci času pomocí internetu a běh jednoduchého webového server

Použité knihovny:

- ESP8266WiFi.h – připojení k Wi-Fi síti
- WiFiUdp.h a NTPClient.h – synchronizace reálného času z internetu
- PxMatrix.h – ovládání LED matice
- ESP8266WebServer.h – jednoduchý webový server běžící přímo na ESP

Členění programu

Kód je rozdělen do několika částí:

Inicializace systému (setup()):

- Naváže Wi-Fi spojení a vypíše IP adresu do sériové konzole.
- Inicializace LED matice, nastavení jasu a rychlosti obnovování obrazu.
- Spouští NTP klienta,
- Zapne jas
- Zobrazení IP adresy
- Spuštění webového serveru a definice jednotlivých HTTP handlerů.

Hlavní smyčka (loop()):

- Zajišťuje aktualizaci času.
- Obsluhuje webové požadavky klienta.
- Kontroluje alarm (jestli je aktivován) a vykresluje zprávu na LED matici.

Zobrazení času a zprávy:

- `showTime()` – vykresluje aktuální čas na střed obrazovky.
- `displayIP()` – posouvá IP adresu přes celou obrazovku.
- `showMessage()` – posuvná zpráva zadaná uživatelem.
- `toggleBlink()` – blikání dvojtečky každých 500 ms.

Webové rozhraní:

- Obsahuje HTML formulář pro zadání zprávy a nastavení alarmu.
- Odeslané hodnoty jsou zpracovány na pozadí a uloženy do proměnných v programu.

Alarm:

- Čas budíku je porovnáván s aktuálním časem každou sekundu.
- Pokud nastane shoda, na displeji se zobrazí speciální hláška.

Dokumentace v kódu

Kód je doplněn o popisné komentáře které:

- Uvádějí, k čemu slouží jednotlivé sekce
- Funkce jsou odděleny a strukturovány tak, aby byly co nejvíce přehledné
- Klíčové proměnné jsou pojmenovány, co nejjednoznačněji (např. `showTime`)

Postup vývoje:

1. Test fungování LED matice – zobrazení jednoduchého textu, jako inspiraci jsem použil tutoriál [\[1\]](#)
2. Test funkce hodin s počátečním napevno zadaným časem, výběr fontu pro čísla a písmena, výběr velikosti písma a výběr velikosti mezery mezi čísly
3. Synchronizace času přes NTP – přidání knihovny `NTPClient`
4. Přidání blikání dvojteček
5. Nastavení Wi-Fi a zobrazování IP – připojení k síti a výpis IP adresy posuvem.
6. Implementace webového rozhraní – zobrazení HTML stránky s formulářem pro nastavení alarmu a textu.
7. Přidání alarmu a posuvného textu – logika pro vyhodnocení alarmu a posun textu po displeji.

8. Ladění okrajových případů – například problémy se zobrazením delší IP adresy, výběr menšího fontu, přesné zarovnání textu na střed.

Doporučení pro zopakování projektu:

- Je dobré si jednou za čas zkontrolovat, jestli je zapojení GPIO pinů mezi ESP8266 a LED maticí správné a jestli jsem ho omylem nezměnil
- Když program běží tak kontroluji, aby náhodou nevypadl žádný z GPIO pinů (povedlo se mi tak usmažit jedno ESP)
- Před každým zapnutím je dobré si zkontrolovat, jestli je vše zapojeno správně a jestli jsem náhodou nepopletl kontakty (abych si náhodou neodpálil matici nebo ESP). Také je dobré si vždy zkontrolovat jestli
- Funkce jako scrollování textu nebo blikání časovače je dobré řešit pomocí neblokujících funkcí (Ticker nebo časové značky `millis()`), ne `delay()`.

Testování projektu

1. Test: Spuštění a nasazení aplikace

Cíl: Ověřit, zda se zařízení po zapnutí úspěšně inicializuje, připojí k Wi-Fi, získá čas z NTP serveru a zobrazí IP adresu na displeji

Postup:

Připojit napájení k ESP8266, LED matici a následně pozorovat sériový výstup (Serial Monitor) i samotnou LED matici

Očekávaný výsledek:

Na displeji se zobrazí IP adresa, poté se začne zobrazovat aktuální čas

Skutečný výsledek:

IP adresa, která se zobrazila na LED matici byla stejná jako IP adresa v sériovém výstupu a přešla plynule na aktuální čas

Výsledek: Test úspěšný

2. Test: Přístup k webové stránce

Cíl: Ověřit, zda je možné z jiného zařízení v síti přistoupit na webový server běžící na ESP8266.

Postup:

Získám IP adresu ESP8266, otevřu webový prohlížeč a zadám tuto IP adresu, přičemž tím ověřím načtení HTML stránky

Očekávaný výsledek:

V prohlížeči se zobrazí webová stránka s políčky pro zadání alarmu a zprávy

Skutečný výsledek:

Webová stránka se načetla správně, bylo možné zadat nový čas alarmu a vlastní zprávu

Výsledek: Test úspěšný

3. Test: Nastavení a spuštění alarmu

Cíl: Ověřit, zda zařízení spustí alarm v určený čas

Postup:

Přes webové rozhraní nastavit čas alarmu (např. 12:01), aktivovat alarm, počkat na dosažení tohoto času

Očekávaný výsledek:

Jakmile se aktuální čas shoduje s nastaveným časem alarmu, na displeji se zobrazí zpráva alarmu

Skutečný výsledek:

Ve zvolený čas se zobrazil alarm

Výsledek: Test úspěšný

4. Test: Zadání vlastní zprávy

Cíl: Ověřit možnost zadat vlastní zprávu, která se bude ukazovat na LED displeji

Postup:

Přes webové rozhraní zadám novou textovou zprávu, např. „Budíček!“ a následně potvrdím a začnu sledovat displej

Očekávaný výsledek:

Zpráva se zobrazí po alarmu

Skutečný výsledek:

Zpráva se objevila na displeji a po určité době se displej opět přepnul na zobrazení času.

Výsledek: Test úspěšný do jisté míry, delší zprávy se nezobrazí celé

5. Test: Alarm s textem v jiném písmu

Cíl: Ověřit, že ESP správně zpracuje a zobrazí text napsaný uživatelem v jiném písmu než standardní latinka – např. v azbuce

Postup:

Nejdříve se připojím se na webové rozhraní zařízení a následně nastavím alarmový text např. na: „это русский“

Očekávaný výsledek:

Na displeji se zobrazí nově zadaný text v azbuce

Skutečný výsledek:

Místo toho, než aby se na obrazovce zobrazil text v azbuce se na obrazovce neukázalo nic. Po chvíli však obrazovka začala znovu ukazovat čas

Výsledek: Test neúspěšný

Nasazení projektu

Pro úspěšné nasazení projektu je třeba mít:

- ESP8266 (já osobně použil NodeMCU, lze použít i Wemos D1 mini)
- LED panel 64x32 s hub75 konektorem
- Napájecí adaptér 5V pro displej
- Propojovací kabely Female/Female
- MikroUSB kabel pro nahrání programu a napájení ESP8266

Postup nasazení projektu:

1. Stáhněte prostředí Arduino IDE na <https://www.arduino.cc/en/software>
2. V nastavení Arduino IDE přidejte podporu pro ESP8266:
 - a. Soubor> Předvolby> Dodatkové URL pro Správce desek:
http://arduino.esp8266.com/stable/package_esp8266com_index.json
 - b. Nástroje> Správce desek> ESP8266 by ESP8266 Community
3. Vyberte správnou desku:
 - a. Nástroje> Deska> NodeMCU 1.0 (ESP-12E Module) nebo odpovídající
4. Po stažení projektu z Github vezměte soubory ze složky libraries a vložte je do ve složce Arduino do složky libraries
5. Připojte ESP8266 k LED panelu následovně:
 1. ESP – D0 -> hub75 in – Latch
 2. ESP – D1 -> hub75 in – A
 3. ESP – D2 -> hub75 in – B
 4. ESP – D4 -> hub75 in – NC
 5. ESP – D5 -> hub75 in – Clock
 6. ESP – D6 -> hub75 in – D
 7. ESP – D7 -> hub75 in – Red0
 8. ESP – D8 -> hub75 in – C
 9. ESP – Ground -> hub75 in – Ground
 10. hub75 in – Green1 -> hub75 out – Green0
 11. hub75 in – Blue0 -> hub75 out – Green1
 12. hub75 in – Red1 -> hub75 out – Red0
 13. hub75 in – Blue1 -> hub75 out – Blue0
 14. hub75 in – Green0 -> hub75 out – Red1
6. Připojte ESP8266 k PC pomocí USB
7. Změňte heslo a název sítě (SSID, password) na Vaši síť
8. Vyberte správný port: Nástroje > Port
9. Klikněte na Upload (Šipka vlevo nahoře)
10. Po připojení ESP k Wi-Fi se zobrazí IP adresa přes kterou se můžete připojit k webové stránce, na LED matici se následně ukáže aktuální čas
11. Webová stránka umožňuje: nastavit čas alarmu, zadat vlastní zprávu, která se zobrazí, změnit časové pásmo

GIT odkaz

<https://github.com/OvcaTV/Led-Matrix-ESPClock>

Závěr

V této ročníkové práci se mi podařilo udělat plně funkční digitální hodiny na základě mikrokontroleru ESP8266 a LED matice 64×32 s roztečí 2,5 mm. Za pomoci knihoven PxMatrix, NTPClient a ESP8266WebServer jsem udělal přesné časové řízení synchronizované přes internet s čitelným zobrazením času ve formátu hh:mm:ss k tomu jsem ještě udělal jednoduché webové rozhraní pro nastavení alarmu, text alarmu a změnu časové zóny. Díky upravenému rozmístění číslic, plynulému blikání dvojtečky poskytuje zařízení pro uživatele co nejvěrnější podobu digitálních hodin a díky automatickému přepínání mezi režimy (zobrazení IP adresy, času, popř. zprávy) dává uživateli pocit chytrých hodin.

Projekt mi umožnil prohloubit znalosti práce s LED maticemi a ESP od konfigurace GPIO pinů a správy dynamického skenování matice až po návrh jednoduché webové aplikace v HTML/CSS. Naučil jsem se, jak správně neodpálit ESP, jak získávat čas skrz Wi-Fi připojení. Díky testování na různých scénářích byla ověřena robustnost řešení a odhalena i výraznější slabina – zobrazení speciálních znaků mimo latinku.

Do budoucna bych rád rozšířil podporu pro více znakových sad (UTF-8), přidal možnost ovládání barev a jasu přímo z webu. Tento projekt tak nejen prokázal praktickou využitelnost mikrokontrolerů, ale zároveň posloužil jako komplexní ukázka integrace onoho mikrokontroleru, displeje a webového rozhraní, kterou lze dále rozšiřovat a přizpůsobovat dalším účelům.