



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY

A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

DEPARTMENT OF CONTROL AND INSTRUMENTATION

DIGITRONKOVÉ HODINY

NIXIE CLOCK

SEMESTRÁLNÍ PRÁCE BSPC

SEMESTRAL THESIS BSPC

AUTOŘI PRÁCE

AUTHORS

Martin Štastný, Ondřej Vybíral

BRNO 2019

ZADÁNÍ SEMESTRÁLNÍ PRÁCE

1. Připojit hodiny k Wi-Fi síti
2. Běh hodin bez použití externích periférií (externí RTC)
3. Možnost aktualizace času z NTP serveru
4. Nastavení budíčku
5. Webové rozhraní pro změnu parametrů
6. Zabezpečit přístup do webového rozhraní
7. Navrhnout a realizovat hodiny

Obsah

1	Úvod.....	1
2	Hardware.....	1
3	Software.....	4
4	Závěr.....	6

Seznam obrázků

Obrázek 1: Hlavní obrazovka (webové rozhraní)	5
Obrázek 2: Inicializace programu.....	7
Obrázek 3: Smyčky.....	8

1 ÚVOD

Cílem projektu bylo vytvořit hodiny s použitím elektronek. Pro jejich ovládání byl použit mikrokontrolér WeMos D1 Mini ESP8266 WiFi modul. O veškerou logiku se tedy staral zmíněný mikrokontrolér.

2 HARDWARE

Jako řídicí člen byl použitý zmíněný WeMos D1 Mini ESP8266 WiFi modul. Jak název napovídá obsahuje Wi-Fi modul, který pracuje na frekvenci 2,4 GHz a podporuje standardy 802.11 b/g/n. Maximální rychlost je 72,2 Mbps. Veškeré piny jsou v 3,3 V logice.

Tabulka 1: Specifikace WeMos D1 Mini ESP8266 WiFi modul

Mikrokontrolér	ESP8266EX
USB – Sériový Převodník	CH340G
Digitální I/O piny	9
Analogové vstupy	1
Flash	4MB
Taktovací frekvence	80MHz/160MHz
Rozměry	
Délka	34,2mm
Šířka	25,6mm
Výška	7mm
Váha	8,26g

Tabulka 2: Piny a jejich využití WeMos D1 Mini ESP8266 WiFi modul

WeMos piny	Funkce	ESP8266 piny
TX	TXD	TXD
RX	RXD	RXD
A0	Analogový vstup	A0
D0	I/O	GPIO16
D1	I/O, SCL	GPIO5
D2	I/O, SDA	GPIO4
D3	I/O, 10k pull-up	GPIO0
D4	I/O, 10k pull-up, LED	GPIO2
D5	I/O, SCK	GPIO14
D6	I/O, MISO	GPIO12
D7	I/O, MOSI	GPIO13
D8	I/O, 10k pull-down, SS	GPIO15
G	Země	GND
5V	5V	
3V3	3.3V	3.3V
RST	Reset	RST

Hodiny se skládají ze dvou desek. Horní deska obsahuje digitrony, jejich drivery a posuvné registry. Digitron je výbojka plněná plynem a slouží k zobrazovacím účelům. Na spodní desce je realizováno jak řízení, tak i další rozšiřující periférie. Tyto dvě desky jsou propojeny deseti pinovým propojem, kterým se přenáší anodové napětí, datové piny posuvných registrů a zapínání/vypínání oddělovacích teček.

Spodní deska obsahuje řídicí Wemos D1 mini, který má rozšířené analogové vstupy multiplexerem DG408. Ten se stará o přepínání vstupů:

- Měření napětí pro digitrony
- Měření proudu digitrony
- Přetížení proudu nebo napětí na digitronech (otevřený kolektor)
- Měření úrovně okolního hluku
- Stavů přepnutí hlavního vypínacího tlačítka
- Nastavitelné tlačítko 1
- Nastavitelné tlačítko
- Měření vstupního napětí

O rozšíření výstupů se stará posuvný registr 74HC595. Ten ovládá:

- Bzučák
- Relé
- Nastavitelnou LED
- Spodní a horní oddělovací LED na horní desce
- Přepínání multiplexoru

Na desce jsou přítomny čtyři zdroje napětí: zdroj pro digitrony, zdroj 3,3V, 5V, 12V. Digitronový zdroj a 5V zdroj je realizován měniči. Naopak napětí 3,3V a 12V je realizováno lineárními stabilizátory. Zdroj pro digitrony má implementované zapínání/vypínání z dedikovaného pinu Wemosu. Dále je v designu implementována možnost jak měření, tak nastavení napětí i proudu. Přetížení měřících kanálů je zvlášť detekováno analogovým vstupem.

Mezi rozšiřující periferie řídicí desky patří:

- Detektor hluku
- Bzučák
- Relé
- Dvě tlačítka
- Indikační LED
- Výstup pro servomotor
- Pinheader pro +5V, TX, RX, GND

3 SOFTWARE

Vývoj softwaru lze využít vývojové prostředí Arduino IDE (velmi rozšířené u mikrokontroléru vycházející z konceptu Arduina). Toto prostředí je multiplatformní (Linux, Windows, macOS, webové verze), desktopová verze je programována v jazyce Java. Další alternativou je vývoj v prostředí Visual Studio Code vyvíjené společností Microsoft. Toto prostředí je též dostupné pro výše zmíněné tři operační systémy. Pro úspěšné kompilování souboru v tomto rozhraní je třeba doinstalovat rozšíření pro Arduino.

Samotný program využívá knihovny pro připojení k Wi-Fi, práci s časem, HTTP server, multicast DNS, obsluhu posuvných registrů pro rozšíření periférií, a nastavení časovače skrze přerušení. Viz tab. 3. Jednotlivé diagramy jsou v příloze. Všechny jsou veřejně dostupné na internetu a lze je stáhnout a použít. Reset mikrokontroléru vede k vyrestování nastavení, má tak nahrazovat tlačítko pro tovární nastavení. V knihovně ESP8266WiFi je třeba upravit v souboru ESP8266WiFiGeneric.cpp u funkce `hostByName()` parametr na timeout 100 (ms). Konkrétně na `hostByName(aHostname, aResult, 100)`.

Tabulka 3: Použité knihovny

Použité knihovny	Využití
TimeLib	Čas
ESP8266WiFi	Připojení k Wi-Fi sítím
WiFiUdp	Práce s UDP pakety
ESP8266WebServer	Provoz http serveru
ESP8266mDNS.h	Multicast DNS
ShiftRegister74HC595.h	Posuvný registr pro rozšíření vstupů
Ticker.h	Časovač skrze přerušení

Nastavení hodin mělo být, pokud možno snadno dostupné. Bylo zde několik variant, které bylo možné využít. Namátkou je to například specializovaný software, který by jen komunikoval s hodinami přes sériovou linku. My jsme zvolili elegantnější řešení, kdy komunikace probíhá skrze webové rozhraní (přes Wi-Fi). Tím je vyloučena potřeba aplikace a ovládání je tak multiplatformní. Výchozí účet a heslo je: `admin`, `admin`. Po přihlášení se uživatel dostane na hlavní stránku (`/main`), která se každou 1 vteřinu aktualizuje. Viz obr. 1.

Jednotlivé stánky a s jejich příponami jsou v tab. 4. Zabezpečení je u všech stránek řešeno pomocí porovnání zadaného hesla a účtu, které se ukládá do dočasných pracovních proměnných `tmpLogin` a `tmpPassword`. To znamená, že pokud je rozhraní odemčené, tak je dostupné pro všechny uživatele, dokud se jeden z nich neodhlásí. Potom vyžaduje znovu po všech uživatelích autorizaci.

Jednotlivé vstupy formulářů pro hodiny a podobně jsou ošetřeny proti nežádoucím vstupům jako jsou znaky: `&x@` a jiné.



Obrázek 1: Hlavní obrazovka (webové rozhraní)

Tabulka 4: Seznam použitých stránek

Funkce pro jednotlivé stránky	Účel	Metoda	Přípona
<code>void handleRoot();</code>	Přihlašovací okno	GET	/
<code>void handleMain();</code>	Hlavní obrazovka	GET	/main
<code>void handleNotFound();</code>	Chybný parametr	-	-
<code>void handleLogin();</code>	Kontrola autorizace	POST	/login
<code>void handleSetTime();</code>	Nastavení času	GET	/settime
<code>void handleSetMessage();</code>	Zpracování dat z času	POST	/setmessage
<code>void handleAlarm();</code>	Nastavení budíčku	GET	/alarm
<code>void handleAlarmMessage();</code>	Zpracování dat z budíčku	POST	/alarmmessage
<code>void handleAlarmOff();</code>	Zpracování dat - vypnutí budíčku	GET	/alarmoff
<code>void handleChLogin();</code>	Změna hesla	GET	/chlogin
<code>void handleChLoginMessage();</code>	Zpracování dat - změna hesla	POST	/chloginmessage
<code>void handleLogout();</code>	Odhlášení	GET	/logout
<code>void handleAccess();</code>	Změna přihlašovacích údajů k Wi-Fi	GET	/access
<code>void handleAccessMessage();</code>	Zpracování dat	POST	/accessmessage
<code>void handleNTP();</code>	Nastavení parametrů NTP	GET	/ntp
<code>void handleNTPMessage();</code>	Zpracování dat z NTP	POST	/ntpmessage
<code>void handleDixi();</code>	Za/vypnutí digitron	GET	/dixi

Čas se při zapnutí defaultně počítá od 00:00:00 1.1.1970. Program se snaží pravidelně dotazovat NTP (Network Time Protocol) serveru pro aktuální čas. Výchozí server je `ntp.cesnet.cz`. Interval dotazů na NTP server je ve výchozím nastavení 180 sekund (kvůli demonstraci) v makru `NTPSYNC`. Časovou zónu je též možné měnit, výchozí je +2.

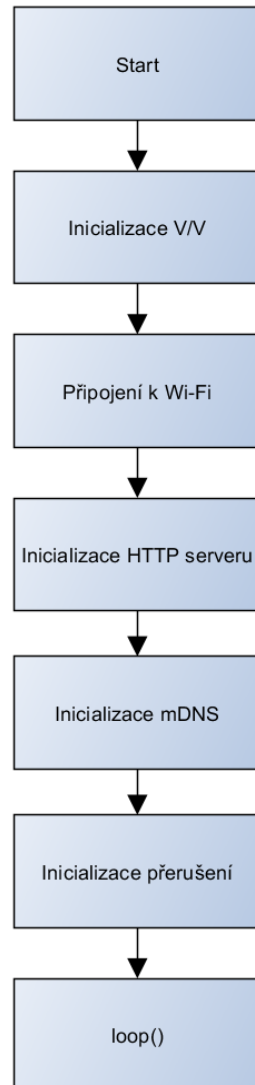
V programu probíhají paralelně k hlavní smyčce dvě vedlejší pomocné smyčky s nastavitelnou četností spouštění. Vzhledem k tomu, že je na čipu ESP8266 dostupný pouze jeden HW timer je druhá smyčka spouštěna první rychlejší. Rychlejší smyčka se stará především o čtení analogových vstupů. V pomalé smyčce je update digitronů, obsluha tlačítek: přepínání datum/čas a vypínání budíku.

4 ZÁVĚR

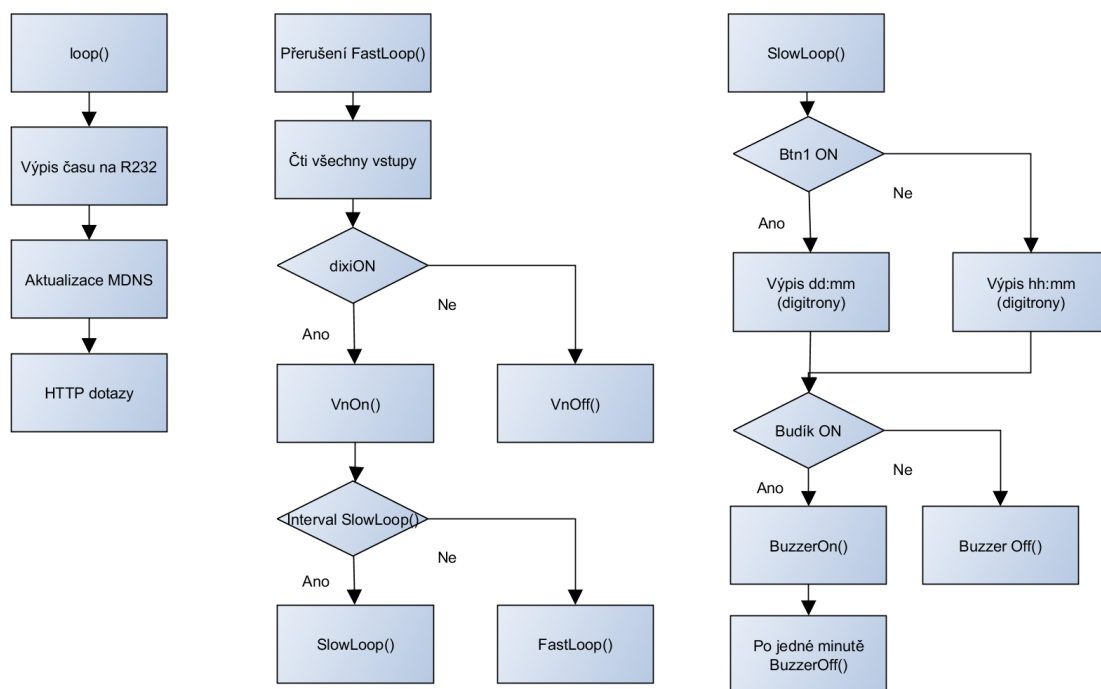
V rámci práce byl naprogramován mikrokontrolér Wemos D1 mini v jazyce C++. Ten obsahoval HTTP server vč. rozhraní pro ovládání nastavení hodin. V něm bylo například možné měnit čas, nastavení budíku a jiné.

Dále v rámci práce byl navrhnout hardware, který obsahoval digitrony pro zobrazování času. Zároveň zde byly tlačítka, kterými bylo možné zapínat nebo vypínat budík a měnit zobrazení času nebo datumu. Následně byl návrh realizován.

Přílohy



Obrázek 2: Inicializace programu



Obrázek 3: Smyčky