



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Mikroprocesorové a vestavěné systémy

S - Řízení spotřeby energie mikrokontroléru

Řešitelka:
Nadzeya Antsipenka (xantsi00)

10. prosince 2024

Obsah

1	1 Úvod	2
2	Rozbor a návrh řešení	4
3	Vlastní řešení	5
4	Závěrečné hodnocení	6
	4.1 Autoevaluace	6
5	Reference	7

1 1 Úvod

Zadáním projektu bylo pro mikrokontrolér zvolené platformy shrnout klíčové informace o jím podporovaných režimech činnosti, o spotřebě mikrokontroléru v těchto režimech, o možných přechodech mezi nimi, o jejich ceně (latence) a mechanismech těchto přechodů. Jako vestavnou platformu jsem vybrala ESP32.

Mikrokontrolér ESP32, jako například Wemos D1 R32, který mám já, podporuje pět režimů činnosti s odlišnými charakteristikami, přechody a souvisejícími stavy zdrojů. Podle potřeby výkonu může čip přepínat mezi různými režimy napájení. Tyto režimy jsou:

- **Active Mode (Run)**
Spotřeba: v závislosti na aktivních perifériích a frekvenci procesoru (až 260 MHz) se obvykle pohybuje od 160 mA do 240 mA. Bylo také zjištěno, že čip občas odebírá více než 790 mA, zejména při současném používání WiFi a Bluetooth
Dostupné zdroje: všechny zdroje, včetně procesoru, pamětí, periférií a komunikačních modulů, jsou plně funkční
- **Modem Sleep Mode**
Spotřeba: přibližně 3 mA při nízkých otáčkách a 20 mA při vysokých otáčkách
Dostupné zdroje: je aktivní vše kromě WiFi, Bluetooth a rádia, protože vysílačky Wi-Fi nebo Bluetooth se mezi intervaly vysílání vypínají. Procesor zůstává aktivní a hodiny jsou konfigurovatelné
Zdroje probuzení: aby připojení zůstalo živé, Wi-Fi, Bluetooth a rádio se probouzejí v předem definovaných intervalech. Tento postup se označuje jako Association Sleep Pattern. Během tohoto schématu spánku ESP32 přepíná mezi aktivním režimem a režimem spánku modemu. Za tímto účelem se ESP32 připojuje ke směrovači v režimu stanice pomocí DTIM beacon mechanism. Modul Wi-Fi je mezi dvěma intervaly signálů DTIM vypnut a poté automaticky zapnut těsně před příchodem dalšího signálu. To vede k úspoře energie
Latence: žádná, okamžitý přechod
- **Light Sleep Mode**
Spotřeba: 0,8-1,2 mA, v závislosti na aktivovaných perifériích
Dostupné zdroje: během režimu lehkého spánku je CPU pozastaveno tím, že jsou zakázány jeho hodinové impulsy. Naproti tomu RTC a ULP-procesor zůstávají aktivní
Zdroje probuzení: Časovače, GPIO, UART a další periferie.
Latence: Rychlé probuzení (1 ms)
- **Deep Sleep Mode**
Spotřeba: 10-150 μ A, v závislosti na konfiguraci.
Dostupné zdroje: CPU, většina paměti RAM a všechny digitální periferie

jsou vypnuty. V provozu zůstávají pouze následující části čipu: ULP koprocessor, Radič RTC, Periferie RTC, Rychlá a pomalá paměť RTC

Zdroje probuzení: GPIO, časovače a touch piny **Latence:** mírně delší doba probouzení (5 ms)

- Hibernation Mode

Spotřeba: 2-8 μ A

Dostupné zdroje: pouze jeden časovač RTC (na pomalých hodinách) a několik GPIO RTC pro buzení čipu zůstávají běžet

Zdroje probuzení: omezeno na GPIO nebo časovač RTC

Latence: významné zpoždění způsobené opětovnou inicializací

Shrnutí aktivit dostupných v různých režimech je v tabulce 1. Tabulka byla převzatá z Shrnutí spotřeby energie v různých režimech je v tabulce 2. Tabulka byla převzatá z

Tabulka 1: Resource States Across Power Modes

Power Mode	Active	Modem-sleep	Light-sleep	Deep-sleep	Hibernation
Sleep Pattern	-	Assoc. sleep	-	ULP sensor-mon.	-
CPU	ON	ON	PAUSE	OFF	OFF
Wi-Fi/BT Baseband	ON	OFF	OFF	OFF	OFF
RTC Memory/Periph.	ON	ON	OFF	ON	OFF
ULP Co-Processor	ON	ON	ON	ON/OFF	OFF

Tabulka 2: Power Consumption and Mode Descriptions

Power Mode	Description	Power Consumption
Active	Wi-Fi Tx/Rx, Bluetooth Tx/Rx	Refer to Table 5-4
Modem-sleep	CPU powered, radios off: 240 MHz (dual-core): 30–68 mA 160 MHz (dual-core): 27–44 mA Normal (80 MHz): 20–31 mA	 30–68 mA 27–44 mA 20–31 mA
Light-sleep	CPU paused, peripherals off	0.8 mA
Deep-sleep	ULP coprocessor on; RTC timer and memory active RTC timer + RTC memory	150 μ A (ULP) 10 μ A
Hibernation	RTC timer only	5 μ A
Power off	CHIP_PU at low level, full shutdown	1 μ A

2 Rozbor a návrh řešení

Rozbor a návrh řešení poskytuje přehled prostředků použitých pro realizaci projektu.

- Rotační enkodér
Rotační enkodér je zapojen tak, že CLK je připojen na GPIO25, DT na GPIO26 a SW na GPIO27. Rotační enkodér zjišťuje směr otáčení (ve směru hodinových ručiček a proti směru) a stisk tlačítka. Na výstupu je pak druhá mocnina vstupu.
- Bluetooth komunikace
Pro umožnění práce z Bluetooth je použita knihovna `BluetoothSerial.h`. Aplikace přijímá příkazy k ovládání LED, režimů spánku a jiných funkcí:
 - `led_on` – Zapne LED.
 - `led_off` – Vypne LED.
 - `Go_to_modem_sleep` – Přepne do modem-sleep režimu.
 - `Wake_up_from_light_sleep` – Přepne zpět do aktive-mode režimu.
 - `Go_to_light_sleep` – Přepne do light-sleep režimu.
 - `go_to_sleep` – Přepne do deep-sleep režimu.
- Dotyková čidla
Na piny GPIO12 a GPIO14 jsou připojena dotyková čidla. Dotyk do GPIO12 spouští modem-sleep. GPIO14 spouští light-sleep a dotyk do GPIO12 a GPIO14 zároveň spouští deep-sleep. Hodnoty se čtou pomocí funkce `touchRead()`.
- LED dioda
LED dioda je připojena na GPIO13 a reaguje na Bluetooth příkazy (`led_on` / `led_off`).

- Wake-up GPIO

Na GPIO2 je navázáno tlačítko, které má za úkol probudit systém z light a deep-sleep.

3 Vlastní řešení

Program spravuje čtyři režimy:

- Active Mode
Procesor a periferie jsou aktivní, což umožňuje plné spuštění programu. Používá se pro čtení rotačního enkodéru, komunikaci Bluetooth a ovládání LED. Výchozí režim po spuštění systému.
- Modem Sleep Mode
Procesor a periferie jsou aktivní. Komunikace přes Bluetooth stále probíhá díky Association Sleep Pattern a rotační enkodér snímá data. Spustit Modem Sleep Mode lze pomocí `esp_wifi_set_ps(WIFI_PS_MIN_MODEM)` a přejít zpátky do Active mode `esp_wifi_set_ps(WIFI_PS_NONE)`
- Light Sleep Mode
CPU je pozastaveno, nelze komunikovat přes Bluetooth, a rotační enkodér nesnímá žádná data, ale jeho stav bude zachován po probuzení. Vstup do Light Sleep Mode zajišťuje `esp_light_sleep_start()` a výstup stejně jako výstup z Deep Sleep Mode `esp_sleep_enable_ext0_wakeup()`
- Deep Sleep Mode
Jenom paměť RtC není vypnuta a všechny inicializované hodnoty kromě hodnot zachovaných v `RTC_DATA_ATTR` jako například flag `wasInDeepSleep`, který ukazuje ze zařízení se vrátilo z deep-sleep, a proměna `bootCount`, která ukazuje kolikrát se provedla inicializace, tedy kolikrát zařízení bylo v deep-sleep + 1. Vstup do deep-sleep provede `esp_deep_sleep_start()`.

Tabulka dolu zahrnuje dostupnost klíčových prostředků mikrokontroléru v mikrokontrolérem podporovaných režimech činnosti

Legenda

- **A** – Aktivní režim (Active).
- **M** – Modem-sleep režim (Modem Sleep).
- **L** – Light-sleep režim (Light Sleep).
- **D** – Deep-sleep režim (Deep Sleep).
- **N/A** – Nedostupné v daném režimu.
- **A** – Dostupné v daném režimu.

Komponenta / Proměnná	A	M	L	D
CPU	A	A	N/A	N/A
Bluetooth	A	A/N/A	N/A	N/A
RTC časovač	A	A	A	A
ULP koprocessor	N/A	N/A	N/A	A
LED	A	A	N/A	N/A
Rotary encoder	A	A	N/A	N/A
Dotyková čidla	A	A	A	A
Bluetooth příkazy	A	A/N/A	N/A	N/A
Proměnná fromLightSleep	false	false	true	N/A
Proměnná wasInDeepSleep	false	false	false	true
Externí probuzení (GPIO)	N/A	A	A	A

4 Závěrečné hodnocení

Shrnutí vlastností řešení Cílem projektu bylo navrhnout a implementovat systém na bázi mikrokontroléru ESP32, který umožní správu napájení prostřednictvím různých režimů spánku (Active, Modem Sleep, Light Sleep, Deep Sleep). Projekt splňuje požadavky zadání tím, že implementace pokrývá všechny klíčové spánkové režimy ESP32 a podporuje více zařízení, jako jsou čidla a Bluetooth, zajišťují škálovatelnost řešení.

4.1 Autoevaluace

E

1 – Zadání přišlo mně trochu matoucí a místo toho, abych se zeptala u vedoucího, ptala jsem se svých spolužáků, jak mám zadání pochopit. A tento postup nelze nazvat úplně správným, na druhé straně jsem prostudovala vhodnou literaturu a přišla jsem na vhodné řešení.

F

4-5 – Projekt plně splnil požadavky na funkčnost, protože implementoval všechny klíčové režimy spánku ESP32 (Active, Modem Sleep, Light Sleep, Deep Sleep) a správně reaguje na externí události, jako jsou přepínače a dotyková čidla.

Q

2 – Kód je strukturovaný do přehledných funkcí, každá odpovídá konkrétnímu úkolu (generování signálu, výpočet vzdálenosti, aktualizace displeje). Kód je komentovaný, takže údržba je jednoduchá.

P

N/A – Ještě neproběhla

D

3 – Dokumentace je dobře zpracována, zahrnuje podrobný popis implementace a vysvětlení kódu, což umožňuje snadné pochopení fungování celého systému.

5 Reference

- Tabulka 1 <https://randomnerdtutorials.com/esp32-deep-sleep-arduino-ide-wake-up-sources/>
- Schéma zapojení desky ESP32 D1 R32
https://www.halloweenfreak.de/arduino/pdfs/D1_R32_ENG.pdf
- Přehled spotřeby energie v různých režimech
[https://lastminuteengineers.com/esp32-sleep-modes-power-consumption/?utm_content = cmp - true](https://lastminuteengineers.com/esp32-sleep-modes-power-consumption/?utm_content=cmp-true)
- ESP32 datasheet a Tabulka 2
https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf
- zapojení rotačního enkodéru
https://esp32io.com/tutorials/esp32-rotary-encodergoogle_vignette