**A black background with a black square

Description automatically generated with medium confidence**

**Kauno technologijos universitetas**

Elektros ir elektronikos fakultetas

**Baterijos išsikrovimo laiko skaičiavimas**

Įterptinės sistemos T170B417  
Laboratorinis darbas Nr. 2

|  |  |
| --- | --- |
|  |  |
| **Dovydas Liutkus**  Studentas / Studentė | (parašas) (data) |
|  |  |
| **Vytautas Daunoras**  Dėstytojas | (parašas) (data) |
|  |  |

**Kaunas, 2024**

Variantas 39

39. Baterija: Li-SOCL2 (AAA700); Matavimo periodas Ts: 0,2s; Vidurkinimo ir atvaizdavimo periodas Tv: 3s.

A screenshot of a computer

Description automatically generated

*1 pav. Naudota būsenų diagrama*

## Aktyvaus ir energijos taupymo režimų trukmių įvertinimas

**Probleminė užduotis 1**

Kaip įvertinti programos darbo laikus aktyvaus (RUN) ir energijos taupymo (SLEEP) miego režimuose?

* Paleisti taimerį, kiekvieną kartą kai valdiklis nueina miegoti ir atsikelia nuskaityti taimerio vertes. Taip galime apskaičiuoti intervalus kada valdiklis miega ir kiek kartų miega per vieną ciklą. Ciklo pabaigoje dar kartą nuskaitome taimerį, kad gautume bendrą ciklo trukmę. Išsiunčiame duomenis per UART (kadangi negalime naudoti debug režimo kai programoje panaudojamas miegas).
* Kitas, panašus, sprendimas - indikuoti nuėjimą miegoti ir atsikėlimą per GPIO išvadą. Pavyzdžiui visada laikant aukštą lygį, prieš einant miegot išvesti žemą lygį ir atsikėlus vėl aukštą. Šiam sprendimui reiktų osciloskopo su kuriuo stebėtume ir išmatuotume miego trukmę.

Pasirinktas pirmasis variantas, siekiant neturėt priklausomybės nuo papildomos įrangos.

**Lentelė 1. Miego trukmės matavimo rezultatai.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Matavimo nr | 1 | 2 | 3 | 4 | 5 |
| Ciklo trukmė,s | 2,79692041 | 2,79692041 | 2,79692065 | 2,79692045 | 2,79692045 |
| Miego suma,s | 2,78354242 | 2,78354266 | 2,78354266 | 2,78354242 | 2,78354266 |

## Mikrovaldiklio vidutinės vartojamos srovės skaičiavimas su CubeMX

Baterijos Li-SOCL2 (AAA700) maksimali srovė yra 10mA, o valdiklis RUN režime naudoja 24,78mA. Todėl pasirenkame kitą bateriją. Pasirinkta Li-SOCL2(A3400)(1x1) turinti 100mA srovę.

A screenshot of a computer

Description automatically generated

*2 pav. Vidutinės srovės įvertinimas*

Kaip matome iš 2 pav. Su rastom miego ir veikimo trukmėm valdiklis turėtų vidutiniškai vartoti 7,79 mA ir su Li-SOCL2(A3400) baterija turėtų veikti 18 dienų ir 4 valandas.

## Mikrovaldiklio taktinio dažnio parinkimas

**Probleminė užduotis 2**

Parinkite mikrovaldiklio taktinį dažnį. Kuris minimizuoja vidutinę suvartojamą srovę ir užtikrina kuo ilgesnį įrenginio veikimą iki baterijos išsikrovimo.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

*3 pav. Dvi vartojamos galios estimacijos su skirtingais RUN dažniais (100MHz viršuje ir 60MHz apačioj) simuliacijoje*

Keičiant miego dažnį gyvavimo trukmė žymiai pasikeičia, o keičiant RUN dažnį įtaka maža, nes procesorius labai mažą dalį (mažiau nei 1%) praleidžia miege. Nors aukštesnis RUN dažnis sutrumpina valdiklio aktyvų veikimą tai daro mažą įtaką galios suvartojimui.

## Sistemos vartojamos srovės matavimas

Kadangi naudojau STM32F429-I disco maketą su integruotu ekranu ir pilnai neatjungiamu ST-LINK srovės suvartojimas buvo didesnis nei didžiausia X-NUCLEO-LPM01A tiekiama srovė (apie 70mA). Todėl matavimą atlikau su laboratoriniu maitinimo šaltiniu, osciloskopu ir 0,9 Omo šunto varža.

A diagram of a circuit

Description automatically generated

*4 pav. Matavimo schema*

Šunto varža buvo prijungta vietoj JP3 jungiklio (4 pav).

A screenshot of a computer

Description automatically generated

*5 pav. Srovės matavimo oscilograma (įtampa ant šunto varžos)*

A screen shot of a computer

Description automatically generated

*6 pav. Srovės matavimo oscilograma*

Iš oscilogramų galima ryškiai atskirti momentus kai yra atnaujinamas ekranas - aukšti ir ilgiau trunkantys impulsai. Trumpiau trunkantys impulsai - tai matavimo momentai (kas 0.2s) . Likusį laiką mikrovaldiklis miega.

**Lentelė 2. Srovės vartojimas prie skirtingų taktinių dažnių**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Taktinis dažnis,MHz | 100 | 50 | 25 | 10 |
| Vidutinė srovė,mA | 11 | 9 | 8 | 7 |
| Maksimali srovė,mA | 161 | 144 | 139 | 144 |

Mažinant taktinį dažnį vidutinė vartojama srovė taip pat mažėjo. Maksimali srovė mažėjo, tačiau šis išmatavimas nėra itin tikslus, nes maksimalios srovės vertė nebuvo stabili.

## Stop režimo realizacija

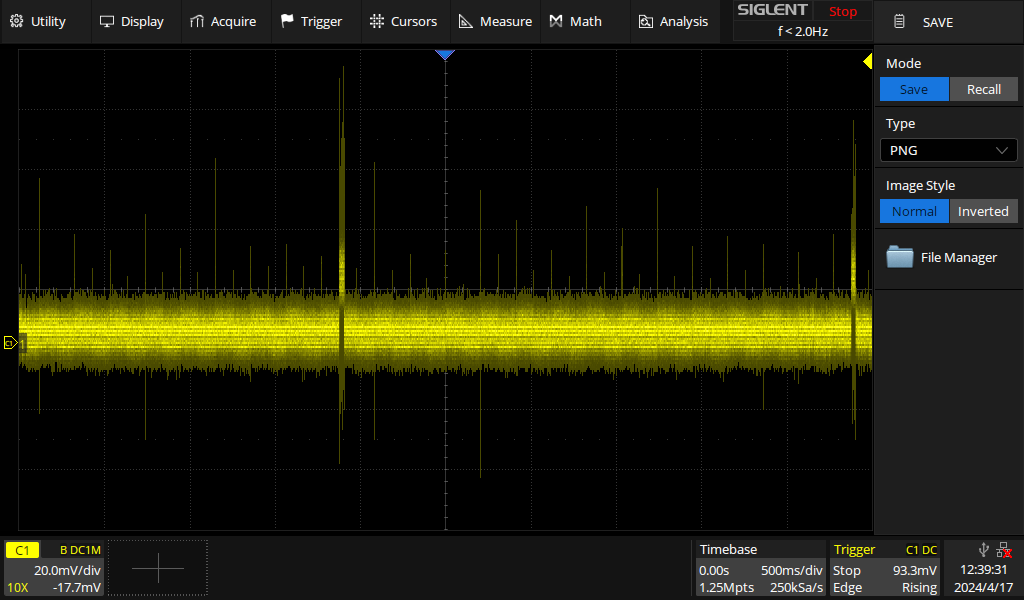
Stop režimą realizavau pakeičiant TIM6 į RTC. Įėjimas į miega atliekamas su šia funkcija:

HAL\_PWR\_EnterSTOPMode(PWR\_LOWPOWERREGULATOR\_ON, PWR\_STOPENTRY\_WFI);

Matavau srovę su 0,9 Omo šunto varža kaip praeituose skyriuose.

|  |  |
| --- | --- |
| Taktinis dažnis,MHz | 100 |
| Vidutinė srovė,mA | 4 |
| Maksimali srovė,mA | 100 |

Matosi, kad vidutinis srovės suvartojimas su STOP režimu buvo ženkliai mažesnis negu su paprastu miego režimu.



*7 pav. Šunto varžos įtampa Stop režime*

**Projektas:**

<https://github.com/dovydasliutkus/Iterptiniu_Lab2>