



Kauno technologijos universitetas

Elektros ir elektronikos fakultetas

Inžinerinis projektas

Įterptinės sistemos, T170B417

Linas Beinorius

Studentas

Prof. dr. Žilvinas Nakutis

Dėstytojas

Kaunas, 2025

Turinys

1. Teorinė analizė.....	4
1.1. Matavimo reikalavimai.....	4
1.2. Mikroprocesorius.....	4
1.3. Indikatorius.....	4
1.4. Apšvietumo jutiklis.....	5
2. Principinė schema ir BOM	9
3. Mikroprocesoriaus programos aprašas.....	11
4. Kompiuterio programa	13
5. Paklaidų matavimas	14
6. Galutinis projektas	18
7. Išvados	20
Šaltiniai.....	21
Priedas	22

Paveikslų sąrašas

1 pav.	Transimpedasinis stiprintuvas apšviestumui išgauti	5
2 pav.	Naudojamo jutiklio atvirkštinės srovės priklausomybė nuo apšviestumo	7
3 pav.	Aprašyta jutiklio atvirkštinės srovės priklausomybė nuo apšviestumo	8
4 pav.	Įterptinės sistemos blokinė diagrama	9
5 pav.	Principinė projekto schema	10
6 pav.	BOM.....	10
7 pav.	Programos būsenų diagrama	11
8 pav.	Sujungtos schemos nuotrauka	12
9 pav.	Pagrindinis programos langas	13
10 pav.	Parametrų nustatymo langas.....	13
11 pav.	Detektorių jautrumo kreivės[2]	14
12 pav.	Varžos tolerancijos ir fotodiode jautrumo koeficiento įtakos nuo išmatuojamos įtampos .	16
13 pav.	Mikroprocesoriaus ADC paklaidos	16
14 pav.	Apšviestumo matuoklis	18
15 pav.	Atvaizduojami parametrai kompiuteryje.....	18
16 pav.	Duomenų saugojimo pavyzdys (grafikas sukurtas rankiniu būdu vizualizacijai).....	19

1. Teorinė analizė

Darbo tikslas – suprojektuoti ir realizuoti įterptinę sistemą, kuri yra sudaryta iš STM32 valdiklio. Sistema turi atitikti šiuos reikalavimus:

- Du apšviestumo jutikliai
- LCD indikatorius (16x2)
- Duomenų perdavimo sąsaja į personalinį kompiuterį ir jų saugojimas (PK)

Mano užduoties variantas yra 5, tai reiškia, jog įrenginys turi atlikti tokius matavimus:

- 2 kanalų apšviestumas.
- Vidurkio ir kanalų skirtumo skaičiavimai.
- Signalo pralaidumo juostas yra 0 – 100 Hz.
- Indikatoriuje vidurkis atvaizduojamas kas 2 sekundes.
- Į kompiuterį siunčiami duomenys yra 0.2 s periodu.

1.1. Matavimo reikalavimai

Skaičiavimo reikalavimams reikalinga iki 100 Hz pralaidumo juosta, tai reiškia, jog diskretizuoti reikia bent 2 kartais daugiau vieną kanalą. Tai nėra didelis dažnis, galima sudaryti taimerį, kurio diskretizavimo periodas yra 400 (dvigubai daugiau nei teoriškai reikalingas) Hz ir kas kart reikalauti kanalų reikšmes. Paprastumo dėlei galima turėti du atskirus taimerus, kurių kitas būtų atsakingas už duomenų atvaizdavimą ir siuntimą į personalinį kompiuterį, kurį galima aprašyti 200 ms periodu, taip indikatoriuje atvaizduoti vidurkį ir skirtumą kas 10 pertrauktį. Duomenų siuntimas į kompiuterį turi būti įvykdytas kas 200 ms, tai reiškia, jog vidurkintos dviejų kanalų reikšmės turės apytiksliai 80 naujų reikšmių kiekvieną kartą, prieš siunčiant duomenis į personalinį kompiuterį. Indikatoriaus atvaizdavimui taip pat reikalingas vidurkinimo skaičius, todėl jis bus atvaizduojamas nuo paskutinės iki atvaizdavimo pertraukties gautos reikšmės.

1.2. Mikroprocesorius

Šiuo atveju nėra didelio duomenų apdorojimo, dėl kurio procesorius nesugebėtų suspėti visko suskaičiuoti, atvaizdavimui ir duomenų išgavimui. Projekto realizavimui reikalingi bent 2 taimeriai, 1 analogas–kodas keitiklis (ASK) su keliais kanalais, 1 UART sąsaja. Dėl mažesnio mikroprocesoriaus apkrovimo duomenų perdavimui bus naudojamas taimeris, kuris aparatiškai po savo pertraukties nusiųs komandą ASK pradėti nuskaitinėti abiejų kanalų įtampą. Po sekvencinio nuskaitymo per tiesioginį mainų kanalą duomenys bus siunčiami į atmintį, į buferį, kuriuos jau galės apdirbti procesorius. Darbui galima naudoti STM32F103 mikroprocesorių, kuris turi visas projektui reikalingas periferijas, tačiau dėl patogumo iš pradžių sistema bus kuriama STM32L452RE NUCLEO plokšte, jog būtų paprastesnis programavimas ir prieiga prie personalinio kompiuterio duomenų siuntimui (viduje jau egzistuoja keitiklis, todėl užtenka duomenis perduoti per UART2 išvadus ir peržiūrėti per terminalą asmeniniame kompiuteryje).

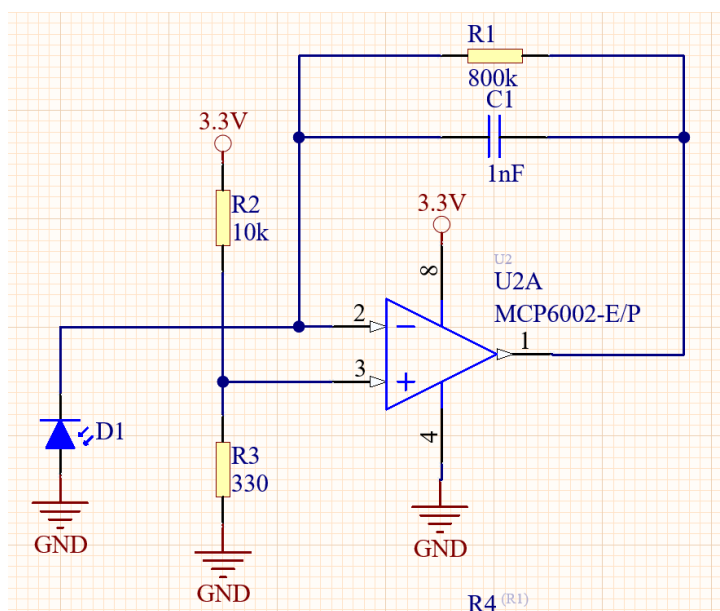
1.3. Indikatorius

Matavimams ir jų atvaizdavimams nėra reikalingas grafinis indikatorius. Kadangi nėra galimybės atlikti šiems matavimams kalibravimo funkcijos, nėra būtinybės turėti mygtukų matricos bei grafinės sąsajos. Maža duomenų atvaizdavimo sparta (200 ms) yra pakankama LCD indikatoriui realizuoti.

Kadangi yra numatyta matuoti iki 10000 liuksų, tai naudojant dvi linijas kiekvienam kanalui atskirai yra pakankamai ilgio (numatyta atvaizduoti tokiu pavidalu „CHx=XX lx“ iš viso bent 9 simboliai) ir atvaizdavimo perstūmimai nereikalingi. Pasirenkamas naudoti 16x2 LCD indikatorius. Kanalų skirtumas gali būti kaip atvaizdavimo funkcija, kuri yra įjungiamo nuspaudus mygtuką arba slenkant indikatoriaus dinaminę atmintį. Yra dar viena galimybė, tai 100 ms skirti atvaizdavimui kanalų, o kita pusė – kanalų skirtumui. LCD apšvietimas yra nuo 5V, tačiau tai problemų nekelia, nes maitinimas yra tiekiamas iš personalinio kompiuterio. Duomenų linijos yra pritaikytos naudoti 3.3 V ir 5 V siuntimui. Naudojamas duomenų perdavimo protokolas yra lygiagrečiai 4 bitų sąsaja. Priverstinis bus vėlinimas bus panaudojant vieną iš taimerio skaitiklių, kadangi panaudojant blokuojančią „HAL_Delay“ funkciją yra per ilgas laiko tarpas, dėl kurios galima pastebėti LCD mirgėjimą, duomenų įrašymo slinkimą.

1.4. Apšvietumo jutiklis

Yra reikalingas apšvietumo skaičiavimas, todėl reikia naudoti šviesos priklausomybę turinčius komponentus. Rinkoje tai gali būti fotorezistoriai („LDR“), fotodiodai ir fototranzistoriai. Taip pat rinkoje yra pilnai apdirbtų produktų, modulių, kurie jau yra kalibruoti ir atvaizduoja apšvietumo signalą keičiant išėjimo dažnį arba išsiuntimas vyksta per protokolą. Jų principas yra toks pat kaip ir šiuo atveju naudojant fotodiodus, tačiau norint išgauti kuo tikslensį spektro jautrumą, naudojami lygiagrečiai keli ar keliasdešimt fotodiodų. Lėtas fotorezistoriaus reakcijos laikas ir nestatūs frontai gali būti nepakankami atvaizduoti 100 Hz kintančio signalo, dėl ko bus naudojami fotodiodai. Jų apšvietumo priklausomybė nuo atvirkštinės srovės yra tiesinė logaritminėje skalėje, todėl patogiu matuoti. Kadangi iš jų yra gaunama tik srovė, ją reikia konvertuoti į įtampą, kurią mikroprocesorius gali apdoroti ASK (analoginis-skaitmeninis keitiklis). Tai yra padaroma transimpedansiniu stiprintuvu.



1 pav. Transimpedasinis stiprintuvas apšvietumui išgauti

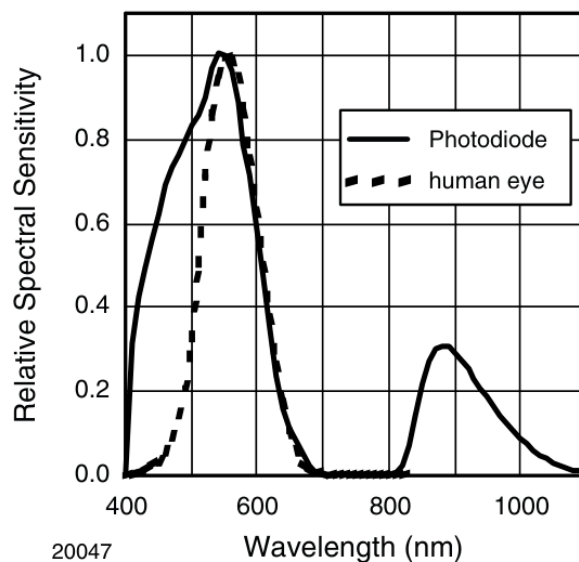
Fotodiodas veikia atvirkštinės srovės režime, kurio srovė priklauso nuo apšvietumo. R_1 rezistorius nustato tekančios srovės stiprinimą. Fotodiodai yra linkę užsigeneruoti, todėl naudojamas kondensatorius C_1 , kuris veikia kaip žemų dažnių filtras ir apriboja aukštus dažnius pagal formulę[1]:

$$f_p = \frac{1}{2\pi R_1 C_1} \approx 194 \text{ Hz}$$

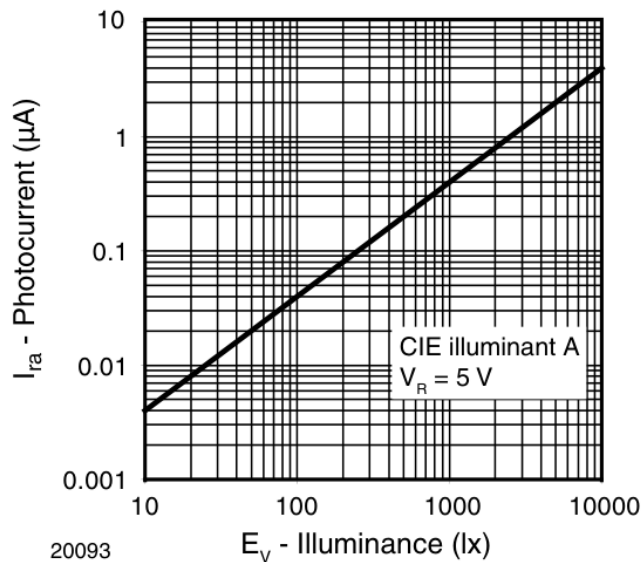
Šis taip pat veikia kaip ir antisanklodinis filtras, kadangi signalai po to bus perduodami į diskretizuojantį keitiklį. Kadangi maitinimas yra vienpolis, reikalingas operacinio stiprintuvo virtualaus nulio pakėlimas (pasirinkta 100 mV), kad nebūtų stiprintuvo išotinimas. Šiuo atveju pasirinktas operacinis stiprintuvas turi įėjimo įtampų ruožą, kuris yra 300 mV žemiau už bendrą išvadą, todėl norint gauti didesnio tikslumo apšvietos reikšmes prie mažo apšvietimo, galima neinvertuojantį įėjimą prijungti ir prie žemės. Stiprinimo varža yra apskaičiuojama pagal formulę:

$$R_{gain} = \frac{V_{maksimali} - V_{minimali}}{I_{inmax}} = \frac{3.3 - 0.1}{4\mu A} \approx 800k$$

Naudojamas PIN fotodiodas yra SMD tipo ir jo spektro maksimumas yra artimas 555 nm dažniui. Lyginant su žmogaus spektriniu jautrumu, kainos ir panašumo atžvilgiu rinkoje, šis buvo tinkamiausias, tačiau pastebime, jog iš pateikto jautrumo, fotodiodas taip pat sugeria ir dalį infraraudonojo spektro, bei visiškai nesugeria raudonos spalvos, todėl norint gauti tikslesnę apšvietą, lyginamą su žmogaus jautrumu, reiktų turėti filtrą, kuris sugeria visus infraraudonuosius spindulius.



Kadangi fakultete nėra kalibravimo galimybių, srovės tekėjimo duomenys yra paimami iš duomenų lapo, nors pats srovės statusas gali būti ir kitoks.



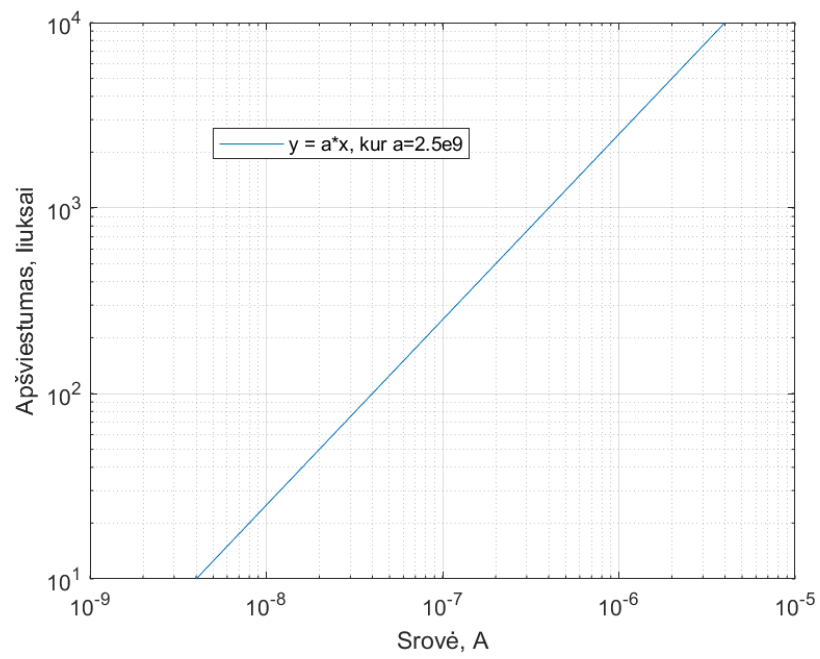
2 pav. Naudojamo jutiklio atvirkštinės srovės priklausomybė nuo apšviestumo

Iš čia yra paimama maksimali srovė $4 \mu\text{A}$. Tai reiškia, jog mūsų matuojamas diapazonas bus iki 10 kiloliuksų (lx arba lux matavimo vienetai). Naudojamas operacinis stiprintuvas – MCP6002, kuris turi rail-to-rail įėjimus ir išėjimą, taip pat svarbu tai, jog yra pagamintas KMOP technologijos. Pasirinktas toks stiprintuvas, kadangi reikalingas mažas srovės tekėjimas į įėjimus, jog nebūtų užgožiamos fotodiodo srovės. Šio stiprintuvo srovė – 1 pA kambario temperatūroje ir to užtenka, kadangi fotodiodo atbulinė srovė tamsioms sąlygomis yra 0.1 nA , 100 kartų mažesnė nei įtekama. Stiprintuvo maitinimas bus tiekiamas iš mikroprocesoriaus bendros paskirties išvadų, kadangi srovės yra mažos ir nesiekia 15 mA , kas yra nurodyta mikroprocesoriaus duomenų lape prie maksimalių ištekamų srovių. Dėl šio jungimo galima prijungti papildomą kanalą maitinimo įtampos nuskaitymui, taip gaunant tikslesnius matavimo rezultatus.

Kadangi naudojamas mikroprocesorius, išsivedama formulė, kuri yra skaičiuojama iš **2 pav.** Ši formulė bus naudojama po įtampos konvertavimo į srovę, tekančią per diodą, nes tada turėsime vieną iš ašių reikšmių. Pastebime, kad logaritminėje skalėje ši formulė yra tiesė, vadinasi ji gali būti aproksimuojama nuolatine dedamąja ir skalės statumu. Statumas yra skaičiuojamas pagal formulę:

$$a = \frac{\log(I_2) - \log(I_1)}{\log(E_2) - \log(E_1)} = \frac{3}{3} = 1$$

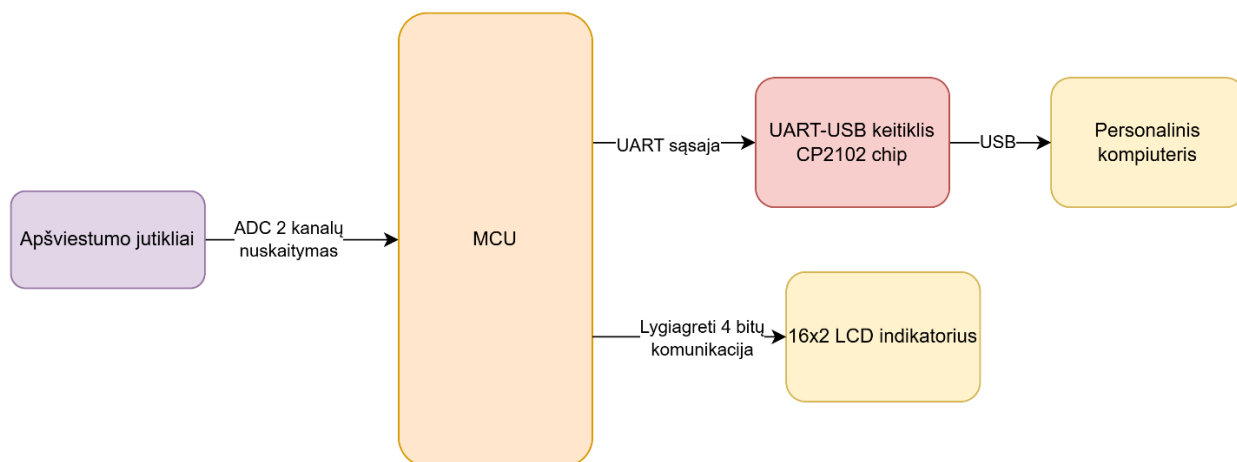
Gaunamas rodiklio pagrindas – 1, tai reiškia, jog ši formulė gali būti aproksimuojama tiesės pavidalu. Šiuo atveju užtenka dviejų lygčių iš kurių pagal tiesinės lygties formulę yra išgaunami koeficientai pateikti žemiau.



3 pav. Aprašyta jutiklio atvirkštinės srovės priklausomybė nuo apšviestumo

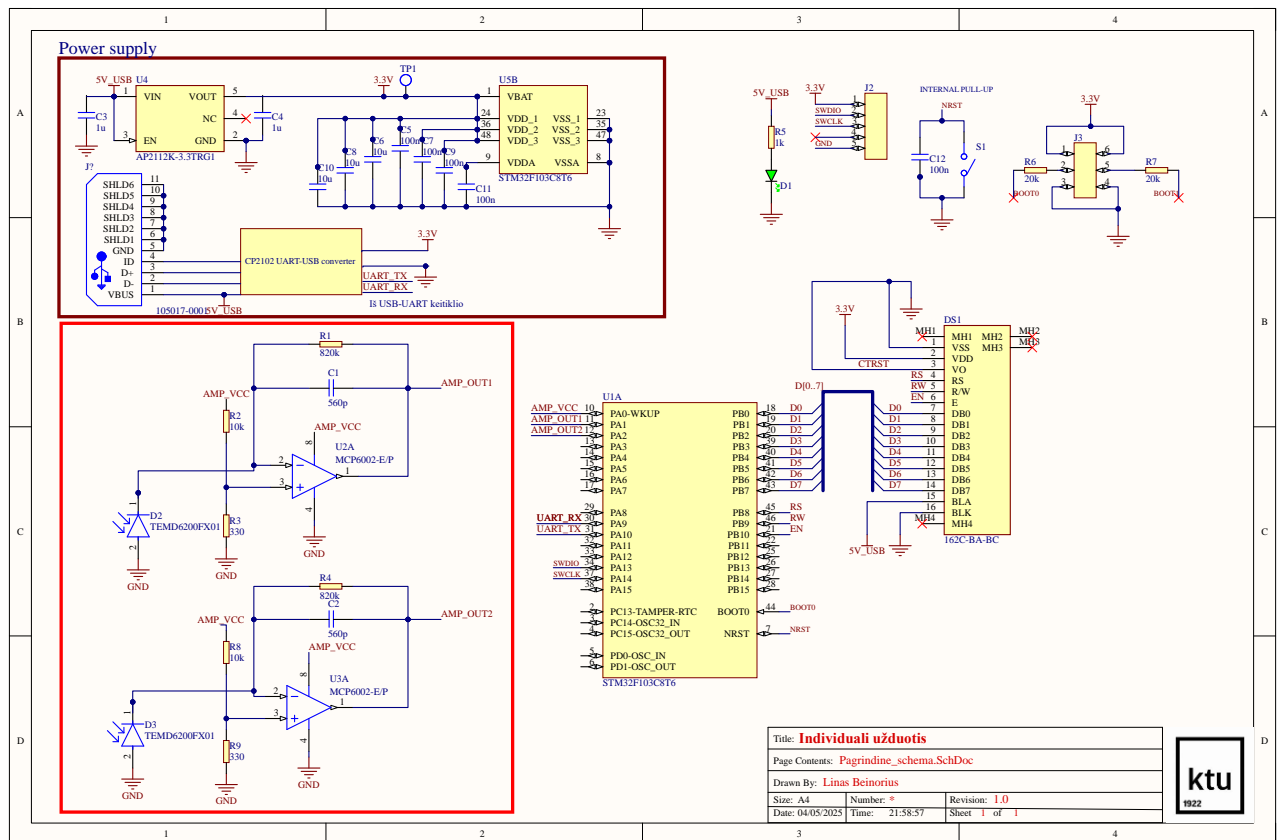
2. Principinė schema ir BOM

Žemiau pateikiama sistemos blokinė diagrama.



4 pav. Įterptinės sistemos blokinė diagrama

Sistemos blokinė diagrama yra nuosekli ir paprasta. Apšviestumo matavimui naudojami 2 kanalai, kuriems reikalingas nuskaitymas atliekamas viduje mikroprocesoriaus esančiu keitikliu. Nuskaityti duomenys yra apdorojami mikroprocesoriuje ir kas 2 sekundes yra atvaizduojama į LCD indikatorių atskiro kanalo apšviestumas, nuspaudus mygtuką vietoj dviejų kanalų reikšmių yra nustatoma skirtuminė jų reikšmė. Kas 200 ms duomenys yra perduodami per UART sąsają į išorinį keitiklį, kuris apdoroja duomenis taip, jog būtų išsiunčiami į personalinį kompiuterį per USB sąsają, pasinaudojant dedikuotu pagal integrinę grandyną draiveriu.

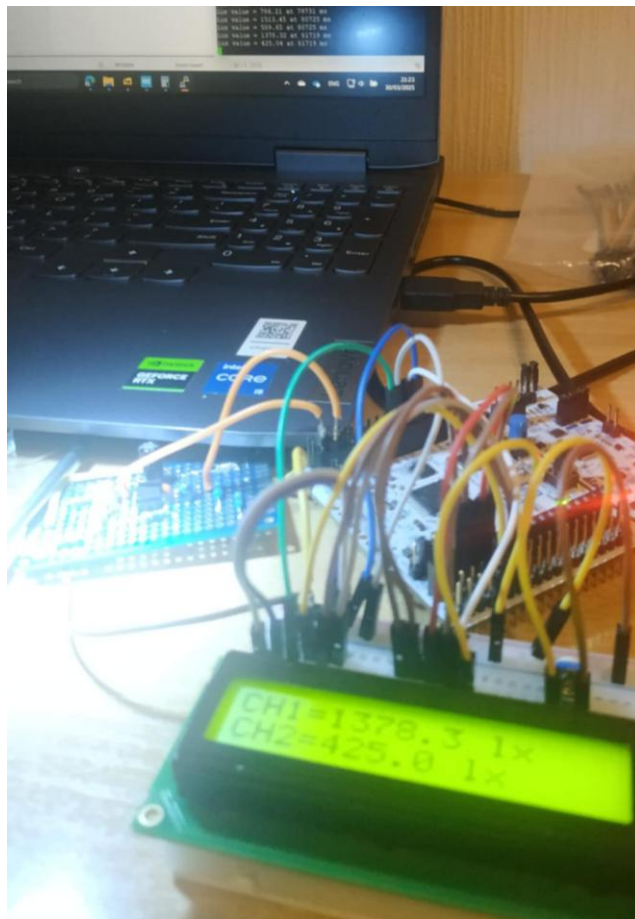


5 pav. Principinė projekto schema

Qty	MPN	Manufacturer	Part Description	Distributor	Unit Price	Line Total
2	C0805C561J5GACTU	KEMET	560pF ±5% 50V COG 0805 Multilayer Ceramic Capacitors MLCC - SMD/SMT RoHS	Mouser	0.114	0.228
1	AP2112K-3.3TRG1	Diodes Inc.	AP2112 Series 0.6 A 3.3 V Fixed Output SMT LDO Linear Regulator - SOT23-5	Mouser	0.504	0.504
2	STM32F103C8T6	STMicroelectronics	ARM Cortex-M3 STM32 F1 Microcontroller 32-Bit 72MHz 64kB FLASH LQFP48 STM32F103C8T6	Farnell	5.25	10.5
5	C0402C104K8PACTU	KEMET	Capacitor; Ceramic; Cap: .100uF; Tol: 10%; SMT; Vol-Rtg 10V; X5R; Tape and Reel KEMET C0402C104K8PACTU	DigiKey	0.09	0.45
2	CC0805KKX7R68B105	Yageo	Ceramic Capacitor, Ceramic, 10V, 10% +Tol, 10% -Tol, X7R, 15% TC, 1uF, Surface Mount, 0805	Distrelec	0.02	0.04
1	68000-106HLF	Amphenol Communications Solutions	Conn Unshrouded Header HDR 6 POS 2.54mm Solder ST Top Entry Thru-Hole BergStik® Bag	DigiKey	0.57	0.57
1	68000-106HLF	Amphenol Communications Solutions	Conn Unshrouded Header HDR 6 POS 2.54mm Solder ST Top Entry Thru-Hole BergStik® Bag	DigiKey	0.57	0.57
1	68000-106HLF	Amphenol Communications Solutions	Conn Unshrouded Header HDR 6 POS 2.54mm Solder ST Top Entry Thru-Hole BergStik® Bag	DigiKey	0.57	0.57
2	CRCW0805820K0FKEA	Vishay	CRCW FIXED RESISTOR 0805 (2012Metric) Chip Resistor ±1% ±100ppm/°C 820kOhm 0.125W 1/8W	Distrelec	0.01	0.02
2	MBB02070C1002FCT00	Vishay	Fixed Resistor, Metal Film, 0.4W, 10000ohm, 350V, 1% +/-Tol, 50ppm/Cel, Through Hole Mount	DigiKey	0.09	0.18
2	MCP6002-E/P	Microchip	MCP6002 Series 6 V 1 MHz Through Hole Low-Power Operational Amplifier - PDIP-8	DigiKey	0.46	0.92
3	GRM21BR61E106KA73L	Murata	Multilayer Ceramic Capacitors Mlcc - Smd/Smt 10 Uf 25 Vdc 10% 0805 X5R	Distrelec	0.04	0.12
2	CRCW0805820K0FKEA	Vishay	Res Thick Film 0805 20K Ohm 1% 1/8W ±100ppm/°C Molded SMD Paper T/R	DigiKey	0.09	0.18
3	CFR50J330R	TE Connectivity	Resistor, Carbon Film, Res 330 Ohms, Pwr-Rtg 0.66 W, Tol 5%, Axial TE Connectivity CFR50J330R	DigiKey	0.09	0.27
2	TEMD6200FX01	Vishay	TEMD6200 Series 540 nm 60A° 40 nA (Ira) Ambient Light Sensor 0805	Farnell	0.745	1.49
1	TS02-66-60-BK-260-LCR-D	Same Sky	TS02 Series SPST-NO 0.05A 12VDC Through Hole Tactile Switch	DigiKey	0.09	0.09
1	150080GS75000	Würth Elektronik	WURTH ELEKTRONIK - 150080GS75000 - LED, Green, SMD, 0805, 30 mA, 3.2 V, 520 nm	DigiKey	0.16	0.16
1	162H BA BW	Displaytech	LCD Character Display Modules & Accessories 16x2 Char Display STN Y/G 6 oclock	Mouser	15.15	15.15
						32.00966599

6 pav. BOM

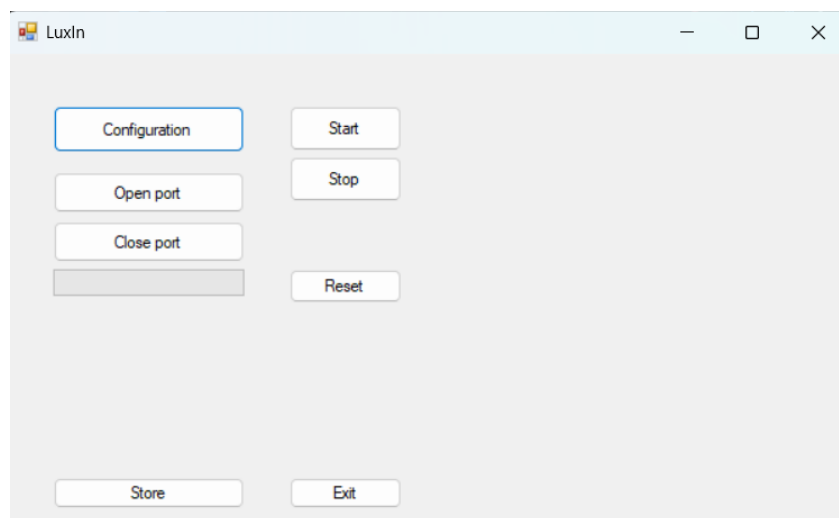
Preliminari šio projekto schema yra 32 eurai, tačiau šiuo atveju nėra įtraukiamas USB–UART išorinis keitiklis, kurį galima įsigyti už kelis eurus. Į kainą taip pat neįtraukiama plokštės gamintojo ir surinkimo kaštai.



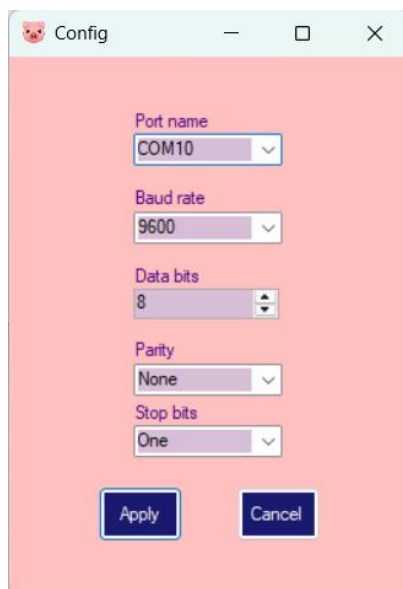
8 pav. Sujungtos schemos nuotrauka

4. Kompiuterio programa

Kompiuteryje pasinaudojame grafinės sąsajos įrankiu „Windows form“ Visual Studio aplinkoje. Duomenų siuntimas ir sustojimas bus inicializuoti kompiuteriu, paspaudus „Start“ ir „Stop“ mygtukus, kuriais bus siunčiamas informacijos paketas. Šis paketas yra nuskaitomas mikroprocesoriumi ir yra nustatoma vėliavėlė, kuri leidžia siųsti apšviestumo duomenis per sąsają. Duomenų atvaizdavimui yra sudaromas grafinis laiko ašyje langas, kuriame, priklausomai nuo to, kas pasirinkta siųsti, yra dviejų kanalų reikšmės arba skirtuminis signalas. Kadangi duomenys yra siunčiami per UART sąsają, reikalinga duomenų konfigūracija, todėl visų pirma reikia pasirinkti, kuriame „COM“ išvade yra prijungtas mikroprocesorius ir jam reikalingi parametrai. Patogumui yra dar sudaromi keli kiti mygtukai, bei minimalus klaidų atvaizdavimas.



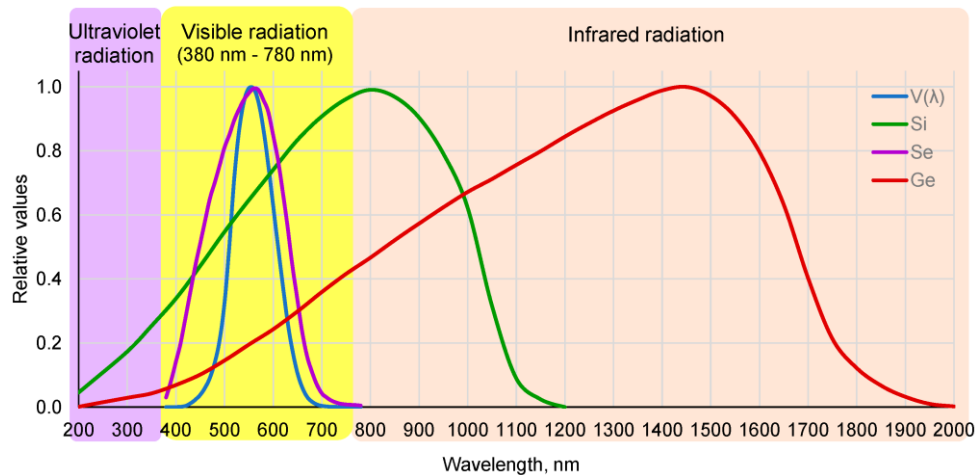
9 pav. Pagrindinis programos langas



10 pav. Parametrų nustatymo langas

5. Paklaidų matavimas

Reikalavimai vidutinėms darbinio apšvietimo vertėms reglamentuojami atitinkamuose standartuose.[2].



11 pav. Detektorių jautrumo kreivės[2]

Kadangi nėra kalibravimo galimybių, sunku nusakyti, kokia yra mūsų paklaida, lyginant su standartizuotu dydžiu. Taip pat nėra galimybių nusakyti ir mūsų aproksimuotos logaritminės tiesės teisingumo, kadangi kiekvienas fotodiodas skiriasi savo statumu, tą galima pastebėti ir matuojant mūsų prietaisu. Dėl šių priežasčių mes galime dalinai nusakyti skaičiavimo metu daromas paklaidas, kurios pasireiškia dėl verčių apvalinimo, skyros.

Mūsų matuojamas dydis yra netiesioginis, kadangi šviesos apšviestumą (šviesos kiekį plote) reikia surinkti, mūsų atveju surinkimas vyksta dviejų kanalų fotodiodais, kurie, priklausomai nuo jos kiekio, leidžia tekėti srovei. Ši srovė yra konvertuojama transimpedansiniu stiprintuvu (pirma apsakoma neapibrėžtis). Konvertuota įtampa yra perduota į mikroprocesoriaus keitiklį, kuris turi kvantuotas reikšmes (antra apsakoma neapibrėžtis). Šioji įtampos vertė atgal konvertuoja tekančios srovės vertę mikroprocesoriaus pagalba ir pagal duomenų lape nurodytą funkciją išskaičiuoja apšviestumą. Standartinė suminė netiesiogiai matuojamo dydžio neapibrėžtis, kai visi įėjimo dydžiai nesusiję (nekoreliuoti) yra skaičiuojama pagal formulę[3]:

$$u_q = \sqrt{\left(\frac{\delta L}{\delta V_{ADC}} u_{V_{ADC}}\right)^2 + \left(\frac{\delta L}{\delta R_{ref}} u_{R_{ref}}\right)^2 + \left(\frac{\delta L}{\delta S} u_S\right)^2}$$

, kur dalinės išvestinės yra įtakos koeficientai, o u dydžiai – įėjimų standartinės neapibrėžtys.

Įtampa skaičiuojama pagal formulę:

$$U_F = R_{ref} \cdot I_F$$

, kur R_{ref} – varža, o I_F – atvirkštinė fotodiodo srovė.

Tada apšviestumas išskaičiuojamas pagal formulę:

$$L = k \cdot I_F \rightarrow k \cdot \frac{V_{ADC}}{R_{ref}} \rightarrow \frac{1}{S} \frac{V_{ADC}}{R_{ref}}$$

k yra koeficientas, kuris išvedamas iš jautrumo koeficiento S (A/lux). Dėl šios priežasties gauname tokią standartinę suminę neapibrėžtį:

$$u_q = \sqrt{\left(\frac{\delta L}{\delta V_{ADC}} u_{V_{ADC}}\right)^2 + \left(\frac{\delta L}{\delta R_{ref}} u_{R_{ref}}\right)^2 + \left(\frac{\delta L}{\delta S} u_S\right)^2}$$

Iš čia susirandame standartinės neapibrėžties žinomų verčių:

$u_{V_{ADC}}$ yra vieno žingsnio (kurio skirstinys yra stačiakampis) standartinis nuokrypis:

$$u_{ADC} = \frac{\Delta}{\sqrt{12}} \rightarrow \frac{3.3V}{4095 \cdot \sqrt{12}} \approx 0.232 \text{ mV}$$

$u_{R_{ref}}$ yra varžos, kurią naudojame kaip atraminę įtampos gavimui, absoliutinė neapibrėžtis. Kadangi tariame, jog tai yra stačiakampis skirstinys, tai gauname vertę:

$$u_{R_{ref}} = 820k\Omega \cdot 0.05 = 41000 \text{ k}\Omega$$

u_S yra fotodiodo gamyklinis išsibarstymas, kuris nėra nurodytas, todėl tariame, jog tai yra 15% nuo mūsų gauto jautrumo koeficiento S.

$$\frac{\delta L}{\delta V_{ADC}} = \frac{1}{S \cdot R_{ref}} = 3.05 \cdot 10^3$$

$$\frac{\delta L}{\delta R_{ref}} = -\frac{1}{S} \frac{V_{ADC}}{R_{ref}^2} = -3.72 \cdot 10^{-3} \cdot V_{ADC}$$

$$\frac{\delta L}{\delta S} = -\frac{1}{S^2} \frac{V_{ADC}}{R_{ref}} = -7.62 \cdot 10^{12} \cdot V_{ADC}$$

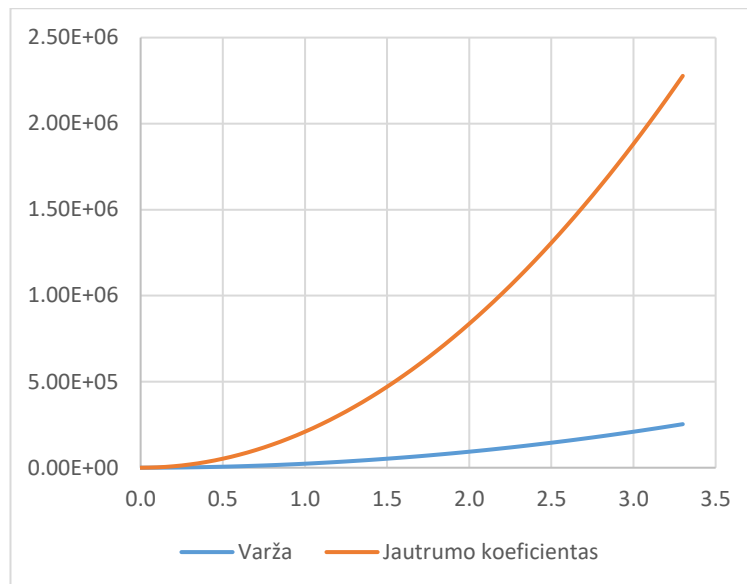
Pastebime, jog vertės turi priklausomybę nuo išmatuojamos įtampos. Iš čia viską įsistatome į suminę neapibrėžtumo formulę ir suskaičiuojame (kai V_{ADC} yra 3.3V):

$$u_q = \sqrt{(3.05 \cdot 10^3 \cdot 0.232 \times 10^{-3})^2 + (-3.72 \cdot 10^{-3} \cdot 3.3 \cdot 41000)^2 + (-7.62 \cdot 10^{12} \cdot 3.3 \cdot 0.15 \cdot 2.5 \times 10^9)^2} \approx 1590 \text{ lux}$$

Tuo tarpu, kai V_{ADC} yra 0:

$$u_q = \sqrt{(3.05 \cdot 10^3 \cdot 0.232 \times 10^{-3})^2 + (-3.72 \cdot 10^{-3} \cdot 0 \cdot 41000)^2 + (-7.62 \cdot 10^{12} \cdot 0 \cdot 0.15 \cdot 2.5 \times 10^9)^2} \approx 0.7 \text{ lux}$$

Matome, jog prie mažų įtampų mūsų suminė paklaida yra nedidelė, tačiau didėja, didėjant apšvietai. ADC keitiklio paklaida sudaro nedidelę dalį prie didelės įtampos, tačiau yra labai svarbus, kai apšvieta yra maža.



12 pav. Varžos tolerancijos ir fotodiode jautrumo koeficiento įtakos nuo išmatuojamos įtampos

Pastebime, jog jautrumo koeficiento svertinis narys sudaro didžiąją dalį paklaidos prie didesnių įtampų, todėl, norint gauti geresnę paklaidą, reikalingas tiksli jautrumo koeficiento priklausomybė (duomenų lape nėra nurodytas išsibarstymas, todėl pasirinkta vertė buvo 15%). Taip pat bėda yra triukšmas ir paklaidos prie mažų įtampų, kurios daro įtaką įtampos šuoliams, kurie staigiai gali pakeisti atvaizduojamą vertę. Šią paklaidą bandoma sumažinti vidurkinant.

Symbol	Parameter	Conditions ⁽⁴⁾		Min	Typ	Max	Unit
ET	Total unadjusted error	Single ended	Fast channel (max speed)	-	4	5	LSB
			Slow channel (max speed)	-	4	5	
		Differential	Fast channel (max speed)	-	3.5	4.5	
			Slow channel (max speed)	-	3.5	4.5	
EO	Offset error	Single ended	Fast channel (max speed)	-	1	2.5	
			Slow channel (max speed)	-	1	2.5	
		Differential	Fast channel (max speed)	-	1.5	2.5	
			Slow channel (max speed)	-	1.5	2.5	
EG	Gain error	Single ended	Fast channel (max speed)	-	2.5	4.5	
			Slow channel (max speed)	-	2.5	4.5	
		Differential	Fast channel (max speed)	-	2.5	3.5	
			Slow channel (max speed)	-	2.5	3.5	
ED	Differential linearity error	Single ended	Fast channel (max speed)	-	1	1.5	
			Slow channel (max speed)	-	1	1.5	
		Differential	Fast channel (max speed)	-	1	1.2	
			Slow channel (max speed)	-	1	1.2	
EL	Integral linearity error	Single ended	Fast channel (max speed)	-	1.5	2.5	
			Slow channel (max speed)	-	1.5	2.5	
		Differential	Fast channel (max speed)	-	1	2	
			Slow channel (max speed)	-	1	2	

13 pav. Mikroprocesoriaus ADC paklaidos

Matome, jog pats mikroprocesorius turi keletos mažiausios skilties bitų paklaidas, kurios nėra tik absoliutinės. Kadangi matuojama iki 10 klux, tai reiškia, jog viena skiltis atitinka apytiksliai 2.5 lux(10000/4095). Pasirenkant diapazoną reikia atsižvelgti ir į tai, kiek triukšminga yra aplinka, kadangi bet kokia indukuota įtampa per laidus ar maitinimo įtampos šuoliai (nėra atraminės įtampos vertės) sudaro pokytį. Pasirenkame taip, jog šie neturėtų santykinai didelio pokyčio liuksmetro

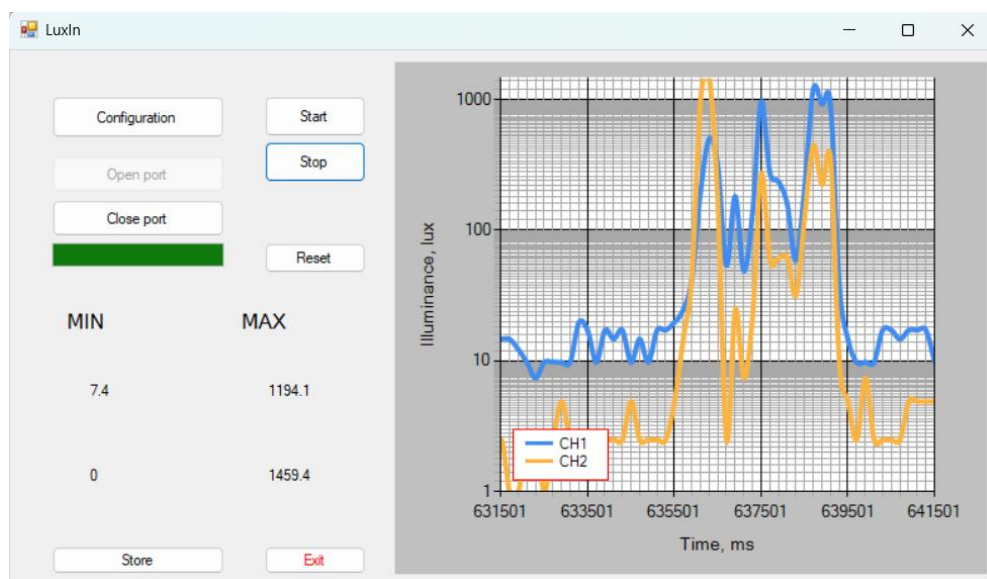
matavimams, todėl 50 lx galėtų būti minimali diapazono vertė. Duomenų atvaizdavime paliekamos vertės vizualizacijai.

6. Galutinis projektas

Pratestuotas su L4 serijos mikrovaldiklių procesas veikė, tačiau buvo reikalingas patvarumas, kadangi laidams pajudinus būdavo nenuspėjamas indikatorius atvaizdavimui ir apšviestumo skaičiavimui. Dėl šios priežasties sudarytas ant projektavimo plokštės įrenginys su F1 serijos mikrovaldikliu, kurio nepastovioji atmintis siekia 64 kB (programa be optimizacijos viršija 40 kB ribą, todėl šiam darbui šis mikroprocesorius yra tinkamesnis kainos atžvilgiu).

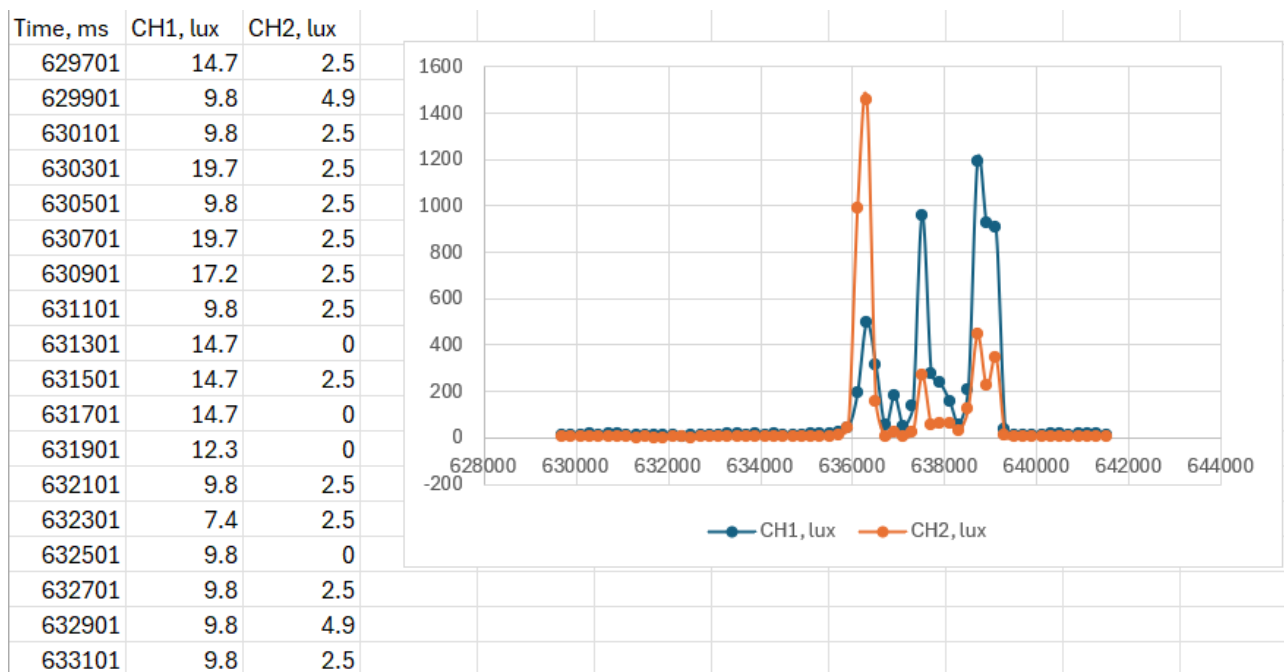


14 pav. Apšviestumo matuoklis



15 pav. Atvaizduojami parametrai kompiuteryje

Kadangi duomenys yra paduodami iš mikroprocesoriaus, tuo pačiu metu siunčiami apšviestumo matavimo ir absoliutinio mikroprocesoriaus laiko, taip žinant tikslų momentą, kuriame įvyksta matavimas. Duomenys yra slankūs, t.y. grafike yra atvaizduojama 10s paskutiniųjų duomenų, kurie gali būti išsaugomi Excel faile žemiau pateiktu formatu.



16 pav. Duomenų saugojimo pavyzdys (grafikas sukurtas rankiniu būdu vizualizacijai)

Įrenginio kodas ir kita informacija yra prieinama per „GitHub“ talpyklą:
https://github.com/xKaLiOx/Iterptines_sistemas

7. Išvados

Pagamintas įrenginys matuoja dviejų kanalų apšviestumo absoliutines vertes, taip pat mygtuko paspaudimu yra perjungiamas į kanalų skirtumo atvaizdavimą. Šis įrenginys gali būti pritaikytas patalpų apšviestumo įvertinimui, dviejų skirtingų terpių ar patalpų apšviestumo skirtumo įvertinimui, siekiant terpes palaikyti vienalytiškoms. Tolesniam tobulinimui yra galimybė sukurti dedikuotą PCB bei korpusą, tiesioginę USB sąsają į kompiuterį. „F1“ serija palaiko USB protokolą, tačiau reiktų didesnės maksimalios srovės įtampos reguliatoriaus ant plokštės, nes tiesiogiai kišant į USB lizdą, reguliatorius perkais, jeigu bus parinktas didelis skystųjų kristalų indikatoriaus kontrastas.

Šaltiniai

1. J. Caldwell, „1 MHz, Single-Supply, Photodiode Amplifier Reference Design“, 2014.
2. P. Tabaka ir J. Wtorkiewicz, „Analysis of the Spectral Sensitivity of Luxmeters and Light Sensors of Smartphones in Terms of Their Influence on the Results of Illuminance Measurements—Example Cases“, *Energies*, t. 15, nr. 16, Art. nr. 16, saus. 2022, doi: 10.3390/en15165847.
3. P. Kaškonas, modulio „Matavimai ir metrologija“ paskaita „Neapibrėžtis“, 2024 m.

Priedas

main.c

```
/* USER CODE BEGIN Includes */
#include <stdio.h>
#include "LCD_16x2_PARALLEL.h"
#include <string.h>
/* USER CODE END Includes */

/* USER CODE BEGIN PD */
#define NUM_OF_CHANNELS 3
#define ADC_BUFFER_SIZE 240
#define UART_BUFFER_SIZE 40
#define LCD_BUFFER_SIZE 32
#define UART_RX_SIZE 16

#define ADC_V_Step 8.05861e-4f

#define LCD_UPDATE_CNT 10

#define GAIN_RESISTANCE 820
#define DIODE_coefficient 2.5e6

#define SHOW_ABSOLUTE 1
#define SHOW_DIFFERENCE 0

volatile int TICK=0;
/* USER CODE END PD */

/* USER CODE BEGIN PV */
volatile uint16_t ADC_Samples[ADC_BUFFER_SIZE] = {0};
//AVERAGES
volatile float ADC_Voltages[NUM_OF_CHANNELS] = {0};
volatile double LUX_Values[NUM_OF_CHANNELS-1] = {0};
volatile double LUX_Difference;
volatile uint32_t SUM[NUM_OF_CHANNELS] = {0};
uint8_t pData_RX[UART_RX_SIZE] = {0};

char UART_data[UART_BUFFER_SIZE];

uint8_t ERROR_FLAG = 0;
volatile uint8_t LCD_SHOW_FLAG = 0;
volatile uint8_t UART_UPDATE_FLAG = 0;
volatile uint8_t ADC_CALC_AVG_FLAG = 0;
volatile uint8_t LCD_SHOW_TYPE = SHOW_ABSOLUTE;
volatile uint8_t LCD_CLEAR_ONCE = 0;
volatile uint8_t UART_SEND_FLAG = 0; // flag from PC to start UART data transmission

volatile uint8_t LCD_Delay_cnt;

char LCD_top_text[LCD_BUFFER_SIZE]="Starting...";
char LCD_bottom_text[LCD_BUFFER_SIZE]="LuxIn";
/* USER CODE END PV */

/* USER CODE BEGIN PFP */
double Voltage_to_lux(float Vadc);
void DELAY_US(uint16_t TIME_US);
/* USER CODE END PFP */

int main(void)
```

```

{

    /* USER CODE BEGIN 1 */

    /* USER CODE END 1 */

    /* MCU Configuration-----*/

    /* Reset of all peripherals, Initializes the Flash interface and the Systick. */
    HAL_Init();

    /* USER CODE BEGIN Init */

    /* USER CODE END Init */

    /* Configure the system clock */
    SystemClock_Config();

    /* USER CODE BEGIN SysInit */

    /* USER CODE END SysInit */

    /* Initialize all configured peripherals */
    MX_GPIO_Init();
    MX_DMA_Init();
    MX_ADC1_Init();
    MX_TIM2_Init();
    MX_USART2_UART_Init();
    MX_TIM3_Init();
    /* USER CODE BEGIN 2 */
    HAL_Delay(500);

    if(HAL_ADCEx_Calibration_Start(&hadc1) != HAL_OK)
    {
        sprintf(LCD_top_text, "ERR CALIBRATION");
        sprintf(LCD_bottom_text, "RESET MCU");
        ERROR_FLAG = 1;
    }

    if(HAL_ADC_Start_DMA(&hadc1, (uint32_t*)ADC_Samples, ADC_BUFFER_SIZE) != HAL_OK)
    {
        sprintf(LCD_top_text, "ERROR ADC");
        sprintf(LCD_bottom_text, "RESET MCU");
        ERROR_FLAG = 1;
    }

    if(HAL_TIM_Base_Start_IT(&htim2) != HAL_OK)
    {
        sprintf(LCD_top_text, "DISPLAY TIM ERR");
        sprintf(LCD_bottom_text, "RESET MCU");
        ERROR_FLAG = 1;
    }

    if(HAL_TIM_Base_Start(&htim3) != HAL_OK)
    {
        sprintf(LCD_top_text, "ERROR MAIN TIM");
        sprintf(LCD_bottom_text, "RESET MCU");
        ERROR_FLAG = 1;
    }

    if(HAL_UARTEx_ReceiveToIdle_IT(&huart2, pData_RX, UART_RX_SIZE) != HAL_OK)
    {

```

```

    sprintf(LCD_top_text, "ERROR UART");
    sprintf(LCD_bottom_text, "CHECK CABLE");
    ERROR_FLAG = 1;
}
//HAL_GPIO_WritePin(LUX_VCC_GPIO_Port, LUX_VCC_Pin, GPIO_PIN_SET); // turn on LCD DATA
transmission power
LCD_Init();
HAL_Delay(10);
LCD_SEND_COMMAND(LCD_DISPLAY_OFF);
HAL_Delay(5);
LCD_SEND_COMMAND(LCD_Entry_MODE);
HAL_Delay(5);
LCD_SEND_STR(LCD_top_text, 0x3, 0);
LCD_SEND_STR(LCD_bottom_text, 5, 1);
LCD_SEND_COMMAND(LCD_SET_DISPLAY_DEFAULT);
HAL_Delay(2000);
LCD_CLEAR();
LCD_SEND_COMMAND(LCD_RETURN_HOME);
if(ERROR_FLAG) Error_Handler(); //REIKIA PALEISTI TAIMERI TIM6 PRIES ATVAIZDAVIMA, us
delay
/* USER CODE END 2 */

/* Infinite loop */
/* USER CODE BEGIN WHILE */
while (1)
{
    if(ADC_CALC_AVG_FLAG)
    {
        for(uint8_t x = 0; x < ADC_BUFFER_SIZE; x++)
        {
            SUM[x%3] += ADC_Samples[x];
        }

        for(uint8_t x = 0; x < NUM_OF_CHANNELS; x++)
        {
            SUM[x] /= (ADC_BUFFER_SIZE/NUM_OF_CHANNELS);
            ADC_Voltages[x] = SUM[x]*ADC_V_Step;
            LUX_Values[x] = Voltage_to_lux(ADC_Voltages[x]);
        }
        ADC_CALC_AVG_FLAG=0;
        if(!LCD_SHOW_TYPE)
        {
            LUX_Difference = LUX_Values[0]-LUX_Values[1];
        }
    }
    if(UART_UPDATE_FLAG && !ADC_CALC_AVG_FLAG && UART_SEND_FLAG)
    {
        if(LCD_SHOW_TYPE)
        {
            sprintf(UART_data, "CH1 value = %1.1f at %d
ms\r\n", LUX_Values[0], (int)HAL_GetTick());
            HAL_UART_Transmit(&huart2, (uint8_t*)UART_data, strlen(UART_data), 100);
            sprintf(UART_data, "CH2 value = %1.1f at %d
ms\r\n", LUX_Values[1], (int)HAL_GetTick());
            HAL_UART_Transmit(&huart2, (uint8_t*)UART_data, strlen(UART_data), 100);
            UART_UPDATE_FLAG = 0;
        }
        else
        {
            sprintf(UART_data, "CH1-CH2 value = %1.1f at %d
ms\r\n", LUX_Difference, (int)HAL_GetTick());

```



```

        HAL_UART_Transmit(&huart2, (uint8_t*)UART_data, strlen(UART_data), 100);
        UART_UPDATE_FLAG = 0;
    }
}

if(LCD_SHOW_FLAG && !ADC_CALC_AVG_FLAG)
{
    HAL_GPIO_TogglePin(USER_LED_GPIO_Port, USER_LED_Pin);
    if(LCD_SHOW_TYPE)
    {
        sprintf((char*)LCD_top_text, "CH1=%1.1f lx", LUX_Values[0]);
        LCD_SEND_COMMAND(LCD_RETURN_HOME);
        LCD_SEND_STR((char*)LCD_top_text, 0, 0);
        sprintf((char*)LCD_bottom_text, "CH2=%1.1f lx", LUX_Values[1]);
        LCD_SEND_STR((char*)LCD_bottom_text, 0, 1);
        LCD_SHOW_FLAG = 0;
    }
    else
    {
        if(LCD_CLEAR_ONCE)
        {
            LCD_CLEAR_DRAM();
            LCD_CLEAR_ONCE = 0;
            sprintf((char*)LCD_top_text, "DIFFERENCE");
            LCD_SEND_STR((char*)LCD_top_text, 3, 0);
            sprintf((char*)LCD_bottom_text, "%1.1f lx", LUX_Difference);
            LCD_SEND_STR((char*)LCD_bottom_text, ((16-
(int)strlen(LCD_bottom_text))/2), 1);
            LCD_SHOW_FLAG = 0;
        }
        else
        {
            sprintf((char*)LCD_bottom_text, "%1.1f lx", LUX_Difference);
            LCD_SEND_STR((char*)LCD_bottom_text, ((16-
(int)strlen(LCD_bottom_text))/2), 1);
            LCD_SHOW_FLAG = 0;
        }
    }
}

/* USER CODE END WHILE */

/* USER CODE BEGIN 3 */
}
/* USER CODE END 3 */
}

/* USER CODE BEGIN 4 */
double Voltage_to_lux(float Vadc)
{
    // Vdda maitinimas operacinio Vadc nuskaityta isejimo reiksme
    volatile double Current=0;
    static double LUX=0;

    //GAIN RES 820 k DIODE_COEF 2.5 e9, galima rodiklius suprastinti per 3
    Current = Vadc/GAIN_RESISTANCE;
    LUX = (double)(DIODE_coefficient*Current);
    return LUX;
}

void DELAY_US(uint16_t TIME_US)
{
    uint32_t old_timer_value = TIM3->CNT;

```

```

    uint32_t target_time = (old_timer_value + TIME_US) % (TIM3->ARR + 1);

    if (target_time < old_timer_value) // overflow
    {
        while (TIM3->CNT >= old_timer_value);
    }

    while (TIM3->CNT < target_time);
}
void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim)
{
    if(htim == &htim2)
    {
        UART_UPDATE_FLAG = 1;
        ADC_CALC_AVG_FLAG = 1;
        if(LCD_Delay_cnt<LCD_UPDATE_CNT) LCD_Delay_cnt++;
        else
        {
            LCD_Delay_cnt %=LCD_UPDATE_CNT;
            LCD_SHOW_FLAG = 1;
        }
    }
}
void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)
{
    if(GPIO_Pin == Button_SET_Pin)
    {
        LCD_SHOW_FLAG = 1;
        ADC_CALC_AVG_FLAG = 1;
        UART_UPDATE_FLAG = 1;
        LCD_CLEAR_ONCE = 1;
        if(LCD_SHOW_TYPE) LCD_SHOW_TYPE = SHOW_DIFFERENCE;
        else LCD_SHOW_TYPE = SHOW_ABSOLUTE;
    }
    for(uint8_t x = 0;x<UART_BUFFER_SIZE;x++)
    {
        UART_data[x] = '\0';
    }
    for(uint8_t x = 0;x<LCD_BUFFER_SIZE;x++)
    {
        LCD_top_text[x] = ' ';
        LCD_bottom_text[x] = ' ';
    }
}
void HAL_UARTEx_RxEventCallback(UART_HandleTypeDef *huart, uint16_t Size)
{
    char Start_string[] = "Start\r\n";
    char Stop_string[] = "Stop\r\n";
    if(huart == &huart2)
    {
        if(Size == strlen(Start_string) && strncmp(Start_string,(char*)pData_RX,Size)==0)
        {
            UART_SEND_FLAG = 1;
        }
        else if(Size == strlen(Stop_string) &&
        strncmp(Stop_string,(char*)pData_RX,Size)==0)
        {
            UART_SEND_FLAG = 0;
        }
    }
}

```

```
    HAL_UARTEx_ReceiveToIdle_IT(&huart2, pData_RX, UART_RX_SIZE); // See more Start and  
stop flags  
}  
/* USER CODE END 4 */
```