

Львівський національний університет імені Івана Франка

Фізичний факультет

Кафедра експериментальної фізики

Магістерська робота

на тему:

«Система управління монохроматором

на базі платформи Arduino»

Виконав студент II курсу групи Фзф-62
спеціальності «Фізика та астрономія»

Ткачук О. С.

Керівник: проф. Вістовський В. В.

Рецензент: доц. Бовгира О.В

м. Львів – 2022 р.

Вступ

Сучасні електронні пристрої, які використовуються у різних галузях науки і техніки, здебільшого містять інтегральні схеми, що підвищує надійність приладу та знижує його вартість порівняно з використанням дискретних напівпровідникових елементів. Окрім цього зменшення розмірів і ваги пристрою розширює межі його практичного застосування. Різноманітність інтегральних схем на сьогодні випускаються такими компаніями як Microchip, On Semiconductor, Texas Instrument, Analog Device, Linear Technology та багато інші, які є лідерами на світовому ринку. Одними з інтегральних мікросхем, які здатні виконувати різноманітні функції є мікроконтролери. Сьогодні стрімко розвиваються інформаційні і комп'ютерні технології, що дозволяє створювати автоматизовані технологічні лінії, якими можна керувати з допомогою комп'ютера. Одним з вузлів, що забезпечує взаємозв'язок комп'ютера з периферійним пристроєм є мікроконтролер. Для використання мікроконтролера у вузлах електроніки необхідно написати програмний код для реалізації тієї чи іншої задачі. Програмне налаштування параметрів мікроконтролера дозволяє керувати функцією відповідного пристрою без зміни параметрів електричної схеми, що є важливим фактором надійності пристрою в цілому. Одним з можливих застосувань мікроконтролерів є автоматизація фізичного експерименту, що дозволяє проводити вимірювання фізичних величин і передавати дані на комп'ютер.

Зміст

Розділ 1: Принцип роботи та оптичні схеми градкових монохроматорів	4
1.1 Монохроматор Bentham ТМс150.....	4
1.2 Монохроматор МДР-206	5
1.3 SpectraPro-150	6
Розділ 2: Функціональні можливості сучасних мікропроцесорних платформ.....	8
2.1 Системи на мікропроцесорі Atmega	8
2.1.1 Arduino UNO R3	8
2.1.2 Arduino NANO	12
2.1.3 ARDUINO MEGA	15
2.2 Системи на мікропроцесорі Broadcom BCM2711	18
2.2.1 Raspberry Pi 4.....	18
2.3 Огляд драйверів для біполярних крокових двигунів	19
2.3.1 L6219	19
2.3.2 DRV8825	21
2.3.3 TMC2209	24
2.3.4 A4988.....	26
Розділ 3: Розробка системи управління монохроматора SpectraPro-150.....	28
Використані джерела	40

Розділ 1: Принцип роботи та оптичні схеми градкових монохроматорів

1.1 Монохроматор Bentham ТМс150



Рис.1 Зовнішній вигляд Bentham ТМс150

Займаючи не великі розміри монохроматор ТМс150 має двохпозиційну турель з однією або двома дифракційними градками, три типи щілин на вибір, та моторизований механізм зміни 6-ти оптичних фільтрів розташованих усередині приладу на входному порті. Керувати ним можна через USB порт.

Характеристики

Таблиця 1

Оптична система

Конфігурація	Симетрична система Черні-Тернер
Фокусна відстань	150мм
Дифракційний механізм	Двохпозиційна турель
Кількість дифракційних решіток	1 або 2 шт.
Розмір решіток	30 x 30 мм
Відносний опір(апертура)	f/4.2
Кількість входних/вихідних портів	1 вхід, 1 вихід

Таблиця 2

Оптомеханічна система

Роздільна здатність механічного приладу	0,000072°/крок
Тип щілини	FSS – фіксована VSS – регульована мікрометричним гвинтом MVSS - моторизована
Розмір щілини	ширина 10мкм – 10мм, висота 20мм
Порядок сортування	6 – позиційний диск для оптичних фільтрів включаючи затвор
Діаметр фільтру	25мм

Оптична продуктивність

Спектральний діапазон	В межах 200нм – 50 мкм
Лінійна дисперсія	5,4 нм/мм
Точність встановлення довжини хвилі	± 0.3 нм
Повторюваність довжини хвилі	$\pm 0,01$ нм
Повна роздільна здатність/зменшена висота щілини	0,3 / 0,2 нм
Відторгнення розсіяного світла при 2,5 FWHM	10^{-5}

1.2 Монохроматор МДР-206



Рис.2 Зовнішній вигляд МДР-206

Монохроматор МДР-206 призначений для використання як джерело монохроматичного випромінювання в спектральному діапазоні від 190 до 5000 нм. Монохроматори можуть оснащуватися додатковими пристроями, що розширюють можливості у вирішенні науково-дослідних завдань. У тому числі: джерелами світла, фотоприймачами, туреллю автоматичного перемикавання фільтрів, узгодженим дзеркальним конденсором тощо. Вбудований в монохроматор мікропроцесорний контролер забезпечує автоматичну установку будь-якої заданої довжини хвилі в робочому спектральному діапазоні.

Характеристики

Таблиця 4

Тип монохроматора	Схема Еберта-Фасті
Розмір щілини	1/4
Спектральний діапазон	190-5000нм
1200шт/мм	190-1250
600шт/мм	380-2500
300шт/мм	760=5000

1.3 SpectraPro-150



Рис.3 Зовнішній вигляд SpectraPro-150

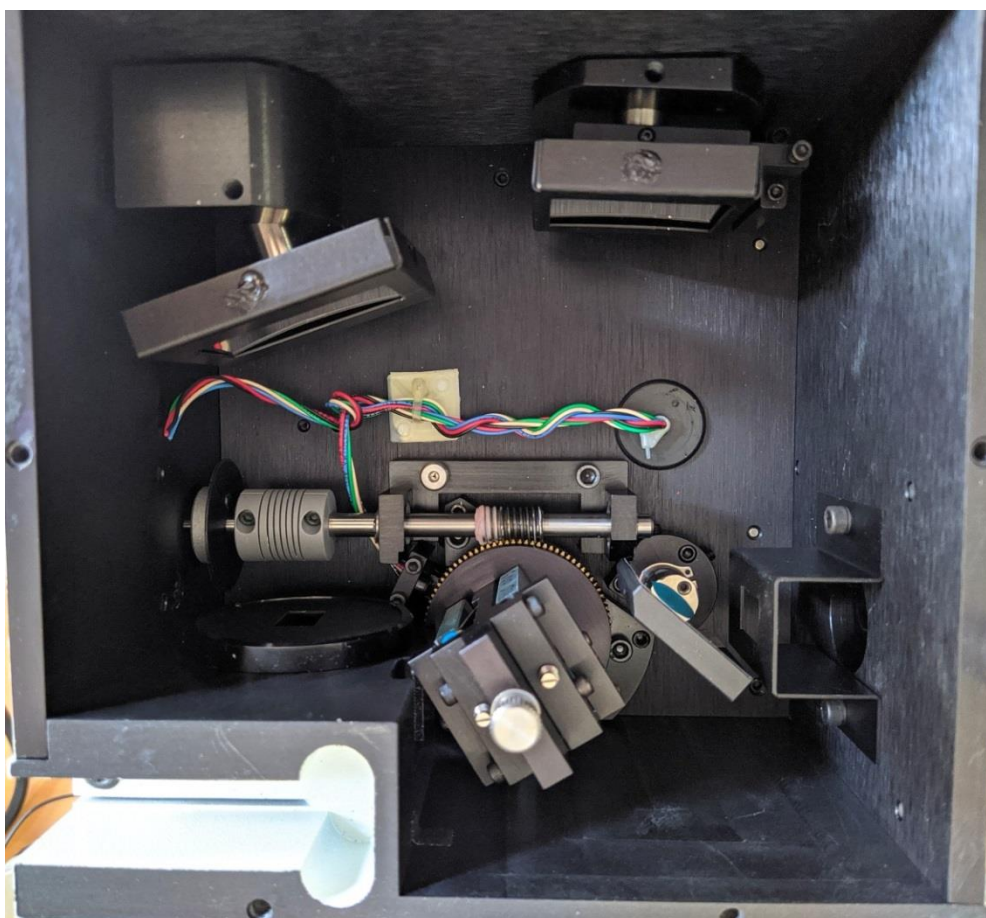


Рис.4 Вигляд зверху

Характеристики

Таблиця 5

Фокусна відстань	150мм
Апертура	f/4
Оптична система	Черні-Тернера
Спектральний діапазон	0 – 1400нм
Роздільна здатність	0,4нм
Дисперсія	5 нм/мм
Точність	±0,25нм
Повторюваність	±0,05нм
Крок крокового двигуна	0,005нм
Розмір фокальної площини	25мм ширина x 10мм висота
Розмір градки	32x32 мм
Робочі діапазони	0-1400 нм для градки 1200 шт/мм. 0-2800 нм для градки 600 шт/мм.

Монохроматор SpectraPro-150 двох градковий монохроматор який має компактний розмір. Маючи можливість підключення до комп'ютера можна керувати його градками посилаючи сигнал на кроковий двигун. Сам прилад працює за оптичною системою Черні-Тернера, яка розкладає вхідне світло на декілька порядків. Присутня можливість в ручну керувати розмірами вхідної та вихідної щілини

Розділ 2: Функціональні можливості сучасних мікропроцесорних платформ

2.1 Системи на мікропроцесорі Atmega

2.1.1 Arduino UNO R3

Зовнішній вигляд та характеристики



Рис.5. Зовнішній вигляд Arduino UNO R3

Контролер UNO R3 побудований на мікропроцесорі ATmega328, які дає нам такі можливості для роботи з ним:

- 14 цифрових входів виводів (6 з яких підтримують режим ШИМ модуляції).
- 6 аналогових входів.
- Частоту 12 Мгц.
- USB порт.
- Роз'єм для програмування.
- Кнопку скидання.

Характеристики

Мікроконтролер	АТmega328
Робоча напруга	5V

Напруга живлення (граничне)	6-20V
Цифрові входи / виходи	14 (з них 6 можуть використовуватися в якості ШІМ-виходів)
Аналогові входи	6
Максимальний струм одного виведення	40мА
Максимальний вихідний струм виводу 3.3V	50мА
Flash-пам'ять	32 КБ(АТmega328) з яких 0.5 КБ використовується завантажувачем
SRAM	2 КБ (АТmega328)
EEPROM	1 КБ (АТmega328)
Тактова частота	16МГц

Додаткові можливості

- **Послідовний інтерфейс:** виходи 0 (Rx) та 1 (Tx) використовуються для прийому і передачі послідовних даних логічного рівня TTL. Ці виводи підключені до виходів передачі даних мікросхеми АТmega328, використовуючи USB-UART.
- **Зовнішні переривання:** виходи 2 і 3, можуть бути використані як входи зовнішніх переривань, Програмно можуть бути установлені на переривання на низькому рівні.
- **ШИМ виходи:** 3, 5, 6, 9, 10, 11 можуть працювати в режимі з розширенням 8 розрядів.
- **Світлодіод:** 13 ніжка, світиться при високому рівні сигналу на виході.

- **AREF:** Опорна напруга АЦП мікроконтролера.
- **RESET:** Низький рівень на цьому виході викликає скидання мікроконтролера.

Схема Arduino UNO R3

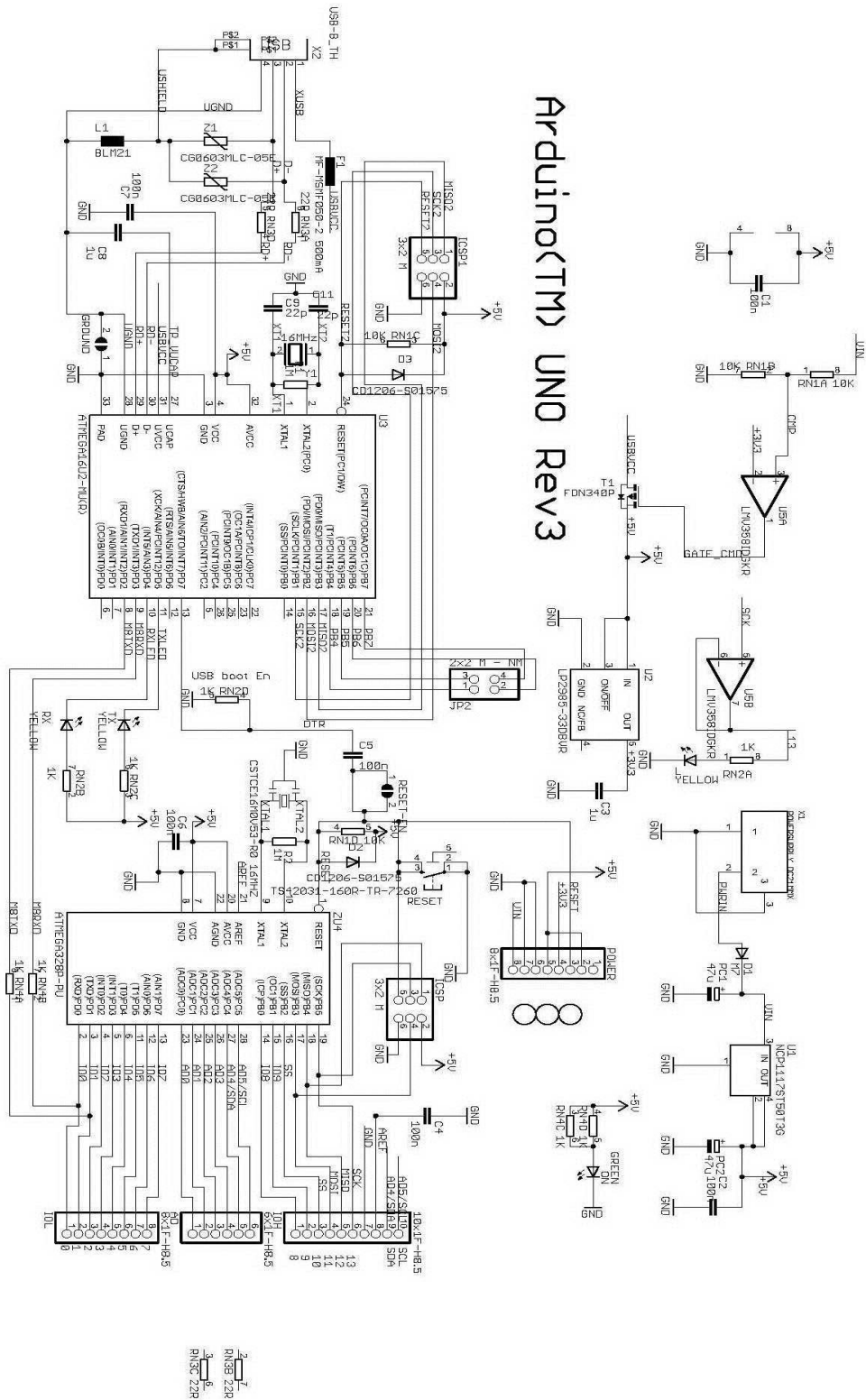


Рис.6. Схематичне зображення UNO R3

2.1.2 Arduino NANO

Зовнішній вигляд та характеристики



Рис.8. Зовнішній вигляд Arduino NANO

Таблиця 7

Характеристики

Мікроконтролер	ATmega328, ATmega168
Робоча напруга	5V
Напруга живлення (рекомендована)	7-12V
Напруга живлення (граничне)	6-20V
Цифрові входи / виходи	14 (з них 6 можуть використовуватися в якості ШІМ-виходів)
Аналогові входи	8
Струм живлення	20mA
Максимальний вихідний струм виводу	40mA (но не більше 200mA)
Flash-пам'ять	32 КБ(ATmega328) , 16 КБ (ATmega168)
SRAM	2 КБ (ATmega328), 1 КБ (ATmega168)
EEPROM	1 КБ (ATmega328), 512Б (ATmega168)
Тактова частота	16МГц

Схема Arduino NANO

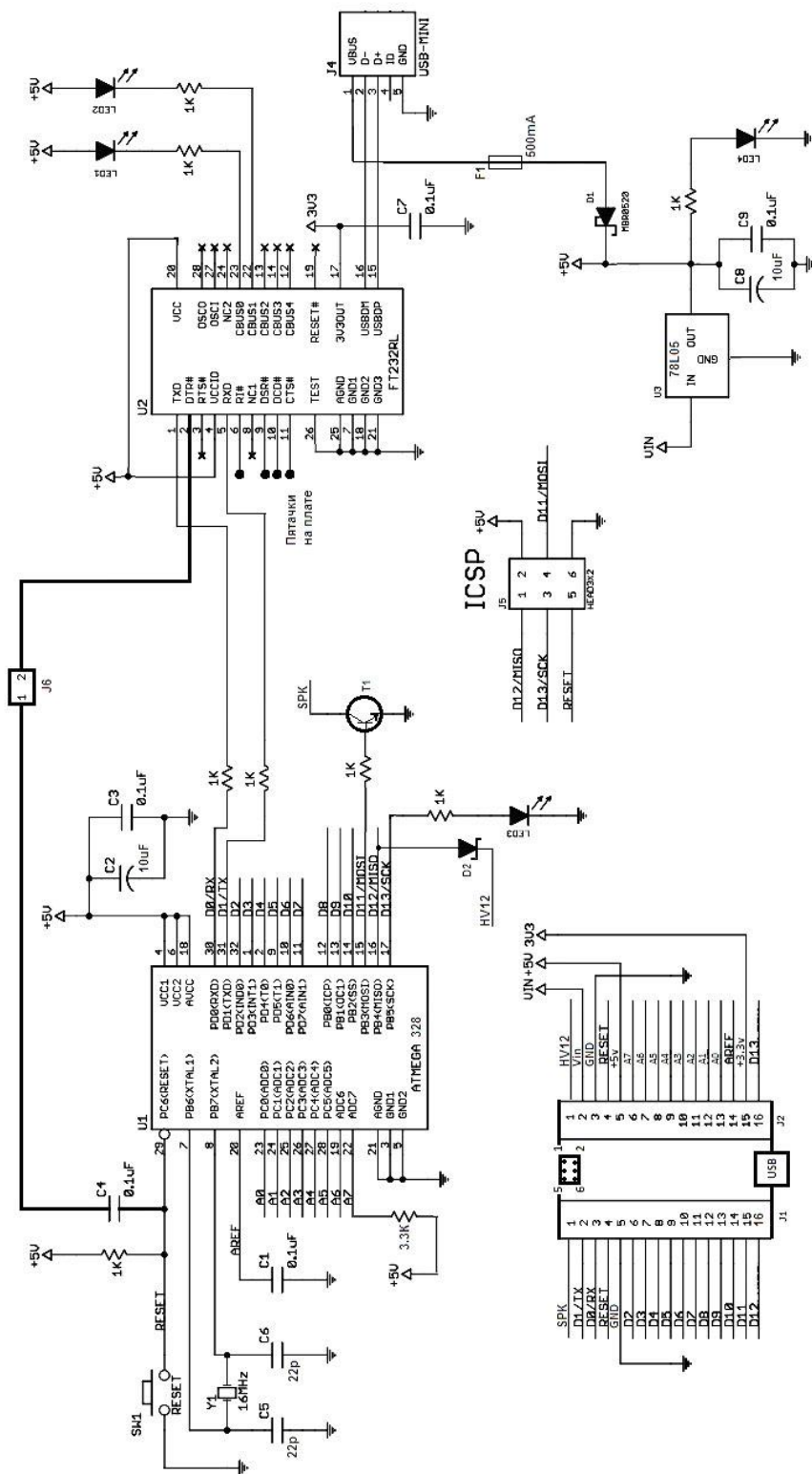
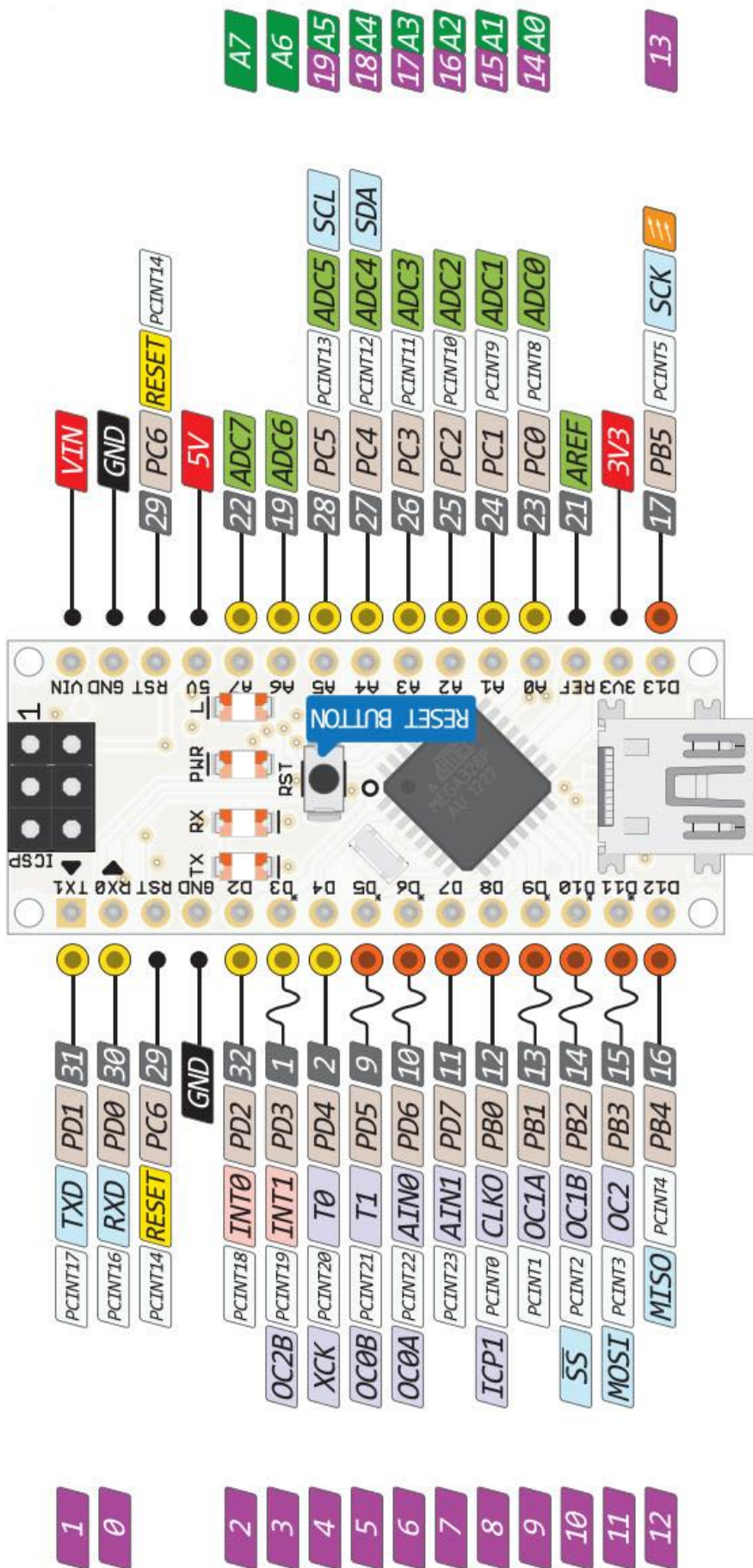


Рис.9. Схематичне зображення Arduino NANO



2.1.3 ARDUINO MEGA

Зовнішній вигляд та характеристики



Рис.11. Зовнішній вигляд ARDUINO MEGA

Таблиця 8

Характеристики

Мікроконтролер	ATmega2560
Робоча напруга	5V
Напруга живлення (рекомендована)	7-12V
Напруга живлення (граничне)	6-20V
Цифрові входи / виходи	54 (з них 15 можуть використовуватися в якості ШІМ-виходів)
Аналогові входи	16
Максимальний струм одного виведення	40mA
Максимальний вихідний струм виводу 3.3V	50mA
Flash-пам'ять	256 КБ(ATmega328) з яких 8 КБ використовується завантажувачем

Продовження табл. 8

EEPROM	4 КБ
Тактова частота	16МГц

Схема Arduino MEGA

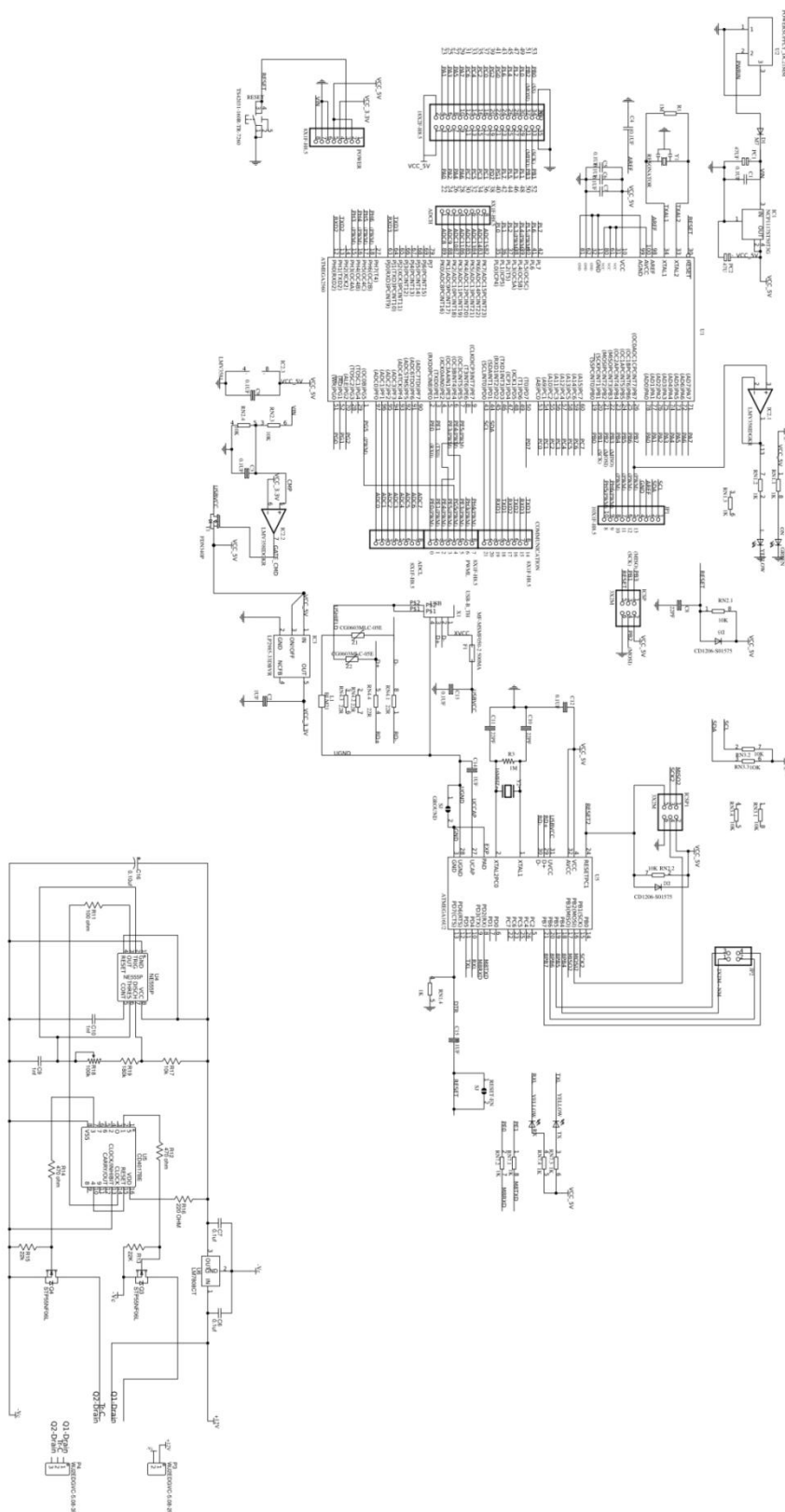


Рис.12. Схематичне зображення контролера

MEGA PINOUT

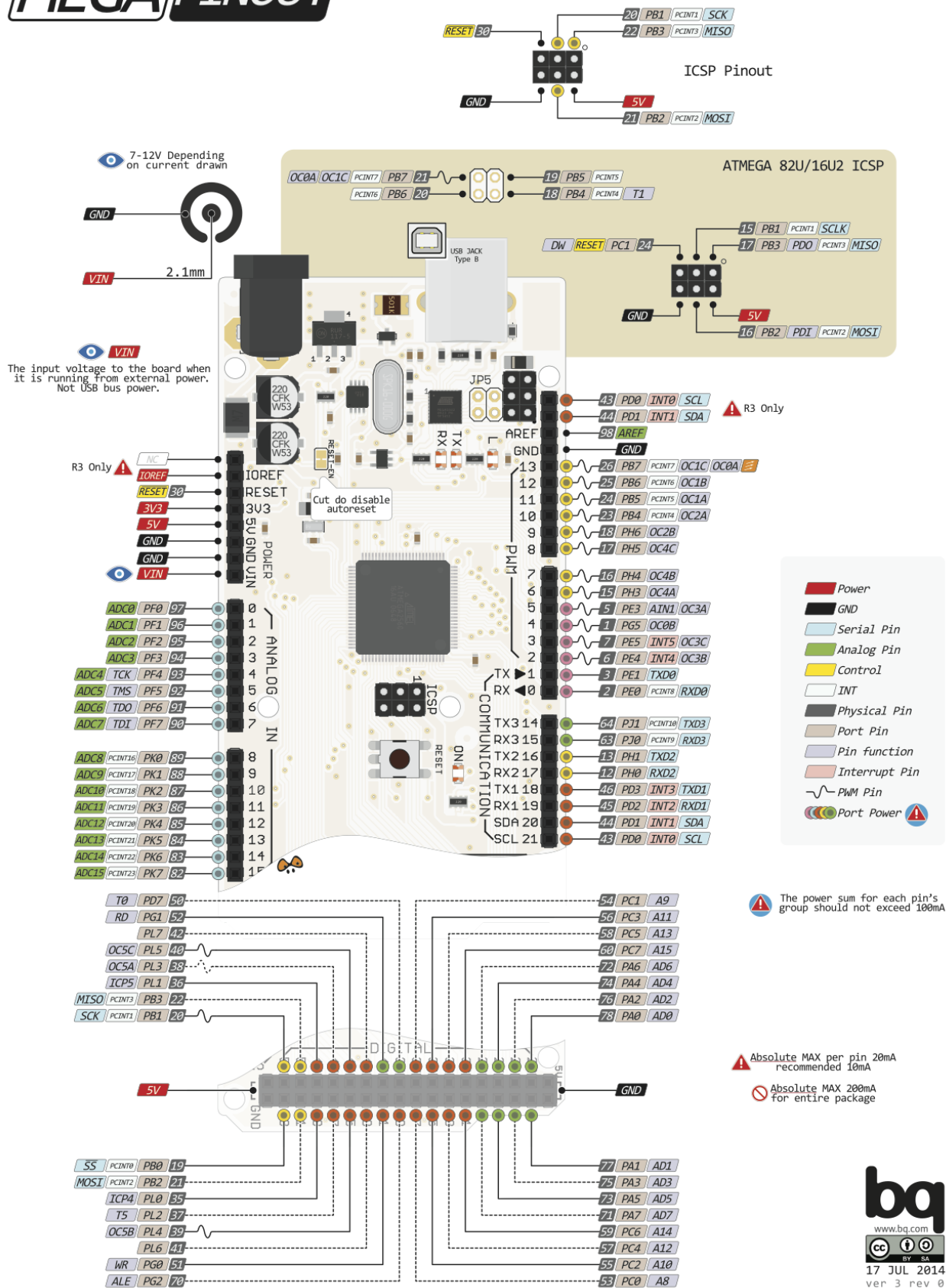


Рис.13. Розпіновка контактів

2.2 Системи на мікропроцесорі Broadcom BCM2711

2.2.1 Raspberry Pi 4

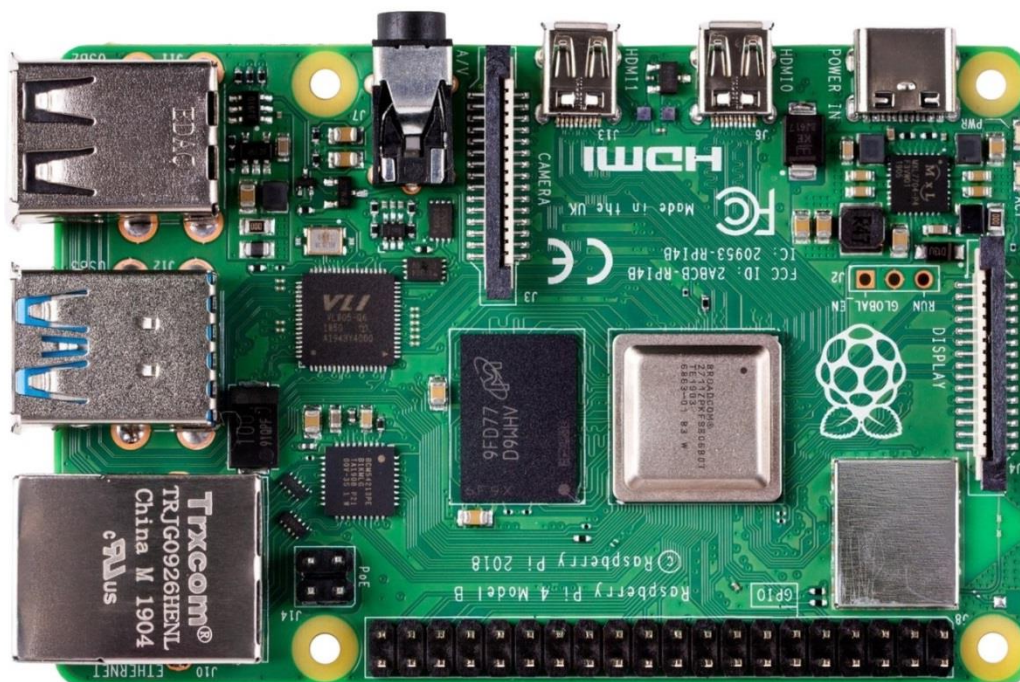


Рис.14 Зовнішній вигляд Raspberry Pi 4

Характеристики

У Raspberry Pi 4 вбудований 64 бітний чотирьох ядерним процесором частотою 1.5 ГГц (ARM Cortex-A72), підтримка двох дисплеїв з розширенням до 4к та 60 к/с, оперативною пам'ятю до 4Гб, двох діапазонною безпроводною антеною 2,4/5.0 ГГц, Bluetooth 5.0, USB 3.0 та можливість PoE

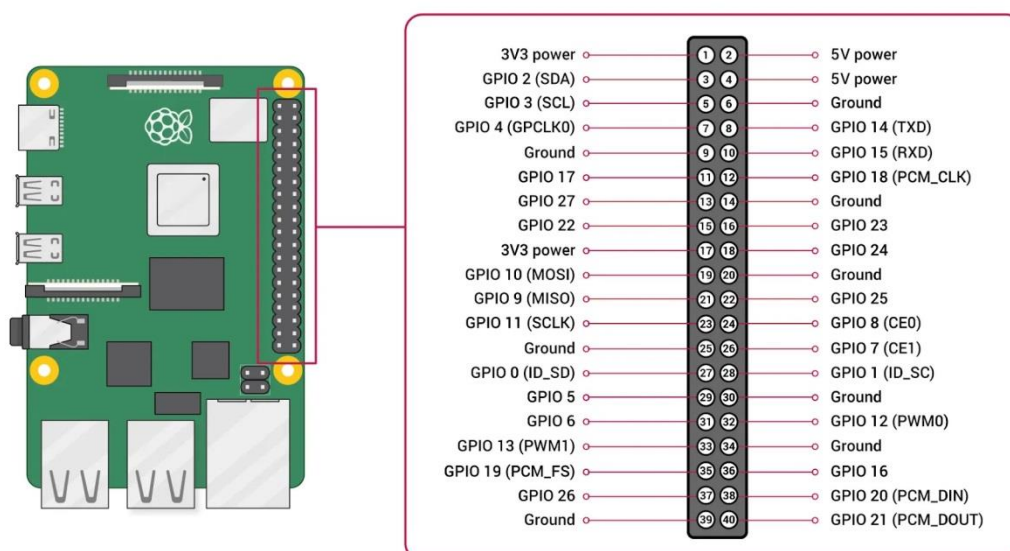


Рис.15. Розпіновка контактів

2.3 Огляд драйверів для біполярних крокових двигунів

2.3.1 L6219

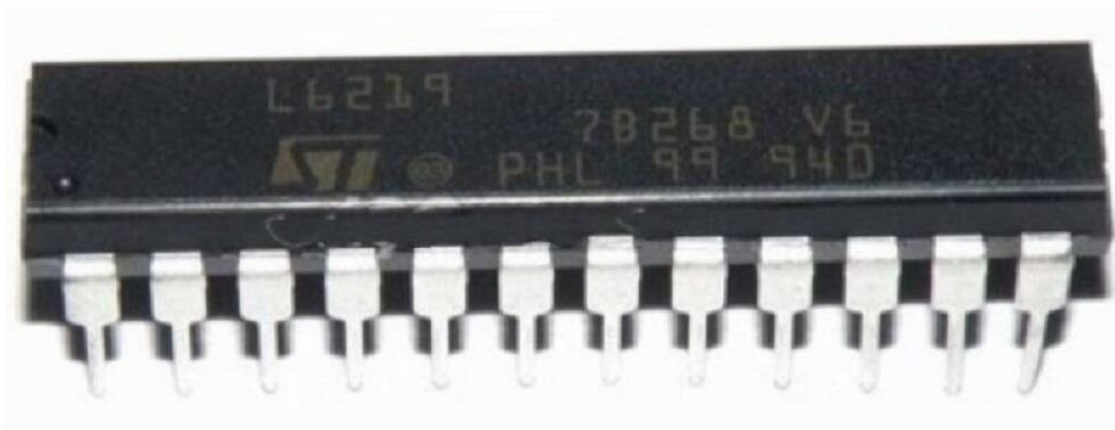


Рис.16. Зовнішній вигляд L6219.

Драйвер L6219 призначений для керування біполярних крокових двигунів, подаючи напругу на обидві обмотки. Підтримує подвійний міст напруги, та підтримує до 46В, включає чотири діоди для рециркуляції струму. Захист перехресної провідності присутній щоб уникнути одночасної перехресної провідності під час перемикавання напрямку струму. Внутрішня широтно-імпульсна модуляція контролює вихідний струм до 750mA з піковим струмом 1А. Широкий діапазон регулювання струму від 750mA(кожен міст) досягається за допомогою двох логічних входів і зовнішнього опорного джерела напруги. Фазовий вхід до кожного мосту визначає напрямок струму навантаження. Основні характеристики драйвера наступні:

Корпус: 24-SO.

Напруга живлення: 4,75V – 5,25 V.

Напруга яка подається на мотор: 10V – 45V.

Вихідний струм: 750mA.

Мікрокроки: 1\3, 2\3, 1.

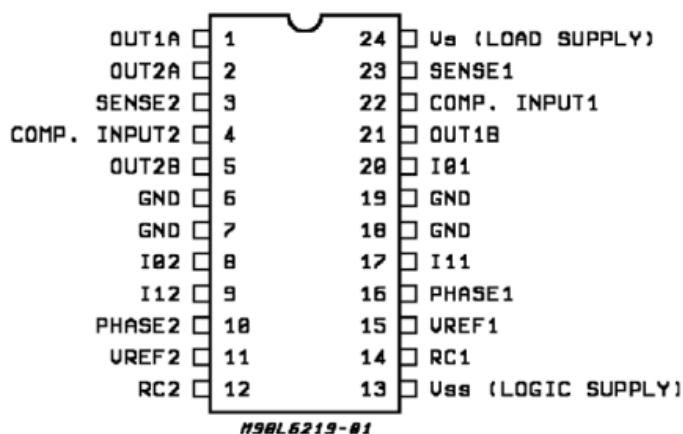


Рис.17. Розпіновка контактів драйвера

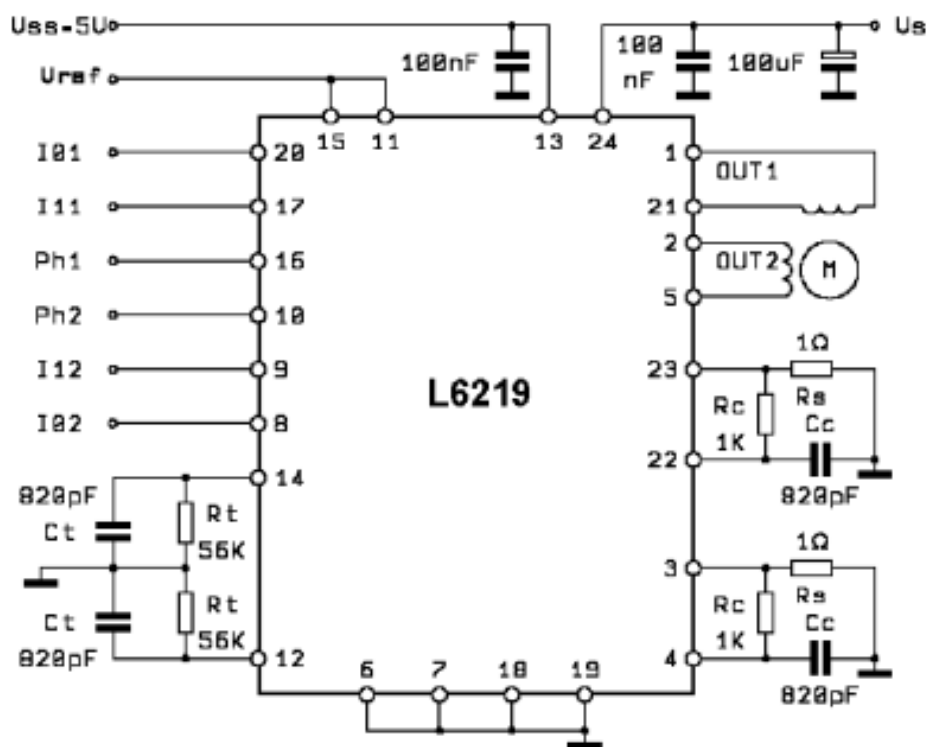


Рис.18. Схема підключення драйвера

Драйвер дозволяє працювати в 1/3, 2/3 та 3/3 мікрокроках. Керуючі контактами I_0 та I_1 ми можемо змінювати режими його роботи, як видно на цій хемі:

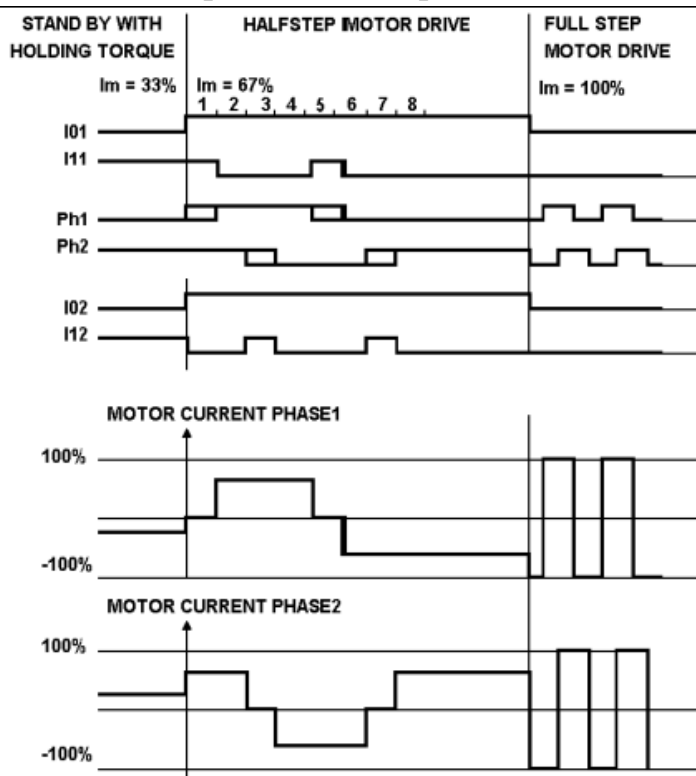


Рис.19. Принцип роботи драйвера

2.3.2 DRV8825

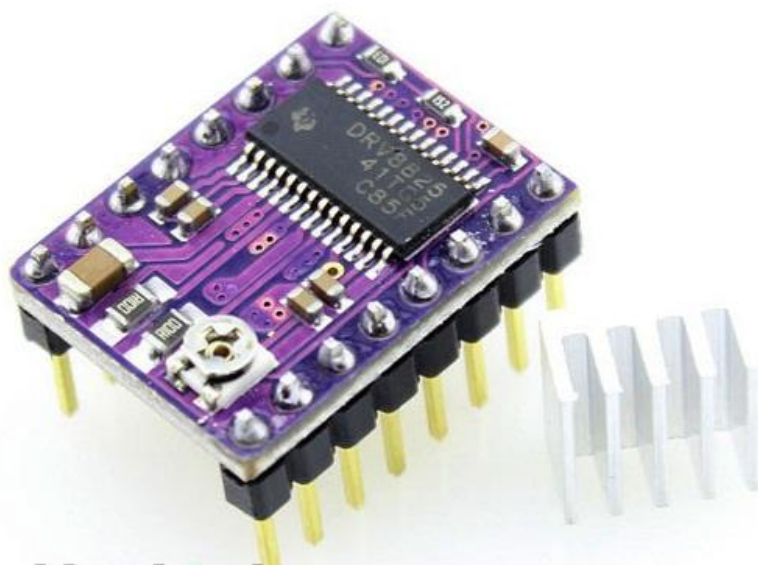


Рис.20 Зовнішній вигляд DRV8825

Характеристики

Цей драйвер для мікрокрокових двигунів має регульоване обмеження струму, захист від перевантаження та перегріву, а також шість мікрокрокових роздільних можливостей (до 1/32 кроку). Він працює від 8,2 В до 45 В і може видавати приблизно 1,5 А на фазу без радіатора (розрахований на 2,2 А на котушку з достатнім додатковим охолодженням).

Крокові двигуни зазвичай мають специфікацію розміру кроку (наприклад, 1,8° або 200 кроків на оберт), що стосується повних кроків. Мікрокроковий драйвер, такий як DRV8825, забезпечує більш високу роздільну здатність, дозволяючи розташування проміжних кроків, які досягаються шляхом живлення котушок проміжними рівнями струму. Наприклад, керування двигуном у чверть кроковому режимі дасть двигуну з 200 кроками на оберт 800 мікрокроків на оберт за допомогою чотирьох різних рівнів струму. Входи перемикача роздільної здатності (розмір кроку) (MODE0, MODE1 і MODE2) дозволяють вибрати з шести кроків роздільної здатності відповідно до таблиці нижче.

Таблиця 9

MODE0	MODE1	MODE2	Режим роботи
Low	Low	Low	Повний крок
High	Low	Low	Половина кроку
Low	High	Low	1/4 кроку
High	High	Low	1/8 кроку
Low	Low	High	1/16 кроку
High	Low	High	1/32 кроку
Low	High	High	1/32 кроку

Усі три селекторних входи мають внутрішні резистори 100 кОм, що на вихід, тому, якщо ці три контакти вибору мікрокроків відключені, це призводить до повнокрокового режиму. Щоб мікрокрокові режими функціонували правильно, обмеження струму має бути встановлене досить низьким, щоб спрацювало обмеження струму. В іншому випадку проміжні рівні струму не будуть підтримуватися належним чином, і двигун буде пропускати мікрокроки. Мікросхема має три різні входи для управління станами живлення: RESET, SLEEP і ENBL. Ці стани RESET і SLEEP за замовчуванням перешкоджають роботі драйвера; обидва ці контакти повинні мати високий рівень, щоб увімкнути драйвер (їх можна підключити безпосередньо до логічної «високої» напруги від 2,2 до 5,25 В, або ними можна динамічно керувати через підключення до цифрових виходів MCU).

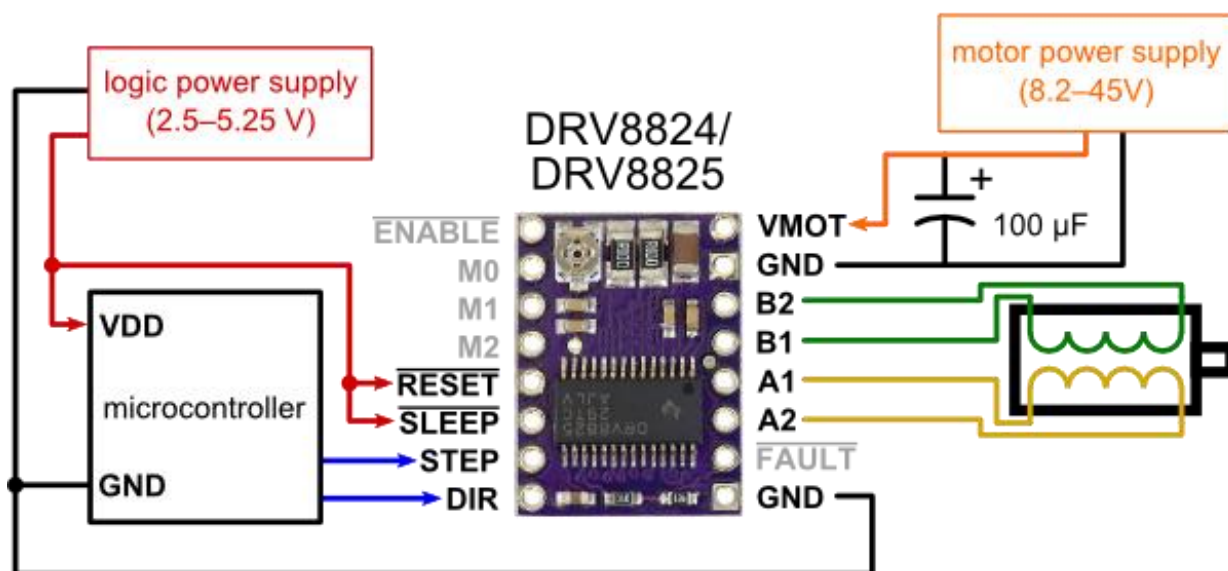


Рис.21. Схема підключення до мікроконтролера.

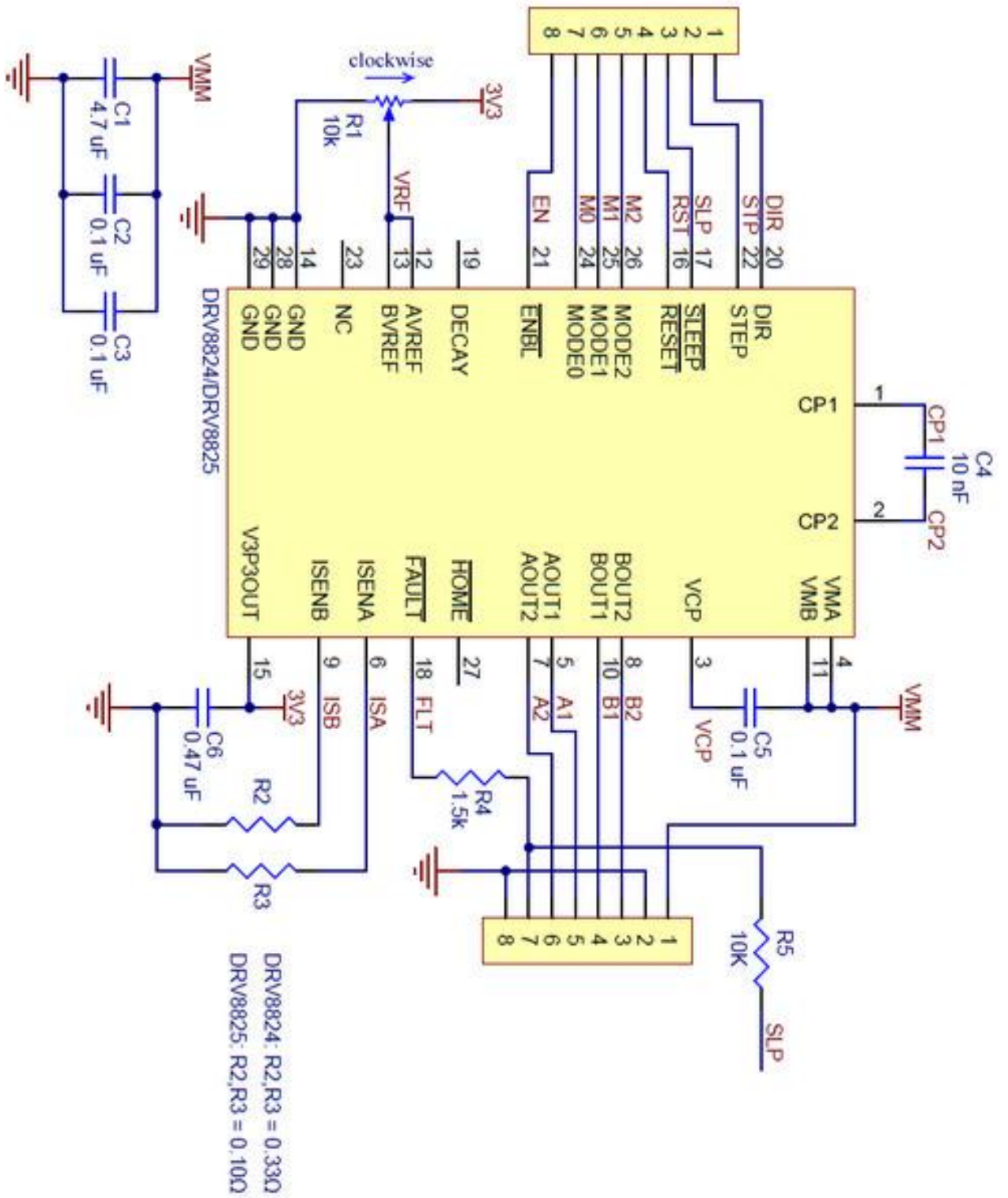


Рис.22. Розпіновка . DRV8825

2.3.3 TMC2209

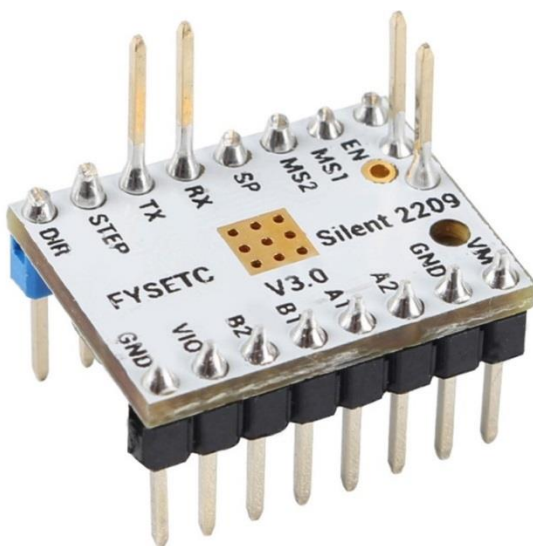


Рис.23 Зовнішній вигляд TMC2209

Характеристики

TMC2209 позиціонує себе як безшумний драйвер для керування двофазних крокових двигунів. Він працює від 5,5 В до 28 В і може видавати приблизно 1,25 А на фазу без радіатора (розрахований на 2,5 А на котушку з достатнім додатковим охолодженням). Сам драйвер може працювати з крокам аж до 1/256.

Таблиця 10

MS2	MS1	Режим роботи
GND	GND	1/8 кроку
GND	+5	1/32 кроку
+5	GND	1/64 кроку
+5	+5	1/16 кроку

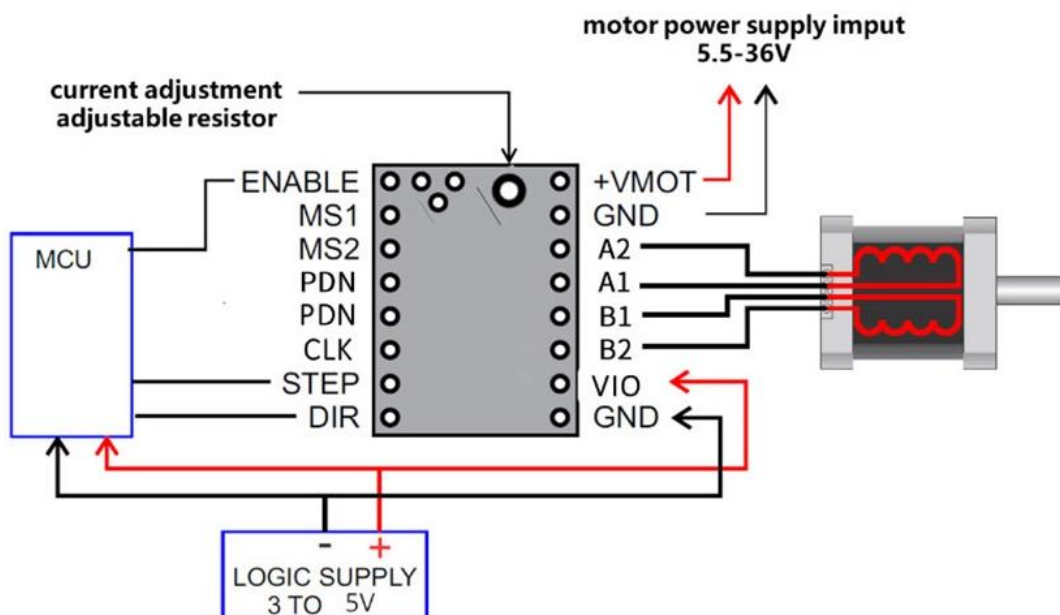


Рис.24 Схема підключення TMC2209

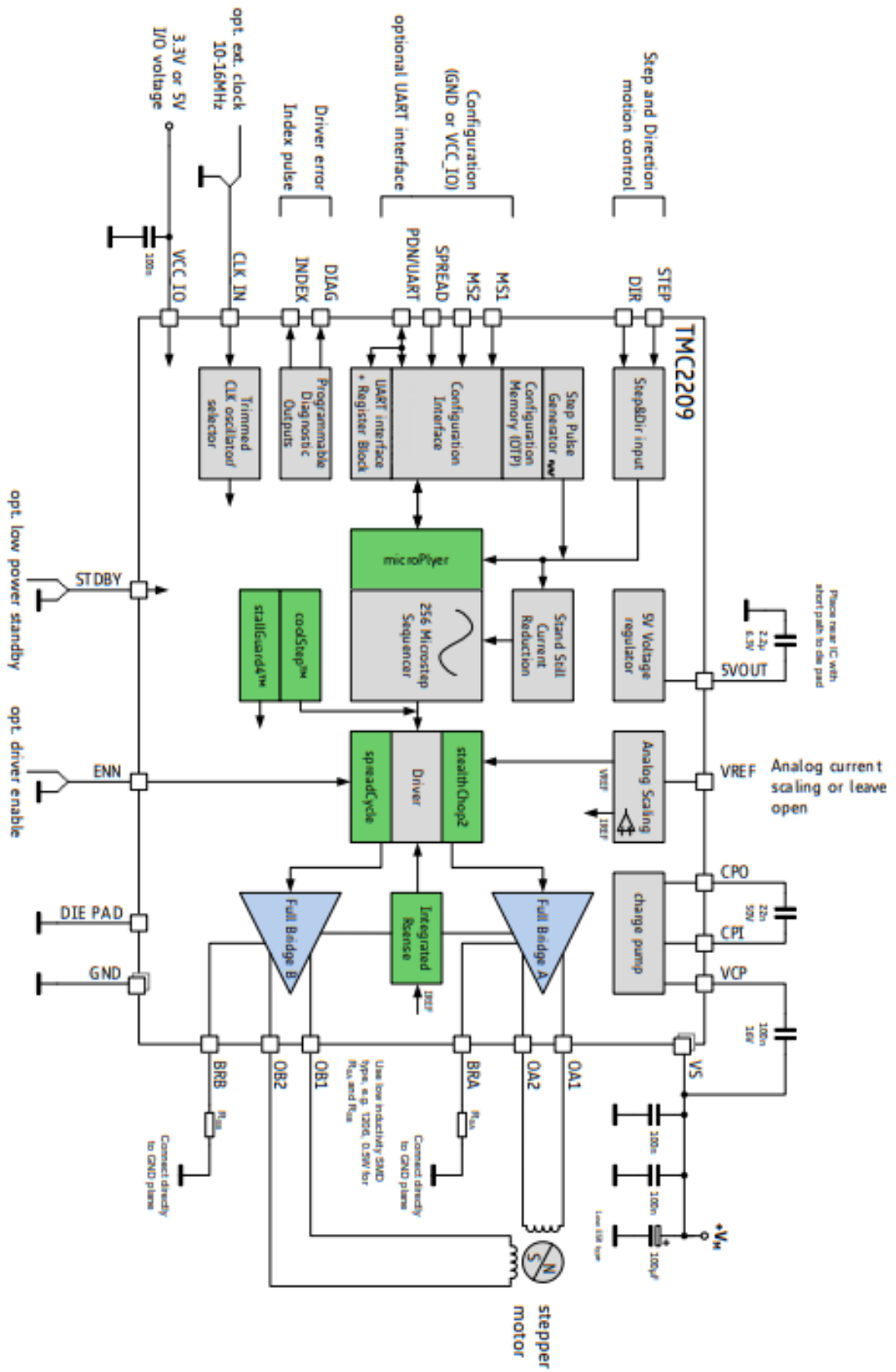


Рис.25 Блок схема TMC2209

2.3.4 A4988

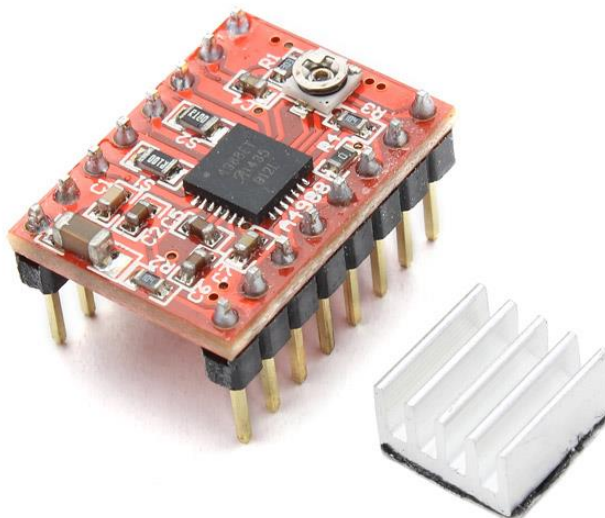


Рис.26 Зовнішній вигляд A4988

Драйвер A4988 позиціонує себе як мікрокроковим драйвером з вбудованим перемикачем опору. Він призначений для експлуатації біполярних крокових двигунів та може працювати у режимі 1/16 кроку. Максимальна напруга 35В а сила струму до 2А(з охолодженням)

Таблиця 11

MS1	MS2	MS3	Режим роботи
L	L	L	Повний крок
H	L	L	1/2 кроку
L	H	L	1/4 кроку
H	H	L	1/8 кроку
H	H	H	1/16 кроку

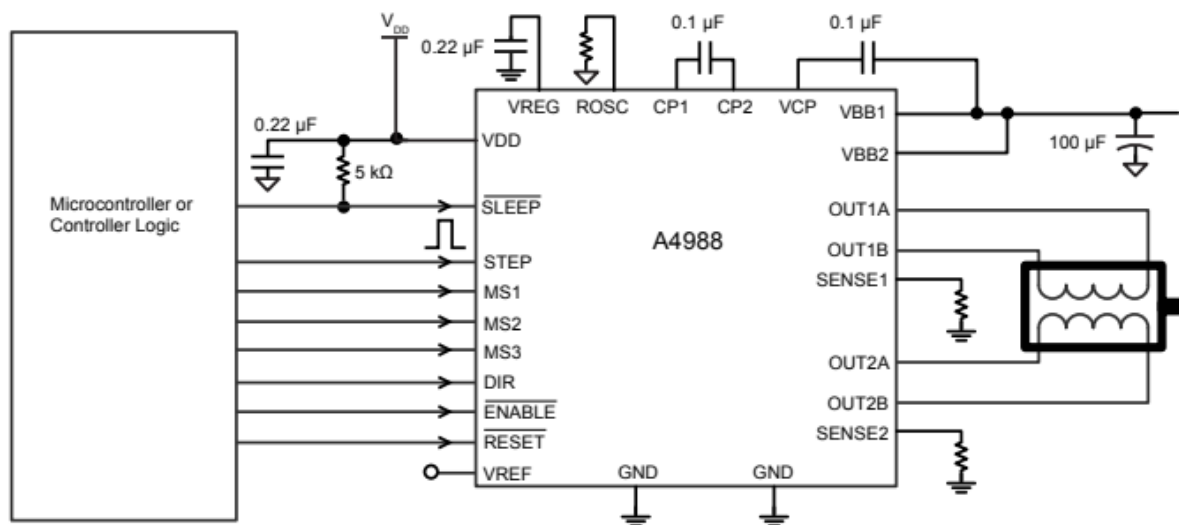


Рис.27 Схема підключення A4988

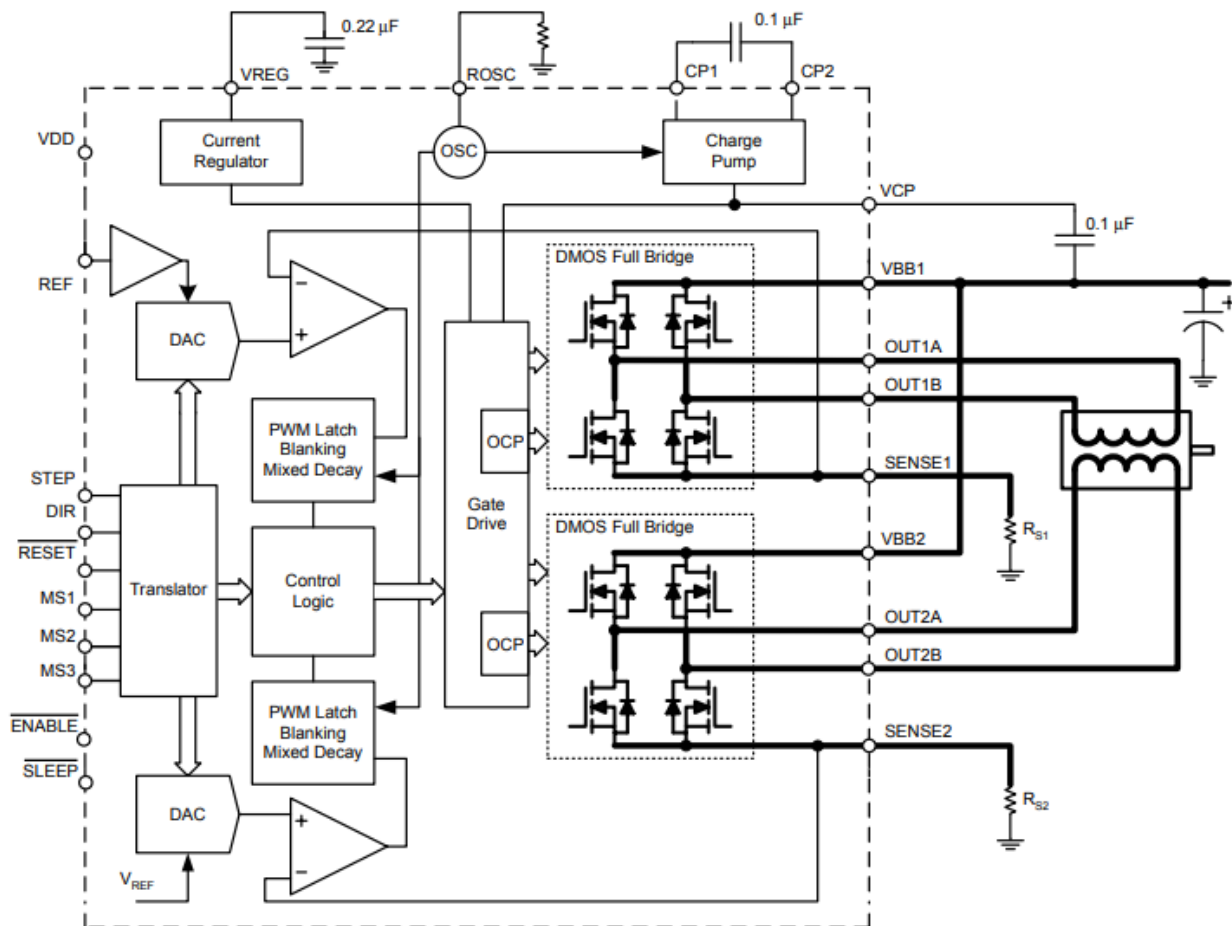


Рис.28 Блок схема А4988

Розділ 3: Розробка системи управління монохроматора SpectraPro-150

За результатами огляду літератури ми визначили що для наших цілей підійде платформа Arduino UNO R3 для керування даним монохроматором, використовуючи різноманітні драйвера для крокових двигунів, ми обрали оптимальний який дає найкращу точність і плавність руху ґраток з драйвером TMC2209. Підчас тестів перевірили на пропуск кроків він нас задовільнив. Для позиціонування ґраток у монохроматорі встановлено дві оптопари, яка задає грубу позицію і точну. Перша фіксує положення турелі, і знаходиться під нею, друга знаходиться біля крокового двигуна і задає точніше положення.

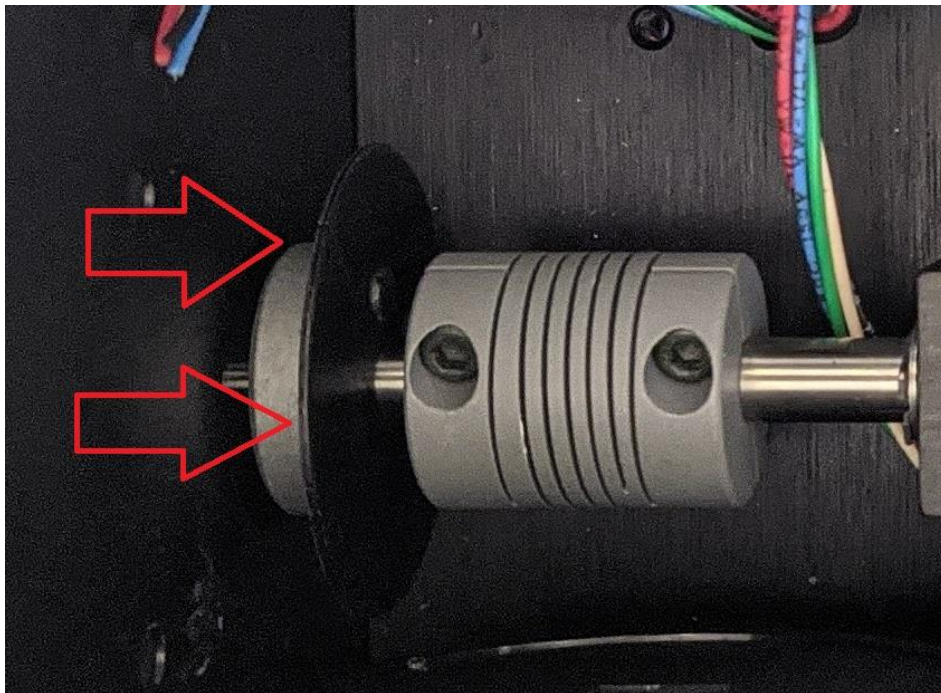


Рис.29 Оптопара біля крокового двигуна

На основі цього ми розробили програму для управління мікроконтролером

```
#include <FlexiTimer2.h>
```

```
const int led = 13;
```

```
const int grating_indicator = 4;
```

```
const int motor_indicator = 5;
```

```
const int dirPin = 2;
```

```
const int stepPin = 3;
```

```
long int current_step_num = 0;
```

```
long int needed_step_num = 0;
```

```

int sped = 30;

byte dr = 0;

long int steps = 0;

byte ind_1_state = 0;

byte ind_1_prestate = 0;

byte ind_2_state = 0;

byte ind_2_prestate = 0;

char command;

bool initialising = false;

bool stopRegim = false;

```

В цій частині програми ми налаштуємо та вказуємо нашій програмі якому номеру ніжки контролера відповідають значення:

13 – світловий індикатор.

4 та 5 - ніжка це наші оптопари які зчитують сигнал.

2 та 3 - задають роботу крокового двигуна.

Далі ідуть змінні `current_step_num` та `needed_step_num` які містять номер кроку, `sped` – задає часову затримку яку має кроковий двигун(в мікросекундах), `steps` - буферна змінна яка зберігає значення в яку сторону треба крутити. Далі ми ввели змінні які потрібні для роботи з оптопарами (`ind_1_state` та `ind_1_prestate`), вони містять поточний стан та попередній. За допомогою `command` можна керувати через COM порт нашим контролером. Присутня також можливість перевірки на ініціалізацію(`initialising`) та припинення роботи (`stopRegim`).

```

void setup()
{
  pinMode(stepPin, OUTPUT);

  pinMode(dirPin, OUTPUT);

  pinMode(led, OUTPUT);

  pinMode(grating_indicator, INPUT);

  pinMode(motor_indicator, INPUT);

```

```
digitalWrite(dirPin, LOW);
digitalWrite(led, LOW);
Serial.begin(2000000);
FlexiTimer2::set(60000, timerInterrupt);
}
```

Тут ми задаємо наші піни на вхід/вихід, щоб контролер розумів які потрібно зчитати а які відправити. Також підключаємо обмін даними по СОМ порту зі швидкістю 2000000, та задаємо нашому таймеру час на 60 секунд.

```
void loop()
{
  if (Serial.available())
  {
    if (!isalpha(Serial.peek())) Serial.flush();
    command = Serial.read();
```

Зчитання та перевірка того що першим прийшло на СОМ порт, якщо першим прийшло число то воно видаляється і вважається за сміття, а якщо буква тоді виконуються команди.

```
  if (command == 'i')
  {
    goto_zero();
    steps = 0;
    current_step_num = 0;
    Serial.print('i');
```

Якщо ми надіслали команду і, то наш монохроматор прямує в нульове значення тобто ініціалізується.

```
  }
  else if (command == 'd')
  {
    steps = Serial.parseInt();
    Serial.read();
```

```
motor_run_nsteps();

Serial.println(current_step_num);
```

Якщо прийшла команда d то ми крутимо нашу градку на задану кількість кроків. При використанні даної команди потрібно спершу написати команду d, потім кількість кроків, та закінчити будь-якою буквою, щоб програма розуміла де кроки а де команда, якщо в кінці не задати букву то він буде через кожне значення зупинятись.

```
}

else if (command == 'a') //перевірка чи підключений

{

Serial.print('a');
```

При використанні цієї команди, ми перевіряємо чи працює наш мікроконтролер.

```
}

else if (command == 'g') {

needed_step_num = Serial.parseInt();

Serial.read();

steps = needed_step_num - current_step_num;

motor_run_nsteps();

Serial.println(current_step_num);
```

Дана команда дозволяє нам виставляти градку на точно заданий крок (повне коло для прокрутки градками становить 576000). Працює так само як і команда d.

```
}

}

}
```

У функцію нижче задана логіка для крутіння в ліво або в право нашу градку, яка записана в змінну steps. Можна задавати як додатні так і від'ємні значення.

```
inline void motor_run_nsteps()

{

digitalWrite(led, HIGH);

if (steps < 0) dr = 1; else dr = 0;

digitalWrite(dirPin, dr);
```

```

steps = abs(steps);
while (steps)
{
  makeStep();
  steps--;
  if (!dir) current_step_num++; else current_step_num--;
  if (current_step_num > 576000) current_step_num -= 576000;
  if (current_step_num < 0) current_step_num += 576000;
}
digitalWrite(led, LOW);
}

```

Функція ініціалізації. Основний принцип роботи тут в зчитуванні сигналів з наших оптопар та записуємо ці змінні, запускаємо наш таймер FlexiTimer2 щоб ініціалізація тривала не довше 60 секунд, якщо більше 60 то відбувається переривання і програма зупиняється. Далі по кроково крутимо наш двигун і перевіряємо чи не отримали ми сигнал на одній з оптопар, якщо отримали то зменшуємо нашу швидкість обертання двигуна та очікуємо на сигнал на другій оптопарі. Сигнали які міняють свої значення після спрацювання на обох оптопарах вважаємо за нульове значення.

```

void goto_zero()
{
  digitalWrite(led, HIGH);
  digitalWrite(dirPin, 0);
  ind_1_state = !digitalRead(grating_indicator);
  ind_1_prestate = ind_1_state;
  initialising = true;
  FlexiTimer2::start();
  while (true)
  {
    if (stopRegim)
    {

```



```
digitalWrite(led, LOW);
initialising = false;
stopRegim = false;
FlexiTimer2::stop();
return;
}
makeStep();
ind_1_state = !digitalRead(grating_indicator);
if ((!ind_1_state) && ind_1_prestate)
{
    makeStep();
    if (!ind_1_state) break;
    else
    {
        while (ind_1_state) makeStep();
        break;
    }
}
ind_1_prestate = ind_1_state;
}
sped *= 10;
ind_2_state = digitalRead(motor_indicator);
ind_2_prestate = ind_1_state;
while (true)
{
    if (stopRegim)
    {
        digitalWrite(led, LOW);
        initialising = false;
```

```
stopRegim = false;
FlexiTimer2::stop();
return;
}
makeStep();
ind_2_state = digitalRead(motor_indicator);
if ((!ind_2_state) && ind_2_prestate)
{
    if (stopRegim)
    {
        digitalWrite(led, LOW);
        initialising = false;
        stopRegim = false;
        FlexiTimer2::stop();
        return;
    }
    makeStep();
    if (!ind_2_state) break;
    else
    {
        while (ind_2_state)
        {
            if (stopRegim)
            {
                digitalWrite(led, LOW);
                initialising = false;
                stopRegim = false;
                FlexiTimer2::stop();
                return;
            }
        }
    }
}
```

```

    }
    makeStep();
}
break;
}
}
ind_2_prestate = ind_2_state;
}
speed /= 10; // return high speed
digitalWrite(led, LOW);
initialising = false;
stopRegim = false;
FlexiTimer2::stop();
}

```

Функція яка крутить наш кроковий двигун з заданою затримкою (мікросекундах)

```

inline void makeStep()
{
    digitalWrite(stepPin, HIGH);
    delayMicroseconds(speed);
    digitalWrite(stepPin, LOW);
    delayMicroseconds(speed);
}

void timerInterrupt()
{
    if (initialising) stopRegim = true;
}

```

Програма дозволяла переміщати турель ґратки у будь яке положення. Експериментально було встановлено що повний оборот становить 576000 кроків (для ґраток), тому наступне завдання було визначення областей кутів при яких спостерігається нульовий, перший та другий порядок. Визначали за допомогою гелій неоновового лазера.

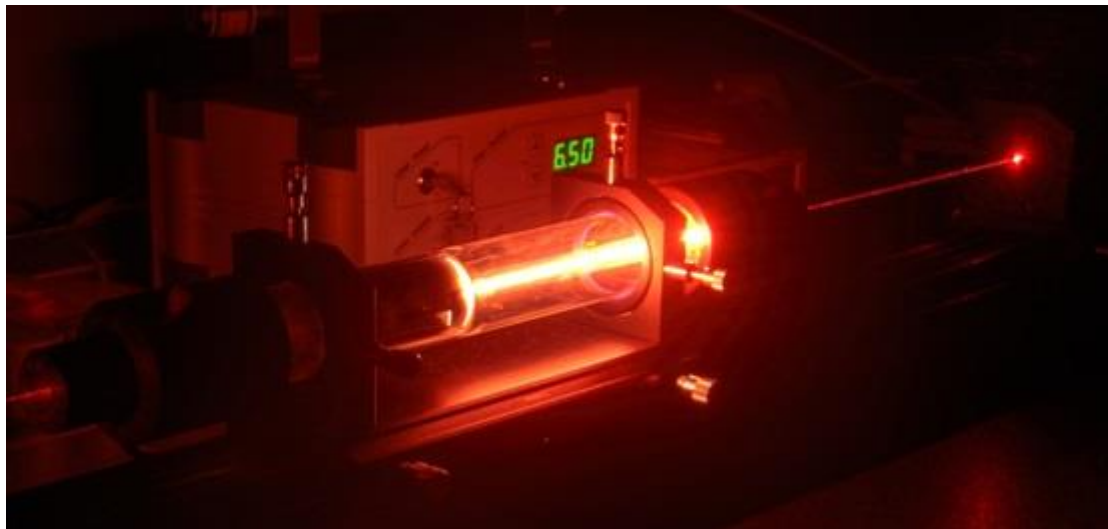


Рис.30 Вигляд гелій-неоновового лазера

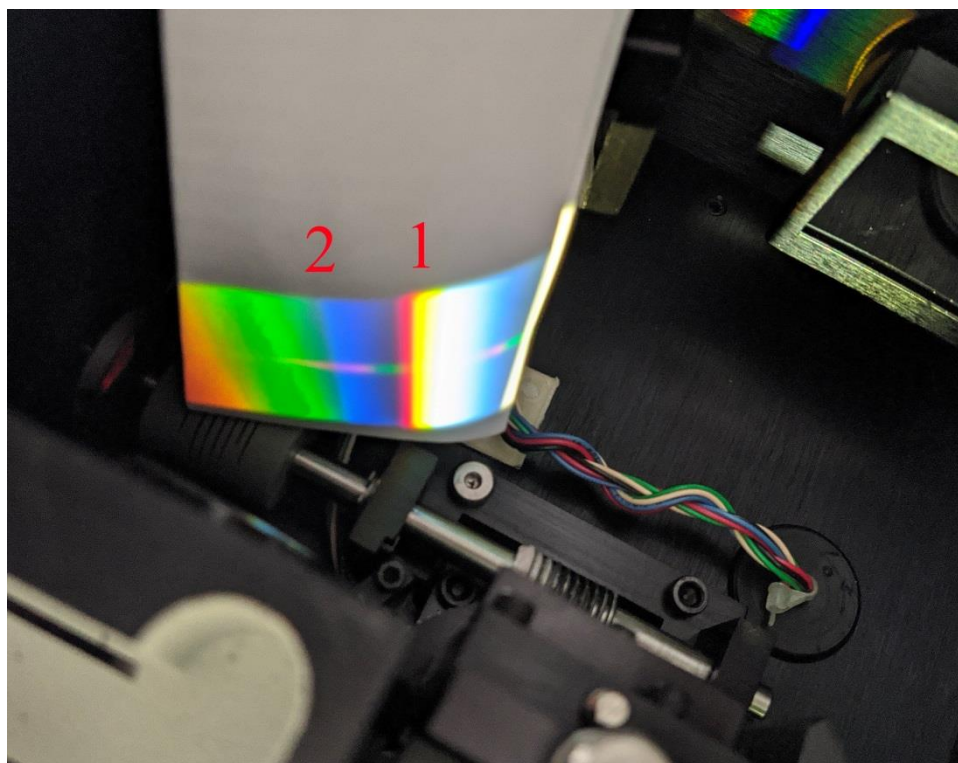


Рис.31 Нульовий та перший порядок отриманий за допомогою білого світла

1: нульовий порядок (біла лінія), 2: перший порядок (починається з червоного кольору)

Спочатку потрібно було встановити нульове положення або положення старту для ґраток. Програма перед запуском встановлюється в нульове положення, яке відповідає спрацюванню двох термопар. Тепер відносно цього положення маючи можливість в

програмі задавати переміщення на будь-яку кількість кроків та крутити в обидві сторони, маючи таку програму ми змогли визначити, якому номеру відповідає для кожної ґратки нульовий порядок, перший, другий порядок.

$$d(\sin\varphi_0 - \sin\varphi) = m\lambda \quad (3.1)$$

Таким чином ми визначити приблизний діапазон кутів для першого порядку який складає від 5 до 10 градусів. Зважаючи на ці данні ми вирішили в подальшому, залежність довжини хвилі на виході номера кроку двигуна монохроматора, задавати лінійною. Таким чином для нас юстувати монохроматор у відповідність крокам-довжин хвиль ця проблема стала проблемою визначення коефіцієнтів в лінійній формулі:

$$n = a + b\lambda \quad (3.2)$$

Для того щоб знайти ці два коефіцієнти, нам достатньо знати дві довжини хвилі. З цією задачею нам допоміг перший порядок з гелій-неонового лазера (638nm):

$$n_1 = a + b\lambda_1 \quad (3.3)$$

і ми взяли ще зелений лазер, напівпровідниковий (532nm):

$$n_2 = a + b\lambda_2 \quad (3.4)$$

Маючи приблизні значення А і В та номер кроку ми отримали спектр в першому наближенні:

$$n = a_0 + b_0\lambda \quad (3.5)$$

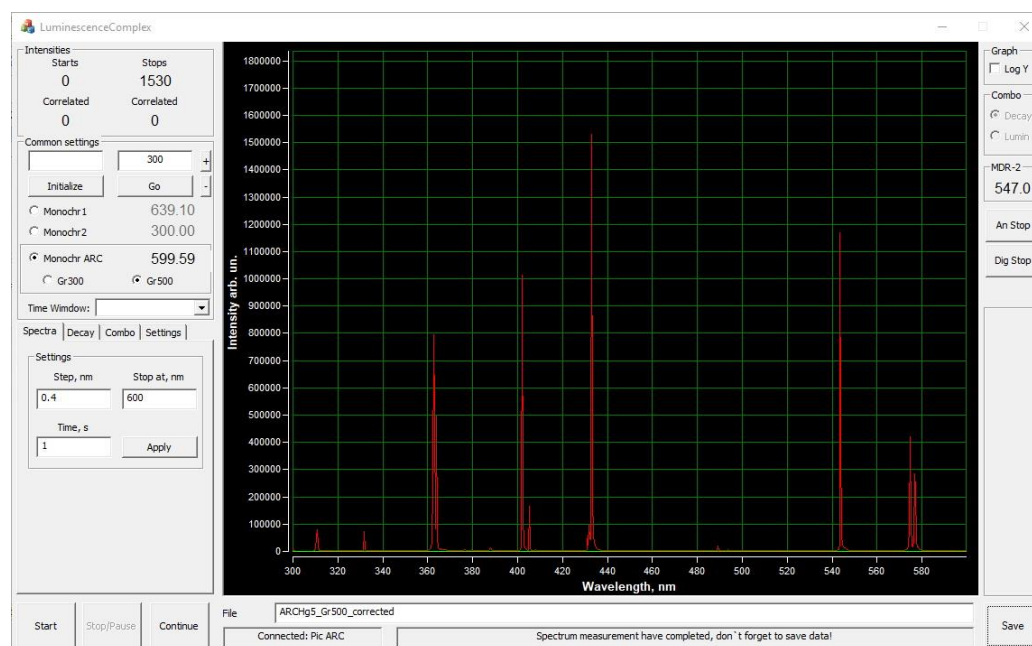


Рис.32 Спектр ртутної лампи низького тиску за допомогою монохроматора ARC

Провели апроксимацію для правильної довжини хвилі і не правильної.

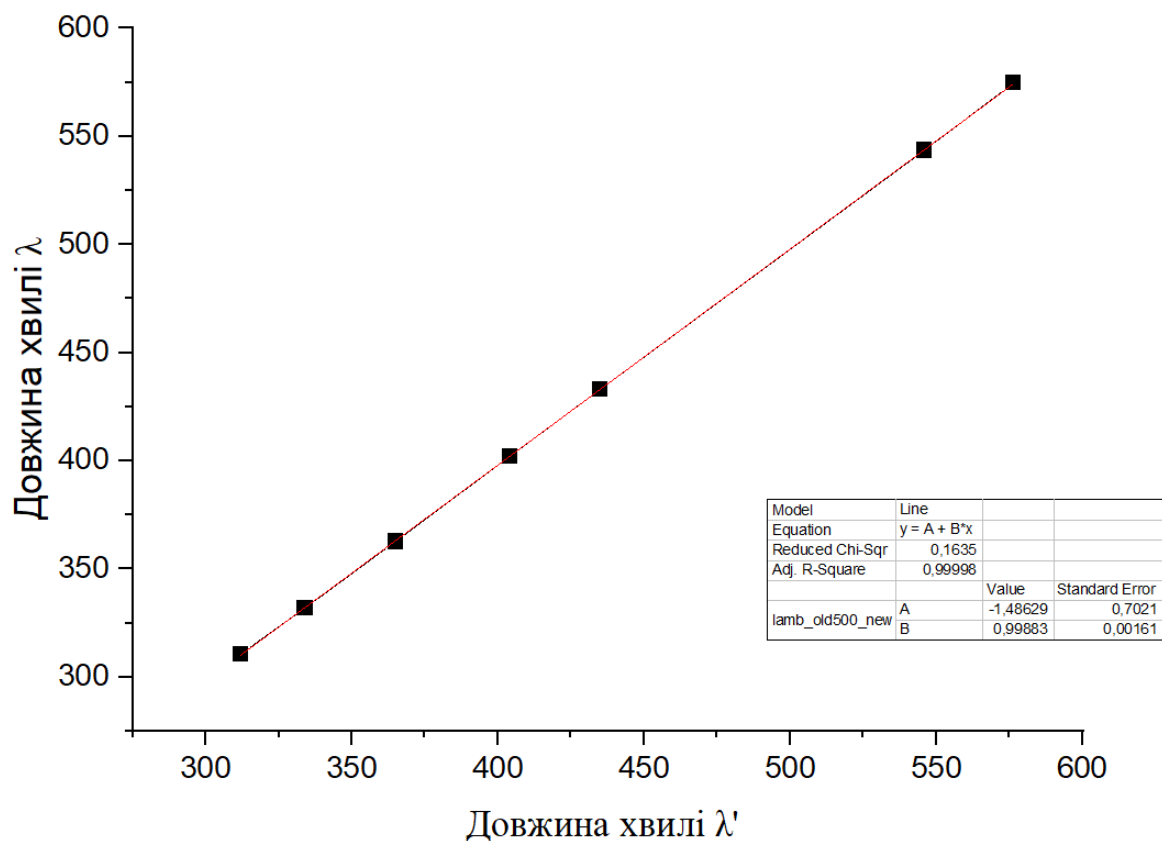


Рис.33 Апроксимація правильної довжини хвилі від не правильної

І на основі цих даних побудували графік:

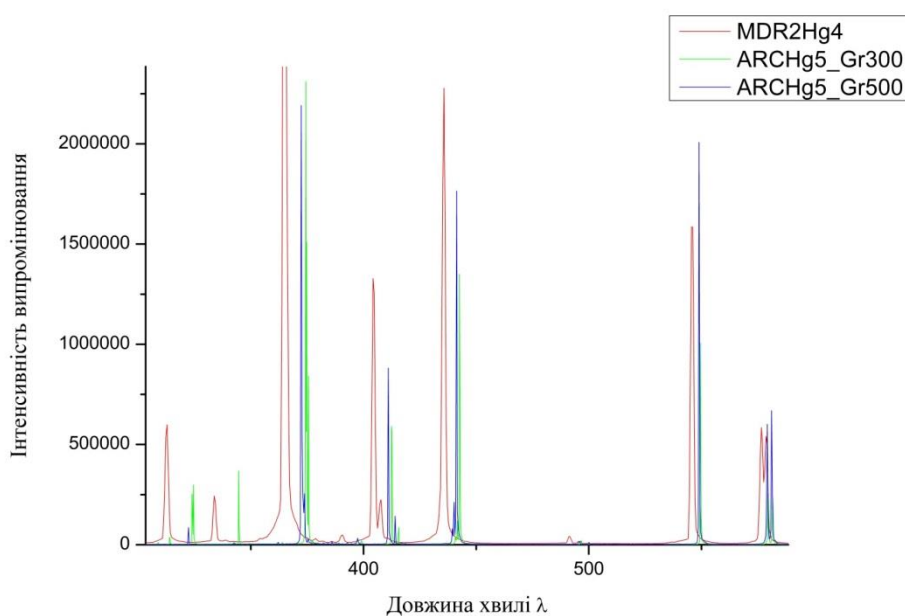


Рис.34 Спектр після приблизного юстування монохроматора

Для другого наближення ми використали данні з апроксимації:

$$\lambda' = c + d\lambda \quad (3.6)$$

І для остаточної корекції будемо мати:

$$n = (d^0 + b_0c) + b_0d\lambda \quad (3.7)$$

На основі фінальної формули для корекції графік спектру буде мати вигляд:

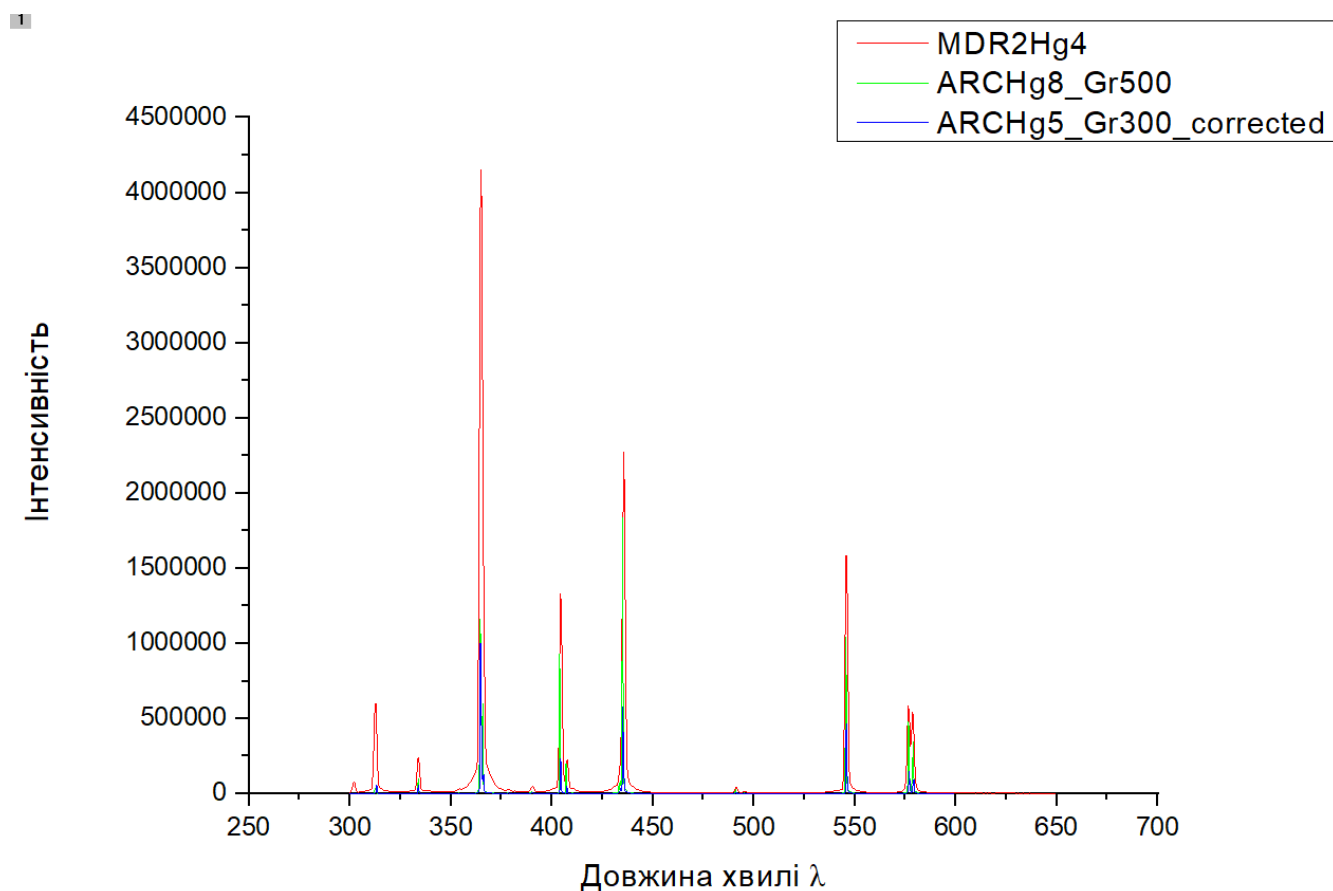


Рис.35 Спектр після приблизного юстування монохроматора

Як видно зі спектру, наші обидві ґрадки повністю співпали, таким чином можна сказати що монохроматор проґрадуваний.

Використані джерела

1. Пархоменко Д. А., Смирнов Є. М. Розробка радіоелектронних схем на основі мікроконтролерів (на прикладі AVR мікроконтролерів фірми Atmel) 2013, 73 ст
2. В.О. Поджаренко, В.Ю. Кучерук, В.М. Севастьянов ОСНОВИ МІКРОПРОЦЕСОРНОЇ ТЕХНІКИ 2006, 228 ст
3. Пушак А.С., Вістовський В.В. ПРОГРАМУВАННЯ PIC-КОНТРОЛЕРІВ 2021, 118 ст
4. Johannes Wild Arduino | Step by Step: The Ultimate Beginner's Guide with Basics on Hardware, Software, Programming & DIY Projects 2022. 94 с
5. Jeff Cicolani Beginning Robotics with Raspberry Pi and Arduino: Using Python and OpenCV 2021. 369 с
6. Operating Instructions Acton Research Corporation SpectraPro-150 – Режим доступу : <http://ridl.cis.rit.edu/products/manuals/Acton/old/MANUAL/Sp-150.pdf>
7. Monochromators and Spectrographs . . . Higher Throughput, Resolution, and Precision – Режим доступу : http://ridl.cfd.rit.edu/products/manuals/Acton/old/CATALOG/SpectruPro_bro.pdf
8. Nicholas Stuart Arduino Programming: 3 in 1- Beginner's Guide+ Tips and tricks+ Advanced methods to learn Arduino programming 2021. 536 с
9. Blatt Jan Arduino Programming: What Is The Need For The Arduino Music Project?: Arduino Music Player With Display 2021. 188 с