**ЛАБОРАТОРНА РОБОТА №3**

***Мета заняття:*** ознайомитися з основними поняттями. Встаановити середовище розробки. Отримати практичні навички по роботі з відлагоджувальною платою та середовищем розробки.

**Початок роботи**

**Завдання №1.** Ознайомитися з основними поняттями.

Розпочнемо виконання лабораторної роботи. Для початку опишемо короткі теоретичні відомості про платформу Arduino.

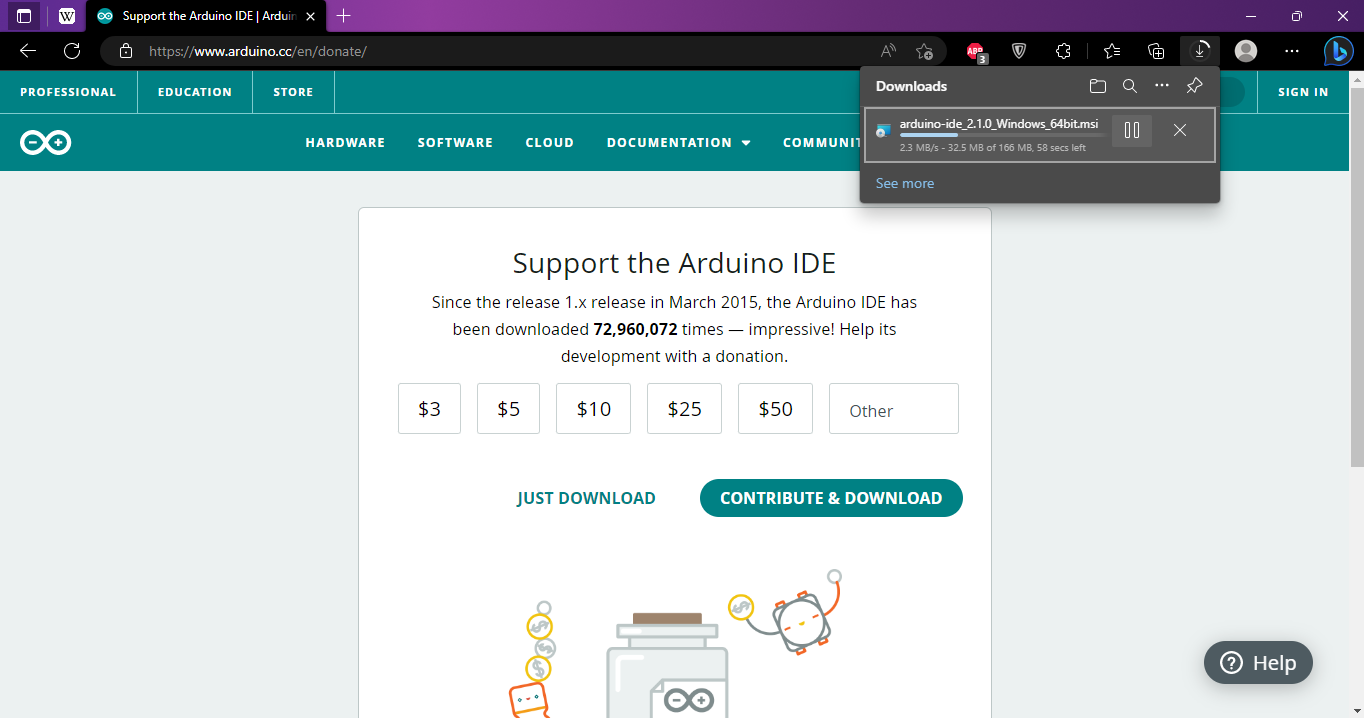
В першу чергу Arduino апаратна обчислювальна платформа для аматорського конструювання, основними компонентами якої є плата мікроконтролера з елементами вводу/виводу та середовище розробки Processing/Wiring на мові програмування, що є спрощеною підмножиною C/C++. Arduino може використовуватися як для створення автономних інтерактивних об'єктів, так і підключатися до програмного забезпечення, яке виконується на комп'ютері. Ардуїно може використовуватися для розробки інтерактивних систем, керованих різними датчиками та перемикачами. Такі системи, у свою чергу, можуть керувати роботою різних індикаторів, двигунів та інших пристроїв. Проекти Ардуїно можуть бути як самостійними, так і взаємодіяти з програмним забезпеченням, що працює на персональному комп'ютері (наприклад, програмами Flash, Processing, MaxMSP). Будь-яку плату Ардуїно можна зібрати вручну або купити готовий пристрій; середовище розробки для програмування такої плати має відкритий вихідний код і є повністю безкоштовним. Мова програмування Ардуїно є реалізацією схожої апаратної платформи "Wiring", яка базується на середовищі програмування мультимедіа "Processing".

ArduinoIDE — інтегроване середовище розробки для Windows, MacOS та Linux, розроблене на *С* та *C++*, призначене для створення та завантаження програм на Arduino-сумісні плати, а також на плати інших виробників.

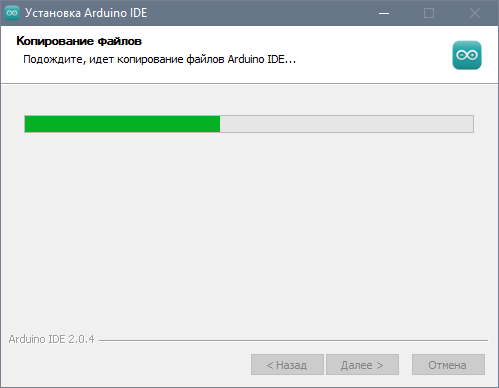
Вихідний код середовища випущений під загальнодоступною ліцензією GNU версії 2. Підтримує мови *С* та *C++* з використанням спеціальних правил структурування коду. *ArduinoIDE* надає бібліотеку програмного забезпечення з проекту *Wiring*, яка надає безліч загальних процедур введення та виведення. Для написаного користувачем коду потрібні лише дві базові функції для запуску ескізу та основного циклу програми, які скомпільовані та пов'язані з заглушкою програми *main()* у виконувану циклічну програму з ланцюжком інструментів GNU, також включеною до дистрибутиву IDE. Використовує програму *avrdude* для перетворення коду, що виконується, в текстовий файл у шістнадцятковому кодуванні, який завантажується в плату Arduino програмою-завантажувачем у вбудованому програмному забезпеченні плати. Зі зростанням популярності Arduino, інші постачальники, в якості програмної платформи, почали впроваджувати компілятори та інструменти з відкритим вихідним кодом (ядра), які можуть створювати і завантажувати ескізи в інші мікроконтролери, що не підтримуються офіційною лінійкою мікроконтролерів Arduino.

**Завдання №2.** Встановити середовище розробки Arduino IDE.

Розпочнемо процес встановлення програмного середовища ArduinoIDE. Для виконання цього завдання виконаємо завантаження відповідного істаляційного пакету з офіційного сайту розробника даного програмного забезпечення.



Запустимо на виконання істаляційний пакет, після чого виконаємо усі необхідно налаштування для виконання подальшого процесу становлення даного ПО.



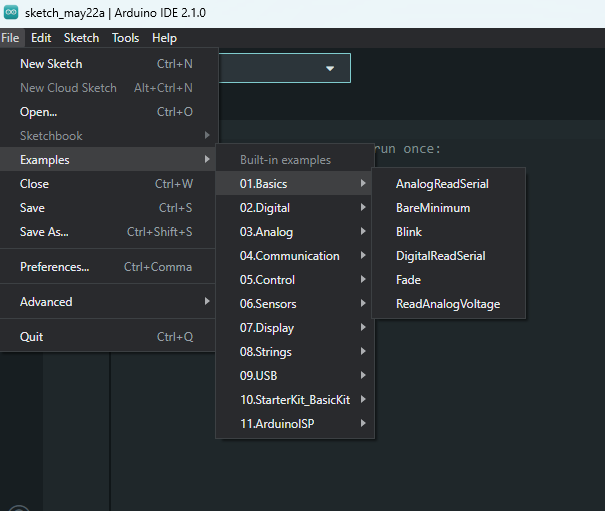
Після виконання процесу встановлення, програма являється повністю готовою до її експлуатації.



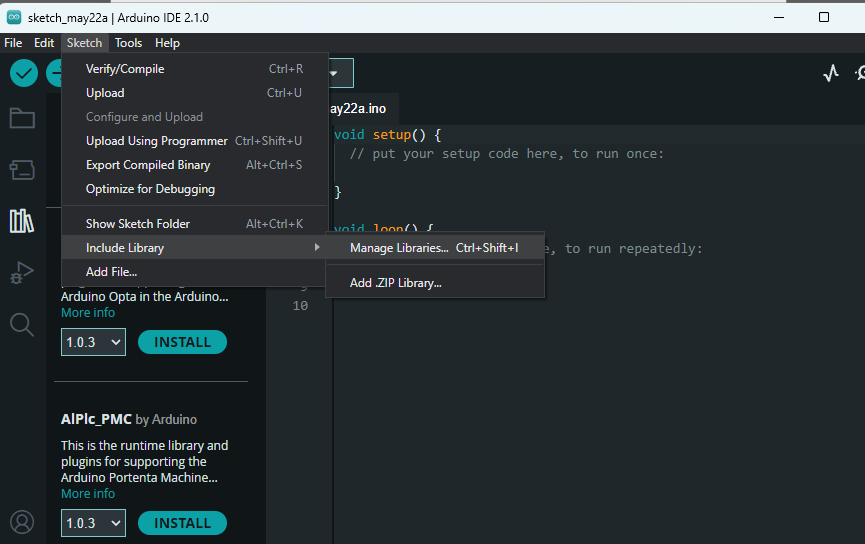
**Завдання №3.** Ознайомитися з роботою в середовищы розробки Arduino IDE та виконати приклади з лістингів 3.1 – 3.3. Розібратися з роботою прикладів.

Розпочнемо виконання процесу ознайомлення з роботою в програмному середовищі ArduinoIDE. Дане середовище програмування призначене для виконання розробки програмного забезпечення для платформи Arduino або для мікроконтролерів, що використовуються на базі цих платформ.

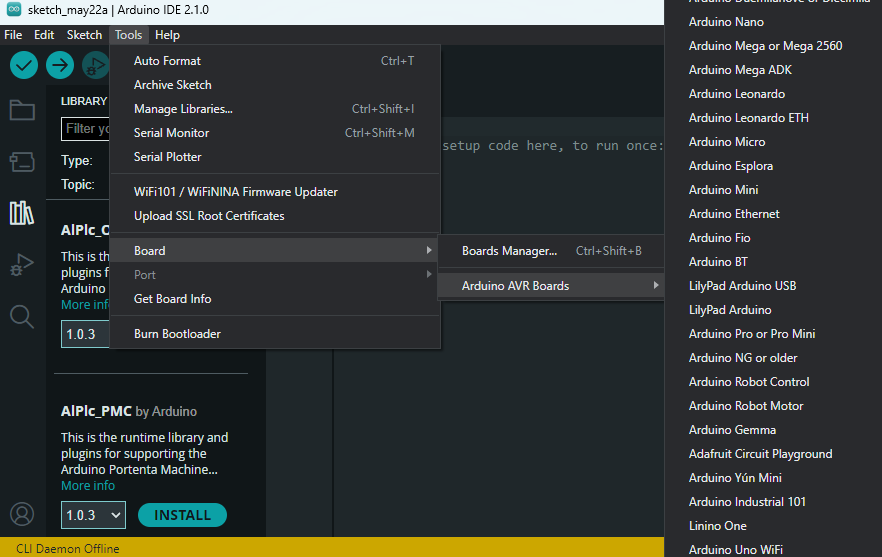
Середовище програмування ArduinoIDE працює на базі мови програмування C/C++ и скомпонований з бібліотекою AVR Libc. Дане середовище програмування не лише призначене для коду, але й яляється непоганим середовищем для виконання ознайомлення з основами написання програмного коду для мікроконтролерів цієї платформи. Для відкриття відповідних прокладів достатньо перейти до вкладки «File» та обрати пункт «Exanples» після чого у відповідному меню варто обрати, котрий з представлених прикладів цікавитиме саме вас.



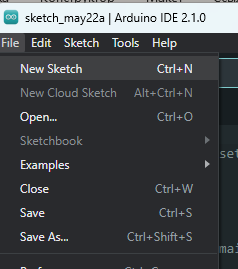
Є наявність підтримки стороніх бібліотек з наборами відповіних інструкцій для роботи з багатьма функціями та компонентами, що можуть значно розширювати почаковий функціонал використання мікроконтролерів на базі платформи Arduino.



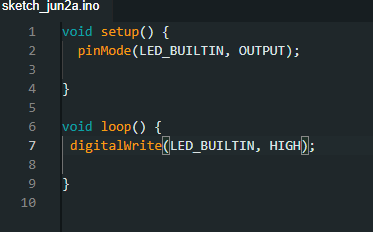
Перед початком роботи з програмним кодом впевнимося у відповіності встановлених параметрів роботи компілятора програмного середовища, для того щоб мати повноцінну можливість перевірити роботу скетчів. Для цього перейдемо на вкладку «Tools».



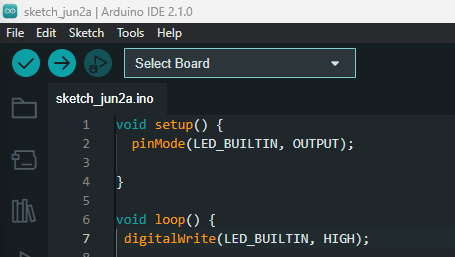
Оскільки я не маю Arduino UNO я лише можу імітувати процес роботи тому створимо новий скетч



В результаті було скетч. Лістинг програми зображений на рисунку нижче.



Виконаємо процес компілації скетч, та перевіримо роботу даного програмного коду на базі платформи Arduino.



**Завдання №4.** Згідно з заданим викладачем варіантом запишіть досконалу диз'юнктивну нормальну форму (ДДНФ) для логічної функції F чотирьох змінних, яка задана таблицею істинності (таблиця 3.1). Реалізувати програмно функцію логічного виразу. Вхідним параметром функції є однобайтова змінна. Введення виконати за допомогою програми монітора. Результат. Роботи функції відобразити на світлодіоді.

Відповідно до наданого викладачем завдання виконаємо пошук ДДНФ для логічної функції F чотирьох змінних, яка задана таблицею істинності №4.1.



*Таблиця №4.1*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| x1 | x2 | x3 | x4 | F |
| 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

– 0000

– 0010

-0100

-0110

x1x4 – 1001

x1x2x4 – 1101

x1x2x3 – 1110

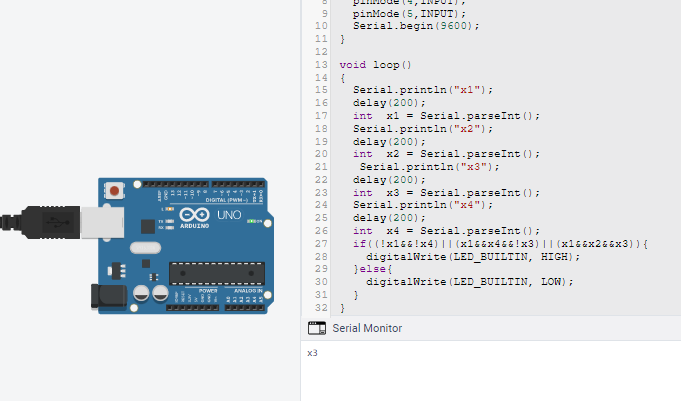
x1x2x3x4-1111

Використовуючи таблицю істинності для нашої функції виконаємо процес її мінімізації на основі метода карт Карно. Для цього виконаємо побудову карти та на її основі знайдемо ДДНФ для нашої функції.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***х1х2\х3х4*** | **00** | **01** | **11** | **10** |
| **00** | 1 | 0 | 0 | 1 |
| **01** | 1 | 0 | 0 | 1 |
| **11** | 0 | 1 | 1 | 1 |
| **10** | 0 | 1 | 0 | 0 |

Отримуємо наступну **мінімізовану функцію методом карт Карно**:

Таким чином, нами було отримано досконалу диз’юнктивну нормальну форму для нашої початкової функції. Перейдемо до процесу моделювання роботи даної функції на базі платформи Arduino. Для цього напишимо відповідну програмне реалізацію нашої функції, щоб виконати її моделювання. Лістинг програми зображено нижче.



Відкриваючи моніторинг послідовного порту та вводячи відповідні значення було отримано наступні результати перевірки роботи програми.

Лістинг коду:

// C++ code

//

void setup()

{

pinMode(LED\_BUILTIN, OUTPUT);

pinMode(2,INPUT);

pinMode(3,INPUT);

pinMode(4,INPUT);

pinMode(5,INPUT);

Serial.begin(9600);

}

void loop()

{

Serial.println("x1");

delay(200);

int x1 = Serial.parseInt();

Serial.println("x2");

delay(200);

int x2 = Serial.parseInt();

Serial.println("x3");

delay(200);

int x3 = Serial.parseInt();

Serial.println("x4");

delay(200);

int x4 = Serial.parseInt();

if((!x1&&!x4)||(x1&&x4&&!x3)||(x1&&x2&&x3)){

digitalWrite(LED\_BUILTIN, HIGH);

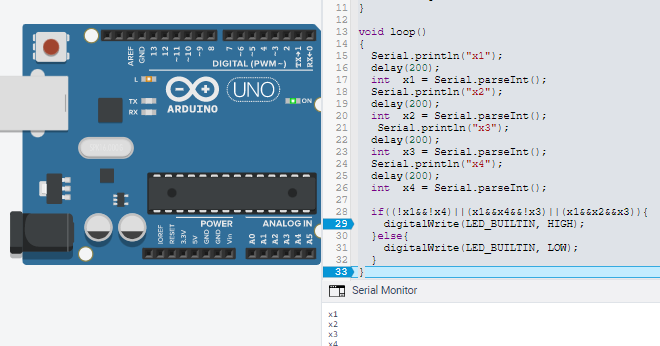
}else{

digitalWrite(LED\_BUILTIN, LOW);

}

}

Результат роботи при введені 1110:



Посилання на схему: <https://www.tinkercad.com/things/hLMCDjygvnT>

Виконавши тестування роботи побудованої нами ДДНФ, що було спрощена на основі карт Карно, помилок в її роботі виявлено не було. Побудована функція працює коректно.

**Завдання №5.** Детально пояснити схему багатофункціонального макету, схема якого зображена на рисунку №5.1, відповідно до завдання.

Почнімо виконання завдання. Для цього викладачем було надано відповідну електричну схему деякого пристрою, що зображена на рисунку №5.1.

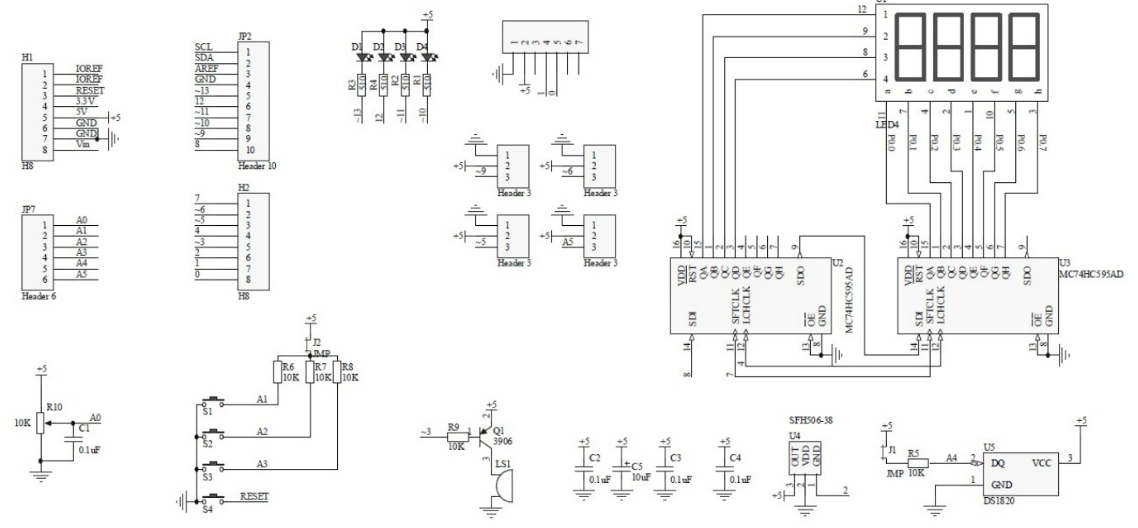
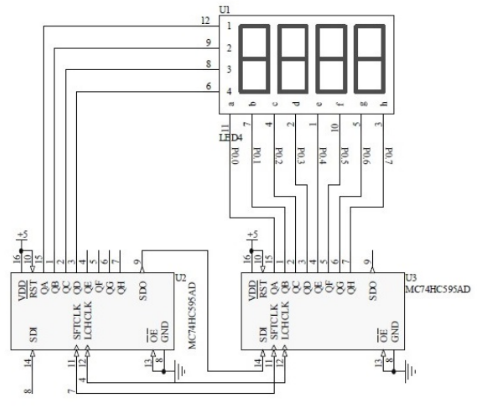


Рис. 5.2 – Схема багатофункціонального макету

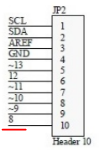
Використваши технічну документацію на мікросхему MC74HC595AD вдалося встановити, що це мікросхема восьмибітного регістра зсуву, що здатна приймати інформацію послідовному вигляді та перетворювати її у паралельний. Також вивід інформації можливий у послідовного вигляді.

Оглянемо частину схеми.

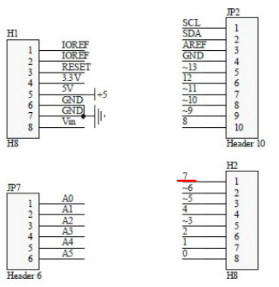


На 14 вхід мікросхеми U2 приходить послідовний код значення. Під впливом сигналів керування 11 вивід та 12 вивід в регістри виконується запис послідовної інформації. Частина бітів, застосовується для керування збіркою семисегментними індикаторам. Далі, з мікросхеми U2, з виводу 9 виконується вивід інформації на мікросхему U3, яка в свою чергу виконує керування значенням одного з чотирьох семисегментним індикатором.

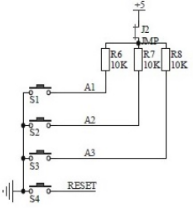
Відповідно до схеми цей послідовний сигнал буде надходити з JP2 а саме з сигналу 8.



Також було визначено, що керування мікросхемами U2 та U3 займається сигнал під номером 7, що також надходить зовні на одни з роз’ємів схеми.

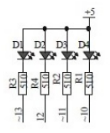


Розглянемо інші частини схеми. Оглянувши схему, було виявлено наявність перемикачів, схема яких зображена нижче на рисунку.



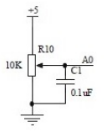
В нашому випадку кнопок на платі 4. Значення з 3 перемикачів позначено у вигляді сигналів A1, A2, A3.

Розглянемо іншу частину схеми, яка відповідає за індикацію наявності деяких сигналів.

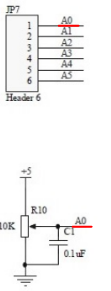
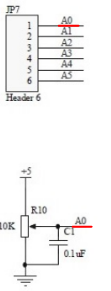


Світлодіоди D1, D2, D3 та D4. Світлодіоди підключені через навантаження, у вигляді резисторів. Керування ними відбувається на основі 4 сигналів. Дані сигнали 10, 11, 12 та 13 надходять з роз’єму JP2.

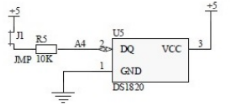
Оглянувши схему, було виявлено наявність потенціометра, який може бути застосований у вигляді елементу керування.



Значення з даного потенціометру виводиться у вигляді сигналу A0 на роз’єм JP7.



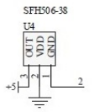
Перейдемо до огляду іншої частини схеми. Для прикладу оглянемо частину схеми з використанням комопонета під маркуванням DS1820.



Це цифровий термодатчик, який можна використовуватися як елемент керування.

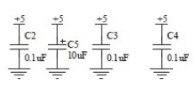
Через резистор виконається його підключення до деякої схеми, яка займається обробкою цифрового сигналу з цього датчика.

Було виявлено інфрачервоний датчкий, що представлений у вигляді інтегральної мікросхеми SFH506-38 на схемі.

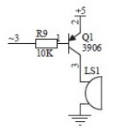


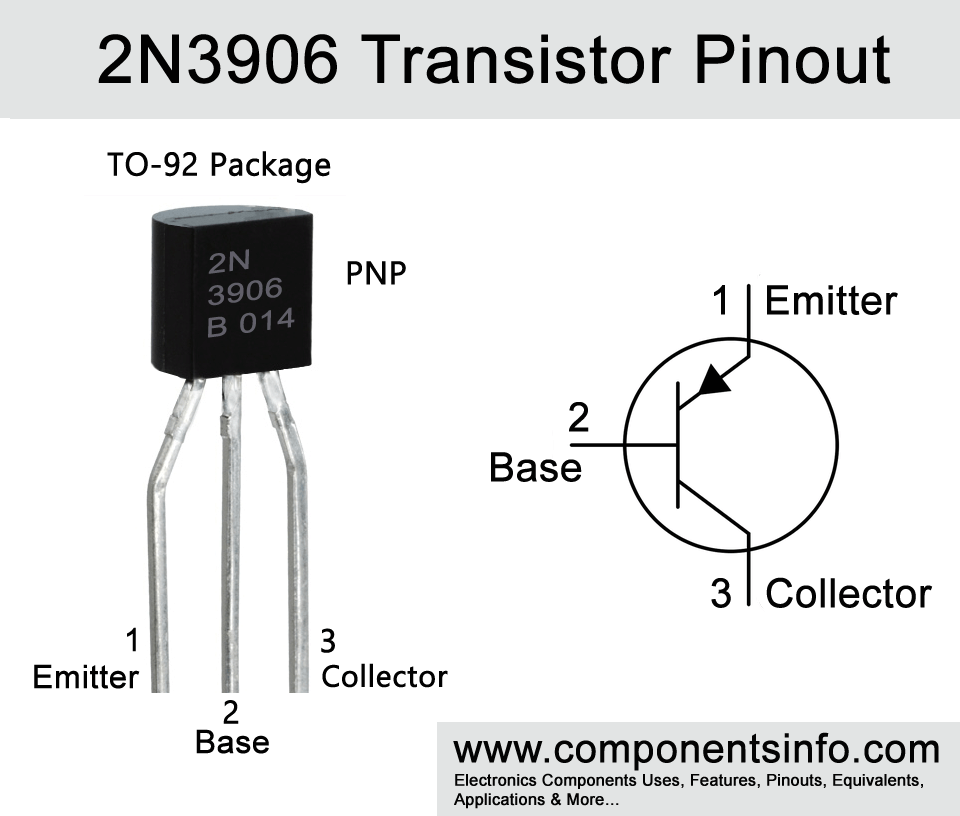
Цифровий сигнал з цього датчику подається на відповідний роз’єм на нашій схемі до H2.

Також, на схемі було виявлено конденсатори в колі живлення схеми. Конденсатори використовуються для фільтрації можливих завад.

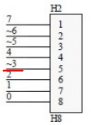


Було виявлено схему керування навантаженням через біполярний P-N-P транзистор 3906.





Керування транзистором в схемі виконується на основі знайомої нам ШИМ. Використовуючи різну сквіжність сигналу можемо виконувати контроль ступенем відкриття транзистора. Керування виконується на основі сигналу 3, що через резистор підлючений до бази транзистора. Сам сигнал керування надходить з відповідного роз’єму на схемі.



**Висновки:** В ході виконання лабораторної роботи було проведено ознайомлення з роботою програмного середовища Arduino IDE, а також безпосередньо платформою Arduino. Під час виконання роботи було проведено процес програмування мікроконтролера та тествання працеспроможності деяких програмних застосунків. Закріпив здобуті навички та вміння, щодо роботи з даною платформою, а також основними принципами програмування платформи.