

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

Факультет радіофізики, електроніки та комп'ютерних систем

Кафедра комп'ютерної інженерії

**Розробка демонстраційного макету для виміру відстані
за допомогою технології LIDAR**

Кваліфікаційна робота бакалавра
студента 4 року навчання
спеціальність: 123 «Комп'ютерна інженерія»
В'ячеслава ЗАВАДСЬКОГО

Науковий керівник
канд. фіз.-мат. наук Олександр БАУЖА,
доцент кафедри комп'ютерної інженерії

Рецензент
канд. фіз.-мат. наук Богдан СУСЬ,
доцент кафедри нанофізики конденсованих середовищ
Навчально наукового інституту високих технологій

До захисту допускаю:

Завідувач кафедрою

Юрій БОЙКО

Ухвалено на засіданні кафедри “ _____ ” _____ 2022 р., протокол № _____

Реферат

Кваліфікаційна робота бакалавра: містить сторінок, який містить: 58 сторінок, 13 рис, 3 додатків, 9 джерел .

У даній бакалаврській роботі розроблено макет для виміру відстані за допомогою лазерного датчика виміру відстані (LIDAR). Даний макет планується використовувати в лабораторній роботі з курсу «Периферійні пристрої». Також розроблена документація до лабораторної роботи, що використовує даний макет.

Розроблений макет використовується для виміру відстані на короткі дистанції, на базі апаратно-обчислювальної платформи Arduino, переваги якої розглянуті цій роботі.

Зміст	
РЕФЕРАТ	2
Перелік умовних скорочень.....	5
Мета дипломної роботи.....	6
Вступ	7
1 Мікроконтролери	8
1.0 Історія розвитку мікроконтролерів	10
1.1 Мікроконтролери AVR	12
1.1.1 Мікропроцесор	13
1.1.2 Пам'ять	14
1.1.2.1 Пам'ять програм (Flash ПЗУ)	14
1.1.2.2 Регістрова пам'ять	15
1.1.2.3 Оперативна пам'ять	15
1.1.2.4 Незалежна пам'ять даних	15
1.1.3 Периферія	16
1.1.3.1 Порти введення / виводу (І / О)	16
1.1.3.2 Переривання	16
1.1.3.3 Таймери / лічильники	17
1.1.3.4 Сторожовий таймер	17
1.1.3.5 Тактовий генератор	18
1.1.3.6 Живлення	18
1.2 Огляд платформ розробки	19
1.2.1 Мікроконтролер Arduino Uno R3	20
2 Практична частина	24
2.0 Вибір інтегрованого середовища розробки	24
2.1 Основні функції для роботи з Arduino	25
2.2 Технологія LIDAR	26
2.2.1 Принцип дії	26
2.2.2 Метод знаходження відстані	29

2.2.3 Лазерний датчик відстані GY-530 на VL53L0X	29
2.2.4 Світлодіодна матриця дисплей 8x8 на MAX7219.....	30
2.2.5 Ультразвуковий датчик відстані Arduino HC SR04.....	32
2.2.6 Похибка при вимірюванні	33
2.3 Результати роботи	33
Висновки	38
Використана література.....	39
Додаток А	40
Додаток Б	42
Додаток В	45

Перелік умовних скорочень

ЕОМ – електронно-обчислювальна машина

МК – мікроконтролер

АЦП – аналого-цифровий перетворювач

ОЗП – оперативно запам'ятовуючий пристрій

МП – мікропроцесор

EEPROM (Electrically Erasable Programmable Read-Only Memory) – постійний запам'ятовувальний пристрій, що програмується та очищається за допомогою електрики, один з видів енергонезалежної пам'яті

RAM (Random-access memory) – оперативна пам'ять

АЛП – арифметико-логічний пристрій

РЗП – регістри загального призначення

ОС – операційна система

IDE (Integrated Development Environment) – інтегроване середовище розробки

LIDAR (Light Identification, Detection and Ranging) - технологія отримання та обробки інформації

Мета дипломної роботи

Метою даної роботи є розробка демонстраційних макетів для виміру відстані які будуть використовуватися як лабораторна робота в курсі “Периферійні пристрої”.

Даний демонстраційний макет буде використовуватися в якості навчання студентів проектувати пристрої на основі платформи Arduino Uno, у середовищі Arduino IDE та керувати ними в курсі “Периферійні пристрої”.

Вступ

Перші прилади для визначення відстані з'явилися вже давно, а за допомогою технології LIDAR відомо з 20 століття. За допомогою цієї технології відстань вимірювали точно та швидко, також вимірювали відстань на великі відстані.

Поява інтегральних мікросхем, у тому числі великої кількості електронних елементів, і їх використання в комп'ютерах третього та наступних поколінь збільшилося, що спрощує процес спілкування людини з комп'ютером. Саме це максимально наблизило до об'єкта управління та контролю. Подальше зростання рівня інтеграції дозволив розмістити на кристалі мікросхеми вже не окремі прості вузли або фрагменти пристроїв EOM, а цілі EOM

Зростання рівня інтеграції призвело до створення мікроконтролерів і абсолютно нової обчислювальної техніки, яка здатна обробляти, і зберігати інформацію в одному або кількох чіпсетах. Використання мікроконтролерів в продуктах покращують такі показники, як вартість, надійність, енергоспоживання, форм-фактор тощо. Скорочують час розробки продукту та роботи його придатним для модифікації.

1. Мікроконтролери

Мікроконтролер (англ. microcontroller), або однокристална мікро ЕОМ — Виконано у вигляді мікропроцесорної системи, специфічної для мікропроцесора, що містить:

ALU(*arithmetic logic unit*) - арифметико-логічний пристрій;

RAM (*Random Access Memory*) - оперативна пам'ять;

ROM(*read-only memory*)- постійна пам'ять;

I/O ports - порти вводу/виводу;

Clock - таймер-лічильник;

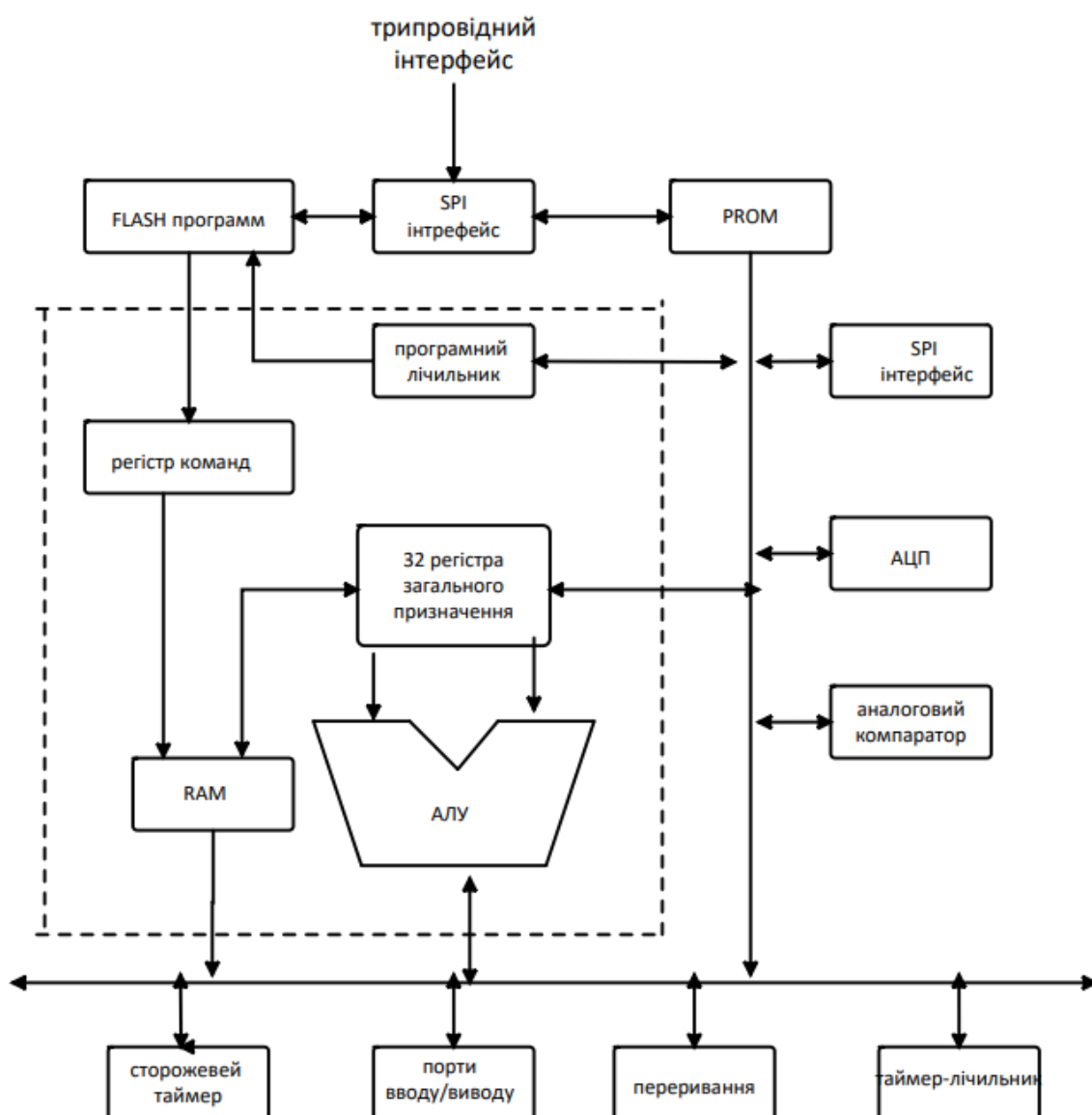


Рис. 1. Спрощена структурна схема мікроконтролера

Використовуються для керування електронними пристроями, як однокристальний комп'ютер здатний для виконання простих задач. Через те, що використовується одна мікросхема значно знижується розміри та енергоспоживання, вартість пристроїв, які були побудовані на базі мікроконтролерів.

Використовуються в продуктах у яких необхідно автоматичне управління, а саме в всій побутовій техніці, автомобілях, персональних комп'ютерах. Тобто більшість процесорів, що використовуються в приладах є мікроконтролерами.

Для розробки мікроконтролерів потрібно дотримуватися балансу, а саме: розмір, собівартість мікроконтролера, гнучкість використання та продуктивність. Залежно від галузі використання ці параметри можуть сильно відрізнятися один від одного, через це існує велике число різноманітних мікроконтролерів, які відрізняються розміром, типом вбудованої пам'яті, архітектурою процесорного модуля, набором периферійних пристроїв та типом корпусу.

У той час як 8-розрядні процесори загального призначення були повністю замінені більш ефективними моделями, 8-розрядні мікроконтролери продовжують широко використовуватися. Це відбувається тому, що існує велика кількість додатків, які не вимагають високої продуктивності, але для яких важлива низька вартість. У той же час існують мікроконтролери з більшою обчислювальною потужністю, наприклад, цифрові сигнальні процесори.

Обмеження ціни та споживання енергії також перешкоджають зростанню тактової частоти контролера. Незважаючи на те, що виробники наполегливо працюють, щоб їхня продукція працювала на високих частотах, вони також пропонують клієнтам вибір за допомогою модифікацій, розроблених для різних частот і напруг живлення.

Багато моделей мікроконтролерів використовують статичні дані для оперативної пам'яті та внутрішніх регістрів. Це дозволяє контролеру працювати на нижчій частоті без втрати даних, навіть коли годинник повністю зупинений. Як правило, передбачені різні режими енергозбереження, в яких деякі периферійні пристрої та обчислювальні модулі вимикаються.

На додаток до оперативної пам'яті мікроконтролери також можуть мати вбудовану незалежну пам'ять для зберігання програм і даних. Багато контролерів взагалі не мають шини для підключення зовнішньої пам'яті. Найдешевший тип пам'яті дозволяє робити лише одноразові записи. Такі пристрої підходять для масового виробництва в тих випадках, коли програма контролера не буде оновлюватися. Інші модифікації контролера мають можливість багаторазового перезапису зовнішньої пам'яті. На відміну від процесорів загального призначення, мікроконтролери зазвичай використовують архітектуру Гарвардського комп'ютера.[1]

1.0 Історія розвитку мікроконтролерів

Створення мікропроцесора є великим досягненням мікроелектроніки та обчислювальної техніки, оскільки МП є одночасно і елементом електроніки, і одним з основних вузлів ЕОМ, тобто процесором, призначенням якого є виконання двох основних функцій:

- 1) обробка цифрової інформації
- 2) управління процесом обробки інформації та роботою решти вузлів обчислювальної машини.

Історію МП можна простежити з 1971 року, коли американська компанія Intel створила перший мікропроцесор при розробці калькуляторів для задоволення потреб японської компанії Busicom. Це 4-розрядний МР І4004, призначений для обробки 4-розрядних двійкових даних. Він містить 2300 транзисторів, працює на тактовій частоті 108 кГц і адресує 640 байт

пам'яті зі швидкістю 0,06 MIPS. Розробниками є Тед Хофф і Фредерік Феггін[2].

У 1972 році Intel випустила 8-розрядний i8080 MP. Він має тактову частоту 2 МГц, адресує 64 КБ пам'яті та містить 6000 транзисторів. З тих пір на ринку з'явилася велика кількість типів МП від різних компаній. Треба сказати, що принципи, покладені в основу першого Intel MP, були дуже вдалим. Тож МП інших компаній схожі на клонів МП Intel. Щоб удосконалити та розвивати принципи першого MP, у майбутньому Intel зуміла залишатися попереду у своєму виробництві до сьогодні.[3].

16-розрядний мікропроцесор I8086/8088 компанії представлений на ринку з 1978 року. Це майже ідентичні МП з однаковим ланцюгом команд, загалом 156. Частота - 5 МГц, ємність 0,33 MIPS, 29000 транзисторів, пам'ять адрес 1 МБ. У 1982 році вийшов MP i80286. 134 000 транзисторів, 16 МБ адресної пам'яті, захищений режим і віртуальна пам'ять.

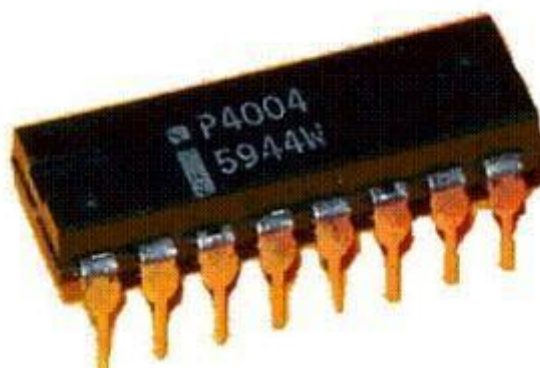


Рис. 2. Один з перших мікропроцесорів

Починаючи з 1985 року, Intel почала випускати 32-розрядний мікропроцесор i80386 (275 000 транзисторів, 4 ГБ адресації пам'яті). У 1989 році був випущений МП i80486 - 1,2 мільйона транзисторів, інтегрований кеш і математичний співпроцесор. У 1992-1993 роках Intel почала випускати 64-розрядний Pentium MP, який сьогодні випускається з різними модифікаціями. Крім розглянутих вище МП, ще один важливий напрямок

розвитку мікропроцесорної техніки, вбудовані мікроконтролери (embedded microcontrollers), вперше зосередився на виконанні функцій управління різними об'єктами. Звідси їх назва - Мікроконтролери (МК).

Їх застосування дуже різноманітне – від складної техніки (холодильники, пральні машини, кухонні комбайни тощо) до найскладніших систем керування технічними процесами та роботизованими системами. Розробка перших МК і МП належала Intel, яка випустила 8-розрядний МК 8048 загального призначення в 1976 році. Ще в 2002-2003 роках у світі вироблялося 3,2 мільярда мікросхем на рік, а процесорів для ПК - 200 мільйонів. Таким чином, у 2006 році в усьому світі було вироблено понад 4 мільярди мікросхем, а їхнє світове виробництво за останні п'ять років зросло приблизно втричі. Кількісний обсяг виробництва сучасної промисловості МК майже в 10 разів перевищує традиційний МП.

1.1 Мікроконтролери AVR

Мікроконтролери призначені для використання у вбудованих системах керування, включаючи побутову техніку. Звідси і назва «мікроконтролер». Сьогодні існує понад 500 різних типів МК, і всі вони використовуються. З них найпоширенішим є сімейство 8-розрядних мікроконтролерів AVR від Atmel. Мікроконтролери AVR мають гарвардську архітектуру (дані та оператори зберігаються окремо) і систему команд, близьку до RISC (обчислювання зі скороченим набором команд). Мікроконтролери AVR має: швидкий процесор RISC, два типи енергонезалежної пам'яті (пам'ять флеш-програм і пам'ять даних EEPROM), оперативну пам'ять, порти вводу-виводу та різні схеми периферійного інтерфейсу[3]

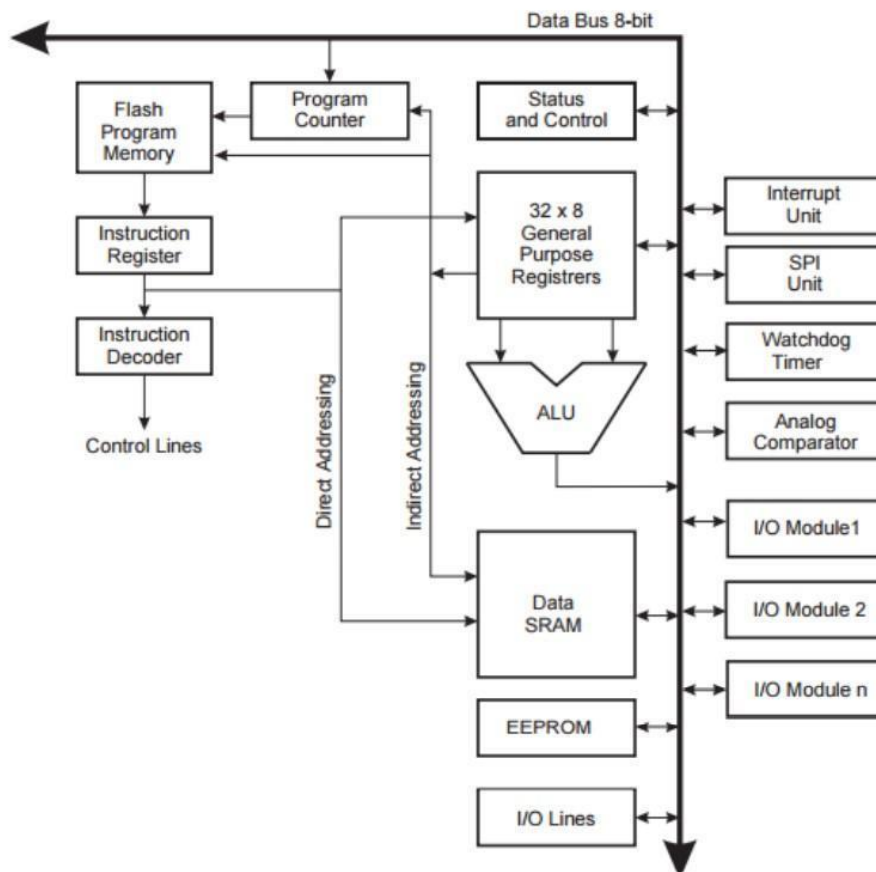


Рис.3. Архітектура мікроконтролерів AVR

1.1.1 Мікропроцесор

Основною частиною мікроконтролера AVR є 8-розрядне ядро мікропроцесора або ЦП побудований на принципах архітектури RISC. Основою цього модуля є арифметичний і логічний прилад. На основі сигналу системного тактового сигналу з пам'яті програм вибирається наступна команда відповідно до лічильника команд (Program Counter - PC) і виконується АЛП. Існує всього 32 регістри загального призначення, і вони мають байтовий формат, тобто кожен регістр складається з 8 біт. РЗП розташований на початку адресного простору RAM, але не є його фізичною частиною. Тому доступ до них можна отримати двома способами (як до реєстрів та як до пам'яті). Це рішення є особливістю AVR, що підвищує продуктивність і ефективність мікроконтролера.

Різниця між регістрами і оперативною пам'яттю полягає в тому, що перший може виконувати будь-які операції (арифметичні, логічні, бітові), другий може записувати дані лише з регістрів[4].

1.1.2 Пам'ять

Мікроконтролери AVR використовують архітектуру Гарварда, яка розділяє не тільки адресний простір пам'яті програм і пам'яті даних, а й їх шини доступу. Кожна область пам'яті даних (ОЗП і EEPROM) також розташована в своєму адресному просторі.

1.1.2.1 Пам'ять програм (Flash ROM)

Пам'ять програм призначена для зберігання командних послідовностей, які керують роботою мікроконтролера, і має 16-розрядну структуру. Усі AVR мають флеш-програми різного розміру - від 1 до 256 КБ. Її головна перевага полягає в тому, що він побудований за принципом електронного перепрограмування, що дозволяє багаторазово стирати і записувати інформацію. Програми зберігаються у Flash-пам'яті AVR через загальний програматор і інтерфейс SPI, в тому числі безпосередньо на монтажній платі. За винятком Tiny11 і Tiny28, всі мікроконтролери AVR можуть бути запрограмовані в ланцюзі (функція ISP) через комунікаційний інтерфейс SPI.

Усі мікроконтролери серії Mega здатні до самопрограмування, тобто самостійно змінювати вміст своєї програмної пам'яті. Ця функція дозволяє створювати дуже гнучкі системи, поверх яких алгоритм буде змінювати сам мікроконтролер на основі будь-яких внутрішніх умов або зовнішніх подій. Гарантована кількість циклів для накладення перезапису в мікроконтролерах AVR другого покоління становить не менше 10 000 циклів із типовим значенням 100 000 циклів. (Офіційна технічна документація від Atmel Corp. показує значення 10000 циклів.)[4]

1.1.2.2 Регістрова пам'ять

Регістрова пам'ять включає 32 реєстри загального призначення об'єднаних у файл, і службові реєстри введення / виводу. І ті й інші розташовані в адресному просторі ОЗП, але не є частиною цього адресного простору. У області реєстрів вводу-виводу є різноманітні службові реєстри (керуючі реєстри для мікроконтролерів, реєстри стану тощо), а також реєстри керування периферійними пристроями, що входять до складу мікроконтролера. По суті, управління мікроконтролером полягає в управлінні цими реєстрами.

1.1.2.3 Оперативна пам'ять

Внутрішня пам'ять має байтовий формат і використовується для оперативного зберігання даних. Обсяг оперативної пам'яті на різних мікросхемах може варіюватися від 64 байт до 4 КБ. Кількість циклів читання та запису в оперативній пам'яті не обмежена, але при відключенні напруги вся інформація втрачається. До деяких мікроконтролерів можна підключити до 64 КБ зовнішньої статичної ОЗП[4].

1.1.2.4 Незалежна пам'ять даних

Пам'ять EEPROM використовується для тривалого зберігання різної інформації, яка може змінюватися під час роботи системи мікроконтролера. Усі AVR мають блок від 64 байт до 4 КБ енергонезалежної електрично перезаписуваної пам'яті даних EEPROM. Цей тип пам'яті доступний безпосередньо під час виконання програми мікроконтролера, що зручно для зберігання проміжних даних, різних констант, коефіцієнтів, порядкових номерів, ключів. EEPROM можна завантажити ззовні через інтерфейс SPI і звичайний програматор. Час стирання та запису - не менше 100 000 разів

1.1.1.3 Периферія

Периферійні пристрої мікроконтролера AVR включають: порти (від 3 до 48 ліній вводу/виводу), підтримку зовнішніх переривань, лічильник таймера, сторожовий таймер, аналоговий компаратор, 10-розрядний 8-канальний АЦП, UART, JTAG та SPI інтерфейси, скидання джерела живлення пристрою і широтно-імпульсний модулятор.

1.1.3.1 Порти введення / виводу (І / О)

Порти введення/виводу AVR мають багато незалежних ліній введення/виводу, від 3 до 53. Кожна лінія порту може бути запрограмована як вхід або вихід. Потужні вихідні драйвери забезпечують здатність передавати струм 20 мА на лінію порту з максимальним значенням 40 мА, що дозволяє, наприклад, підключатися до світлодіодів мікроконтролера та біполярних транзисторів. Загальне струмове навантаження на всі лінії порту не повинно перевищувати 80 мА (всі значення вказані для напруги живлення 5 В).

Архітектурні характеристики побудови портів введення/виведення в AVR полягає в тому, що кожен фізичний вихід має 3 біти контролю/управління, а не 2, як звичайні 8-розрядні мікроконтролери (Intel, Microchip, Motorola тощо). Це дозволяє уникнути необхідності в копії перенос вмісту в пам'ять для безпеки та швидкості мікроконтролера при роботі із зовнішніми пристроями, особливо в умовах коли є зовнішні електричні перешкоди.

1.1.3.2 Переривання

Система переривань є однією з найважливіших частин мікроконтролера. Усі мікроконтролери AVR мають багаторівневу систему переривань. Це перериває звичайний хід програми для виконання пріоритетних завдань, визначених внутрішніми або зовнішніми подіями. Для кожної такої події розробляється окрема програма, яка називається підпрограмою, яка обробляє запити на переривання, і зберігається в пам'яті програми. Коли відбувається подія переривання, мікроконтролер зберігає

вміст лічильника інструкцій, перериває виконання поточної програми центральним процесором і продовжує виконувати обробник переривань.

Після виконання підпрограми переривання попередньо збережений лічильник команд відновлюється, і процесор повертається для виконання перерваної програми.

Для кожної події можна встановити пріоритет. Концепція пріоритету означає, що виконуюча підпрограма переривання може бути перервана лише іншою подією, якщо її пріоритет вищий за поточну подію. В іншому випадку центральний процесор продовжить обробку нових подій після обробки попередньої події

1.1.3.3 Таймери / лічильники

Мікроконтролери AVR складаються з одного до чотирьох 8-розрядних або 16-розрядних таймерів/лічильників, які можуть використовуватися або як таймери від внутрішнього джерела тактової частоти, або як зовнішні лічильники подій. Їх можна використовувати для точного формування часових інтервалів, підрахунку імпульсів на виводах мікроконтролера, формування черги імпульсів і часових приймачів для послідовних каналів зв'язку. У режимі ШІМ таймер/лічильник може бути широкоімпульсним модулятором для генерування сигналу з програмованою частотою та робочим циклом. Таймер/лічильник здатний генерувати запити на переривання, перемикаючи процесор на службу подій, і позбавляє його від необхідності періодичного опитування статусу таймера. Оскільки мікроконтролери в основному використовуються в системах реального часу, таймери/лічильники є одним з найважливіших елементів.

1.1.3.4 Сторожовий таймер

Сторожовий таймер призначений для запобігання катастрофічних наслідків несподіваного збою програми. Він має власний RC-генератор, що працює на частоті 1 МГц. Що стосується основного внутрішнього RC-

генератора, то значення 1 МГц є приблизним і залежить в основному від величини і температури напруги живлення мікроконтролера.

1.1.3.5 Тактовий генератор

Тактовий генератор генерує імпульси для синхронізації всіх операцій вузол мікроконтролера. Внутрішній годинник AVR можна почати роботу від декількох джерел опорної частоти (зовнішній генератор, зовнішній кварцовий резонатор, внутрішній або зовнішній RC-ланцюг). Мінімальні дозволені частоти необмежені (можлива робота до покрокового режиму). Максимум робоча частота визначається конкретним типом мікроконтролера, і в Atmel згадується його характеристики, хоча майже всі мікроконтролер AVR із заданою робочою частотою наприклад, 10 МГц при кімнатній температурі можна легко «прискорити» до 12 МГц та вище.

1.1.3.6 Живлення

AVR працюють від напруги живлення від 1,8 до 6,0 Вольт. Споживання струму в активному режимі залежить від величини напруги живлення та частоти, на якій працює мікроконтролер, і становить менше 1 мА при 500 кГц, 5-6 мА при 5 МГц і 8-9 мА при 12 МГц. AVR можна запрограмувати на один з трьох режимів низької потужності:

1) Режим очікування (IDLE). Тільки процесор перестає працювати і записує вміст пам'яті даних, внутрішній тактовий генератор, таймер, система переривань і сторожовий таймер продовжують працювати. Споживання струму не перевищує 2,5 мА на частоті 12 МГц.

2) Вимкнуті живлення (POWER DOWN). Вміст реєстрового файлу зберігається, але внутрішній тактовий генератор зупиняється, тому всі функції припиняються, доки не буде отримано зовнішнє переривання або сигнал апаратного скидання. Споживання струму в цьому режимі становить близько 80 мкА, коли сторожовий таймер увімкнено, і менше 1 мкА, коли він вимкнений. Усі значення – напруга живлення 5 В.

3) Економічний режим (POWER SAVE). Продовжує працювати тільки генератор таймера, що забезпечує збереження тимчасової бази. Усі інші функції вимкнені.

1.2 Огляд платформ розробки

При виконанні задачі обрано мікроконтролер, як основу платформи на якій буде розроблятися макет. Він одночасно може виконувати всього одну задачу і відмінно з цим справляються, типовим представником є плата Arduino Uno.

Мікроконтролер зазвичай має такі характеристики:

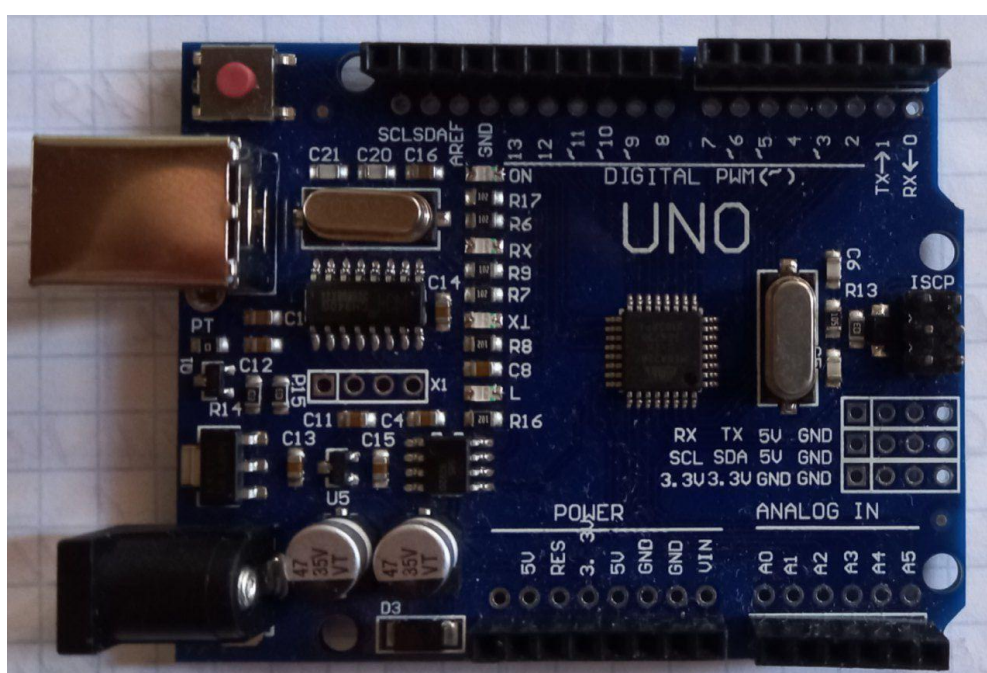
- 1) продуктивність
- 2) виконує лише одну задачу(але якісно)
- 3) вихід в мережу(при додатковому обладнанні)
- 4) довгострокова робота від батареї(споживає малу кількість електроенергії одиниці або десятки мА)
- 5) швидка реакція(завжди присутній контроль над часом і тривалістю подачі сигналів)
- 6) програмується найчастіше мовою C або C++

Платформа Arduino Uno R3:

- 1) розмір 68 x 53 x 15 м
- 2) мікроконтролер ATmega328
- 3) тактова частота 16МГц
- 4) ОЗП 2Кбайт
- 5) Flash-пам'ять 32 Кбайт
- 6) EEPROM 1 Кбайт
- 7) напруга 7 – 12 В
- 8) цифрові входи/виходи 14
- 9) аналогові входи 6
- 10) ШІМ порти 6
- 11) Інтерфейси I2C/TWI, SPI, PWM

1.2.1 Мікроконтролер Arduino Uno

Arduino Uno R3 контролер побудований на ATmega328 мікроконтролері. Платформа має 14 цифрових входів/виходів (6 з яких можна використовувати як ШІМ-виходи), 6 аналогових входів, кварцовий генератор 16 МГц, роз'єм USB, роз'єм живлення, роз'єм ICSP і кнопку скидання. Для роботи необхідно підключити платформу до комп'ютера за допомогою USB-кабелю або використовувати адаптер змінного/постійного струму або живлення від акумулятора[5].



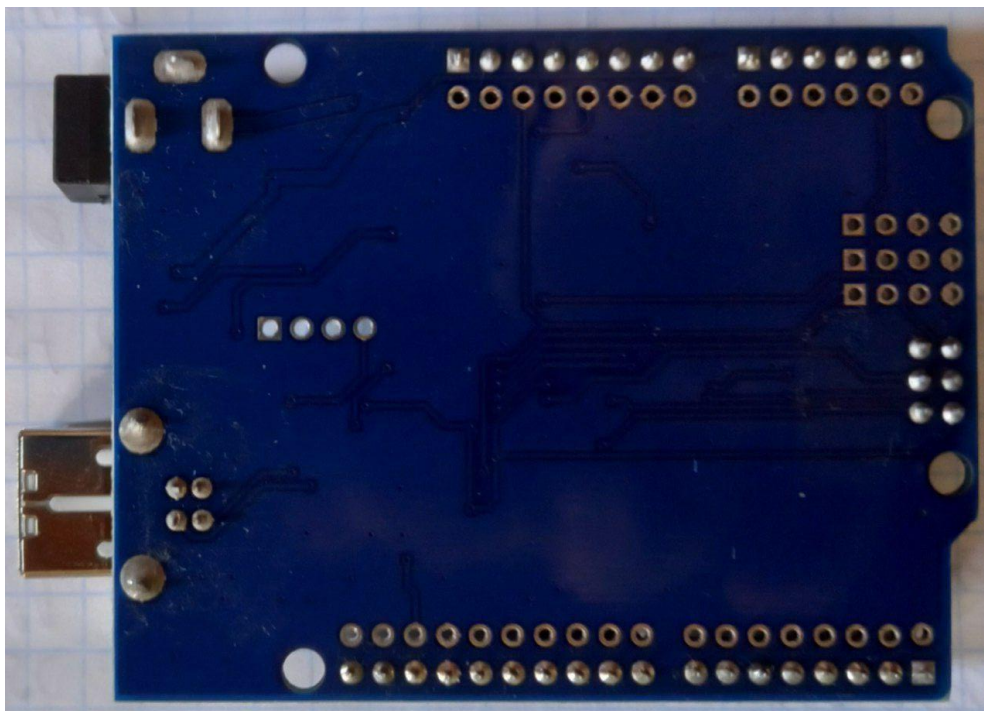


Рис. 4. Плата Arduino Uno R3

Він може житися від USB або зовнішнього живлення, вибір живлення встановлюється автоматично. Якщо останнє, то живлення може бути забезпечено акумуляторною батареєю або перетворювачем напруги змінного/постійного струму (блок живлення). Перетворювач напруги підключається через роз'єм 2,1 мм з центральним плюсовим полюсом. Провід від акумулятора підключаються до виходів Gnd і Vin роз'єму живлення.

Можлива робота від зовнішнього джерела живлення від 6 В до 20 В. Якщо напруга живлення нижче 7 В, то вихід 5 В може видавати менше 5 В, тому контролер працює нестабільно. Якщо подавати напругу вищу за 12 В регулятор може перегрітися і пошкодити плату. Діапазон від 7 В до 12 В.

VIN. Вхід призначений для живлення від зовнішнього джерела живлення (при відсутності 5 В від USB-роз'єму або іншого регульованого джерела живлення). Напруга живлення забезпечується через цей вихід.

5В. Регульоване джерело напруги для живлення мікроконтролера та компонентів на платі. Може живитися від виходу VIN через стабілізатор напруги, роз'єм USB або інше регульоване джерело напруги 5 В

3V3. Напруга на виведення 3.3 В. Максимальне споживання струму 50 мА.

GND. Виходи заземлення.

Мікроконтролер ATmega328 має 32 КБ флеш-пам'яті, з яких 0,5 КБ використовується для зберігання завантажувача, а також 2 КБ оперативної пам'яті (SRAM) і 1 КБ EEPROM. Кожен з 14 цифрових виходів Uno можна налаштувати як вхід або вихід за допомогою функцій `pinMode()`, `digitalWrite()` і `digitalRead()`. Вихід працює при напрузі 5 В. Кожен вихід має навантажувальний резистор 20-50 кОм (за замовчуванням відключений) і може пропускати до 40 мА. Деякі виходи мають спеціальні функції:

- 1) Послідовна шина: 0 (RX) і 1 (TX). Виходи використовуються для прийому (RX) і передачі (TX) даних TTL. Ці виходи підключаються до відповідних виходів мікросхеми послідовної шини ATmega8U2 USB-toTTL
- 2) Зовнішні переривання: 2 і 3. Ці виходи можуть бути налаштовані на виклик переривань з нижчими значеннями, до або після, або під час зміни
- 3) ШІМ: 3, 5, 6, 9, 10, і 11. Будь-який з виходів забезпечує ШІМ з роздільною здатністю 8 біт за допомогою функції `analogWrite`.
- 4) SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). За допомогою даних виходів здійснюється зв'язок SPI.
- 5) LED: 13. Вбудований світлодіод, підключений до цифрового виходу 13.

Також є 6 аналогових входів (позначаються A0 .. A5), кожен має АЦП з роздільною здатністю 10 біт (тобто може приймати 1024 різних значення). Як стандарт, вихідна напруга вимірюється до 5 В відносно землі, але

верхню межу можна змінити за допомогою виходу AREF і функції `AnalogReference()`.

Також встановлено кілька пристроїв для зв'язку з вашим комп'ютером, іншими пристроями Arduino або мікроконтролерами. ATmega328 підтримує послідовний інтерфейс UART TTL (5 В), виходи 0 (RX) і 1 (TX). Мікросхема ATmega8U2, встановлена на платі, завантажує інтерфейс через USB, а програма на стороні комп'ютера «спілкується» з платою через віртуальний COM-порт. Прошивка ATmega8U2 використовує стандартні драйвери USB COM, сторонні драйвери не потрібні, але для підключення Windows потрібен файл `ArduinoUNO.inf`. Послідовний монітор Arduino дозволяє надсилати та отримувати текстові дані при підключенні до платформи. Світлодіоди RX і TX на платформі будуть блимати, коли дані передаються через чіп FTDI або USB-з'єднання (але не при використанні послідовної передачі через виходи 0 і 1).[7]

2 Практична частина

2.0 Вибір інтегрованого середовища розробки

Платформа Arduino має власне середовище розробки Arduino IDE, яке є безкоштовним і повністю відкритим. Arduino IDE — це багатоплатформна програма Java, яка включає редактор коду, компілятор і модуль передачі мікропрограми. Середовище розробки засноване на мові програмування Processing, подібній до мови Wiring. Власне кажучи, це C++, доповнений деякими бібліотеками. Програма обробляється за допомогою препроцесора, а потім компілюється за допомогою AVR-GCC[11].



Рис. 5. Інтегроване середовище розробки Arduino

Програми Arduino написані мовою програмування C або C++, а не Scratch. Середовище розробки Arduino постачається з програмною бібліотекою під назвою Wiring, похідною від проекту Wiring, що полегшує багато стандартних операцій введення-виводу. Користувачам потрібно лише визначити дві функції, щоб створити програму, яка виконується в циклі:

- 1) `setup ()`: функція виконується лише раз при старті програми і дозволяє задати початкові параметри;
- 2) `loop ()`: функція виконується періодично доки плата не буде вимкнено.

2.1 Основні функції для роботи з Arduino

Платформа Arduino має велику кількість функцій для роботи з підключеними пристроями. Ніжче описані лише ті, які були використані у даній роботі[11]:

- 1) `pinMode(pin, mode)` - встановлює режим роботи даного вхід / виходу (`pin`) як входу або як виходу. `mode`: режим одне з двох значення - `INPUT` і `OUTPUT`, встановлює на вхід або вихід відповідно.
- 2) `analogWrite(pin, value)` - видає аналогову величину (ШІМ хвилю) на порт вхід / виходу. Ця функція корисна для керування яскравістю підключених світлодіодів або швидкістю двигунів. Після виклику `analogWrite()` на виході буде згенерована постійна прямокутна хвиля заданої ширини імпульсу до наступного виклику `analogWrite()`. `value` – Період робочого циклу для значення від 0 (повністю вимкнено) до 255 (сигнальна константа).
- 3) `analogRead(pin)` - Функція зчитує значення із зазначеного аналогового входу. Більшість плат Arduino мають 6 каналів (8 в платах Mini і Nano, 16 у платах Mega) і 10-розрядний аналого-цифровий перетворювач (АЦП). Напруга, що подається на аналоговий вхід, зазвичай від 0 до 5 вольт, буде перетворено на значення від 0 до 1023 з 1024 кроками по 0,0049 вольт.
- 4) `Serial.Begin(speed)` - запускає послідовне з'єднання та встановить бітрейт /с. Для зв'язку з комп'ютером використовуються наступні значення: 300, 1200, 2400, 4800, 9600, 14400, 19200, 28800, 38400, 57600 або 115200. При підключенні через порти введення/виводу 0 і

І інші значення можуть використовувати швидкість, необхідну для пристрою, з яким буде здійснюватися обмін даними.[8]

2.1 Технологія LIDAR

LIDAR (Транслітерація LIDAR English Light Identification, Detection and Ranging) - технологія, яка використовує активні оптичні системи для отримання та обробки інформації про віддалені об'єкти, яка використовує явище відбиття та розсіювання світла в прозорих і напівпрозорих середовищах.

LIDAR, як прилад, є принаймні активним оптичним далекоміром. Сканери в системі машинного зору формують 2D або 3D зображення навколишнього простору. «Атмосферний» LIDAR може не тільки визначати відстань до непрозорих цілей, що відбивають світло, а й аналізувати властивості прозорих середовищ, які розсіюють світло. Різні атмосфери є доплерівськими пілотами, які визначають напрямок і швидкість повітряного потоку в різних шарах атмосфери.[6]

2.2.1 Принцип дії

Імпульсне лазерне випромінювання направляється в атмосферу. Потім, розсіяне в протилежному напрямку атмосферою, випромінювання збирається телескопами і реєструється фотодетекторами, які потім оцифровують сигнал.

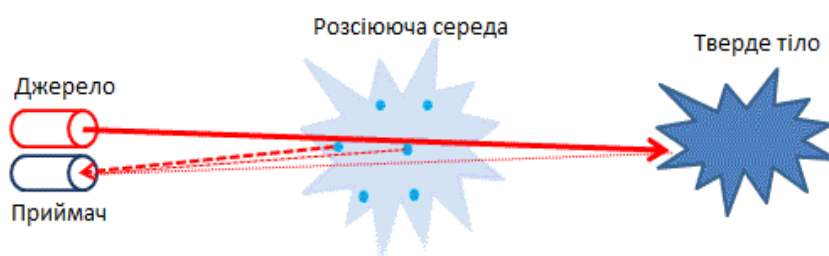


Рис. 6. Принцип дії LIDAR

LIDAR випускає короткі імпульси лазерного випромінювання на об'єкти (поверхні) з частотою до 150 000 імпульсів за секунду. Датчик на пристрої вимірює час, необхідний для повернення імпульсу. Світло рухається з постійною і відомою швидкістю, тому LiDAR може обчислити відстань від об'єкта з високою точністю.

Джерело LIDAR — це лазер і оптична система, що використовуються для формування вихідного лазерного променя, тобто. для контролю розміру вихідної плями та кута розбіжності променя. У більшості конструкцій передавач являє собою лазер, який виробляє короткі імпульси світла з високою миттєвою потужністю.

Частота імпульсів або частота модуляції вибирається таким чином, щоб пауза між двома послідовними імпульсами була не меншою за час відгуку цілі (яка фізично може бути далі, ніж розрахункова дальність дії пристрою). Вибір довжини хвилі залежить від функції лазера та вимог щодо безпеки та конфіденційності обладнання. Найчастіше використовувани Nd:YAG лазери та довжини хвилі:

1550 нм - інфрачервоне випромінювання, невидиме для людського ока або типових приладів нічного бачення. Око не може сфокусувати ці хвилі на поверхні сітківки, тому поріг травмування для хвиль 1550 нм набагато вищий, ніж для коротких хвиль. Однак ризик пошкодження очей насправді вищий, ніж у випромінювачів видимого світла – оскільки око не реагує на інфрачервоне (інфрачервоне) випромінювання, природний захисний рефлекс людини не спрацьовує.

1064 нм - ближнє інфрачервоне випромінювання від неодимових та ітербієвих лазерів, невидиме неозброєним оком, але помітне обладнанням нічного бачення.

532 нм - випромінювання зеленого неодимового лазера, яке ефективно «проникає» великі обсяги води 355 нм - ближнє ультрафіолетове випромінювання

Приймальна частина складається з лінз (телескопів), спектральних та/або просторових фільтрів, поляризаційних елементів та фотоприймачів. Відбите і розсіяне від досліджуваного об'єкта випромінювання концентрується приймальною оптикою (телескопами), а потім пропускається через аналізатор спектра. Пристрій використовується для вибору інтервалу довжин хвиль, на якому проводяться спостереження, тим самим відсікаючи фонове випромінювання на інших довжинах хвиль. Аналізатор може бути складним, ретельно налаштованим монохроматором або поліхроматором, або набором вузькосмугових фільтрів, у тому числі фільтрів випромінювання на довжині хвилі випромінювача лазера.

Блоки передавача і приймача можуть бути розташовані далеко один від одного або створені як один блок, що є поширеним останнім часом. Осі передавача і приймача можуть бути з'єднані (коаксіальні схеми) або рознесені (біаксіальні схеми).

Система управління виконує такі завдання:

Управління режимом LIDAR;

Регулювання частоти лазерного зонда;

Вимірювання енергії випромінювання вихідних і вхідних двочастотних лазерних променях на двох частотах;

Управління системами лідерства для суб'єктів дослідження.

2.2.2 Метод знаходження відстані

Відстань визначається за допомогою часу, який імпульс проходить від джерела до об'єкта, а потім надходить до приймача

$$D = 0.5 * c * t$$

c - швидкість світла

t - час за який імпульс пройшов відстань від джерела до об'єкта, та від об'єкта до приймача

D - відстань до об'єкта

Час проходження світла до об'єкта

$$t = (\varphi/2\pi) * T$$

φ - фазовий зсув

T - період світлових коливань

2.2.3 Лазерний датчик відстані GY-530 на VL53L0X

Модуль GY-530 оснащений вбудованим ІЧ-фільтром. Його випромінювання повністю невидиме для людського ока, забезпечує більші відстані вимірювання, менш чутливе до зовнішнього рівня світла, більш стійке до перехресних перешкод, викликаних скляними поверхнями, а результати вимірювань не залежать від відбивної здатності поверхні. особливість: Надійна робота в умовах високої інтенсивності зовнішнього інфрачервоного випромінювання. Додаткова оптика не потрібна.



Рис.7. GY-530 на VL53L0X

Характеристики:

- 1) Відстань до 2 м
- 2) Діапазон температур -20 до +70 С
- 3) Напруга 2,8- 5 В
- 4) Струм 19 мА
- 5) Довжина хвилі 940 нм

2.2.4 Світлодіодна матриця дисплей 8x8 на MAX7219

Модуль являє собою точково-матричну світлодіодну матрицю розміром 8 x 8 (64 світлодіоди). Точка на платі модуля блимає червоним. Цей модуль можна використовувати для проектування годинника або інших проектів візуалізації на платформі Arduino. Світлодіодна матрична плата розроблена на основі DIP мікросхеми управління MAX7219.

Драйвер MAX7219 окремо керує 64 світлодіодами на платі із загальним катодом. Модуль має послідовний 3-провідний інтерфейс. Передача даних на мікросхему модуля здійснюється через інтерфейс SPI, який підключається через три контакти. Контакти DIN (введення даних), CS (вибір модуля) і CLK (сигнал тактового сигналу) підключені до вільних

цифрових портів на платі Arduino. Живлення для матриці подається на клеми VCC і GND 5 В.

Виходи модулів аналогічні входам і використовуються для послідовного підключення таких модулів. Крім того, оскільки ширина плати відповідає розміру матриці, модулі можна об'єднувати в ряд, тим самим збільшуючи кількість відображуваної інформації. Щоб продовжити використання світлодіодної матриці MAX7219, вам потрібно буде завантажити бібліотеку LedMatrix та встановити її самостійно.

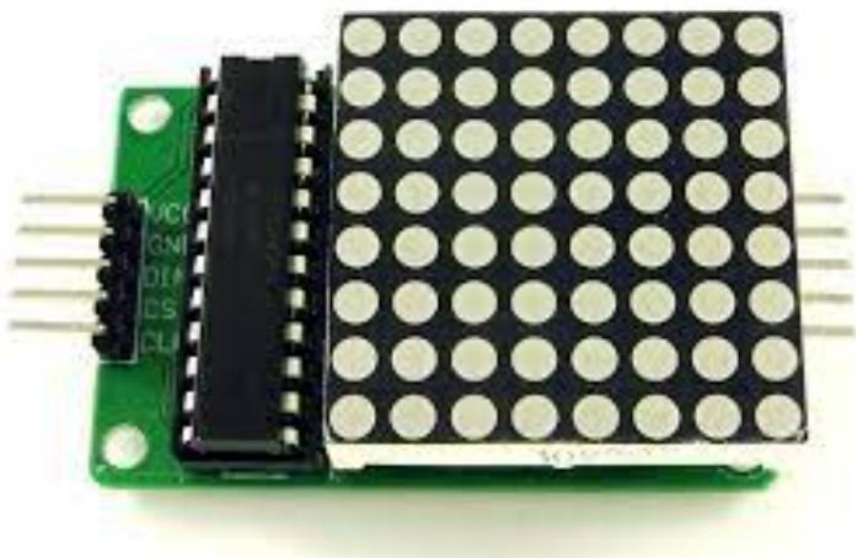


Рис. 8 Світлодіодна матриця дисплей 8x8 на MAX7219

Характеристики:

- 1) Напруга живлення, 4 ... 5,5
- 2) Споживаний струм, ма 150
- 3) Частота тактування, МГц 10
- 4) Мікросхема MAX7219
- 5) Кількість точок 64
- 6) Колір світіння червоний
- 7) Розміри модуля, мм 50 x 32 x 15

2.2.5 Ультразвуковий датчик відстані Arduino HC SR04

Ультразвуковий датчик відстані HC-SR04 - призначений для вимірювання відстаней від 2 до 400 см з межею точності 3 мм. Модуль включає в себе ультразвуковий передавач, приймач і блок управління. На показання датчика практично не впливають сонячне випромінювання та електромагнітний шум. На передній панелі HC-SR04 є два ультразвукових датчика, перший позначений T (передавач) – це ультразвуковий передавач (TCT40-16T), а другий, позначений R (прийом), – це ультразвуковий приймач відбитої хвилі (TCT40-16R). Центрується на вихідному кварцовому генераторі на 27 МГц.



Рис. 9. ультразвуковий датчик відстані Arduino HC SR04

Характеристики:

- 1) діапазон вимірювання відстані 0,03м-4м
- 2) частота ультразвуку 40kHz
- 3) кут зору 30град.
- 4) інтерфейс 2 логічні TTL лінії
- 5) вихідна інформація імпульс 0,15..25mS
- 6) напруга живлення "Vcc" 5V
- 7) струм споживання в активному режимі 15mA

8) розмір модуля 45x20x15mm

2.2.6 Похибка при вимірюванні

При вимірюванні відстані можна отримати різні результати які не будуть сходиться з істинними, а саме таке може відбутися через кут на який датчик посилає промінь (Рис. 9). Для визначення такого кута можна скористатись співвідношенням $A = \arccos L/OB$.

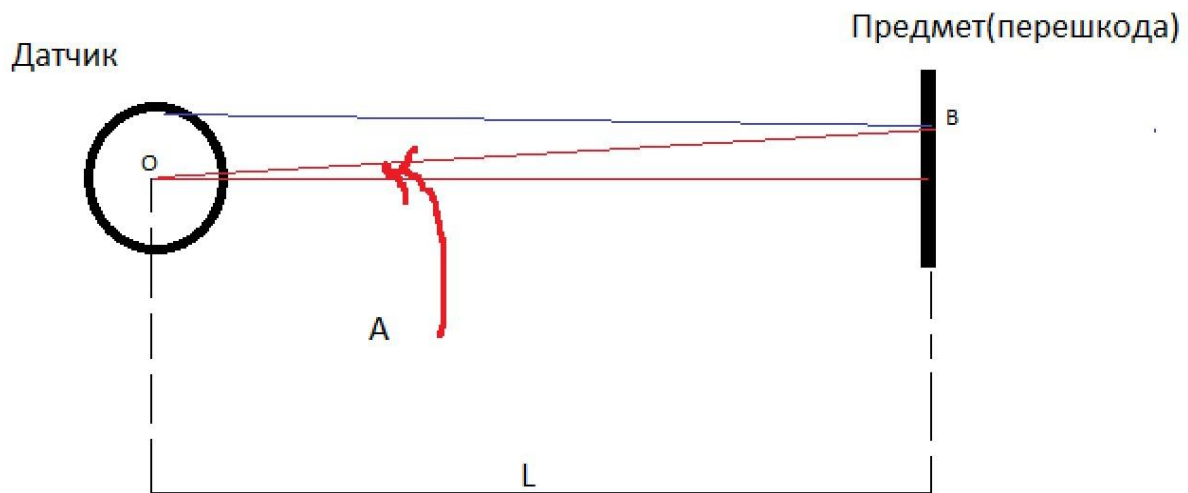


Рис. 10. Кут

2.3 Результати роботи

У результаті виконання роботи зібрано два макети пристрою. Для вимірювання відстані.

У якості мікроконтролера для макету передачі даних був обраний Arduino Uno R3, враховуючи його наступні переваги:

- 1) Програмується безпосереднім підключенням через інтерфейс USB, не потрібен окремий програматор
- 2) Апаратно-програмна платформа Arduino є проектом з відкритим вихідним кодом
- 3) Має кросплатформне середовище розробки

Порівняно низька вартість.

Перший макет складається з:

- 1) Мікроконтролер Arduino Uno R3
- 2) Набір проводів для з'єднання компонентів
- 3) Лазерний датчик відстані GY-530 на VL53L0X
- 4) Світлодіодна матриця дисплей 8x8 на MAX7219

Схема даного макету представлена на рис. 11.

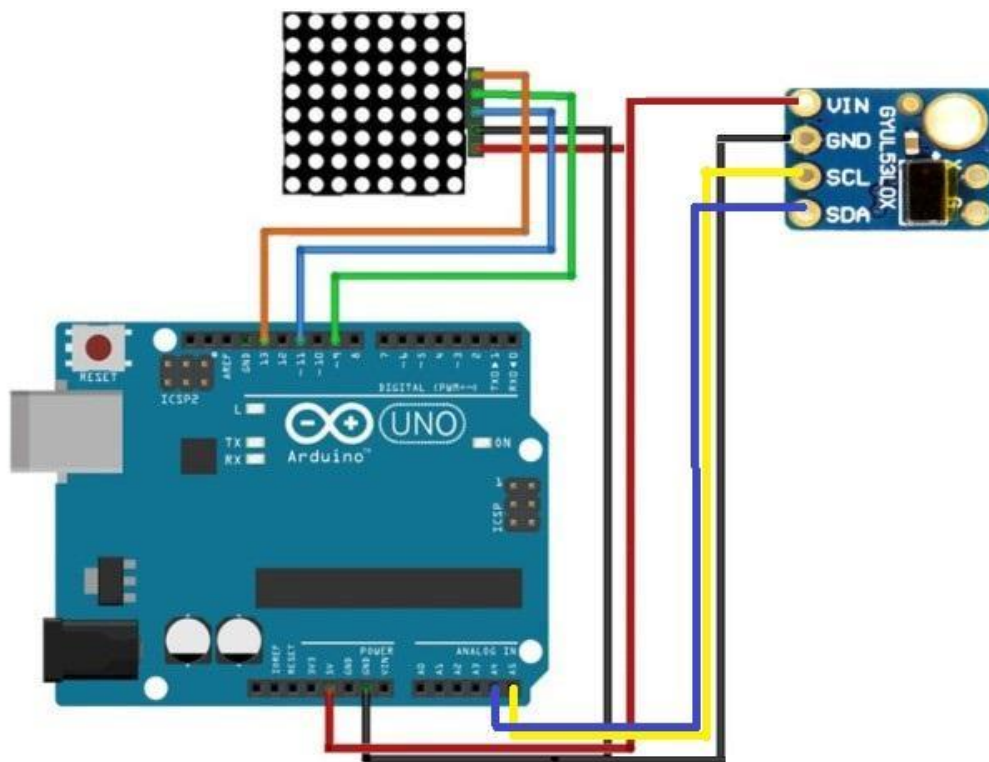


Рис. 11. Макет схеми лазерного датчика і світлодіодною матрицею

Другий макет складається з:

- 1) Мікроконтролер Arduino Uno R3
- 2) Набір проводів для з'єднання компонентів
- 3) Лазерний датчик відстані GY-530 на VL53L0X
- 4) Ультразвуковий датчик відстані Arduino HC SR04

Схема данного макету представлена на рис. 12.

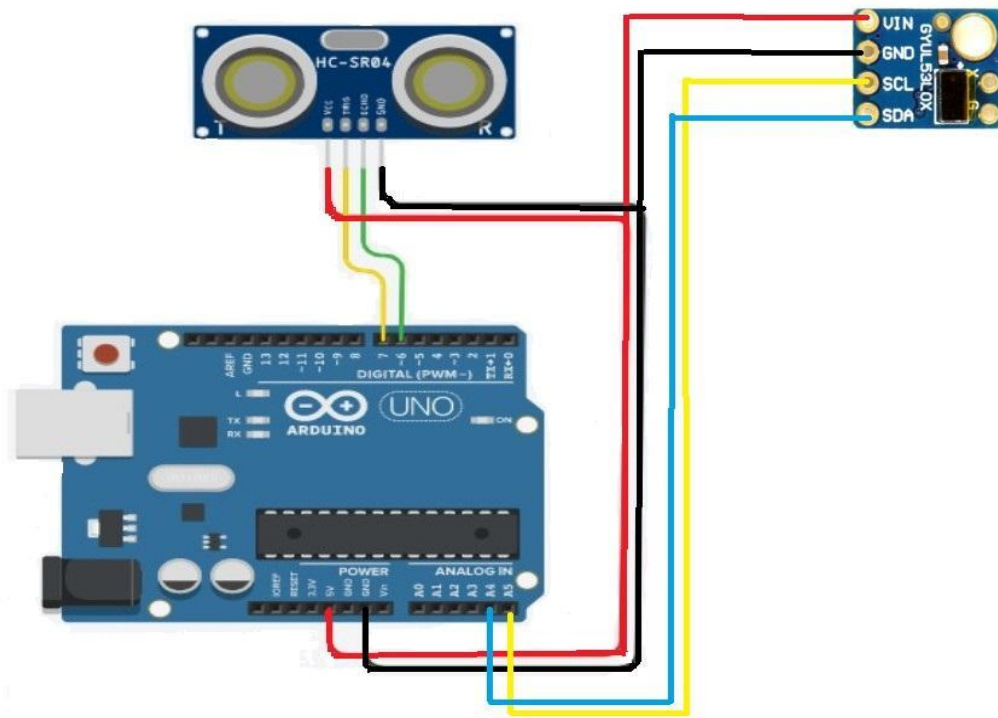


Рис. 12. Макет схеми з ультразвуковим та лазерним датчиком

Код програми для зняття даних про відстань представлена в додатку А. Розроблено опис до виконання лабораторної роботи «Датчики відстані» Додаток В що включає:

- 1) Назву - «Датчики відстані»
- 2) Тему - розробка макету для виміру відстані на платформі Arduino Uno R3
- 3) Мету - навчитися проектувати пристрої на основі платформи Arduino Uno R3 у середовищі Arduino IDE та керувати ними
- 4) Теоретичні відомості з пункта
- 5) Хід роботи
- 6) Індивідуальні завдання з варіанти
- 7) Контрольні запитання

На рис. 12 продемонстровано результат виконання лабораторної роботи варіант 2 результат вимірювання похибки для лазерного датчика. Також наведені результати розрахунку відстані для перешкоди на 30см Додаток Б. З цих результатів виведено середню відстань яку демонструє датчик, максимальну та кут А.

Середня відстань становить 3083,5мм, максимальна 3150мм, а кут 17.23263° .

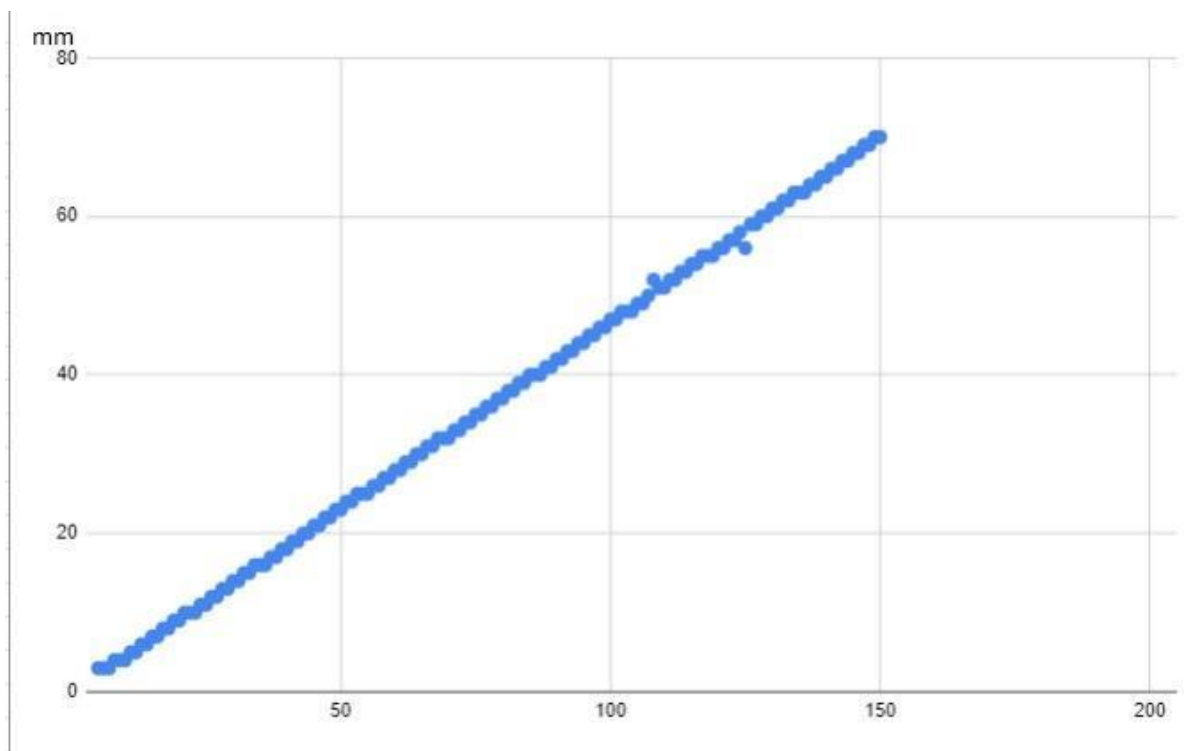


Рис. 13. Похибка вимірювання

Студент при виконанні лабораторної роботи ознайомиться з платформою Arduino та інтегрованим середовищем розробки Arduino IDE, датчиками вимірювання відстані, а саме з лазерним та ультразвуковим датчиками вимірювання відстані.

У результаті виконання лабораторної роботи студент навчиться робити макет для виміру відстані на основі платформи Arduino Uno R3 та

датчиків відстані, програмувати в Arduino IDE з використанням існуючих бібліотек.

Студент отримає навички роботи з платою Arduino Uno R3, датчиками відстані, інтегрованим середовищем Arduino IDE.

Висновок

У ході виконання кваліфікаційної роботи розроблено макет для виміру відстані та документацію до лабораторної роботи «Датчики відстані», що буде використовуватись в курсі «Периферійні пристрої»

Даний макет вимірює відстань від 5см до 1.5м з приблизною похибкою у 5%. У середньому відстань що визначається за допомогою лазерного датчика на 3% більша за істину відстань до об'єкта.

У результаті виконання лабораторної роботи студент навчиться робити макет для виміру відстані на основі платформи Arduino Uno R3 та датчиків відстані, програмувати в Arduino IDE використовуючи існуючі бібліотеки.

Використана література

1. Основи електроніки та мікропроцесорної техніки [Електронний ресурс]. - Режим доступу: URL: <http://repository.kpi.kharkov.ua/handle/KhPI-Press/18457>
2. Ю.В. Новиков и П.К. Скоробогатов. Основи мікропроцесорної техніки. / Иуит; Бинок, 2009
3. Програмування мовою Arduino [Електронний ресурс]. - Режим доступу: URL:<https://doc.arduino.ua/ru/prog/>
4. Atmel - Atmel AVR 8-bit and 32-bit Microcontrollers [Електронний ресурс]. - Режим доступу: URL: <http://atmel.com/products/microcontrollers/avr/>
5. Електронний магазин радіодеталей [Електронний ресурс]. - Режим доступу: URL: <http://arduino-ua.com/>
6. LiDAR Technologies and Systems [Електронний ресурс]. - Режим доступу: URL: <https://www.spiedigitallibrary.org/ebooks/PM/LiDAR-Technologies-and-Systems/eISBN-9781510625402/10.1117/3.2518254?SSO=1>
7. Simon Monk. Getting Started with Sketches. New York. 2012 by The McGraw-Hill Companies. ISBN: 978-0-07-178423-8. MHID: 0-07-178423-3
8. Massimo Banzi co-founder of Arduino & Michael Shiloh. Make: Getting Started with Arduino, 3rd Edition. Published by Maker Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472. 2015. ISBN: 978-1-4493-6333-8

Додаток А

Програмний код для додатку IDE:

```
#include "Arduino.h"
#include "Wire.h"
#include "VL53L0X.h"
//бібліотека для роботи з датчиком VL53L0, на основі цього датчика і мій
датчик GY-530
#include <iarduino_I2C_Matrix_8x8.h>
//бібліотека для роботи за LED матрицею
iarduino_I2C_Matrix_8x8 disp(0x0D);
//адрес підключення LED матриці 8x8
void setup() {

  Serial.begin(9600);
  //приєднатися до шини i2c саме до неї під'єднувався

  Wire.begin();

  sensor.begin(0x52);
  sensor.setMode(sensor.eContinuous,sensor.eHigh);
  //Лазерний далекомір починає працювати

  sensor.start();
}

void loop()
{
  //Отримайте відстань

  Serial.print("Distance: ");
  Serial.println(sensor.getDistance());
```

```
// вивід інформації на командний рядок
//Затримка додається для демонстрації ефекту
//це не вплине на точність вимірювання
disp.print(sensor.getDistance());
//вивід відстані на матрицю 8x8
disp.autoScroll(250);
//для прокрутки значення відстані
delay(1000);
// Затримка при виводі результату в мс
}
```

Дані с лазерного датчика які було зібрано з затримкою 1 секунда(результат виміру відстані надається протягом 1 секунди).

3141 3034 3102 3055 3080 3019 3071 3053 3057 3013 3118 3136 3003 3036
3020 3140 3059 3011 3103 3021 3040 3126 3050 3127 3143 30643144 3089
3027 3025 3091 3085 3138 3051 3008 3076 3076 3038 3042 3106 3139 3036
3128 3115 3112 3070 3028 3058 3090 3011 3019 3021 3142 3087 3139 3080
3079 3065 3101 3035 3072 3012 3047 3106 3122 3026 3033 3015 3073 3136
3012 3093 3086 3014 3038 3068 3143 3076 3019 3021 3126 31073149 3073
3040 3081 3104 3001 3006 3044 3133 3023 3125 3031 3060 3139 3054 3115
3043 3102 3124 3098 3041 3085 3082 3054 3124 3073 3058 3139 3117 3118
3049 3080 3019 3081 3067 3012 3090 3108 3064 3123 3119 3067 3082 3083
3030 3111 3013 3068 3035 3049 3120 3069 3053 3114 3031 3025 3080 3033
3080 3097 3035 3007 3117 3050 3077 3005 3000 3110 3075 30043135 3078
3079 3101 3095 3084 3067 3119 3014 3143 3041 3024 3011 3045 3015 3081
3049 3019 3081 3143 3117 3065 3137 3046 3008 3047 3092 3126 3131 3011
3010 3098 3006 3064 3124 3137 3138 3057 3036 3092 3078 3123 3091 3133
3123 3107 3147 3059 3129 3045 3027 3086 3068 3145 3086 3092 3081 3107
3102 3069 3038 3087 3044 3109 3141 31453058 3091 3021 3060 3150 3094
3057 3088 3060 3103 3078 3137 3137 3099 3101 3129 3018 3025 3138 3102
3120 3112 3061 3007 3054 3126 3055 3148 3101 3063 3113 3053 3021 3128
3008 3113 3144 3092 3070 3083 3028 3032 3052 3069 3138 3047 3005 3033
3064 3107 3081 3104 3020 3052 3068 3148 3016 3078 30953150 3068 3143
3060 3051 3104 3004 3132 3041 3018 3141 3103 3079 3132 3056 3062 3071
3137 3116 3008 3124 3038 3067 3090 3039 3126 3128 3125 3118 3043 3124
3124 3013 3065 3033 3083 3096 3111 3112 3003 3114 3072 3032 3016 3039
3149 3042 3134 3123 3038 3044 3029 3015 3046 31493062 3139 3019 3075
3013 3111 3144 3061 3051 3008 30703074 3142 3018 3065 3048 3091 3070
3142 3046 3053 3060 3108 3060 3014 3073 3095 3120 3119 3024 3124 3118
3105 3084 3012 3074 3002 3100 3033 3079 3022 3136 3069 3116 3042 3086

3146 3125 3025 3048 3143 3045 3037 3105 3094 3139 3138 3041 3007 3140
3069 3021 3009 3137 3124 3146 3145 3070 3011 3139 3150 3098 3082 3111
3024 3079 3049 3065 3008 3079 3127 3008 3121 3003 3137 3076 3098 3124
3044 3068 3125 3005 3150 3098 3063 3146 3134 3009 3101 3108 3101 3047
3100 3032 3027 3088 3121 3101 3016 3138 3082 3066 3093 3093 3108 3111
3139 3095 3001 3120 3053 3068 3054 3073 3004 3019 3118 3111 3078 31073
082 3004 3007 3026 3034 3117 3125 3019 3066 3063 3116 3034 3116 3007
3054 3067 3044 3048 3037 3102 3079 3035 3049 3075 3031 3026 3003 3109
3106 3060 3141 3108 3080 3034 3091 3102 3113 3043 3056 3134 3099 3017
3006 3097 3132 3105 3133 3127 3021 3038 3059 3009 3035 3118 3013 3105
3087 3143 3002 3132 3137 3046 3104 3000 3042 3079 3051 3124 3039 3074
3115 3120 3013 3024 3096 3032 3022 3014 3068 3091 3072 3039 3096 3034
3013 3056 3113 3034 3086 3033 3050 3003 3090 3019 3105 3076 3059 3087
3006 3102 3013 3106 3093 3031 3023 3073 3073 3120 3096 3118 3057 3011
3064 3128 3039 3052 3036 3128 3136 3011 3046 3123 3032 3115 3117 3137
3045 3005 3045 3022 3149 3059 3059 3035 3004 3058 3016 3068 3131 3000
3141 3000 3056 3000 3039 3080 3069 3051 3104 3099 3085 3117 3128 3090
3127 3059 3028 3113 3017 3035 3097 3024 3027 3099 3109 3087 3065 3010
3133 31143051 3013 3114 3143 3050 3113 3097 3048 3052 3119 3132 3149
3111 3145 3006 3057 3124 3028 3095 3111 3074 3094 3022 3131 3118 3107
3141 3126 3050 3003 3054 3110 31123147 3000 3083 3123 3080 3100 3031
3100 3080 3137 3081 30223114 3098 3118 3039 3093 3045 3024 3011 3045
3037 3079 3088 3129 31263142 3132 3128 3046 3094 3122 3069 3014 3125
3146 3049 3110 3078 30473037 3041 3122 3001 3073 3004 3134 3070 3145
3015 3071 3128 3033 3146 3046 3055 3044 3098 3016 3037 3125 3090 3099
3133 3018 3044 3089 3031 3079 3078 3040 3062 3128 3014 3020 3007 3025
3087 3045 3128 3030 3049 3010 3133 3020 3038 3046 3037 3044 3018 3028
3004 3106 3039 3007 3046 3108 3052 3055 3111 3062 3140 3086 3146 3047
3068 3010 3097 3082 3097 3143 3087 3130 3106 3135 3016 3008 3150 3009

3090 3147 3027 3051 3132 3081 3128 3053 3068 3136 3138 3019 3020 3057
3108 3023 3036 3064 3116 3100 3130 3052 3076 3004 3143 3142 3048 3123
3017 3087 3050 3011 3042 3034 3017 3109 3015 3077 3073 3009 3000 3031
3031 3009 3120 3082 3146 3090 3141 3016 3121 3101 3116 3095 3018 3096
3018 3108 3057 3061 3043 3000 3042 3008 3111 31183102 3089 3007 3102
3088 3019 3118 3092 3051 3092 3011 3090 3031 3012 3089 3070 3029 3043
3076 3066 3112 3018 3109 3055 3067 3024 3071 3062 3078 3086 3124 3049
3019 3147 3029 3107 3146 3148 3061 3030 3116 3113 3148 3054 3066 3114
3085 3012 3053 3132 3089 3054 3054 3047 3058 3147 3073 3149 3009 3063
3080 3046 3071 3060 3068 3131 3076 3029 3071 3117 3114 3019 3094 3014
3054 3014 3021 3123 3145 3010 3055 3088 3112 3116 3026

Додаток В

Лабораторна робота

3 курсу «периферійні пристрої»

Датчики відстані

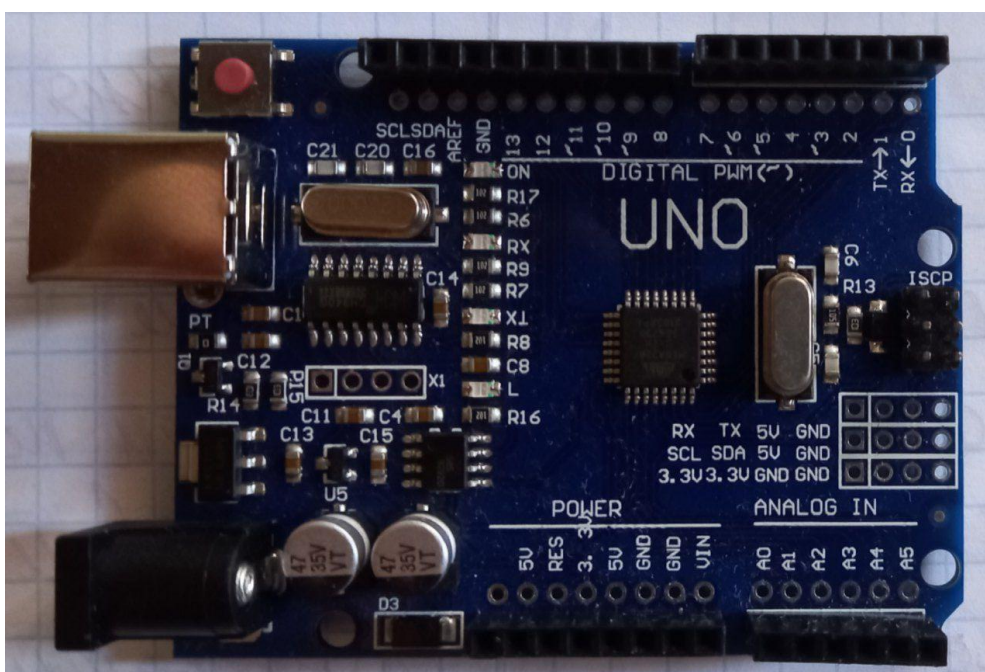
Тема: розробка макету для виміру відстані на платформі Arduino Uno R3.

Мета: навчитися проектувати пристрої на основі платформи Arduino Uno R3 у середовищі Arduino IDE та керувати ними.

Теоретичні відомості:

1. Мікроконтролер Arduino Uno

Arduino Uno R3 контролер побудований на ATmega328 мікроконтролері. Платформа має 14 цифрових входів/виходів (6 з яких можна використовувати як ШІМ-виходи), 6 аналогових входів, кварцовий генератор 16 МГц, роз'єм USB, роз'єм живлення, роз'єм ICSP і кнопку скидання. Для роботи вам потрібно підключити платформу до комп'ютера за допомогою USB-кабелю або використовувати адаптер змінного/постійного струму або живлення від акумулятора.



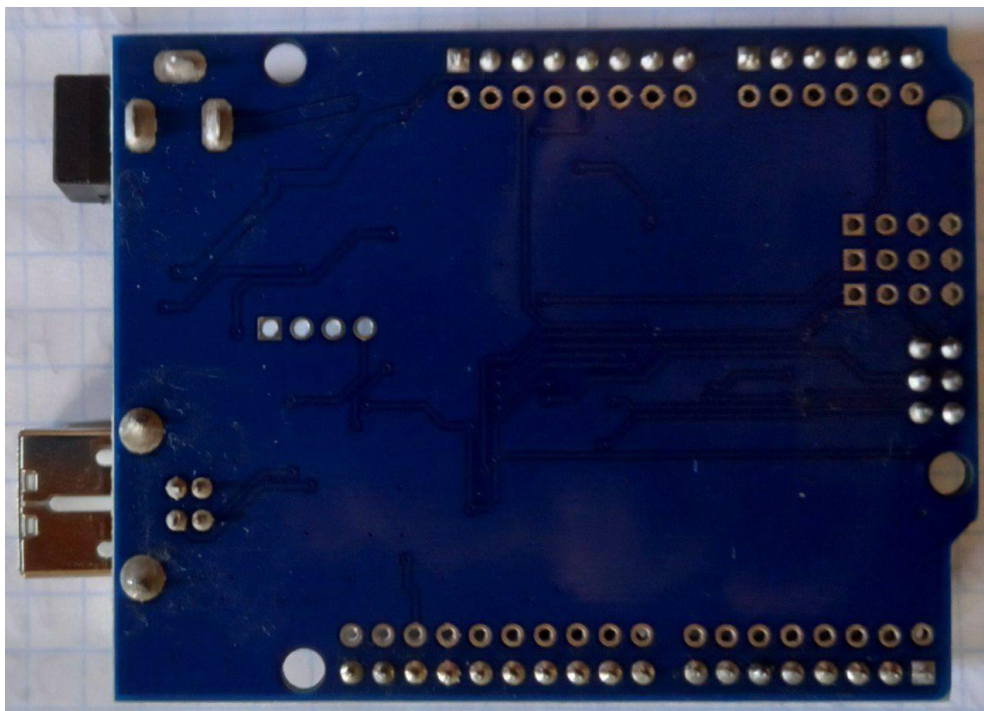


Рис. 1. Плата Arduino Uno R3

Він може житися від USB або зовнішнього живлення, вибір живлення встановлюється автоматично. Якщо останнє, то живлення може бути забезпечено акумуляторною батареєю або перетворювачем напруги змінного/постійного струму (блок живлення). Перетворювач напруги підключається через роз'єм 2,1 мм з центральним плюсовим полюсом. Провід від акумулятора підключаються до виходів Gnd і Vin роз'єму живлення.

Можлива робота від зовнішнього джерела живлення від 6 В до 20 В. Якщо напруга живлення нижче 7 В, то вихід 5 В може видавати менше 5 В, тому контролер працює не стабільно. Якщо подавати напругу вищу за 12 В регулятор може перегрітися і пошкодити плату. Діапазон від 7 В до 12 В.

VIN. Вхід призначений для живлення від зовнішнього джерела живлення (при відсутності 5 В від USB-роз'єму або іншого регульованого джерела живлення). Напруга живлення забезпечується через цей вихід.

5В. Регульоване джерело напруги для живлення мікроконтролера та компонентів на платі. Може живитися від виходу VIN через стабілізатор напруги, роз'єм USB або інше регульоване джерело напруги 5 В

3V3. Напруга на виведення 3.3 В. Максимальне споживання струму 50 мА.

GND. Виходи заземлення.

Мікроконтролер ATmega328 має 32 КБ флеш-пам'яті, з яких 0,5 КБ використовується для зберігання завантажувача, а також 2 КБ оперативної пам'яті (SRAM) і 1 КБ EEPROM. Кожен з 14 цифрових виходів Uno можна налаштувати як вхід або вихід за допомогою функцій `pinMode()`, `digitalWrite()` і `digitalRead()`. Вихід працює при нарузі 5 В. Кожен вихід має навантажувальний резистор 20-50 кОм (за замовчуванням відключений) і може пропускати до 40 мА. Деякі виходи мають спеціальні функції:

- 1) Послідовна шина: 0 (RX) і 1 (TX). Виходи використовуються для прийому (RX) і передачі (TX) даних TTL. Ці виходи підключаються до відповідних виходів мікросхеми послідовної шини ATmega8U2 USB-toTTL
- 2) Зовнішні переривання: 2 і 3. Ці виходи можуть бути налаштовані на виклик переривань з нижчими значеннями, до або після, або під час зміни
- 3) ШІМ: 3, 5, 6, 9, 10, і 11. Будь-який з виходів забезпечує ШІМ з роздільною здатністю 8 біт за допомогою функції `analogWrite`.
- 4) SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). За допомогою даних виходів здійснюється зв'язок SPI.
- 5) LED: 13. Вбудований світлодіод, підключений до цифрового виходу 13.

Також є 6 аналогових входів (позначаються A0 .. A5), кожен має АЦП з роздільною здатністю 10 біт (тобто може приймати 1024 різних значення). Як стандарт, вихідна напруга вимірюється до 5 В відносно землі, але

верхню межу можна змінити за допомогою виходу AREF і функції `AnalogReference()`.

Також встановлено кілька пристроїв для зв'язку з вашим комп'ютером, іншими пристроями Arduino або мікроконтролерами. ATmega328 підтримує послідовний інтерфейс UART TTL (5 В), виходи 0 (RX) і 1 (TX). Мікросхема ATmega8U2, встановлена на платі, завантажує інтерфейс через USB, а програма на стороні комп'ютера «спілкується» з платою через віртуальний COM-порт. Прошивка ATmega8U2 використовує стандартні драйвери USB COM, сторонні драйвери не потрібні, але для підключення Windows потрібен файл `ArduinoUNO.inf`. Послідовний монітор Arduino дозволяє надсилати та отримувати текстові дані при підключенні до платформи. Світлодіоди RX і TX на платформі будуть блимати, коли дані передаються через чіп FTDI або USB-з'єднання (але не при використанні послідовної передачі через виходи 0 і 1).

2. Система інтегрованого середовища розробки

Платформа Arduino має власне середовище розробки Arduino IDE, яке є безкоштовним і повністю відкритим. Arduino IDE — це багатоплатформна програма Java, яка включає редактор коду, компілятор і модуль передачі мікропрограми. Середовище розробки засноване на мові програмування Processing, подібній до мови Wiring. Власне кажучи, це C++, доповнений деякими бібліотеками. Програма обробляється за допомогою препроцесора, а потім компілюється за допомогою AVR-GCC.



Рис. 2. Інтегроване середовище розробки Arduino

Програми Arduino написані мовою програмування C або C++, а не Scratch. Середовище розробки Arduino постачається з програмною бібліотекою під назвою Wiring, похідною від проекту Wiring, що полегшує багато стандартних операцій введення-виводу. Користувачам потрібно лише визначити дві функції, щоб створити програму, яка виконується в циклі:

- 1) `setup ()`: функція виконується лише раз при старті програми и дозволяє задати початкові параметри;
- 2) `loop ()`: функція виконується періодично доки плата не буде вимкнено.

3. Основні функції для роботи з Arduino

Платформа Arduino має велику кількість функцій для роботи з підключеними пристроями. Нижче описані лише ті, які були використані у даній роботі :

- 1) `pinMode(pin, mode)` - встановлює режим роботи даного вхід / виходу (`pin`) як входу або як виходу. `mode`: режим одне з двох значення - `INPUT` і `OUTPUT`, встановлює на вхід або вихід відповідно.
- 2) `analogWrite(pin, value)` - видає аналогову величину (ШІМ хвилю) на порт вхід / виходу. Ця функція корисна для керування яскравістю підключених світлодіодів або швидкістю двигунів. Після виклику `analogWrite()` на виході буде згенерована постійна прямокутна хвиля заданої ширини імпульсу до наступного виклику `analogWrite()`. `value` – Період робочого циклу для значення від 0 (повністю вимкнено) до 255 (сигнальна константа).
- 3) `analogRead(pin)` - Функція зчитує значення із зазначеного аналогового входу. Більшість плат Arduino мають 6 каналів (8 в платах Mini і Nano, 16 у платах Mega) і 10-розрядний аналого-цифровий перетворювач (АЦП). Напруга, що подається на аналоговий вхід, зазвичай від 0 до 5 вольт, буде перетворено на значення від 0 до 1023 з 1024 кроками по 0,0049 вольт.
- 4) `Serial.Begin(speed)` - запускає послідовне з'єднання та встановить бітрейт /с. Для зв'язку з комп'ютером використовуються наступні значення: 300, 1200, 2400, 4800, 9600, 14400, 19200, 28800, 38400, 57600 або 115200. При підключенні через порти введення/виводу 0 і 1 інші значення можуть використовувати швидкість, необхідну для пристрою, з яким буде здійснюватися обмін даними.

Хід роботи:

1. Встановлення програмного забезпечення Arduino IDE
2. Зібрати схему відповідно до свого варіанту.
3. Написати програму для роботи макету.
4. Розрахувати похибку вимірювань будь-яким зручним варіантом.

Приклад створення макету з мигаючим світлодіодом.

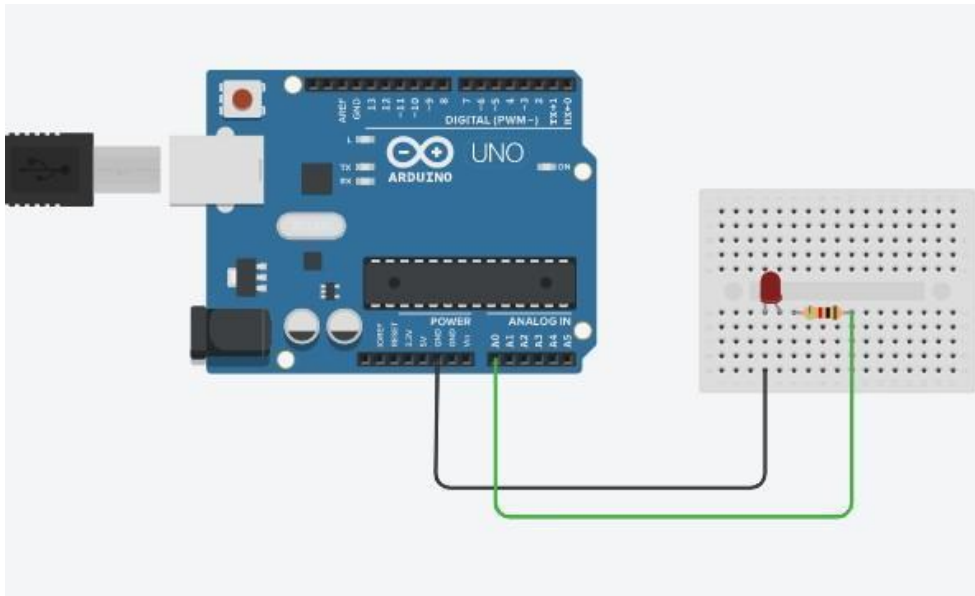


Рис.3. Макет з світлодіодом

У даному макеті не має складнощів потрібно лише правильно під'єднати. Анод даного світлодіода повинне бути підключеним до землі(напруга = 0В), а катод підключений до резистора опором 220Ом та підєднаний до одного за аналогових або цифрових входів/виходів. Анод довша ніжка, а в катода коротша ніжка.

Напишемо код за допомогою якого світлодіод буде світити 5 секунд, а далі переставати світитися на 1,2,3...

Код програми:

```
int LedPin = A0; // аналоговий вхід
```

```
int i =0; // константа для збільшення часу
```

```
void setup()
```

```
{
```

```
pinMode(LedPin, OUTPUT); // напруга на аналоговому вході є 0 В
}

void Loop()
{
    i++; // збільшуємо значення константи для подальшого збільшення
    часу при якому світлодіод не буде світитися

    digitalWrite(LedPin, HIGH); // подаємо на світлодіод напругу, щоб він
    світився

    delay(5000); // час при якому світлодіод буде світитися

    digitalWrite(LedPin, LOW); //перестаємо подавати напругу на
    світлодіод, щоб він не світився

    delay(1000*i); // час при якому світлодіод не буде світитися
}
```

Лазерний датчик відстані GY-530 на VL53L0X

Модуль GY-530 оснащений вбудованим ІЧ-фільтром. Його випромінювання повністю невидиме для людського ока, забезпечує більші відстані вимірювання, менш чутливе до зовнішнього рівня світла, більш стійке до перехресних перешкод, викликаних скляними поверхнями, а результати вимірювань не залежать від відбивної здатності поверхні. особливість: Надійна робота в умовах високої інтенсивності зовнішнього інфрачервоного випромінювання. Додаткова оптика не потрібна.



Рис. 4. GY-530 на VL53L0X

Характеристики:

- 1) Відстань до 2 м
- 2) Діапазон температур -20 до +70 С
- 3) Напруга 2,8- 5 В
- 4) Струм 19 мА
- 5) Довжина хвилі 940 нм

Світлодіодна матриця дисплей 8x8 на MAX7219

Модуль являє собою точково-матричну світлодіодну матрицю розміром 8 x 8 (64 світлодіоди). Точка на платі модуля блимає червоним. Цей модуль можна використовувати для проектування годинника або інших проектів візуалізації на платформі Arduino. Світлодіодна матрична плата розроблена на основі DIP мікросхеми управління MAX7219.

Драйвер MAX7219 окремо керує 64 світлодіодами на платі із загальним катодом. Модуль має послідовний 3-провідний інтерфейс. Передача даних на мікросхему модуля здійснюється через інтерфейс SPI, який підключається через три контакти. Контакти DIN (введення даних), CS (вибір модуля) і CLK (сигнал тактового сигналу) підключені до вільних цифрових портів на платі Arduino. Живлення для матриці подається на клеми VCC і GND 5 В.

Виходи модулів аналогічні входам і використовуються для послідовного підключення таких модулів. Крім того, оскільки ширина плати відповідає розміру матриці, модулі можна об'єднувати в ряд, тим самим збільшуючи кількість відображуваної інформації. Щоб продовжити використання світлодіодної матриці MAX7219, вам потрібно буде завантажити бібліотеку LedMatrix та встановити її самостійно.

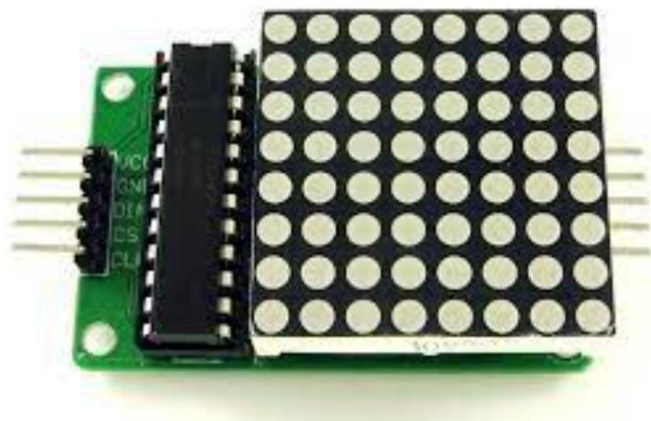


Рис. 5. Світлодіодна матриця дисплей 8x8 на MAX7219

Характеристики:

- 1) Напруга живлення, 4 ... 5,5
- 2) Споживаний струм, ма 150
- 3) Частота тактування, МГц 10
- 4) Мікросхема MAX7219
- 5) Кількість точок 64
- 6) Колір світіння червоний
- 7) Розміри модуля, мм 50 x 32 x 15

Ультразвуковий датчик відстані Arduino HC SR04

Ультразвуковий датчик відстані HC-SR04 - призначений для вимірювання відстаней від 2 до 400 см з межею точності 3 мм. Модуль включає в себе ультразвуковий передавач, приймач і блок управління. На показання датчика практично не впливають сонячне випромінювання та електромагнітний шум. На передній панелі HC-SR04 є два ультразвукових

датчика, перший позначений Т (передавач) – це ультразвуковий передавач (ТСТ40-16Т), а другий, позначений R (прийом), – це ультразвуковий приймач відбитої хвилі (ТСТ40-16R). Центрується на вихідному кварцовому генераторі на 27 МГц.



Рис. 5. ультразвуковий датчик відстані Arduino HC SR04

Характеристики:

- 9) діапазон вимірювання відстані 0,03м-4м
- 10) частота ультразвуку 40kHz
- 11) кут зору 30град.
- 12) інтерфейс 2 логічні TTL лінії
- 13) вихідна інформація імпульс 0,15..25mS
- 14) напруга живлення "Vcc" 5V
- 15) струм споживання в активному режимі 15mA
- 16) розмір модуля 45x20x15mm

Варіанти завдання:

- 1) Зробити макет на основі лазерного датчику відстані GY-530 розрахувати відстань та вивести цю відстань на світлодіодну матрицю дисплей 8x8 на MAX7219.
- 2) Зробити макет для виміру відстані на основі лазерного датчику відстані GY-530, та розрахувати похибку (принаймні для 3 відстаней).
- 3) Зробити макет з ультразвукового датчику відстані Arduino HC SR04 та лазерного датчику відстані GY-530. Порівняти датчики та зробити висновок який більш точний та чому.

Примітка. Бібліотека для користування лазерним датчиком `#include "VL53L0X.h"`, для ультразвукового датчика `#include <HCSR04.h>`, а для матриці `#include <iarduino_I2C_Matrix_8x8.h>`. З приводу встановлення бібліотек дивіться додаток 1.

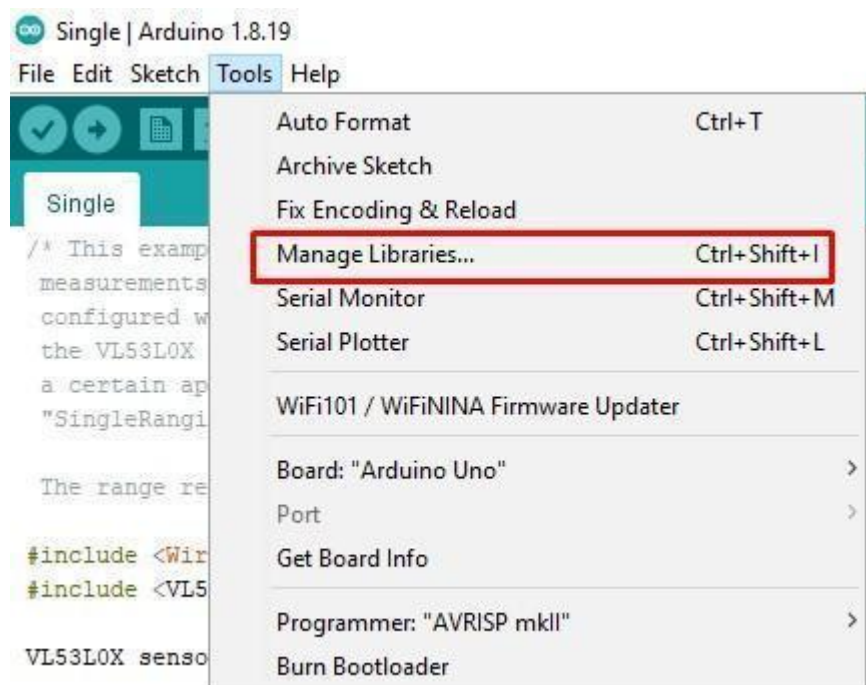
Контрольні запитання:

- 1) З якою швидкістю взаємодіє плата Arduino з ПК?
- 2) Чому результат вимірювань відрізняється від істинного, та як можливо його покращити?
- 3) У чому різниця між аналоговими входами та цифровими?
- 4) Яким чином між собою взаємодіє мікроконтролер та датчики?

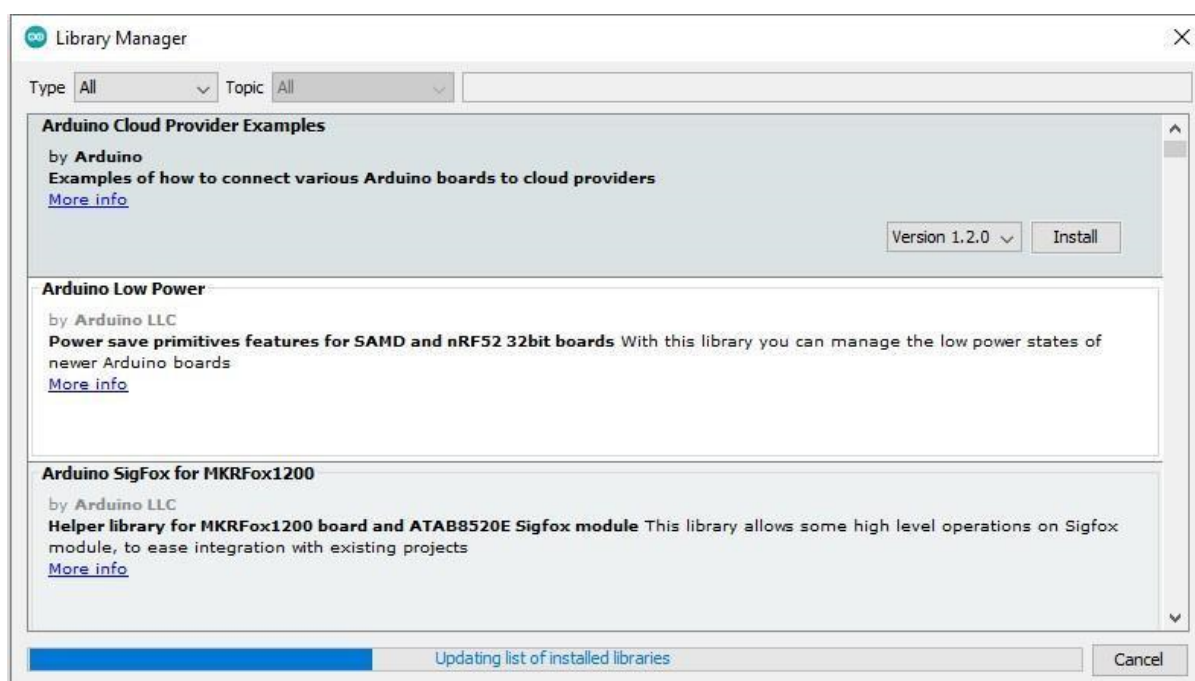
Додаток 1

Пошук бібліотек для в Arduino IDE

Крок 1

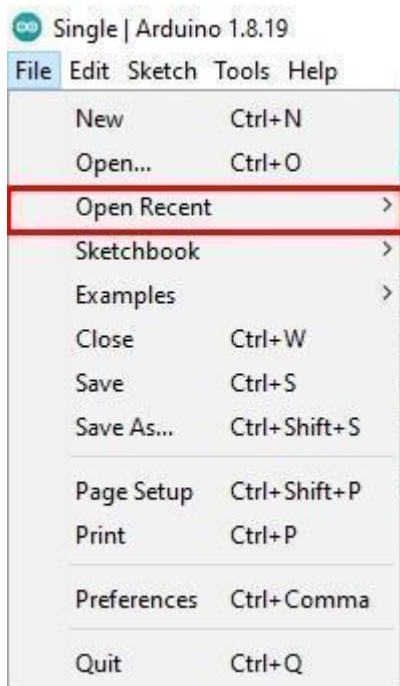


Крок 2



Прошивка бібліотеки

Крок 3



Тут можна обрати та подивитися на екземпляри бібліотек