

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

Сторчак К.П., Тушич А.М., Срібна І.М., Яковенко Н.Д., Кравець Д.В.

ТЕХНОЛОГІЇ ІНТЕРНЕТ РЕЧЕЙ

НАВЧАЛЬНИЙ ПОСІБНИК



Київ-2021

УДК 004.5
ББК 32.973я7

Гриф надано
Державним університетом телекомунікацій
(протокол №2 від 10 вересня 2020 р)

Рецензенти: доц., д.т.н. Бондарчук А.П.
проф., д.т.н. Кунах Н.І.

Навчальний посібник призначений для самостійної роботи студентів вищих навчальних закладів під час поглибленого вивчення дисципліни «Технології Інтернет речей» (ТІР) – циклу дисциплін професійної та практичної підготовки за спеціальністю 126 – Інформаційні системи та технології.

Сторчак К.П., Тушич А.М., Срібна І.М., Яковенко Н.Д., Кравець Д.В.
Технології Інтернет речей. Навч. посібник підготовлено для студентів вищих навчальних закладів – Київ: ДУТ, 2021. – 68 с.

У посібнику розглянуто загальні поняття побудови та використання найпростіших пристроїв на прикладі платформи Arduino.

Навчальний посібник призначений для студентів за напрямком «Інформаційні технології» зі спеціальності 126 – Інформаційні системи та технології, а також може бути корисним для аспірантів, викладачів навчальних закладів відповідних спеціальностей, фахівців, які обслуговують системи IoT.

Редактор:
Інженер I категорії Горбенко А.С.

ЗМІСТ

| | |
|---|----|
| ПЕРЕДМОВА..... | 4 |
| Знайомство з набором Arduino..... | 5 |
| Робота із світлодіодом Arduino | 11 |
| Arduino бігаюча доріжка з світлодіодів..... | 14 |
| Робота зі кнопкою в Arduino | 16 |
| Робота з датчиком рівня рідини Arduino..... | 20 |
| Робота з фотоелементами Arduino | 23 |
| Робота з сервоприводом Arduino | 28 |
| Робота з PIR-датчиком (руху) в Arduino | 33 |
| Дистанційне керування в Arduino..... | 39 |
| Робота з датчиком температури та вологості Arduino..... | 45 |
| Робота з RFID..... | 48 |
| Робота з LCD дисплеєм в Arduino | 54 |
| Робота з ультразвуковим далекоміром HC-SR04 в Arduino..... | 59 |
| Обробка даних від Джойстика. | 63 |
| Управління Pan/Tilt Bracket за допомогою джойстика в Arduino..... | 63 |
| ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ | 67 |

ПЕРЕДМОВА

Посібник з дисципліни «Технології Інтернет речей», яка викладається для студентів спеціальності 126 – Інформаційні системи та технології покликаний допомогти студентам денної та заочної форм навчання здобути навички проектування найпростіших пристроїв для зчитування інформації з різноманітних датчиків та реалізації складних систем розумного будинку.

Зміст та структура посібника відповідає освітньо-професійним програмам підготовки фахівців з спеціальності 126 – Інформаційні системи та технології. Посібник охоплює основний зміст матеріалу дисципліни «Технології Інтернет речей» та допомагає вивчити загальні поняття пристроїв Arduino, розібратись в описах компонентів, проаналізувати електронні схеми приладів та вибрати необхідні складові для створення свого унікального проекту. У підручнику можна знайти приклади використання сенсорів, індикаторів та інтерфейсів передачі інформації. Всі приклади забезпечені списком необхідних складових та прикладами коду з покроковим поясненням.

Знайомство з набором Arduino

Arduino - це платформа, зі зручною і простою архітектурою і програмними кодами для розробки і конструювання електронних пристроїв.

Розглянемо один з наборів для початківців Arduino RFID Kit (рис.1), який містить основні компоненти конструктора.



Рисунок 1. Набір для початківців Arduino RFID Kit

Набір включає основні елементи, в тому числі макетну плату і плату Arduino Uno R3, різні датчики, резистори і діоди.

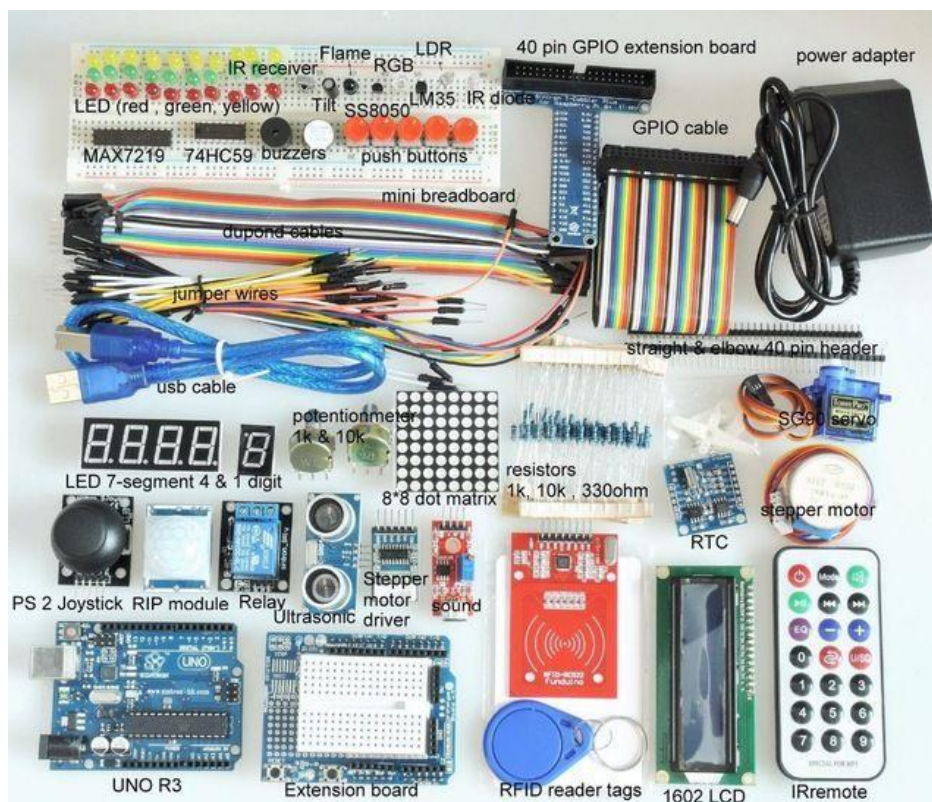


Рисунок 2. Повна комплектація набору для початківців Arduino RFID Kit

Повна комплектація (рис.2) з докладним описом кожного елемента набору зведена в таблицю 1.

Таблиця 1. Повна комплектація набору Arduino

| Назва | Опис |
|---------------------------------------|--|
| Макетна плата МВ-102 | Містить 830 точок, з відстібними краями і на двосторонньому скотчі. Має можливість нарощування по обидва боки, а також підключати живлення (200 входів). |
| Світлодіодна матриця 8*8 | Знизу матриця обладнана двома рядами пінів, стандартний крок - 2,54. Встановлюється в дві макетні плати без спайки. Загальний анод. Діаметр світлодіодів 5 мм. |
| Модуль дисплея 4-х бітовий | Семисегментне відображення, загальний анод. Робоче живлення 5В. |

| | |
|--|--|
| Модуль дисплея однобітовий | Семисегментне відображення, загальний анод. Робоче живлення 5В. |
| I2C - модуль для підключення LCD- дисплея | Забезпечує управління пристроями з восьмибітними шинами. На платі встановлений потенціометр для регулювання контрастності дисплея. |
| LCD-дисплей | Модуль дисплея синій. Має дві строчки по 16 символів в кожній. Габарити 8*3,6 см. Робоче живлення 5 В. |
| RFID - модуль | До складу входить 2 мітки - біла картка і брелок- монетка, а також ніжки для підключення модуля. Працює від живлення 3.3 В, підключається через інтерфейс SPI. Забезпечує читання і запис RFID- міток. |
| Блок клавіатури 4*4 | Матрична клавіатура на 16 тактових кнопок має два кріпильні отвори і виконана на друкованій платі двостороннього типу. Оснащена 8 виводами для підключення до Arduino. |
| Модуль годинника реального часу RTC | Високоточний годинник реального часу, має можливість відліку секунд, хвилин, годин, днів, місяців і років (з урахуванням високосного року). Робота здійснюється через I2C інтерфейс. |
| Модуль мікрофона | Модуль з наявністю на платі підлаштування резистора, компаратора і чутливого мікрофона, аналогового і цифрового виходів. |
| Мікросхема ULN2003 | 4 входи для підключення до мікроконтролеру, 5 виходів для підключення до двигуна (на 5 або на 12 В), два контакти для живлення. На кожен канал є світлодіод, для індикації. |

| | |
|-----------------------------|---|
| Кроковий двигун | Уніполярний модуль 28BYJ-48, з управлінням за допомогою електромагнітних котушок. Номінальне живлення 5 В. 4 фази і 64 кроку. Рекомендовано використовувати в поєднанні з ULN2003. |
| Набори перемичок | Різного типу і довжини для підключення до макетної платі різних модулів (тато-тато, мама-тато). |
| Реле | Одноканальний модуль на чорній платі. Живлення управління 5В. Пряме управління з виводів. Включення - логічний нуль, вимикання - одиниця. |
| Плата Arduino UNO R3 | Для перетворення інтерфейсів використовується АТmega16U2. 14 виводів цифрового типу і 6 - аналогового, роз'єм USB, для підключення до комп'ютера, вивід для подачі живлення, кнопка скидання. Робоче живлення 5В. |
| Пульт | ІК-пульт для дистанційного керування. У комплект поставки батарейка живлення не входить. |
| Світлодіоди | Зелені, жовті, червоні - діаметр 3 мм, інфрачервоний світлодіод. Перетворюють струм в світіння. |
| Резистори | На 220 Ом, 10 кОм, фоторезистори. |
| ІК-приймач | Кут огляду 90 градусів. Приймає випромінювання від ІК-передавача. |
| Пищалка | Активний електромагнітний бубзер на 5В. |
| Температурний датчик | Цифровий датчик температури, розрахований на режим живлення від 3 до 5 В. Вимірює вологість в діапазоні від 20 до 80% |

| | |
|---------------------------------------|---|
| Джойстик | Один цифровий вихід і два аналогових. На платі є два потенціометра. Управління рухом здійснюється в двох вісях. |
| Сервопривод | Аналогового типу. Оснащений нейлоновим редуктором і універсальним S роз'ємом. Напряга живлення 4.8 - 6 В. |
| Перехідник для крони | Конектор для підключення живлення від батареї типу «Крона». Вихідна напруга - 9В. |
| Кабель USB | Для підключення до комп'ютера. |
| Триколірний (RGB) світлодіод | Має окремі виводи, світиться червоним, синім і зеленим. |
| Датчик рівня рідини | Аналоговий. Робоче живлення 5 В. |
| Змінний резистор | Потенціометр на 1 кОм. Забезпечує регулювання опору. |
| Макетна плата без спайки | Має 170 отворів. Призначена для тимчасового проектування невеликих проектів. Крок 2.54 мм. |
| Датчик температури і вологості | Цифровий датчик. Забезпечує калібрування цифрового сигналу на виході. Містить ємнісний датчик вологості і термістор. Перетворює аналогові значення параметрів за допомогою АЦП. |
| Зсувний регістр | Розширювач виводів. Восьмібітний. Робоча напруга від 2.5 В до 3.5 В. |
| Аналоговий датчик температури | Забезпечує вимір температури в межах від -55 до 150 ° С. Калібрування виконується в градусах за шкалою Цельсія. |

Отже, *Arduino KIT* - набір, який включає основні модулі та елементи конструктора, що дає можливість новачкові познайомитися з електронікою, і почати працювати в середовищі *Arduino*. Вхідні в нього компоненти забезпечать створення різних схем і макетів та допоможуть втілити ідеї в

реальність. Такої комплектації буде досить для освоєння перших уроків конструювання, без пошуку додаткових елементів для створення проектів. Набори КІТ значно спрощують життя новачкам, які тільки пізнають ази робототехніки, і роблять процес розробки і створення цікавим і захоплюючим.

Робота із світлодіодом Arduino

1. Основні поняття та фізичний зміст

Пристрої, які називаються індикаторами, дозволяють простому електронному приладу спілкуватися з людиною на мові примітивних сигналів. Існує безліч різних за принципом дії індикаторів, які переслідують одну мету повідомити людині про стан системи. Так, наприклад, індикатор живлення монітора дає зрозуміти підключений останній до джерела енергії.

Найпростіший індикатор, називається *світловипромінюючим діодом*. Це пристрій, являє собою напівпровідниковий прилад, здатний випромінювати світло при пропусканні через нього електричного струму в прямому напрямку (від анода до катода). Нижче наведена схема типового світлодіода з лінзою (рис. 16).

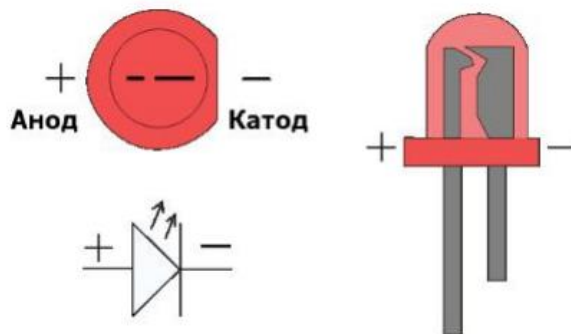


Рисунок 16. Світлодіод

2. Схема підключення

Для того щоб правильно включити світлодіод в електричний ланцюг, необхідно відрізнити катод від анода. Зробити це можна за двома ознаками:

- 1) Анод світлодіода має довший провідник
- 2) З боку катода, корпус світлодіода трохи зрізаний

Принципова схема зображена на рис. 17

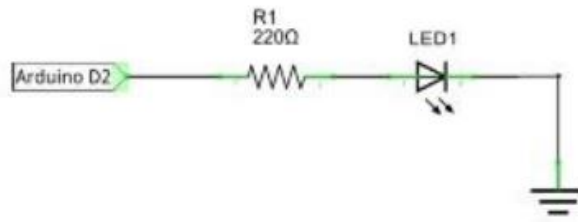


Рисунок 17. Принципова схема

3. Приклад

Нам потрібно написати програму щоб, відразу після запуску програми, світлодіод починав блимати один раз в секунду.

Важливо відзначити, що напруга живлення світлодіода варіюється від 1.85 до 2.5 вольт, при рекомендованій силі струму 20 мА. Для правильної роботи приладу, в ланцюг слід додати обмежувач резистор (від 200 Ом до 500 Ом).

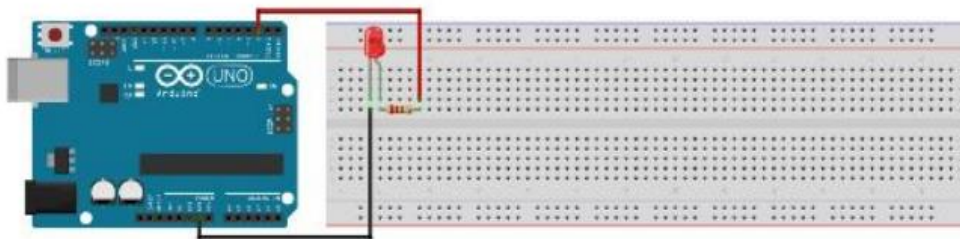


Рисунок 18. Вигляд макета

Щоб блимати світлодіодом (рис. 18), ми будемо послідовно запалювати його, передаючи на ногу №2 сигнал HIGH, а потім гасити за допомогою сигналу LOW. Між включенням і вимиканням світлодіода обов'язково потрібно поставити затримку в кілька сотень мілісекунд, інакше ми не помітимо як світлодіод буде блимати.

```
int led = 2;
```

```
void setup () {  
pinMode (led, OUTPUT); // установка 2-го контакту в режим вывода  
}  
  
void loop () {  
digitalWrite (led, HIGH); // переключить вывод №2 в активный стан  
delay(1000); // пауза 1 секунда  
digitalWrite (led, LOW); // переключить вывод №2 в неактивный стан  
delay(1000); // пауза 1 секунда
```

Arduino бігаюча доріжка з світлодіодів

1. Основні поняття та фізичний зміст

Світлодіоди - це напівпровідникові елементи, які служать для індикації та освітлення. Вони мають полярність (+ та -) і відчують напрямок руху постійного струму. Якщо підключити світлодіод неправильно, то постійний струм не пройде і пристрій не засвітиться. Крім того, світлодіод може вийти з ладу при неправильному підключенні. Анод (довга ніжка світлодіоди) підключається до плюса.

Навчимося робити бігаючу доріжка з світлодіодів на Arduino. В даному прикладі використовується *Arduino Mega 2560*, який потенційно здатен управляти бігаючою доріжкою з 54-х світлодіодів. Але схема і програма не зміняться, якщо ви будете використовувати інші контролери з платформи Arduino такого типу (UNO, Leonardo та інші).

2. Схема підключення

Схема підключення світлодіодів до Arduino Mega 2560 (рис.3).

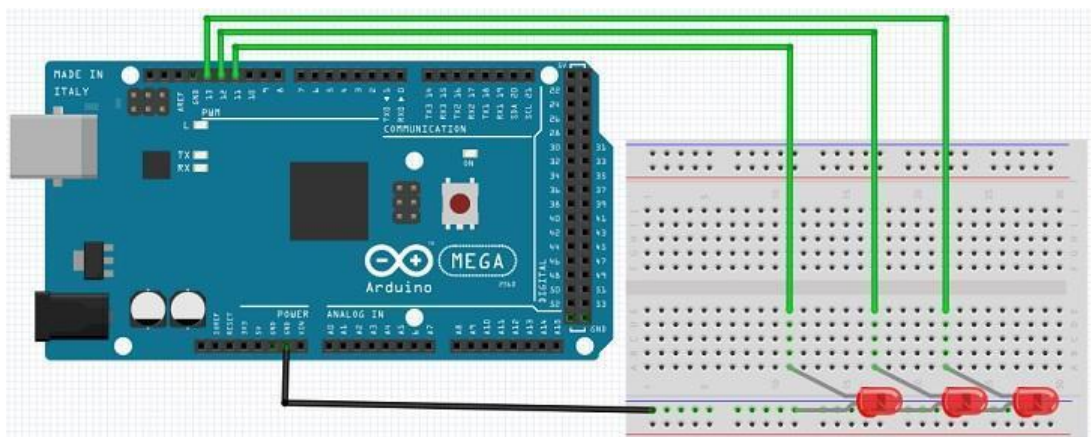


Рисунок 3. Бігаюча доріжка з світлодіодів

3. Приклад

Текст програми з коментарями для реалізації бігаючої доріжки на платформі Arduino:

```

int first_out = 11; // перший дискретний вихід
int last_out = 13; // останній дискретний вихід
int t;
int i;
// блок для ініціалізації входів-виходів і інших вихідних даних
void setup () {
  last_out = last_out + 1; // додаємо одиницю для коректного використання в циклах
  // визначення 11-го, 12-го і 13-го дискретних виводів плати Ардуіно як виходи
  for (i = first_out; i <last_out; i ++) {pinMode (i, OUTPUT); }
}
void loop () {
  for (t = first_out; t <last_out; t ++) { // перебираємо номери дискретних виходів 11,12,13 по
черзі
  digitalWrite (t, HIGH); // запалювання наступного світлодіода
  delay (500); // затримка 500мсек
  for (i = first_out; i <last_out; i ++) {digitalWrite (i, LOW); } // гасимо всі світлодіоди
  }
}

```

Для збільшення кількості керованих світлодіодів в доріжці, в програмі потрібно буде просто замінити значення змінних `first_out` і `last_out`. Перша змінна зберігає початковий дискретний вихід контролера, а друга останній з групи виходів, які йдуть підряд. Наприклад, якщо ми хочемо підключити 10 світлодіодів до доріжки, вводимо такі значення: `first_out = 4`, `last_out = 13`. І Arduino підключить світлодіоди до виводів по порядку з 4-го по 13-й. А перший і другий виводи дискретних входів-виходів краще не чіпати, оскільки їм заважає usb-порт, підключений до комп'ютера.

Робота зі кнопкою в Arduino

1. Основні поняття та фізичний зміст

Кнопка - найпростіший і доступний з усіх типів датчиків. Натиснувши на нього, ви даєте контролеру сигнал, який потім призводить до деяких дій: включаються світлодіоди, видаються звуки і запускаються двигуни. У нашому житті ми часто зустрічаємося з різними перемикачами і дуже добре знайомі з цим пристроєм.

Що таке *кнопка* (рис.19)? По суті, це досить простий пристрій, що замикає і розмикає електричну мережу. Виконувати це замикання / розмикання можна в різних режимах, при цьому фіксувати або не фіксувати своє положення. Відповідно, всі кнопки можна поділити на дві великі групи:

- Кнопки перемикачі з фіксацією. Вони повертаються в початковий стан після того, як їх відпустили. При в залежності від початкового стану поділяють на нормально-замкнуті і нормально-розімкнуті кнопки.
- Кнопки без фіксації (тактові кнопки). Вони фіксуються і залишаються в тому положенні, в якому їх залишили.

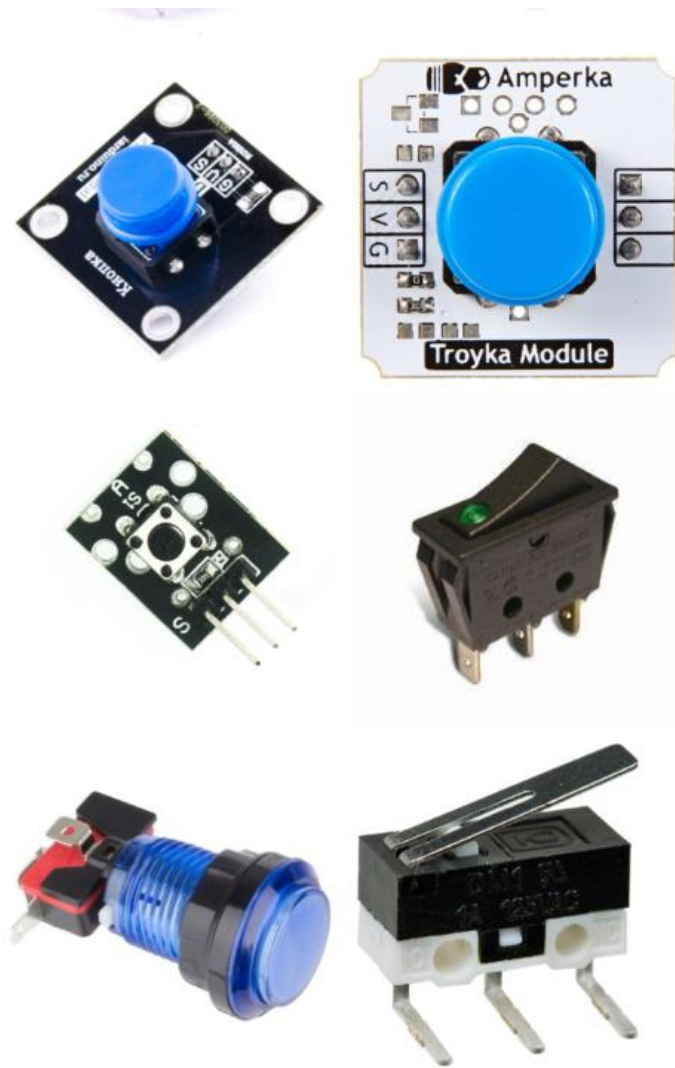


Рисунок 19. Кнопки

2. Схема підключення

Схема підключення кнопки до плати Arduino Uno представлена на рис.20.

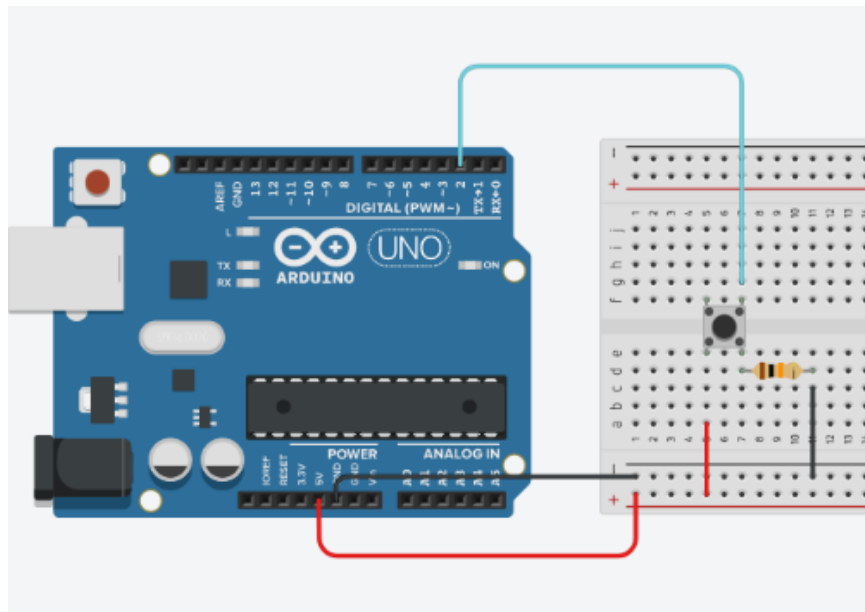


Рисунок 20. Вигляд макету підключення

3. Приклад

Давайте тепер підключимо кнопку до Arduino так, щоб можна було зчитувати в скетчі її стан. Для цього скористаємося наступною схемою (рис.21).

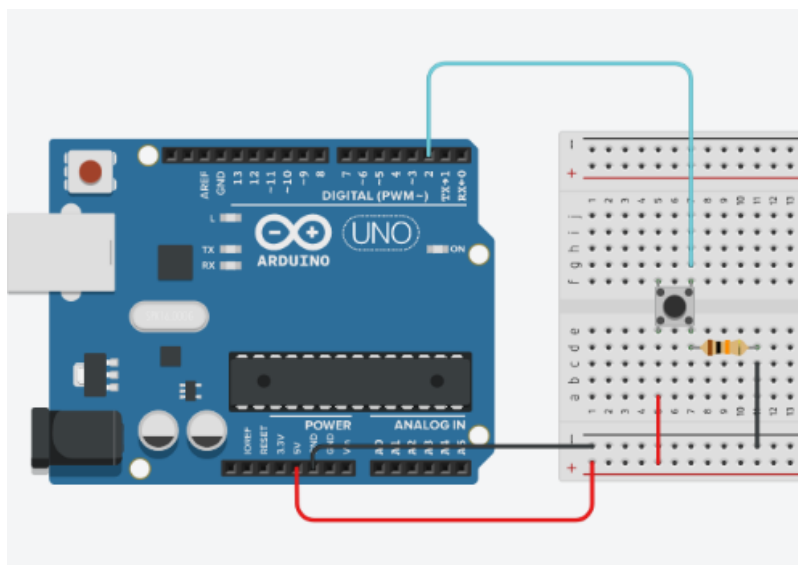


Рисунок 21. Схема підключення

В скетчі ми будемо відстежувати факт натискання і виводити повідомлення в монітор порту. Слід звернути увагу на опір 10 К, який ми додали в цій схемі. Більш докладно про його призначення ми поговоримо пізніше, просто майте на увазі, що такий резистор необхідний для правильної роботи схеми.

Скетч для кнопки Arduino з підтягуючим резистором:

```
const int PIN_BUTTON = 2;

void setup() {
  Serial.begin(9600);
  pinMode(PIN_LED, OUTPUT);
}

void loop() {
  int buttonState = digitalRead(PIN_BUTTON);
  Serial.println(buttonState);
  delay(50);
}
```

Робота з датчиком рівня рідини Arduino

1. Основні поняття та фізичний зміст

Робоча напруга аналогового сенсора-5v. Вихідна напруга залежить від ступеня занурення датчика в рідину і від параметрів, що впливають на коефіцієнт передачі напруги, наприклад, провідність рідини. Це простий у використанні і недорогий *датчик рівня рідини* (рис.22), який широко застосовується в системах автоматизації і при розробці розумного будинку.

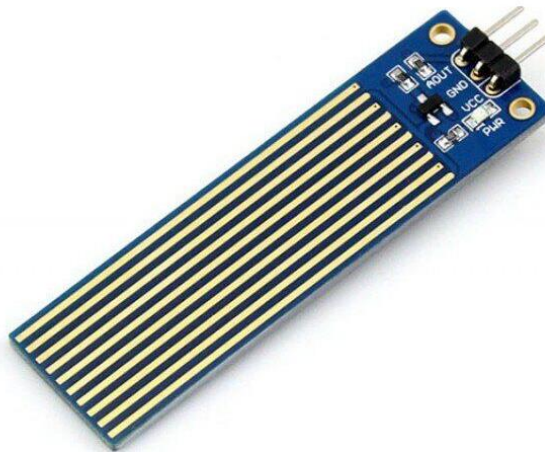


Рисунок 22. Датчик рівня рідини

2. Схема підключення

Схема підключення датчика рідини до плати Arduino (рис.23):

- - до GND
- + до 5 V
- S До аналогового порту

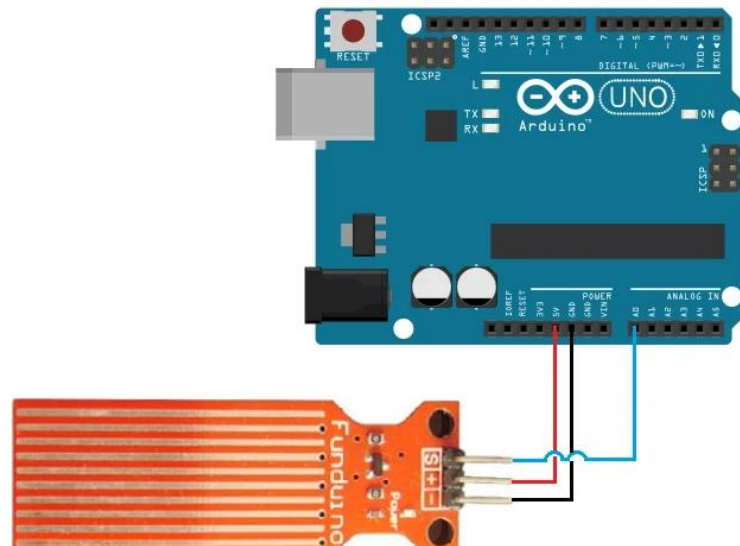


Рисунок 23. Схема підключення

3. Приклад

Напишемо скетч, для виводу значення датчика. Для початку задамо змінну `water` за допомогою оператора `int`. Змінну ми будемо використовувати для зберігання даних в пам'яті і присвоювання їй значення, одержуваного з датчика води.

У процедурі `void setup ()` ми прописуємо режим роботи аналогового порту, як вхід — команда `pinMode (A0, INPUT)` і підключаємо монітор порту за допомогою `Serial.begin(9600)`. Далі в процедурі `void loop ()` отримуємо значення з аналогового порту за допомогою команди `water = analogRead(A0);` і передаємо значення на монітор порту Arduino IDE `Serial.println (water);` і ставимо паузу для нашої зручності.

```
int water;

void setup() {
  pinMode (A0, INPUT);
  Serial.begin(9600);
}
```

```
void loop() {  
  water = analogRead(A0);  
  
  Serial.println(water);  
  delay(1000);  
}
```

Робота з фотоелементами Arduino

1. Основні поняття та фізичний зміст

Датчики освітленості, побудовані на базі фоторезисторів, досить часто використовуються в Arduino проектах. *Фоторезистор Arduino* дозволяє контролювати рівень освітленості і реагувати на його зміну. Отже, розглянемо, що таке фоторезистор, як працює датчик освітленості на його основі, як правильно підключити датчик до плати Arduino.

На відміну від звичайного резистора, фоторезистор (рис.4) може змінювати свій опір в залежності від рівня зовнішньої освітленості. Це означає, що в електронній схемі будуть постійно змінюватися параметри, в першу чергу нас цікавить напруга, падаюча на фоторезистори. Фіксуючи ці зміни напруги на аналогових пінах Arduino, ми можемо змінювати логіку роботи схеми, створюючи тим самим пристрої, що адаптуються під зовнішні умови.

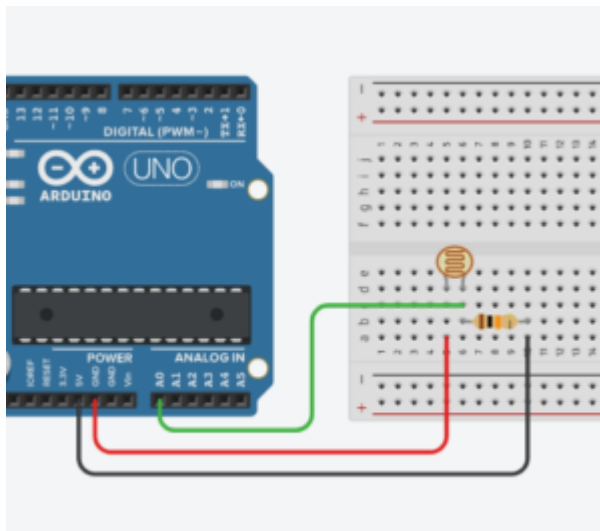


Рисунок 4. Фоторезистор - датчик освітленості Arduino

Основним недоліком фоторезисторів є чутливість до спектру. Залежно від типу падаючого світла опір може змінюється на кілька порядків.

2. Схема підключення

Схема підключення датчика освітленості до Arduino досить проста. Якщо ми використовуємо фоторезистор, то в схемі підключення (рис.5) датчик реалізований як дільник напруги. Одне плече змінюється від рівня освітленості, друге - подає напругу на аналоговий вхід. У мікросхемі контролера ця напруга перетвориться в цифрові дані через АЦП. Оскільки опір датчика при попаданні на нього світла зменшується, то і значення падаючої на нього напруги буде зменшуватися.

Залежно від того, в якому плечі дільника ми поставили фоторезистор, на аналоговий вхід буде подаватися або підвищений або зменшене напруження. У тому випадку, якщо одна нога фоторезистора підключена до землі, то максимальне значення напруги буде відповідати темряві (опір фоторезистора максимальний, майже вся напруга падає на ньому), а мінімальне - гарному освітленню (опір близько до нуля, напруга мінімальна). Якщо ми підключимо плече фоторезистора до живлення, то поведінка буде протилежною.

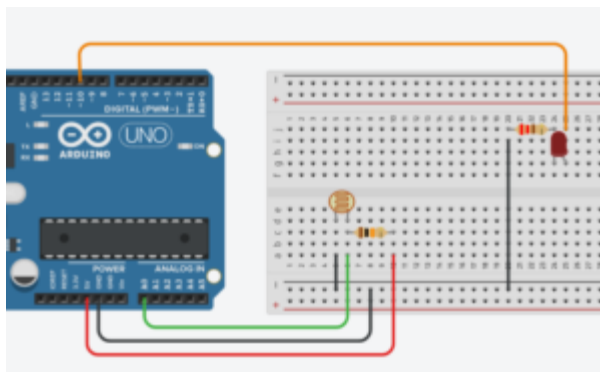


Рисунок 5. Схема підключення датчика освітленості, що включає/вимикає світлодіод

Монтаж плати не повинен викликати труднощів. Так як фоторезистор не має полярності, підключити можна будь-якою стороною, до плати його можна припаяти, під'єднати проводами за допомогою монтажною плати або використовувати звичайні кліпси (крокодили) для з'єднання. Джерелом живлення в схемі є сам Arduino. Фоторезистор приєднується однією ногою до землі, інша підключається до АЦП плати (в нашому прикладі – А0). До цієї ж

ноги підключаємо резистор 10 кОм. Підключати фоторезистор можна не тільки на аналоговий пін A0, але і на будь-який інший.

Якщо ви використовуєте готовий модуль датчика світла, то підключення буде ще більш простим. З'єднуємо вихід модуля VCC з роз'ємом 5В на платі, GND - с землею. Решта виводів з'єднуємо з роз'ємами Arduino. Якщо на платі представлений цифровий вихід, то відправляємо його на цифрові піни. Якщо аналоговий - то на аналогові.

3. Приклад

Ми підключили схему з фоторезистором до Arduino, переконалися, що все зробили правильно. Тепер залишилося запрограмувати контролер.

Написати скетч для датчика освітленості досить просто. Нам потрібно тільки зняти поточне значення напруги з того аналогового піна, до якого підключений датчик. Робиться це за допомогою функції `analogRead()`. Потім ми можемо виконувати якісь дії, в залежності від рівня освітленості.

Давайте напишемо скетч для датчика освітленості, що включає або вимикає світлодіод, підключений за наступною схемою, зображеною на рис.5.

```
#define PIN_LED 13
#define PIN_PHOTO_SENSOR A0
void setup () {
  Serial.begin (9600);
  pinMode (PIN_LED, OUTPUT);
}
void loop () {
  int val = analogRead (PIN_PHOTO_SENSOR);
  Serial.println (val);
  if (val <300) {
    digitalWrite (PIN_LED, LOW);
  } Else {
    digitalWrite (PIN_LED, HIGH);
  }
}
```

}

Алгоритм роботи такий:

- визначаємо рівень сигналу з аналогового піна;
- порівнюємо рівень з граничним значенням. Максимально значення буде відповідати темряві, мінімальне - максимальній освітленості. Граничне значення виберемо рівне 300;
- якщо рівень менше порогового - темно, потрібно включати світлодіод;
- інакше - вимикаємо світлодіод.

Прикриваючи фоторезистор (руками або світлонепроникним предметом), можемо спостерігати включення і виключення світлодіода.

При монтажі постарайтеся розташувати фоторезистор і світлодіод максимально далеко один від одного, щоб на датчик освітленості потрапляло менше світла від яскравого світлодіода.

Можна модифікувати проект так, щоб в залежності від рівня освітленості змінювалася яскравість світлодіода. В алгоритм ми додамо наступні зміни: яскравість лампочки будемо міняти через ШІМ, посилаючи за допомогою `analogWrite ()` на пін з світлодіодом значення від 0 до 255.

Крім того, для перетворення цифрового значення рівня освітлення від датчика освітленості (від 0 до 1023) в діапазон ШІМ яскравості світлодіода (від 0 до 255) будемо використовувати функцію `map ()`.

Приклад скетчу:

```
#define PIN_LED 10
#define PIN_PHOTO_SENSOR A0
void setup () {
  Serial.begin (9600);
  pinMode (PIN_LED, OUTPUT);
}
void loop () {
  int val = analogRead (PIN_PHOTO_SENSOR);
```

```
Serial.println (val);  
int ledPower = map (val, 0, 1023, 0, 255); // Перетворимо отримане значення в рівень  
PWM-сигналу. Чим менше значення освітленості, тим менше потужності ми повинні  
подавати на світлодіод через ШІМ.  
analogWrite (PIN_LED, ledPower); // Міняємо яскравість  
}
```

У разі іншого способу підключення, при якому сигнал з аналогового порту пропорційний ступені освітленості, треба буде додатково «звернути» значення, віднімаючи його з максимального:

```
int val = 1023 - analogRead (PIN_PHOTO_RESISTOR);
```

Використовуючи знання про роботу з фотоелементами ви можете реалізувати безліч цікавих проєктів. Підключення фоторезистора здійснюється за схемою подільника напруги з додатковим опором. Датчик підключається до аналогового порту для вимірювання різних значень рівня освітленості або на іншому цифровому, якщо нам важливий лише факт настання темряви. У скетчі ми просто зчитуємо дані з аналогового (або цифрового) порту і приймаємо рішення, як реагувати на зміни.

Робота з сервоприводом Arduino

1. Основні поняття та фізичний зміст

Сервопривод - це механізм з електромотором з керуванням (рис.6). Ви можете обертати механічний привід на заданий кут із заданою швидкістю або зусиллям.

Найбільш популярні сервоприводи, які утримують заданий кут і сервоприводи, що підтримують задану швидкість обертання.

Сервоприводи мають кілька складових частин. *Привод* - електромотор з редуктором. Найчастіше швидкість обертання мотора буває занадто великою для практичного використання. Для зниження швидкості використовується редуктор: механізм з шестернею, передає і перетворює крутний момент.

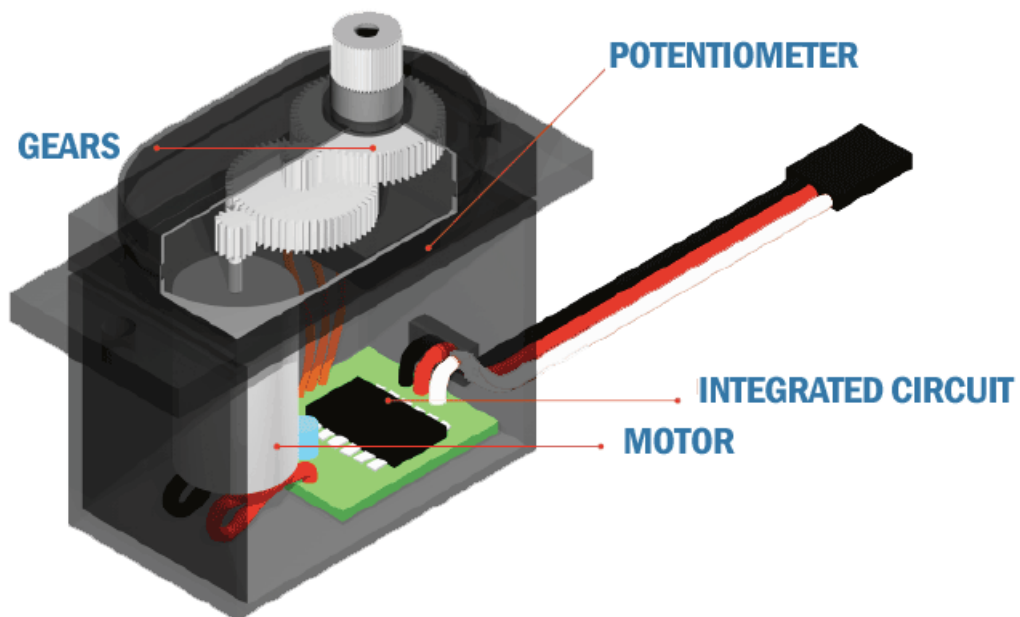


Рисунок 6. Будова сервопривода

Включаючи і вимикаючи електромотор, можна обертати вихідний вал - кінцеву шестерню сервоприводу, до якої можна прикріпити щось, ніж ми хочемо керувати - важіль у формі кола, хрестовини або переключачки для передачі крутного руху на робочий орган. Для контролю положення

використовується датчик зворотного зв'язку - *енкодер*, який буде перетворювати кут повороту назад в електричний сигнал. Для цього часто використовується потенціометр. При повороті бігунка потенціометра відбувається зміна його опору, пропорційне куту повороту. Таким чином, з його допомогою можна встановити поточний стан механізму.

Крім електромотора, редуктора і потенціометра в сервоприводів є електронна начинка, яка відповідає за прийом зовнішнього параметра, зчитування значень з потенціометра, їх порівняння і включення / вимикання двигуна.

До сервоприводу тягнуться три дроти. Два з них відповідають за живлення мотора і землю, третій приносить керуючий сигнал, який використовується для виставлення положення пристрою.

Також, сервоприводи зазвичай мають обмежений кут обертання 180 градусів, їх так і називають «сервопривід 180 °». Але існують сервоприводи з необмеженим кутом повороту осі. Це сервоприводи постійного обертання або «сервоприводи 360 °».

2. Схема підключення

Багато сервоприводів можуть бути підключені до Arduino безпосередньо (рис. 7). Для цього від них йде шлейф з трьох проводів:

червоний - живлення; підключається до контакту 3.3 / 5V або безпосередньо до джерела живлення;

коричневий/чорний – земля;

жовтий/білий - сигнал; підключається до цифрового виходу Arduino.

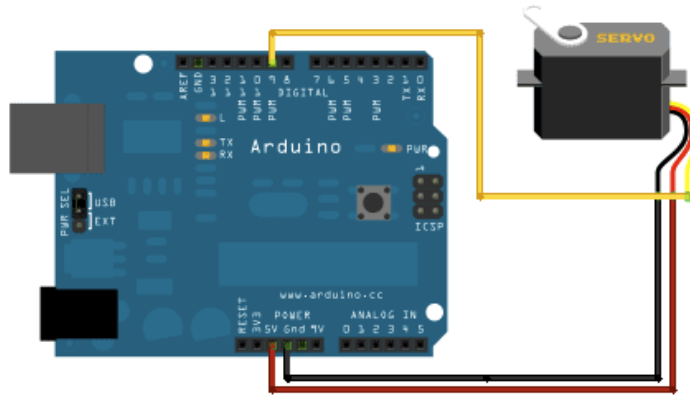


Рисунок 7. Схема підключення сервопривода

На більшості плат Arduino бібліотека Servo підтримує управління не більше 12 сервоприводами, на Arduino Mega - 48. При цьому є невеликий побічний ефект використання цієї бібліотеки: якщо ви працюєте не з Arduino Mega, то стає неможливим використовувати функцію `analogWrite ()` на 9 і 10 пінах незалежно від того, підключені сервоприводи до цих контактів чи ні. На Arduino Mega можна підключити до 12 сервоприводів без порушення функціонування ШІМ / PWM, при використанні більшої кількості сервоприводів ми не зможемо використовувати `analogWrite ()` на 11 і 12 пінах.

Бібліотеки для управління сервоприводами (Servo) і для роботи з приймачами / передавачами на 433 МГц VirtualWire використовують одне і те саме переривання. Це означає, що їх не можна використовувати в одному проєкті одночасно. Існує альтернативна бібліотека для управління сервомоторами - Servo2.

Так, бібліотека Servo дозволяє здійснювати програмне керування сервоприводами. Управління здійснюється наступними функціями:

`attach ()` - приєднує об'єкт до конкретного виводу плати. Можливі два варіанти синтаксису для цієї функції: `servo.attach (pin)` і `servo.attach (pin, min, max)`. При цьому `pin` - номер піна, до якого приєднують сервопривід, `min` і `max` - довжини імпульсів в мікросекундах, що відповідають за кути повороту 0° і

180 °. За замовчуванням виставляються рівними 544 мкс і 2400 мкс відповідно. Значення, що повертається немає.

write () - повертає команду сервоприводу прийняти деяке значення параметра. Синтаксис: `servo.write (angle)`, де *angle* - кут, на який повинен повернутися сервопривід

writeMicroseconds () - повертає команду послати на сервопривід імпульс певної довжини, є низькорівневим аналогом попередньої команди. Синтаксис наступний: `servo.writeMicroseconds (uS)`, де *uS* - довжина імпульсу в мікросекундах. Значення, що повертається немає.

read () - читає поточне значення кута, в якому знаходиться сервопривід. Синтаксис: `servo.read ()`, повертається ціле значення від 0 до 180

attached () - перевірка, чи був приєднаний об'єкт до конкретного піну. Синтаксис наступний: `servo.attached ()`, повертається логічна істина, якщо об'єкт був приєднаний до будь-якого піну, або брехня в зворотному випадку

detach () - виробляє дію, зворотне дії `attach ()`, тобто від'єднує об'єкт від піна, до якого був приписаний. Синтаксис: `servo.detach ()`

```
#include <Servo.h>
// створюємо об'єкти для управління сервоприводами
Servo myservo1;
Servo myservo2;
void setup ()
{
  // підключаємо сервоприводи до виводів 11 і 12
  myservo1.attach (11);
  myservo2.attach (12);
}
void loop ()
{
  // встановлюємо сервоприводи в серединне положення
  myservo1.write (90);
  myservo2.write (90);
}
```

```

delay (500);
// встановлюємо сервоприводи в крайнє лїве положення
myservo1.write (0);
myservo2.write (0);
delay (500);
// встановлюємо сервоприводи в крайнє праве положення
myservo1.write (180);
myservo2.write (180);
delay (500);
}

```

3. Приклад

Наступний скетч постійно повертає насадку на 180 градусів і повертає її назад. У прикладі використовується вбудована бібліотека Servo.

Загальна схема (рис.7) - червоний дріт йде до живлення 5V, чорний або коричневий йде до GND, а жовтий або білий до виводу плати (в нашому прикладі вивід 9).

```

#include <Servo.h>
Servo myservo; // створимо об'єкт сервоприводу
int pos = 0; // початкова позиція
void setup () {
  myservo.attach (9); // сервопривід на виводі 9
}
void loop () {
  for (pos = 0; pos <= 180; pos + = 1) { // від 0 до 180 градусів // крок в один градус
    myservo.write (pos); // просимо повернутися на позицію значення змінної 'pos'
    delay (15); // чекаємо 15ms для досягнення позиції
  }
  for (pos = 180; pos > = 0; pos - = 1) { // від 180 до 0 градусів
    myservo.write (pos); // просимо повернутися на позицію значення змінної 'pos'
    delay (15); // чекаємо 15ms для досягнення позиції
  }
}
}

```


Робота з PIR-датчиком (руху) в Arduino

1. Основні поняття та фізичний зміст

Дослідимо датчик руху на основі піроелектричного ефекту (PIR, passive infrared motion sensor) (рис.8). Такі датчики часто використовуються в охоронних системах і в побуті для виявлення руху в приміщенні. Наприклад, на принципі визначення руху засноване автоматичне включення світла в під'їзді або у ванній. Піроелектричні датчики досить просто влаштовані, недорогі і невибагливі в установці і обслуговуванні.

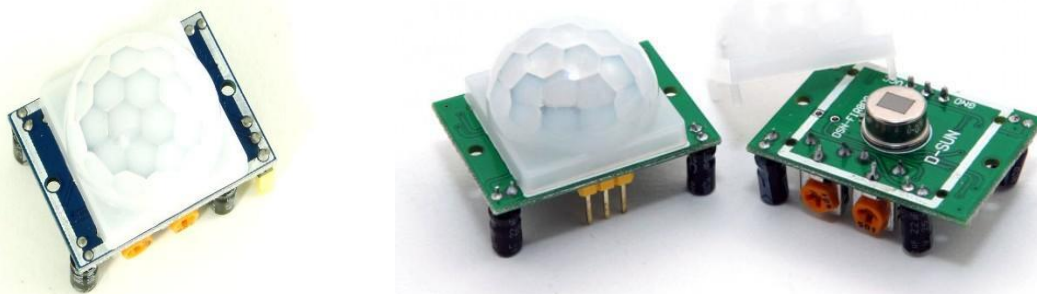


Рисунок 8. PIR-датчики

Піроелектрики - це діелектрики, які створюють електричне поле при зміні їх температури. На основі піроелектриків роблять датчики вимірювання температури. Кожен такий датчик містить два чутливих елемента розміром 1×2 мм, підключених з протилежного полярністю.

Працюватимемо з модулем датчика руху HC-SR501, в якому встановлений один такий піроелектричний датчик. Зверху піроелектрик оточений півсферою, розбитою на кілька сегментів. Кожен сегмент цієї сфери є лінзу, яка фокусує теплове випромінювання на різні ділянки ПІР-датчика. Часто в якості лінзи використовують лінзу Френеля.

Принцип роботи датчика руху наступний. Припустимо, що датчик встановлений в порожній кімнаті. Кожен чутливий елемент отримує постійну

дозу випромінювання, а значить і напруга на них має постійне значення (лівий малюнок на рис. 9).

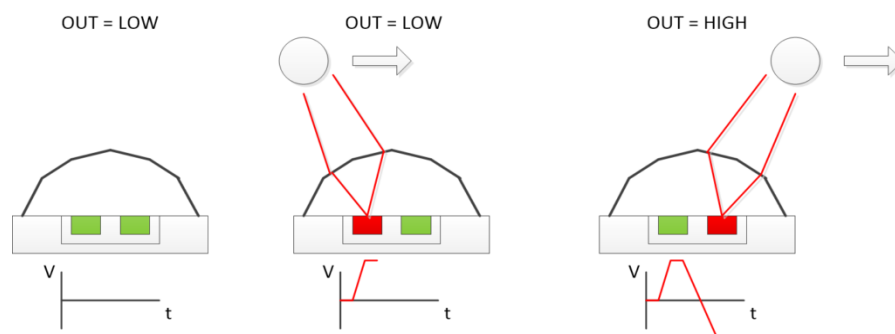


Рисунок 9. Принцип роботи датчика руху

Як тільки в кімнату заходить людина, він потрапляє спочатку в зону огляду першого елемента, що призводить до появи позитивного електричного імпульсу на ньому (центральний малюнок на рис. 9).

Людина рухається, і її теплове випромінювання через лінзи потрапляє вже на другий PIR-елемент, який генерує негативний імпульс. Електронна схема датчика руху реєструє ці різноспрямовані імпульси і робить висновки про те, що в поле зору датчика потрапила людина. На виході датчика генерується позитивний імпульс (правий малюнок на рис. 9).

У модуля датчика HC-SR501 (рис.10) є два змінних резистора і перемичка для настройки режиму. Один з потенціометрів регулює чутливість приладу. Чим вона більше, тим далі «бачить» датчик. Також чутливість впливає на розмір об'єкта, що детектується. Наприклад, можна виключити з спрацювання собаку або кішку.



Рисунок 10. Модуль датчика HC-SR501

Другий потенціометр регулює час спрацьовування T . Якщо датчик виявив рух, він генерує на виході позитивний імпульс довжиною T .

Нарешті, третій елемент управління - перемикач, який перемикає режим датчика. У положенні L датчик веде відлік T від самого першого спрацьовування. Припустимо, ми хочемо керувати світлом у ванній кімнаті. Зайшовши в кімнату, людина викличе спрацьовування датчика, і світло включиться рівно на час T . Після закінчення періоду, сигнал на виході повернеться в початковий стан, і датчик буде чекати наступного спрацьовування.

У положенні H датчик починає відлік часу T кожен раз після виявлення руху. Іншими словами, будь-яке ворухіння людини викличе обнулення таймера відліку T . По-замовчанню, перемикач знаходиться в стані H .

2. Схема підключення та приклад

Для з'єднання з мікро контролером або безпосередньо з реле у HC-SR501 є три виводи. Підключаємо їх до Arduino за схемою на рис.11.

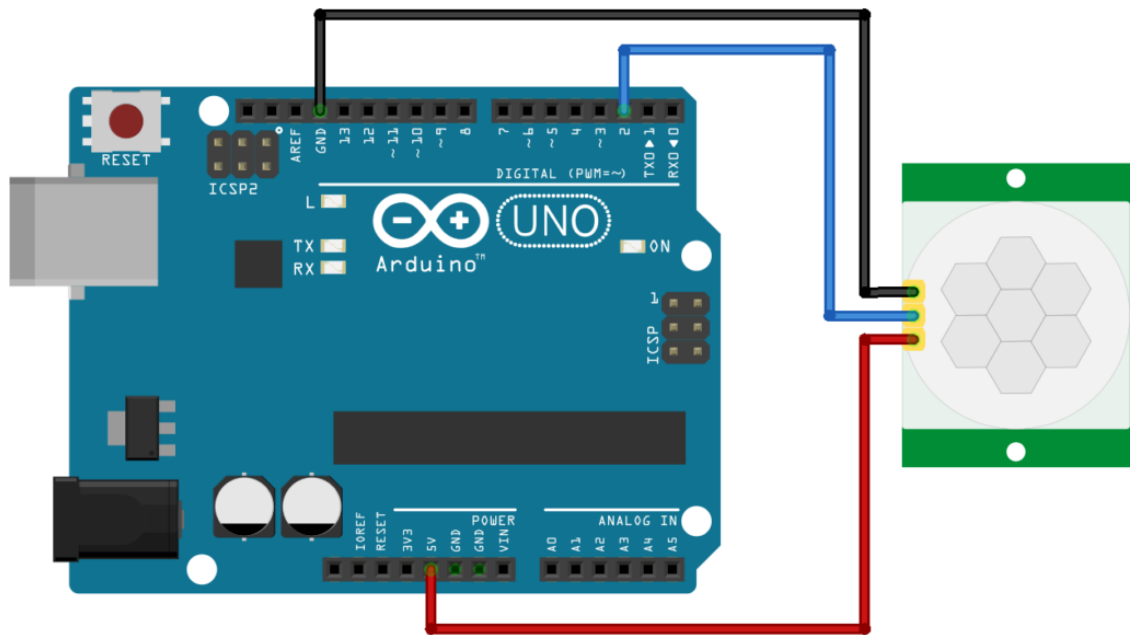


Рисунок 11. Схема підключення

Пам'ятаємо, що цифровий вихід датчика HC-SR501 генерує високий рівень сигналу при спрацьовуванні. Напишемо просту програму, яка буде відправляти в послідовний порт «1» якщо датчик побачив рух, і «0» в іншому випадку.

```
const int movPin = 2
void setup() {
  Serial.begin(9600);
  pinMode(movPin, INPUT);
}
void loop(){
  int val = digitalRead(movPin);
  Serial.println(val);
  delay(100);
}
```

Завантажуємо програму на Arduino і перевіряємо роботу датчика. Можна покрутити налаштування датчика і подивитися як це відіб'ється на його роботі.

Тепер ми можемо створити систему автоматичного включення світла. Для того, щоб керувати освітленням в приміщенні, нам буде потрібно додати в ланцюг реле, як представлено на рис.12.

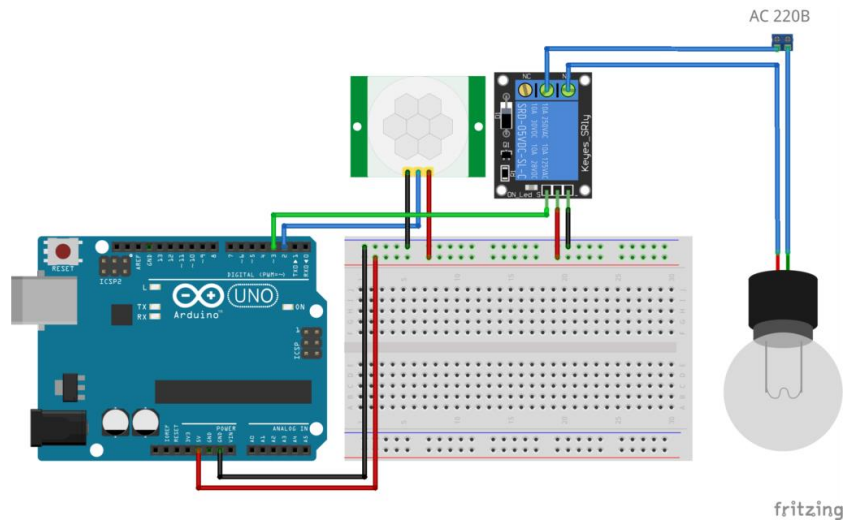


Рисунок 12. Схема системи автоматичного включення світла

Тепер напишемо скетч, який буде при спрацьовуванні датчика включати реле, а отже і освітлення в кімнаті.

```
const int movPin = 2;
const int relPin = 3;
void setup() {
  Serial.begin(9600);
  pinMode(movPin, INPUT);
  pinMode(relPin, OUTPUT);
}
void loop(){
  int val = digitalRead(movPin);
  if (val)
    digitalWrite(relPin, HIGH);
  else
    digitalWrite(relPin, LOW);
}
```

Завантажуємо програму на Arduino, обережно підключаємо схему до побутової мережі і перевіряємо роботу датчика. Дана схема запалює лампу від мережі 220 Вольт. Рекомендується сім раз перевірити всі з'єднання, перш ніж з'єднувати схему з побутовою електромережею.

Як ми бачимо, датчики руху досить корисні і оточують нас всюди. Завдяки охоронним системам, їх можна зустріти практично в кожному приміщенні. Як ми з'ясували, вони дуже прості у використанні і можуть бути легко інтегровані в будь-який проект на Arduino.

Ось кілька ситуацій і місць, де може стати в нагоді датчик руху:

- автоматичне включення світла в під'їзді будинку, у ванній кімнаті і туалеті, перед входними дверима в приміщення;
- сигналізація в приміщенні і на подвір'ї;
- автоматичне відкривання дверей;
- автоматичне включення охоронної відеокамери.

Дистанційне керування в Arduino

1. Основні поняття та фізичний зміст

Зв'язок через ІЧ-порт - поширена, недорога і проста у використанні технологія бездротового зв'язку. Світло ІЧ-діапазону дуже схоже на видиме світло, за винятком того, що воно має дещо більшу довжину хвилі. Це означає, що ІЧ невидиме людським оком - ідеальне рішення для бездротового зв'язку. Наприклад, коли ви натискаєте кнопку на пульті від телевізора (рис.13), інфрачервоний світлодіод вмикається і вимикається не один раз, а 38000 разів на секунду, щоб передавати інформацію до інфрачервоного датчика на телевізорі (наприклад, контролювати гучність або канал).

Для того щоб дізнатися, який протокол використовується в пульті дистанційного управління, потрібно зібрати схемку з ІЧ-приймачем, зняти осциллограму сигналу і порівняти її з відомими протоколами.

ІК-приймач розрахований на роботу з однією несучою частотою, а в деяких протоколах значення несучих частот відрізняються - одні використовують 38 кГц (наприклад, протокол фірми Nec), інші 36 або 40. Тому одна і та ж схема не зможе працювати з усіма без винятку ІК-пультами дистанційного керування.

Одним з найпоширеніших протоколів роботи ІЧ ДК є протокол NEC. Для передачі даних використовується несуча частота 38 кГц. Посилка складається з стартового імпульсу і чотирьох байтів даних - адреса, інвертоване значення команди. Адреса та команда передаються двічі для підвищення надійності.



Рисунок 13. ІЧ-пульт

2. Схема підключення

Вибираючи фотоприймач (рис. 14), розглядаємо лише моделі, в назві якої в кінці буде число 38 (TSOP4838, 31238, 2138, 1738 ...), тобто частота модуляції сигналу 38 кГц. Дану частоту підтримує більшість протоколів ДК.

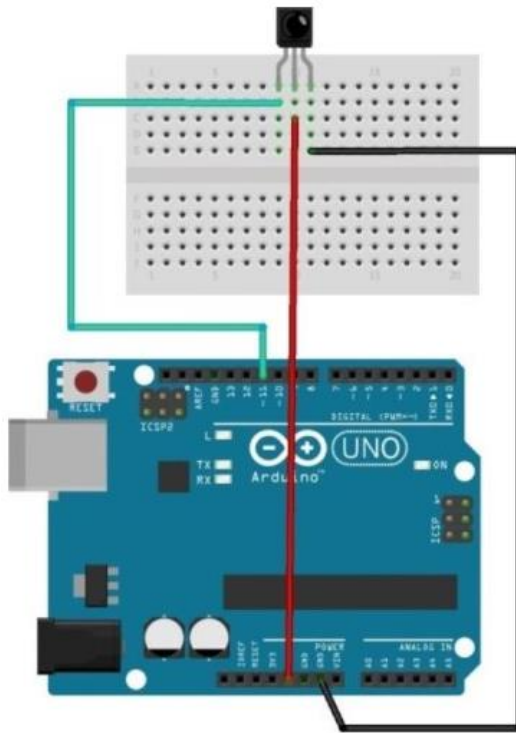


Рисунок 14. Схема підключення фотоприймача

3. Приклад

Створимо скетч, для зчитування кнопок ІЧ-пультя, та виводу їх на монітор порту. Після підключення нам необхідно завантажити бібліотеку для реалізації дистанційного керування. Бібліотека доступна за посиланням: <https://github.com/shirriff/Arduino-IRremote>.

Далі, для інсталяції бібліотеки, в новому скетчі виберіть Sketch->Import Library->IRremote.

Код наведений нижче:

```
#include <IRremote.h>//Підключення бібліотеки IRremote.h
int RECV_PIN = 11;// Назначаємо вивід 11 для приєднання виходу приймача
IRrecv irrecv(RECV_PIN);
decode_results results;
void setup()
{
  Serial.begin(9600);// Задаємо швидкість передачі послідовного порта
```

```

irrecv.enableIRIn();// Дозволяємо роботу приймача
}

// Створюємо серію "dump", в якій програма буде порівнювати результати, отримані з
// даними пультом і даними файлу IRremote.h. Якщо дані збігаються, програма покаже вам,
// який тип протоколу.

void dump(decode_results *results) {
int count = results->rawlen;
if (results->decode_type == UNKNOWN) {
Serial.println("Could not decode message");
}
else {
if (results->decode_type == NEC) {
Serial.print("Decoded NEC: ");
}
else if (results->decode_type == SONY) {
Serial.print("Decoded SONY: ");
}
else if (results->decode_type == RC5) {
Serial.print("Decoded RC5: ");
}
else if (results->decode_type == RC6) {
Serial.print("Decoded RC6: ");
}
Serial.print(results->value, HEX);
Serial.print(" (");
Serial.print(results->bits, DEC);
Serial.println(" bits)");
}
Serial.print("Raw (");
Serial.print(count, DEC);
Serial.print("): ");
for (int i = 0; i < count; i++) { if ((i % 2) == 1) { Serial.print(results-
>rawbuf[i]*USECPERTICK, DEC);

```

```
}  
else {  
Serial.print(-(int)results->rawbuf[i]*USECPERTICK, DEC);  
}  
Serial.print(" ");  
}  
Serial.println("");  
}  
void loop()  
{  
// У цьому циклі програма представлятиме коди, отримані за допомогою натискання  
//клавiш пульта ДК, які потiм можна використати в iншому проєкті  
if (irrecv.decode(&results))// Перевiряємо чи прийнятий код  
{  
Serial.println(results.value, HEX);  
dump(&results);  
irrecv.resume();// Приймаємо наступне значення  
}  
}
```

Результат роботи програми (рис. 15):

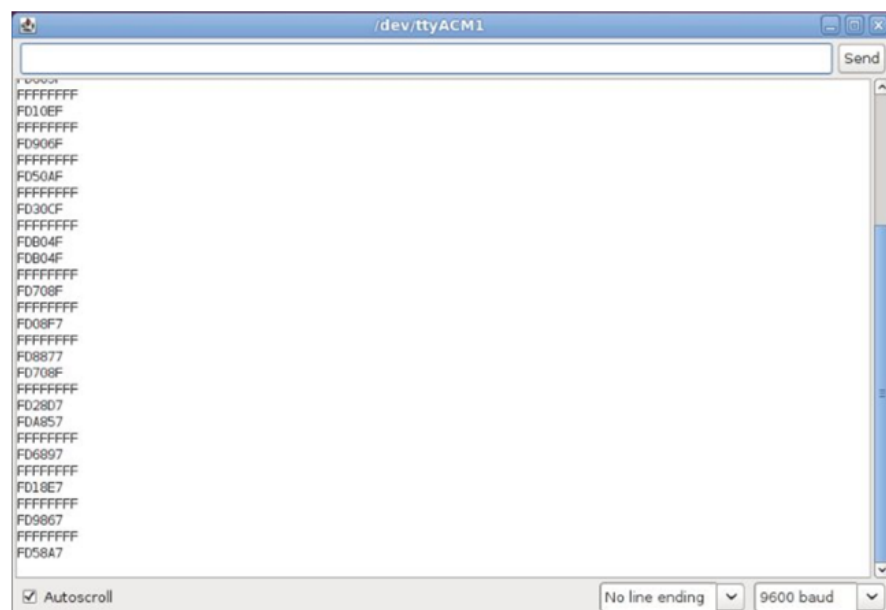


Рисунок 15. Результат роботи програми

Необхідно записати всі числа, які відповідають кнопкам. Наприклад, для пульта ДК, який був зображений на рис.13, коди будуть наступними:

```
PWR FF629D
CH FFE21D
|<< FF22DD
>| FFC23D
>>| FF02FD
- FFE01F
Plus FFA857
EQ FF906F
0 FF6897
100+FF9867
200+FFB04F
1 FF30CF
2 FF18E7
3 FF7A85
4 FF10EF
5 FF38C7
6 FF5AA5
7 FF42BD
8 FF4AB5
9 FF52AD
```

Робота з датчиком температури та вологості Arduino

1. Основні поняття та фізичний зміст

DHT11 - це цифровий датчик, що складається з термістора і ємнісного датчика вологості (рис.24). Поряд з невисокою вартістю DHT11 має наступні характеристики: живлення здійснюється від 3,5-5V, визначення температури від 0 до 50 градусів з точністю 2 градуса, визначення вологості від 20% до 95% з 5% точністю.

Термістор - це термічний резистор, опір якого змінюється з температурою, тобто збільшення температури призводить до падіння його опору. По суті термістор - це термометр опору, виготовлений на основі змішаних оксидів перехідних металів. Відноситься до вимірювальної техніки і може бути використаний для автоматичного вимірювання температури в різних середовищах.

Ємнісний датчик вологості - це конденсатор зі змінною ємністю, який містить струмопровідні обкладання з мідної фольги на текстоліті. Цей конденсатор укладений в герметичний чохол, поверх якого розташований вологи поглинаючий шар. При попаданні частинок води на цей шар, змінюється його діелектрична проникність, що приводить до зміни ємності конденсатора.

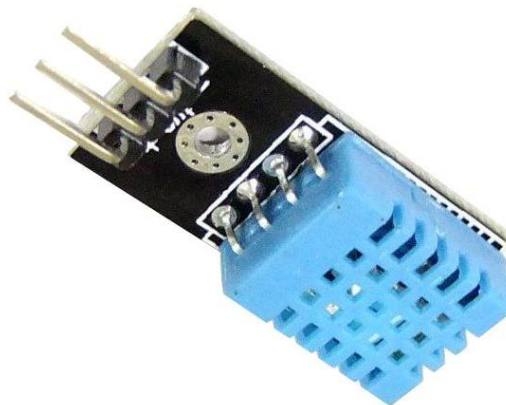


Рисунок 24. DHT11

2. Схема підключення

Для підключення датчика необхідно підключити бібліотеку <DHT.h>. Модуль DHT11 обладнаний трьох піновим роз'ємом і підключається за схемою (рис.25):

- G-підключається до GND
- V-підключається до +5V
- S-підключається до цифрового Pin2

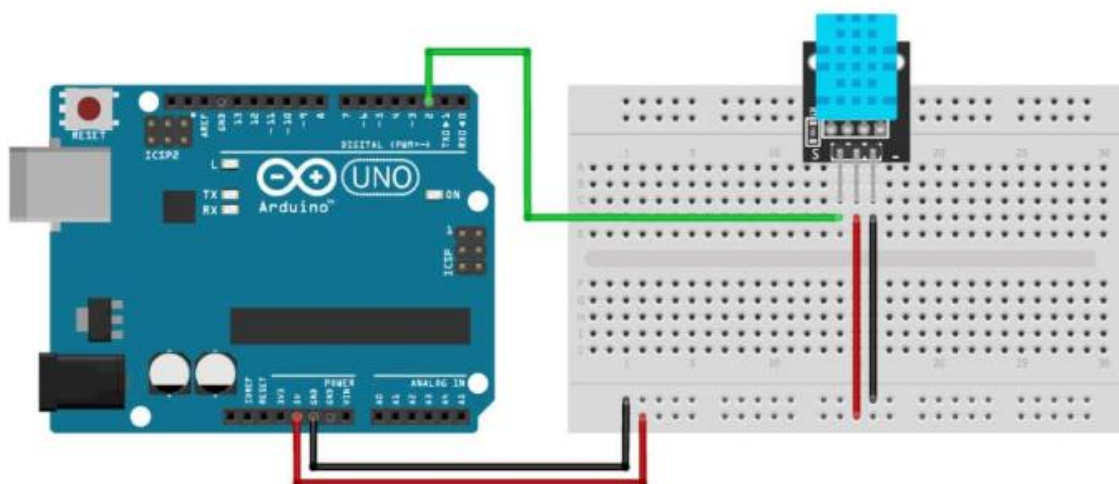


Рисунок 25. Схема підключення DHT11

3. Приклад

Наведемо приклад скетчу, для зчитування температури та вологості з датчика.

Підключаємо бібліотеку:

```
DHT.h.
```

Далі вказуємо порт з якого будемо зчитувати:

```
DHT dht(2, DHT11).
```

Запускаємо датчик:

```
dht.begin();
```

Підключаємо монітор порту:

```
Serial.begin(9600);
```

Зчитуємо температуру і вологість:

```
float h = dht.readHumidity();  
float t = dht.readTemperature();
```

Та виводимо на монітор порту:

```
Serial.print("Humidity: ");  
Serial.println(h);  
Serial.print("Temperature: ");  
Serial.println(t);
```

Робота з RFID

1. Основні поняття та фізичний зміст

Абревіатура RFID розшифровується як Radio Frequency Identification (в перекладі з англійської: радіочастотна ідентифікація). RFID (метод радіочастотної ідентифікації) - технологія, яка для автоматичної ідентифікації об'єктів використовує радіохвилі.

Системи RFID застосовуються в додатках контролю доступу, в додатках контролю і обліку робочого часу, при ідентифікації транспортних засобів, в автоматизації виробництва, в автоматизації складської обробки.

В склад системи RFID входять:

- RFID-Зчитувач;
- RFID-мітка;
- Програмне забезпечення.

Зчитувач займається генеруванням і поширенням електромагнітних хвиль в навколишній простір. Даний сигнал приймається RFID-міткою, яка створює зворотний сигнал, вловлювати антеною пристрою, що зчитує, потім отримана інформація розшифровується і обробляється електронним блоком. Об'єкт, оснащений RFID-міткою, ідентифікується за допомогою унікального цифрового коду, який зберігається в пам'яті електронної мітки. Наприклад, можна в лічені секунди отримати індивідуальні дані користувача або ідентифікаційний номер того чи іншого товару.

Основа роботи технології: взаємодія RFID-мітки (RFID-тега) і RFID-зчитувача (RFID-рідера). RFID-мітка - мініатюрний чіп, який зберігає унікальний номер тега і інформацію і має можливість для передачі даних RFID-рідера. Як тільки RFID-мітка потрапляє в зону дії RFID-рідера, рідер фіксує факт передачі даних, зчитує інформацію з мітки і передає її в облікову систему, яка аналізує дані по заздалегідь заданими алгоритмами.

При цьому між RFID-міткою і RFID-рідером може бути відстань до 300 метрів (системи, що працюють на відстані від 5 до 300 метрів відносять до

систем дальньої ідентифікації, від 20 см до 5 м - ідентифікації середньої дальності, до 20 см - системи ближньої ідентифікації).

Основна використовувана класифікація RFID-міток заснована на джерелі живлення - згідно з нею, теги діляться на пасивні, активні і напівпасивну.

Пасивні RFID-мітки не мають власного джерела живлення і використовують для роботи енергію поля зчитувача. Залежно від архітектури RFID-мітки і типу рідера, пасивні теги працюють тільки на невеликій відстані - до 8 метрів, але при цьому відрізняються компактністю і доступною ціною.

Саме пасивні низькочастотні RFID-мітки найбільш часто зустрічаються нам на товарах в магазинах - над підвищенням компактності тегів і зниженням їх вартості працюють представники провідних світових торгових мереж.

Активні RFID-мітки оснащені власним джерелом живлення, тому можуть отримати додаткові функції, працюють на більшій відстані і менш вимогливі до зчитувача. До їх недоліків, в порівнянні з пасивними мітками, можна віднести великий розмір і обмежений час роботи джерела живлення (правда, на сьогоднішній день мова йде про термін життя батареї до 10 років), однак вони незамінні там, де необхідний великий радіус роботи (до 300 метрів).

Активні RFID-мітки по праву вважаються надійнішими, вони можуть передавати сигнал навіть через воду або метал, а також їх можна оснастити вбудованими сенсорами для оцінки температури, вологості, рівня освітленості та інших параметрів навколишнього середовища. Таким чином, RFID-мітки можуть допомогти відстежувати, наприклад, дотримання умов зберігання певних категорій товарів.

Напівпасивні RFID-мітки працюють за тим же принципом, що і пасивні, але оснащені батареєю для живлення чіпа. Можна сказати, що таке рішення є компромісним в плані вартості, розміру і характеристик RFID-міток.

За виконання RFID-мітки можуть являти собою пластикові картки, брелоки, корпусні мітки, а також самоклеючі етикетки з паперу або

термопластика. Існує також формат «невидимої» етикетки, яка фактично вшивається в упаковку товару безпосередньо на етапі виробництва.

За типом пам'яті RFID-мітки діляться на призначені тільки для ідентифікації (RO, Read Only), розроблені для зчитування блоку інформації (WORM, Write Once Read Many) і перезапису (RW, Read and Write).

RO RFID-мітки використовуються виключно для ідентифікації - дані унікального ідентифікатора записуються при виготовленні тега, тому скопіювати їх і підробити мітку практично неможливо.

WORM RFID-мітки дозволяють одноразово записати будь-які дані, які згодом можна буде багаторазово зчитувати і використовувати. Це дозволяє користувачеві при отриманні доповнити мітку своєю інформацією, яка потім буде використовуватися при зчитуванні.

RW RFID-мітки містять блок пам'яті, який дозволяє багаторазово записувати і зчитувати інформацію. Ідентифікатор RFID-мітки при цьому залишається незмінним.

2. Схема підключення

Призначення виводу інтерфейсу SPI (рис. 26):

- SDA - вибір ведомого;
- SCK - сигнал синхронізації;
- MOSI - передача від master до slave;
- MISO - передача від slave до master;
- RST - вивід для скидання;
- IRQ - вивід переривання;
- GND - земля;
- Vcc - живлення 3.3 В.

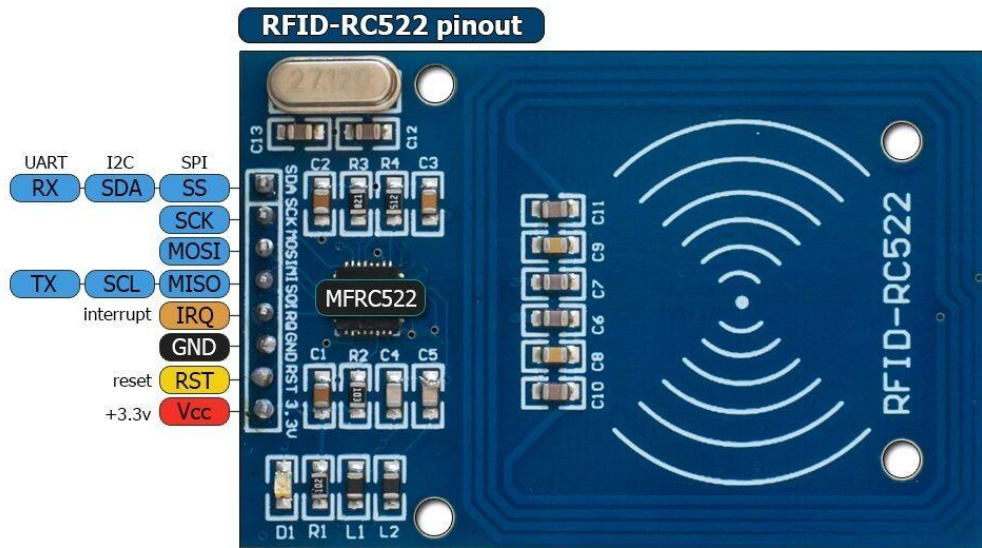


Рисунок 26. RFID модуль RC522 – призначення виводів

Сигнал скидання RST - це сигнал, що надходить від цифрового виходу контролера. При надходженні сигналу LOW відбувається перезавантаження зчитувача. Також рідер установкою на RST низького рівня повідомляє, що знаходиться в режимі сну, для виведення модуля з режиму сну необхідно подати на даний висновок сигнал HIGH.

Схема з'єднань для підключення RFID модуль RC522 до плати Arduino (рис. 27):

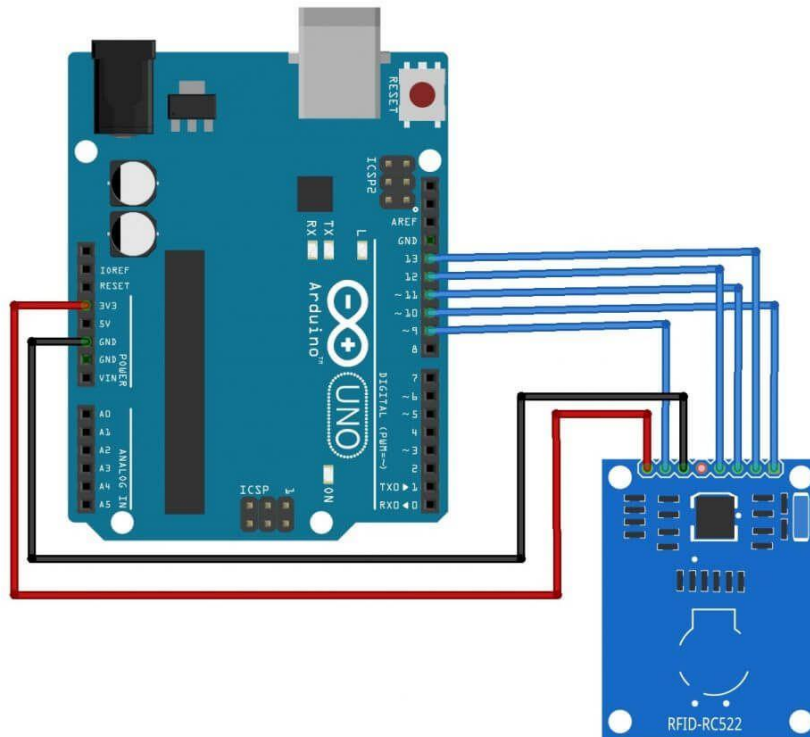


Рисунок 27. Схема з'єднань для підключення RFID модуля RC522 до плати Arduino

3. Приклад

Наведемо приклад скетчу для обзору модуля RFID RC522:

```
// Скетч для обзору модуля RFID RC522
// Вивід UID и типу мітки
// Підключення бібліотек
#include <SPI.h>
#include <MFRC522.h>
// контакти
const int reset  9
const int ss    10
// Створення екземпляру об'єкта MFRC522
MFRC522 rider(reset, ss);
void setup() {
// запуск Serial
```

```
Serial.begin(9600);
// запуск SPI
SPI.begin();
// ініціалізація MFRC522
rider.PCD_Init();
}
void loop() {
// Очікування
if ( ! rider.PICC_IsNewCardPresent())
return;
// читання
if ( ! rider.PICC_ReadCardSerial())
return;
// вивід даних
Serial.print("UID=");
view_data(rider.uid.uidByte,rider.uid.size);
Serial.println();
Serial.print("type=");
byte piccType = rider.PICC_GetType(rider.uid.sak);
Serial.print(rider.PICC_GetTypeName(piccType));
Serial.println();
delay(1000);
}
// перетворення в HEX
void view_data (byte *buf, byte size) {
for (byte j = 0; j < size; j++) {
Serial.print(buf [j]);
Serial.print(buf [j], HEX);
}
}
```

Робота з LCD дисплеєм в Arduino

1. Основні поняття та фізичний зміст

Рідкокристалічний дисплей - це електронний пристрій візуального відображення інформації, принцип дії якого ґрунтується на явищі електричного переходу Фредерікса в рідких кристалах. Дисплей складається з довільної кількості кольорових або монохромних точок (пікселів), і джерела світла або відбивача (рефлектора).

Дисплей є досить невеликий (за мірками дисплеїв) модуль (рис. 28), із приблизними розмірами 8×3 см, закріплений на невеликій підкладці з контактами для підключення.

До його основних переваг можна віднести низьку вартість, чітке відображення, витримку перепадів температур (від -20 до $+70$ °C), широкий кут огляду, регульована контрастність і малі розміри.

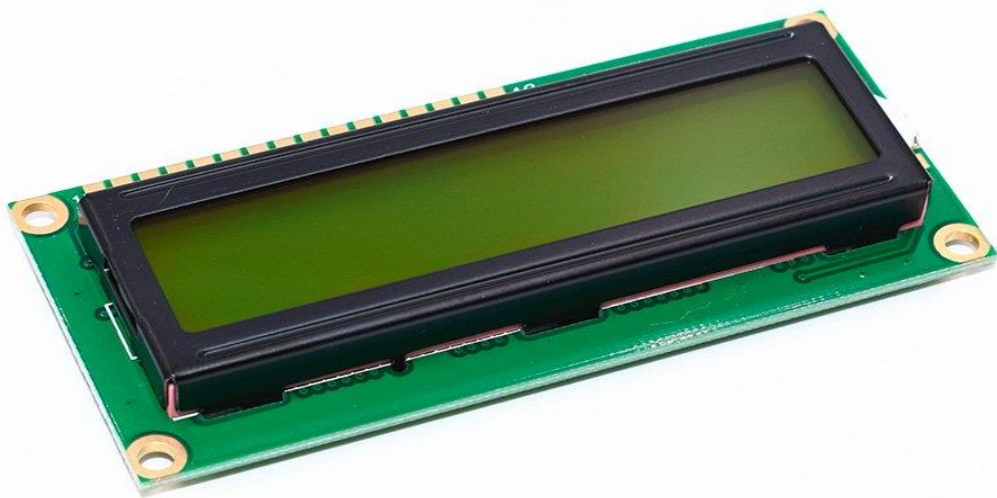


Рисунок 28. Рідкокристалічний дисплей Arduino

2. Схема підключення

Дисплей має 16 контактів для підключення:

1 GND (земля)

- 2 +5 живлення
- 3 VO (установка контрастності)
- 4 RS (команди і дані)
- 5 RW (читання і запис)
- 6 E (Enable)
- 7-14 (DB0 ... DB7) лінія даних
- 15 A (плюс підсвічування)
- 16 K (мінус підсвічування)

Також для підключення знадобиться змінний резистор (потенціометр) на 10 кОм (рис. 29).

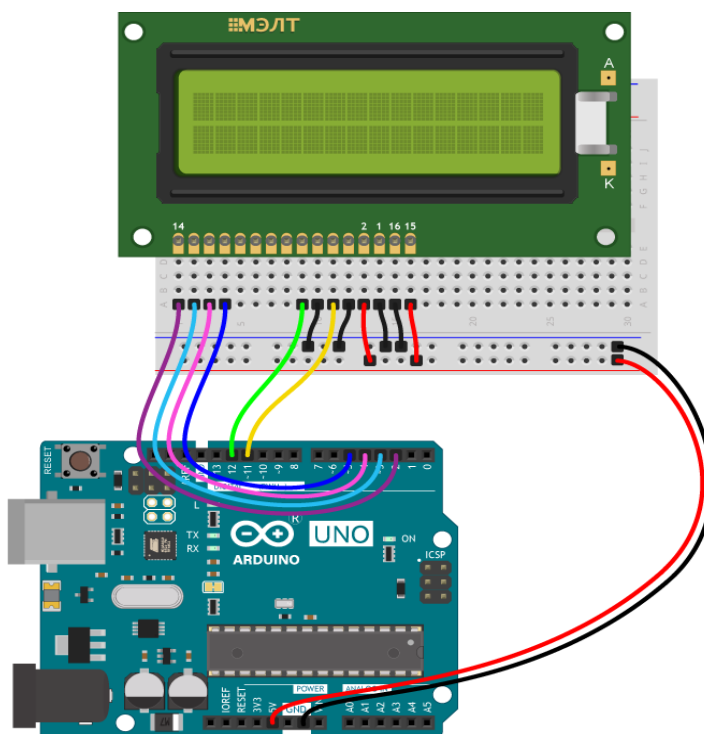


Рисунок 29. Схема підключення

Дисплей є досить невеликий (за мірками дисплеїв) модуль (рис.?), із приблизними розмірами 8×3 см, закріплений на невеликій підкладці з контактами для підключення.

Для зручності підключення прийнято використовувати інтерфейс I2C (рис. 30). Завдяки даній шині число пінів, використовуваних на підключення дисплея, зводиться до чотирьох.

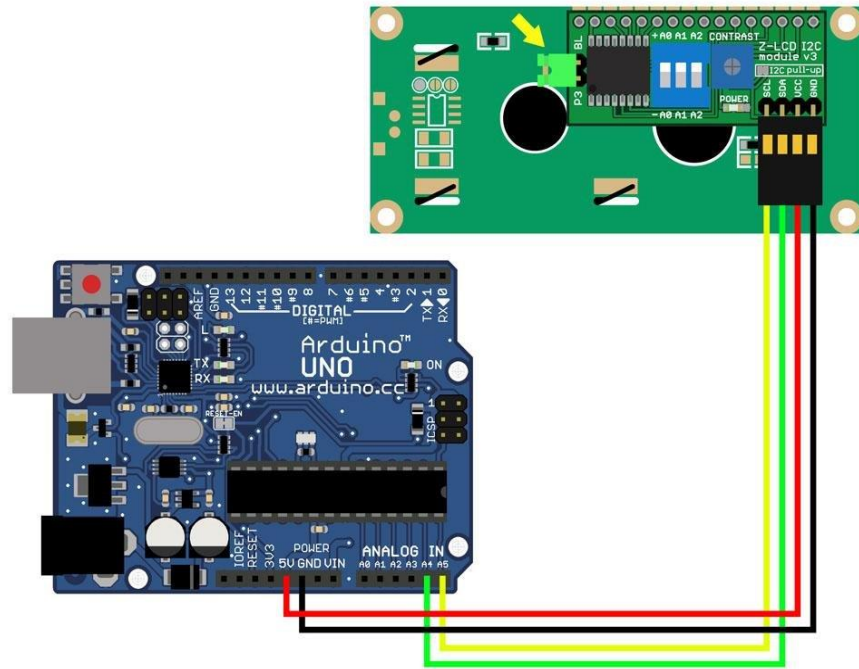


Рисунок 30. Схема підключення з використанням інтерфейсу I2C

Для роботи дисплея використовується вбудована в середу Arduino IDE бібліотека *LiquidCrystal.h*.

Також бібліотека *LiquidCrystal.h* має досить широкий функціонал, який, безсумнівно, стане в нагоді при подальшій роботі з цим модулем:

Робота з курсором:

```

lcd.setCursor(0, 0);           //Встановлюємо курсор (номер комірки, рядок)
lcd.home();                   //Установка курсора в нуль (0, 0)
lcd.cursor();                 //Включити видимість курсора (підкреслення)
lcd.noCursor();              //Прибрати видимість курсора (підкреслення)
lcd.blink();                  //Включити миготіння курсору (5x8)
lcd.noBlink();               //Вимкнути миготіння курсору (5x8)

```

Вивід інформації:

```

lcd.print("1");              //Вивід інформації

```


| | |
|--|--|
| <code>lcd.clear();</code> | //Очищення дисплея, (видалення всіх даних) установка курсору в нуль |
| <code>lcd.rightToLeft();</code> | //Запис проводиться з права на ліво |
| <code>lcd.leftToRight();</code> | //Запис проводиться зліва на право |
| <code>lcd.scrollDisplayRight();</code> | //Зсув всього зображеного на дисплеї на один символ вправо |
| <code>lcd.scrollDisplayLeft();</code> | //Зсув всього зображеного на дисплеї на один символ вліво |

3. Приклад

Якщо ви підключаєте дисплей з i2c протоколу, то краще використовувати бібліотеку *LiquidCrystal_I2C.h*.

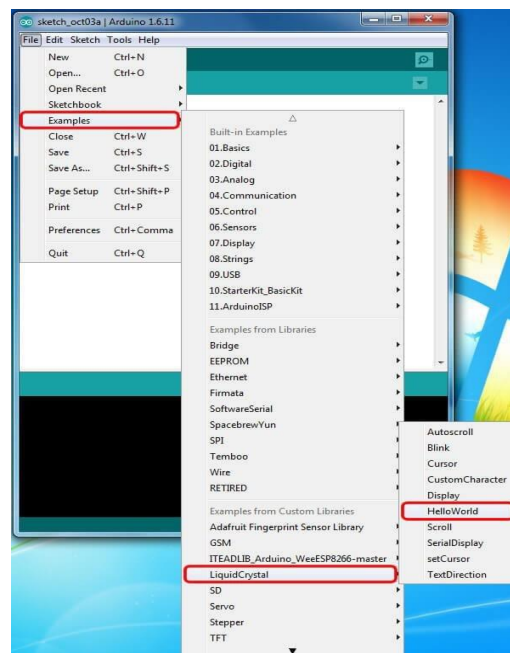


Рисунок 31. Шлях до прикладів

Підключивши даний модуль до Arduino, можна перевірити його працездатність, завантаживши програмний код із прикладів, які вже завантажені в середу розробки (рис. 31).

Програма "HelloWorld" виводить текст в лапках на дисплей і є найпростішою для роботи з дисплеєм.

```
// Додаємо необхідну бібліотеку
#include <LiquidCrystal.h>
// (RS, E, DB4, DB5, DB6, DB7)
LiquidCrystal lcd(7, 6, 5, 4, 3, 2);
void setup(){
// Задаємо розмірність екрану
lcd.begin(16, 2);
// Встановлюємо курсор в початок 1 рядка
lcd.setCursor(0, 0);
// Виводимо текст
lcd.print("Hello, world!");
// Встановлюємо курсор в початок 2 рядка
lcd.setCursor(0, 1);
// Виводимо текст
lcd.print("1");
}
}
```

Робота з ультразвуковим далекоміром HC-SR04 в Arduino

1. Основні поняття та фізичний зміст

Далекомір - це пристрій для вимірювання відстані до певного об'єкта. *Ультразвуковий далекомір HC-SR04* здатний вимірювати відстань від 1-2 см до 4-6 метрів. При цьому точність вимірювання становить 0,5 - 1 см.

Ультразвуковий далекомірний модуль HC-SR04 для Arduino (рис. 32) - це поміщені на одну плату приймач і передавач ультразвукового сигналу. Принцип дії HC-SR04 заснований на добре відомому явищі ехолокації. Випромінювач формує акустичний сигнал, який відбившись від перешкоди, повертається до датчика і реєструється приймачем. Знаючи швидкість поширення ультразвуку в повітрі і час запізнювання між випроміненим і прийнятим сигналом, легко розрахувати відстань до акустичної перешкоди. На відміну від інфрачервоних далекомірів на ультразвуковий датчик HC-SR04 не впливають джерела світла або колір перешкоди.



Рисунок 32. Ультразвуковий далекомір HC-SR04.

Технічні характеристики ультразвукового далекоміра HC-SR04:

- вимірюваний діапазон - від 2 до 500 см;
- точність - 0,3 см;
- ефективний робочий кут - $<15^\circ$;

- кут вимірювань: 30 градусів;
- напруга живлення - 5 В.
- Сила струму спокою: <math>< 2 \text{ mA}</math>;
- Робоча сила струму: 15 мА ;: <math>< 15^\circ</math>;
- ширина імпульсу тригер: 10 мікросекунд;
- розміри: 45 мм x 20 мм x 15 мм.

2. Схема підключення

Підключається ультразвуковий датчик до Arduino за допомогою чотирьох виходів: два виходи живлення (вихід 5 вольт і gnd - заземлення) і два виходи вхідного TRIG (T) і ЕCHO (R) вихідного сигналів. Схему підключення можемо розглянути на рисунку 33.

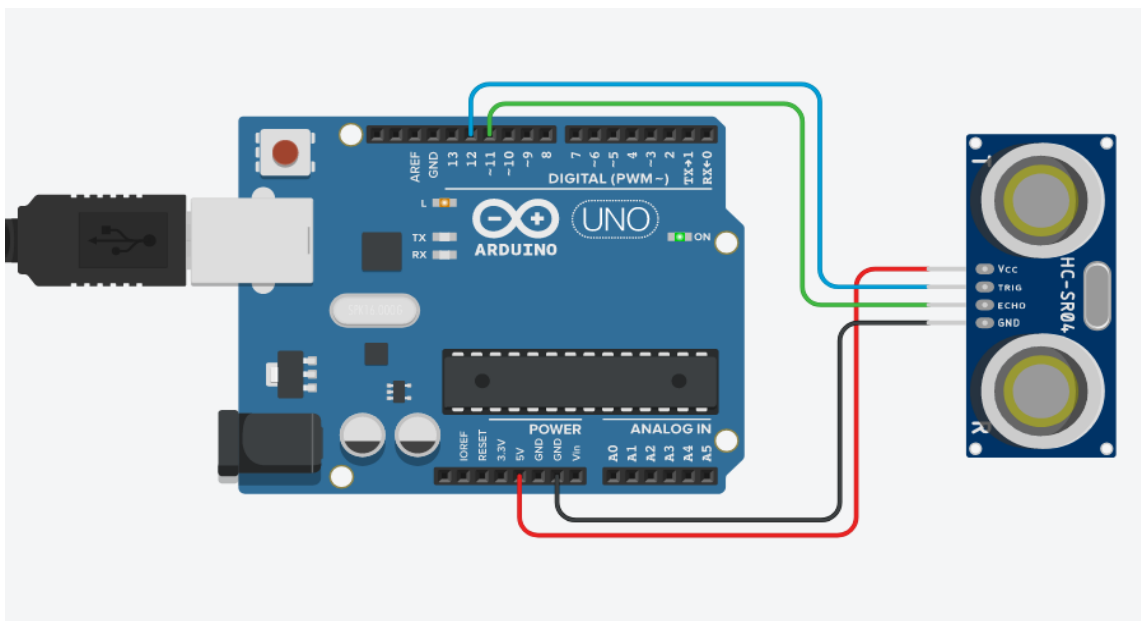


Рисунок 33. Схема підключення ультразвукового далекоміра HC-SR04

3. Приклад

Давайте тепер підключимо ультразвуковий далекомір HC-SR04 до Arduino так, щоб наш далекомір виконував свою функцію – вимірював

відстань до об'єкта. Будемо використовувати схему підключення, зображену на рис.33.

Для початку вимірювання нам необхідно згенерувати на виведення *Trig* позитивний імпульс довжиною 10 мкс. Слідом за цим, датчик випустить серію з 8 імпульсів і підніме рівень на виводі *Echo*, перейшовши при цьому в режим очікування відбитого сигналу. Як тільки далекомір відчує, що звук повернувся, він завершить позитивний імпульс на *Echo*.

Виходить, що нам потрібно зробити всього дві речі: створити імпульс на *Trig* для початку вимірювання, і заміряти довжину імпульсу на *Echo*, щоб потім обчислити дистанцію за простою формулою.

Функція *pulseIn* заміряє довжину позитивного імпульсу на нозі *echoPin* в мікросекундах. У скетчі ми записуємо час польоту звуку в змінну *duration*.

Скетч для стандартної роботи (вимірювання дистанції до об'єкта) ультразвукового далекоміра HC-SR04 на Arduino Uno буде виглядати наступним чином:

```
int trigPin = 11
int echoPin = 12;
long duration, cm, inches;
void setup() {
  Serial.begin (9600);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
}
void loop()
{
  digitalWrite(trigPin, LOW);
  delayMicroseconds(5);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  pinMode(echoPin, INPUT);
  duration = pulseIn(echoPin, HIGH);
```

```
cm = (duration/2) / 29.1;
inches = (duration/2) / 74;
Serial.print(inches);
Serial.print("in, ");
Serial.print(cm);
Serial.print("cm");
Serial.println();
delay(250);
}
```

Обробка даних від Джойстика. Управління Pan/Tilt/Bracket за допомогою джойстика в Arduino

1. Основні поняття та фізичний зміст

Джойстик - пристрій-руків'я керування у відеоіграх: важіль на підставці, який можна відхиляти у двох площинах. На важелі можуть бути різного роду гашетки і перемикачі. Словом «джойстик» в побуті називають орган управління, наприклад, в мобільному телефоні. Подібні графічні маніпулятори з'явилися водночас з електронними дисплеями в радіолокаторах, їх прототипом була ручка керування літаком.

Джойстик є одним з пристроїв для зручної передачі інформації від людини до комп'ютера або мікроконтролеру. Джойстики використовуються для управління рухом роботів, мобільних платформ і інших механізмів.

Модуль двовісний джойстик (рис. 34) має два ступені свободи, являє собою ручку, закріплену на кульовому шарнірі з двома взаємно перпендикулярними осями.



Рисунок 34. Джойстик

При нахилі ручки обертаються рухливі контакти кожного з двох потенціометрів номіналом 10 кОм, які визначають положення осей X і Y. Середній контакт кожного потенціометра виведений на контакти VRX і VRY роз'єму, а крайні підключені до живлення і землі. Також джойстик оснащений

тактовою кнопкою, яка спрацьовує при вертикальному натисканні на ручку, показання знімаються з контакту SW. Після відпускання джойстик зайняв свою попередню центральне стан.

2. Схема підключення

Джойстик ідеально підходить для управління пристроєм на базі Arduino, який рухається в двовимірній системі координат. Для плат Arduino є аналогові джойстикові модулі, які мають X, Y (10 кОм потенційнихометрів) і додаткову кнопку - вісь Z. Джойстик дозволяє плавно і точно відстежувати ступінь відхилення від нульової точки. Сам джойстик пружний, тому повернеться в центральний стан після того, як його звільнить з певної позиції. Таким чином, вам потрібно 5 проводів для підключення: Вісь X, вісь Y, кнопка, потужність VCC і загальний GND. Джойстики є пасивними модулями і не споживають ніякої енергії з дошки Arduino. Висновки VERT і HOR' підключені до аналогових входів A0 і A1 Arduino, SEL до цифрового входного D2. Схема з'єднання відображається на рисунку 35.

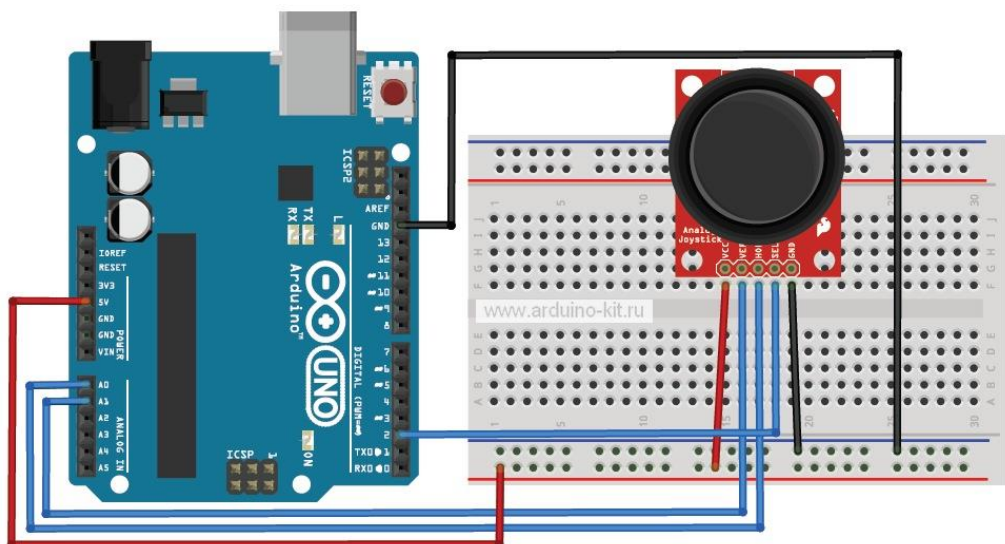


Рисунок 35. Схема підключення джойстика до Arduino

3. Приклад

Давайте підключимо джойстик як на рисунку 36 , та напишемо скетч отримання даних з джойстика після завантаження якого на плату Arduino, зможемо відкривши монітор послідовного порта відслідковувати дані про зміну положення джойстика.

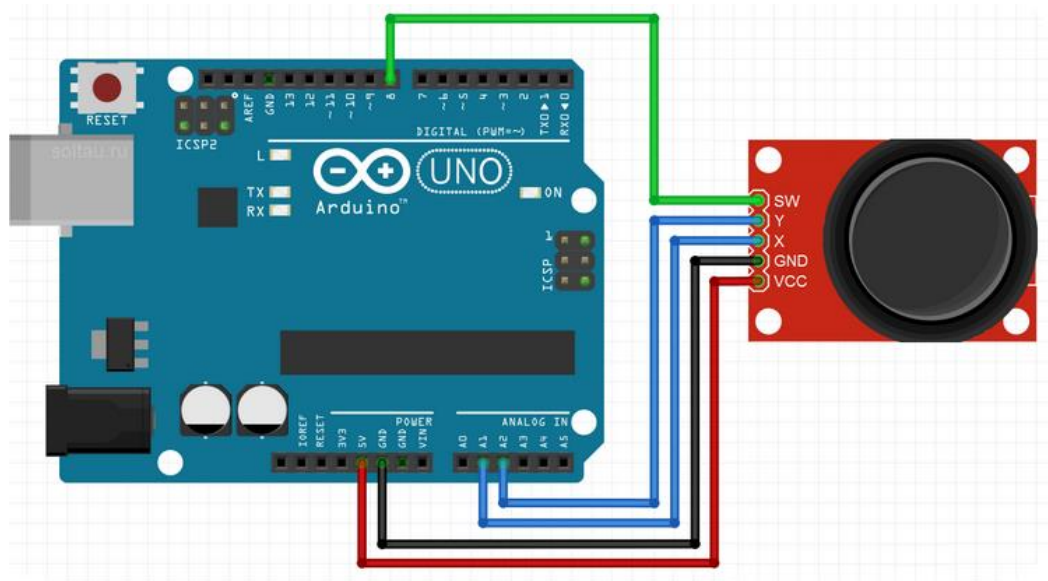


Рисунок 36. Схема підключення модуля джойстика до плати Arduino Uno

Дані з потенціометрів по осях X і Y можуть набувати значень від 0 до 1023. Нерухомому положенні джойстика відповідають значення 511 для кожного потенціометра. При натисканні на кнопку на вході 3 Arduino буде з'являтися 0. Щоб не було наведень, висновок кнопки необхідно підтягнути до +5 В. Дані виводимо в послідовний порт.

```
// пін підключення контакта VRX
#define PIN_VRX A0

// пін підключення контакта VRY
#define PIN_VRY A1

// пін підключення кнопки
#define PIN_BUTTON 3
```

```
void setup () {  
    // запуск послідовного порта  
    Serial.begin (9600);  
}  
void loop () {  
    // Виводимо значення по осі X  
    Serial.print("X = ");  
    Serial.println(analogRead(PIN_VRX));  
    // Виводимо значення по осі Y  
    Serial.print("Y = ");  
    Serial.println(analogRead(PIN_VRY));  
    // Стан кнопки  
    Serial.print("button = ");  
    if (digitalRead(PIN_BUTTON) == HIGH) {  
        Serial.println ("NOT CLICK");  
    }  
    else {  
        Serial.println ("CLICK!");  
    }  
    // Пауза 1 сек  
    delay (1000);  
}
```

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

4. Сторчак К.П. Аналіз вимог до проектування хмарної платформи для Інтернету речей / К.П. Сторчак, А.М. Тушич, О.М. Ткаленко, В.М. Чорна, Т.М. Жила // Науковий журнал «Зв'язок», Київ – №6 2019. – С. 8-11.
5. Срібна І.М. Веб-сервіси AWS для розгортання пристроїв IoT / І.М. Срібна, Є.В. Савчук // Науковий журнал «Зв'язок», Київ – №4 2019. – С. 18-23.
6. Збірник тез Всеукраїнської науково-технічної конференції «Сучасний стан та перспективи розвитку ІОТ». – К.: ДУТ, - 2020. – 184 с.
7. Олещенко Л.М. Програмування пристроїв Інтернету речей / Л.М. Олещенко, Я.В. Хіцко. – К.: КПІ ім. Ігоря Сікорського, 2019, - 47 с.
8. Introduction to IoT (Cisco Networking Academy) // Електронний ресурс. Режим доступу: <https://www.netacad.com>.
9. Посібник // Електронний ресурс. Режим доступу: <https://www.arduino.cc/en/Tutorial/HomePage>
10. Посібник // Електронний ресурс. Режим доступу: <https://www.tutorialspoint.com/arduino/index.htm>
11. Посібник // Електронний ресурс. Режим доступу: <https://core-electronics.com.au/tutorials/arduino-workshop-for-beginners.html>
12. Уроки // Електронний ресурс. Режим доступу: <http://cxem.net/arduino/arduino15.php>
13. Уроки // Електронний ресурс. Режим доступу: <http://www.instructables.com/id/Arduino-Basics-PIR-Sensor/>
14. Уроки // Електронний ресурс. Режим доступу: <https://habr.com/ru/post/357908/>
15. Уроки // Електронний ресурс. Режим доступу: <https://habr.com/ru/post/352806/>

