Міністерство освіти і науки, молоді та спорту України
Сумський державний університет

Методичні вказівки

до виконання лабораторних робіт
з курсу «Програмування систем збору і аналізу даних»
для студентів напряму підготовки 171 «Електроніка»
денної та заочної форми навчання

Суми
Сумський державний університет
2020

Лабораторна робота №1

Тема: «Введення / виведення сигналів у контролерах Arduino»

Мета: навчитись зчитувати дані з дискретних та аналогових входів контролера Arduino. Освоїти генерацію аналогових та дискретних сигналів.

Елементи теорії.

Arduino — апаратна платформа для розробки електронних пристроїв, основними компонентами якої є мікроконтролер (МК) з портами вводу/виводу та середовище розробки ArduinoIDE. Програмування контролера здійснюється на мові Wiring, що є підмножиною C/C++. Усе сімейство контролерів Arduino, окрім версій «Mini» і «ProMini», мають інтегрований на плату перетворювач інтерфейсів USB-UART (USB-TTL), який дозволяє програмувати мікроконтролер без зовнішнього програматора. Дана особливість реалізується також використанням раніше записаної у МК програми-завантажувача (Bootloader). В актуальних насьогодні версіях Arduino використовується контролер ATmega328 компанії Atmel для плат версій Uno, Nano, ProMini з 32 кБайт пам’яті. У платі Mega2560 використовується МК ATmega2560 з 256 кБайт пам’яті.

Зчитування або запис даних на дискретних портах здійснюється за допомогою функцій `digitalRead(порт);` та `digitalWrite(порт, значення);`.Дискретні дані можуть приймати два можливих значення HIGH та LOW, що відповідають високому та низькому рівню сигналу відповідно до ТТЛ логіки. Перед використанням даних функцій порт попередньо необхідно ініціалізувати як вхід або вихід. Дана операція виконується командою `pinMode (порт, режим);`, яка викликається в блоці «setup» основної програми. Параметр «режим» може приймати два значення – INPUT та OUTPUT (введення та виведення відповідно).

Приклад використання функцій дискретного введення / виведення:

```
   void setup() {
      pinMode(13, OUTPUT);        //ініціалізація
цифрового порта №13 як вихід
      pinMode(8, INPUT);      //ініціалізація    цифрового
порта №8 як вхід
   }
```

```
   void loop() {
      int dValue = digitalRead(8);     //присвоюємо
змінній dValue значення, зчитане з порта №8
      digitalWrite(13, HIGH);     записуємо до порта №13
сигнал з високим рівнем
      delay(1000);            // очікування 1000 мс (1 с)
   }
```

Зчитування аналогових даних виконується за допомогою функції `analogRead(порт);`, отримані дані приймають цілочислене значення у бітах. Виходячи з того, що розрядність АЦП – 10 Біт, отримуємо 1024 можливих значення (5 / 1024 = 0,004283 В/Біт), тобто точність вимірювання аналогового сигналу складає 4,883 мВ.

Приклад зчитування аналогових даних

```
  int portValueInBits = analogRead(A0);
    //присвоюємо змінній portValueInBits значення,
зчитане з аналогового порта A0
  float portValueInVolts = 5.0/1024.0* portValueInBits;
    //перетворюємо отримане значення у вольти
```

Для генерації аналогових сигналів використовується широтно-імпульсна модуляція (ШІМ, PWM), котра доступна на цифрових портах №3, 5, 6, 9, 10, 11 плат Uno, Nano і ProMini. Для генерації ШІМ сигналу використовується функція `analogWrite(порт, значення);`, параметр «значення» якої може приймати значення від 0 до 255, 0 буде відповідати сигналу на рівні 0В а 255 – 5В на виході.

Приклад використання ШІМ

```
  analogWrite(9, 128);    //На цифровому порту №9 буде
згенерований сигнал на рівні 2,5В
```

Порядок виконання роботи.

1. Зберіть схему на базі контролера Arduino для реалізації зчитування стану кнопки, підключеної до одного з цифрових портів. Для недопуску хибних спрацювань кнопки замкніть даний порт на землю через високоомний резистор (R > 1кОм).
2. Підключіть до іншого цифрового порта світлодіод. Використовуючи функції `digitalRead()` та `digitalWrite()` реалізуйте запалювання

світлодіода при натисканні на кнопку. Для перевірки стану кнопки використайте умовний оператор `if`. Даний оператор має наступний синтаксис:

```
if(логічний вираз, наприклад, x==0){
    //код, котрий виконається, якщо умова дійсна
    }
    else{
        //код, котрий виконається, якщо умова НЕ дійсна
        }
```

3) Підключіть до одного з виводів ШІМ світлодіод. За допомогою функції `analogRead()` зчитайте напругу з будь-якого аналогового входу, якщо вона більше 4В, виконайте за допомогою функції `analogWrite()` поступове запалювання, а потім згасання світлодіода, підключеного до ШІМ виводу. Поступове запалювання і згасання можна реалізувати у циклі, поступово збільшуючи, або зменшуючи значення, що записується у ШІМ. Синтаксис на прикладі циклу з параметром:

```
for (ініціалізація лічильника; логічна умова виконання; зміна значення лічильника) {
    //Тіло цикла
    }
```

Зміст звіту:

1. Номер, тема та мета роботи.
2. Елементи теорії.
3. Програмний код розробленого додатку.
4. Відповіді на контрольні питання.

Контрольні питання:

1. Назвіть технічні характеристики контролера Arduino Nano.
2. Опишіть методи захисту від дребежання контактів при натисканні кнопок.
3. Яка точність зчитування аналогових даних контролером Arduino. Методи підвищення точності.
4. Що таке широтно-імпульсна модуляція і можливі області її примінення.

Лабораторна робота №2

Тема: «Комунікація у контролерах Arduino. Протокол Serial»

Мета: навчитись реалізовувати комунікацію комп'ютера з контролером Arduino через послідовний інтерфейс.

Елементи теорії.

Протокол UART (Universal asynchronous receiver/transmitter) (універсальний асинхронний прийомо-передатчик) – один з найстаріших і найрозповсюдженіших насьогодні фізичних протоколів передачі даних. Протокол має велику кількість різновидів, найбільш відомий із них – RS-232 (COM-порт). Для усього сімейства UART протоколів спільною є апаратна реалізація – робочі лінії RxD и TxD, або просто Rx и Tx. Лінія Tx (Transmitted Data) використовується для передачі даних, а Rx (Received Data) – приймаюча лінія. Вихід передатчика Tx з'єднується зі входом приймача Rx і навпаки. На електричному рівні UART в сімействі контролерів Arduino підпорядковується стандартній 5-вольтовій TTL логіці а підключення до USB порта комп'ютера здійснюється посередництвом вбудованого на плату USB-UART перетворювача або через цифрові порти «0» та «1». Слід зазначити, що у проектах з ініціалізованим Serial з'єднанням не можна використовувати дані цифрові порти для інших цілей.

На програмному рівні передача даних по UART протоколу здійснюється з використанням бібліотеки «HardwareSerial.h», яка підключається до проекту автоматично на етапі компіляції і додаткової ініціалізації не потребує. Ініціалізація з'єднання виконується за допомогою функції `Serial.begin(швидкість передачі);`, швидкість передачі даних обирається із стандартного набору 300, 1200, 2400, 4800, 9600, 14400, 19200, 28800, 38400, 57600 або 115200 бод (біт / с). передача даних здійснюється командами `Serial.print();` та `Serial.println();` дані передаються у форматі ASCII (як символьний тип). Прийом даних здійснюється за допомогою функції `Serial.read();`, котра зчитує наступний доступний байт з буфера порту. Перевірити наявність даних та кількість доступних для зчитування байт у буфері можна за допомогою функції `Serial.available();`.

Порядок виконання роботи.

1. Зберіть схему на базі контролера Arduino для реалізації зчитування

цифрових даних з датчика руху та аналогових з датчика вологості ґрунту, підключивши сигнальні їх виходи до відповідних портів контролера Arduino.

2. Розробіть програму котра циклічно з періодичністю 2 рази / с опитує датчики та реагує на зміну сигналу з датчика руху з низького рівня на високий (датчик фіксує рух) а також на зменшення вологості ґрунту менше певної критичної межі.

3. Реалізувати відправку даних про події описані в П.2 через серійний протокол на комп’ютер.

4. Реалізувати моніторинг порту на наявність отримуваної інформації. При прийомі з порту «1» необхідно подати високий потенціал на один з цифрових портів з підключеним до нього світлодіодом, і низький при отриманні «0».

Зміст звіту:

1. Номер, тема та мета роботи.
2. Елементи теорії.
3. Програмний код розробленого додатку.
4. Відповіді на контрольні питання.

Контрольні питання:

1. Апаратна та програмна реалізація протоколу UART.
2. Реалізація послідовного інтерфейсу в Arduino.
3. Програмні функції протоколу Serial. Призначення та особливості використання.

Лабораторна робота №3

Тема: «Введення та виведення даних по послідовному протоколу I^2C»

Мета: освоїти навички роботи з пристроями введення та виведення даних, комунікація з якими здійснюється за послідовним протоколом I^2C на прикладі годинника реального часу та символьного дисплея.

Елементи теорії.

Послідовний протокол обміну даними (I^2C – Inter-Integrated Circuits) був розроблений компанією Philips у 80-х роках XX-го ст. для реалізації обміну даними між компонентами електронних схем. Протокол набув широкого розповсюдження в інших областях та галузях електроніки внаслідок простої реалізації та широких можливостей масштабування систем обміну даними на його основі. Протокол використовує для передачі даних дві двонаправлені лінії зв'язку, котрі називаються шина послідовних даних **SDA (Serial Data)** та шина тактування **SCL (Serial Clock)**. Також у даному протоколі використовується дві шини живлення. SDA та SCL підтягуються до шини живлення через резистори. Схема підключення пристроїв до шини I^2C показана на рисунку 7.1. Специфікацією передбачається, що в мережі присутній хоча б один ведучий пристрій **(Master)**, котрий ініціює передачу даних та генерує сигнали синхронізації. У мережі також присутні ведомі пристрої **(Slave)**, які передають дані по запиту ведучого. Кожен ведомий пристрій має свою унікальну адресу за якою ведучий звертається до нього. Більшість пристроїв, які підтримують протокол I^2C мають типову адресу, характерну для кожного типу пристроїв, а також можливість ручної зміни адреси (здебільшого перемичками на платі).До однієї шини може бути одночасно підключено до 127 пристроїв, у тому числі декілька ведучих. Також є можливість підключати пристрої у процесі роботи, тобто реалізована підтримка так званого «гарячого підключення».

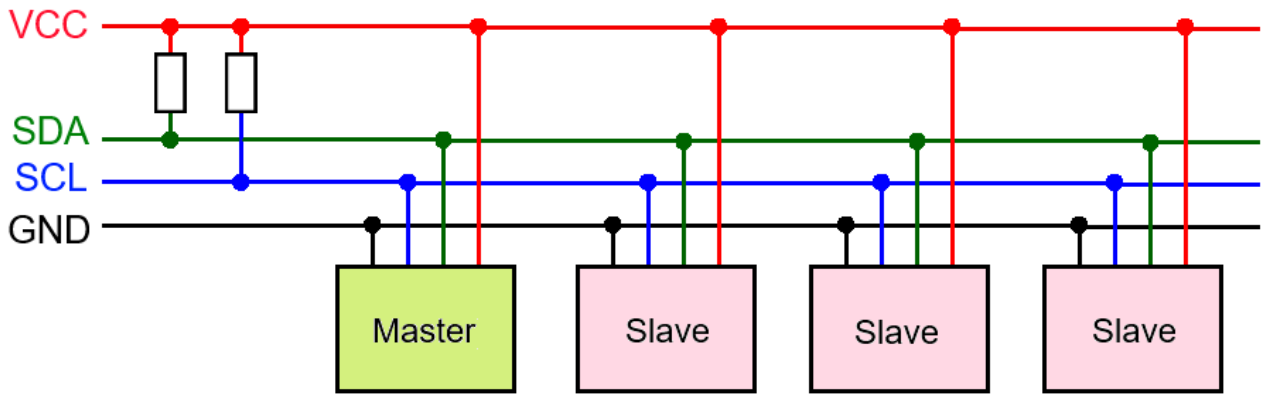

Рисунок 7.1. – Схема підключення пристроїв до шини I^2C

Arduino використовує для роботи по інтерфейсу I^2C два порта. Наприклад, в Arduino UNO та Arduino Nano це аналоговий порт A4, котрий відповідає SDA, та аналоговий порт A5, котрий відповідає SCL.

Для обміну даними з пристроями по шині I^2C для Arduino IDE інтегрована стандартна бібліотека `Wire.h`. Вона має наступні функції:

Функція	Призначення
`begin(address)`	Ініціалізація бібліотеки та підключення до шини I2C. Якщо не вказана адреса, то пристрій вважається ведучим.
`requestFrom()`	Використовується ведучим пристроєм для запиту визначеної кількості байт від ведомого.
`beginTransmission(address)`	Початок передачі даних до ведомого пристрою по заданій адресі.
`endTransmission()`	Припинення передачі даних ведомому.
`write()`	Запис даних від ведомого у відповідь на запит.
`available()`	Повертає кількість байт даних, доступних для прийому від ведомого.
`read()`	Зчитування байта, переданого між пристроями.
`onReceive()`	Вказує на функцію, котра буде викликана при прийомі даних ведомим пристроєм від ведучого.
`onRequest()`	Вказує на функцію, котра буде викликана при прийомі даних ведучим пристроєм від ведомого.

Більшість пристроїв, котрі підключаються за протоколом I^2C для зручності використання мають власні бібліотеки, котрі наслідують функції бібліотеки `Wire.h`.

Порядок виконання роботи.

1. Підключити до шини I^2C контролера Arduino годинник реального часу DS1307 та символьний дисплей з адаптером послідовного порту.
2. Додати бібліотеки для роботи з підключеними пристроями через меню «Скетч –> Підключити бібліотеку –> Додати .ZIP бібліотеку». Для роботи з

годинником реального часу необхідно імпортувати бібліотеки «DS1307RTC.h» та «Time.h». Функції роботи з символьним дисплеєм реалізовані в бібліотеці «LiquidCrystal_I2C.h».

Ініціалізація символьного дисплея здійснюється командою `LiquidCrystal_I2C lcd(адреса, кількість символів у рядку, кількість рядків);`, де адреса за замовчуванням для даного типу дисплеїв – `0x27`, символів у рядку – 16, стрічок – 2. Даною операцією створюється об'єкт `lcd` типу `LiquidCrystal_I2C`.

Основні команди роботи з дисплеєм:

Функція	**Призначення**
`lcd.begin();`	Ініціалізація дисплея.
`lcd.backlight();`	Включення підсвітки.
`lcd.print(дані);`	Виведення на екран даних.
`lcd.setCursor(x,y);`	Установка курсора в задану позицію. x – номер символа, y – номер стрічки.
`lcd.clear();`	Очистка дисплея.

3. Реалізувати зчитування дати та часу з годинника реального часу. Для роботи з годинником реального часу необхідно створити змінну типу `tmElements_t`, перевірка готовності годиннака здійснюється функцією `RTC.read(Ваша змінна)`, якщо вона повертає істину – модуль готовий до роботи. Отримання поточного часу здійснюється через поля доступу до змінної, список полів: `Year`, `Month`, `Day`, `Hour`, `Minute`, `Second`. Всі вони повертають цілочисельне значення параметру, наприклад використавши операцію `int time = tm.Hour` змінній `time` буде присвоєне значення поточної години (`tm` – змінна типу `tmElements_t`). Примітка: для отримання значення поточного року в форматі YYYY можна використати функцію перетворення `(tmYearToCalendar(tm.Year));`.

4. Здійснити виведення поточної дати в 0-ву стрічку дисплея (нумерація символів та стрічок починається з 0) у форматі DD:MM:YYYY та часу в 1-шу стрічку у форматі hh:mm:ss. Оновлення даних на дисплеї здійснювати 1 р/с.

5. Підключити до цифрового входу кнопку і реалізувати при її натисканні одноразове зчитування напруги з аналогових входів A0 та A1 і вивести його на дисплей на 3 с., після чого знову продовжити відображати дату та час.

Зміст звіту:

1. Номер, тема та мета роботи.

2. Елементи теорії.
3. Програмний код розробленого додатку.
4. Відповіді на контрольні питання.

Контрольні питання:

1. Протокол передачі даних I^2C та його реалізація в Arduino.
2. Основні функції бібліотеки `Wire.h`
3. Символьні дисплеї та їх підключення до Arduino по протоколу I2C.
4. Годинник реального часу DS1307. Характеристика.

Лабораторна робота №4

Тема: «Контроль оточуючого середовища за допомогою аналогових і дискретних датчиків»

Мета: освоїти навички роботи з аналоговими та дискретними датчиками фізичних величин та управління силовими пристроями.

Елементи теорії.

Мікроклімат приміщення – це сукупність фізичних чинників та умов навколишнього середовища, які зумовлюють його тепловий стан і впливають на теплообмін людини. Основними і найбільш важливими чинниками, які формують мікроклімат приміщень, є температура та вологість повітря. Для забезпечення нормальної життєдіяльності людини необхідно створити комфортні умови всередині приміщення. Створювані умови залежать від багатьох чинників, таких, як: пора року, час доби, погодні умови поза приміщенням та ін. Корегування мікроклімату приміщень здійснюється за допомогою комплексних та спеціалізованих систем клімат-контролю.

Найпростіша система контролю мікроклімату включає у себе модулі вимірювання температури та вологості і в залежності від їх показань вмикаються системи кондиціонування, зволоження або осушення повітря.

Для контролю параметрів навколишнього середовища, таких, як температура та вологість, за допомогою Arduino можна використати багатофункціональний модуль DHT-11, зовнішній вигляд якого показаний на рис.8.1.

Рисунок 8.1 – Зовнішній вигляд датчика контролю температури і вологості DHT-11

Датчик має наступні робочі характеристики:

Відносна вологість	Температура
Роздільна здатність: 16 біт Діапазон вимірювання: 20 – 90% Точність: ±5% при 25 °C Гістерезис: < ±3% RH	Роздільна здатність: 16 біт Діапазон вимірювання: 0 – 50°C Точність: ±1°C Гістерезис: < ±0,2°C

Для контролю освітленості можна використати дискретний датчик на фоторезисторі (рис. 8.2), котрий реагує на величину світлового потоку, що діє на чутливий елемент.

Рисунок 8.2. – Зовнішній вигляд дискретного датчика освітленості на фоторезисторі

Підключення силових навантажень до Arduino може здійснюватись багатьма свособами, наприклад через відкривання транзистора, тиристора, семістора або оптопати. Одним з найбільш простих і надійних рішень буде використання для даної цілі механічних реле, а між ними і схемою керування для захисту від можливого електричного пробою доцільно буде розмістити оптичну розв'язку.

Порядок виконання роботи.

1. Підключити до одного з цифрових входів (необхідно використовувати саме цифровий, а не аналоговий вхід, оскільки датчик має вбудований аналого-цифровий перетворювач) Arduino багатофункціональний датчик DHT-11 та датчик освітленості на фоторезисторі. Для програмної обробки даних з датчика температури і вологості необхідно підключити бібліотеку `DHT` та ініціалізувати датчик командою `DHT dht(DHTPIN, DHTTYPE);`, де `DHTPIN` – цифровий вхід до якого підключений сигнальний вихід датчика, `DHTTYPE` – тип датчика (у нашому випадку – DHT11). Програмний запуск датчика здійснюється командою `dht.begin();` у тілі функції `Setup`. Для отримання значень температури та вологості в бібліотеці `DHT` описані функції `dht.readHumidity();` та `dht.readTemperature();` відповідно, які повертають значення типу `float`. Для обробки даних з датчика освітленості підключення додаткових бібліотек не потрібне, оскільки він на виході формує дискретний сигнал, а регулювання чутливості здійснюється розміщеним на платі змінним резистором.
2. Реалізувати виведення на символьний дисплей поточних значень температури та вологості, а також часу доби (день або ніч).
3. Здійснити увімкнення вентилятора при досягненні порогового значення температури денної – 20°C, нічної – 24°C. Для запобігання частого спрацювання системи кондиціонування необхідно передбачини гістерезис

температури 2°C. Систему увімкнення вентилятора необхідно здійснити через реле, підключене до цифрового виходу Arduino.

Зміст звіту:

1. Номер, тема та мета роботи.
2. Елементи теорії.
3. Програмний код розробленого додатку.
4. Відповіді на контрольні питання.

Контрольні питання:

1. Які датчики контролю фізичних параметрів навколишнього середовища можна підключити до Arduino?
2. Особливості обробки даних з аналогових та дискретних датчиків фізичних величин.
3. Технічні характеристики датчиків температури і вологості сімейства DHT (DHT-11, DHT-21, DHT-22).

Лабораторна робота №5

Тема: «Керування Arduino за допомогою інфрачервоних сигналів»

Мета: навчитись за допомогою інфрачервоного датчика обробляти сигнали з пультів керування.

Елементи теорії.

Обробка сигналів з інфрачервоних пультів керування в платформі Arduino здійснюється за допомогою приймачів інфрачервоних сигналів. У більшості систем передача сигналів від пульту керування до приймача здійснюється за протоколом «RC5». Принцип кодування управляючого сигналу від пульта має наступну послідовність: спочатку передатчиком надсилається одиничний імпульс тривалістю 9 мс, який є найдовшим і виступає ідентифікатором початку передачі команди; наступним етапом є передача 32-х біт – коду кнопки пульта. Біти коду кнопки кодуються і передаються часовим методом, тобто відправляються опорні імпульси по 500 мкс а час між імпульмами визначає який саме біт був переданий (0 відповідає проміжку близько 500, 1 – 1600 мкс), даний процес показаний на рис. 9.1. Якщо кнопка продовжує бути натиснутою, то формуються імпульси аналогічні початку передачі (9 мс).

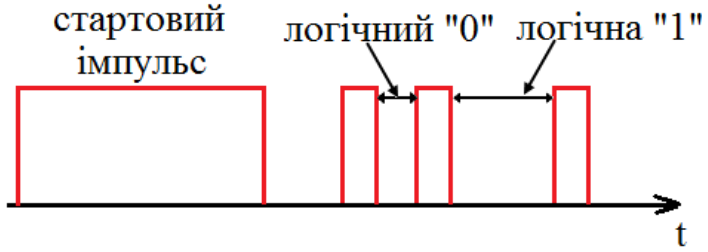

Рисунок 9.1. – Принцип кодування та передачі сигналів за протоколом «RC5»

Порядок виконання роботи.

1. Підключити до Arduino інфрачервоний приймач VS1838B (рис. 9.2). сигнальний вихід датчика підключається до одного з цифрових входів, для живлення необхідно використати джерело напруги 5В.
2. До одного з ШІМ виводів приєднати світлодіод а до цифрового виходу релейний модуль з підключеним навантаженням.
3. Підключити бібліотеку `IRremote.h` та здійснити зчитування кодів чотирьох кнопок пульта керування. Отримані значення записати.

Рисунок 9.2. – Зовнішній вигляд інфрачервоного приймача VS1838B

Ініціалізація бібліотеки та приймача здійснюється створенням змінної-ідентифікатора `IRrecv irrecv(вхід, до якого підключений приймач);`, для накопичення результату прийому коду кнопки необхідно ініціалізувати змінну типу `decode_results` . Запуск зчитування даних виконується командою `irrecv.enableIRIn();` а перевірка доступних для приймання даних функцією, `decode` вашої змінної-ідентифікатора (наприклад `irrecv.decode(&змінна накопичення результату)`), котра повертає значення «істина», якщо є доступні для зчитування дані. Зчитати отримані та накопичені дані можна через функцію `змінна накопичення результату.value`. Після приймання даних необхідно очистити вхідний буфер функцією `irrecv.resume();`

4. Реалізувати програмний код, який би при натисканні однієї пари кнопок вмикав та вимикав підключене через реле навантаження а іншою парою кнопок збільшував та зменшував яскравість світлодіода, котрий підключений до ШІМ виходу. Виведення всіх параметрів (увімкнене чи вимкнене навантаження та напруга на ШІМ виводі) здійснити на символьний дисплей.

Зміст звіту:

1. Номер, тема та мета роботи.
2. Елементи теорії.
3. Програмний код розробленого додатку.
4. Відповіді на контрольні питання.

Контрольні питання:

1. Назвіть технічні характеристики інфрачервоного датчика VS1838B.
2. Основні команди бібліотеки `IRremote.h` та їх призначення
3. Передача даних за допомогою протоколу RC5.
4. Назвіть області використання інфрачервоних систем передачі даних

Література:

Лабораторна робота №6

Тема: «Контроль та керування руху з допомогою контролерів Arduino»

Мета: навчитись контролювати переміщення та рух з використанням ультразвукових далекомірів та контролерів Arduino. Освоїти навички керування рухом з використанням сервоприводів.

Елементи теорії.

Ультрозвуковий далекомір визначає відстань до об'єктів за принципом, схожим до того, як вимірюють відстань деякі тварини (дельфіни, летючі миші), він генерує звукові імпульси частотою 40 кГц і «слухає» його ехо. За часом розповсюдження звукової хвилі до об'єкта і у зворотньому напрямку можна однозначно визначити відстань до об'єкта. Навідміну від інфрачервоних далекомірів на показання ультразвукового далекоміра не впливають засвітки від сонця та колір об'єкта. Недоліком такого далекоміра є складність у розпізнанні тонких та пухких об'єктів.

Сервопривод – механізм, з керуванням через зворотній зв'язок, котрий дозволяє точно управляти параметрами руху. Сервоприводом є будь-який тип механічного приводу, який має у своєму складі датчик (положення, швидкості, зусилля та ін.) і блок керування приводом, що автоматично підтримує необхідні параметри на датчику відповідно до заданих із зовні параметрів. Найбільш поширеними є сервоприводи, що підтримують заданий кут або задану швидкість обертання.

Порядок виконання роботи.

1. Підключити до Arduino ультразвуковий далекомір HR-SC04 та реалізувати скетч для вимірювання відстані до об'єктів. Для роботи з далекоміром існує велика кількість бібліотек котрі мають схожий функціонал. Найбільш поширеною і простою є бібліотека `Ultrasonic.h`, ініціалізація якої відбувається ініціалізацією об'єкта (змінної) типу `Ultrasonic` з параметрами `(Trig PIN, Echo PIN)`, де `Trig PIN` та `Echo PIN` – цифрові входи контролера Arduino, до котрих підключені відповідні виходи далекоміра.

2. Реалізувати виведення на символьний дисплей відстані від далекоміра до об'єкта. Для цього необхідно використати функцію ініціалізованої вами раніше змінної `ВашаЗмінна.distanceRead()`, тип отримуваних даних – float.

3. Підключити сервопривод до одного з ШІМ виводів та реалізувати керування ним у залежності від даних, отриманих з далекоміра. Вважати, що відстань 0 см рівна куту повороту сервопривода 0° а 50 см – 180°. Для роботи з сервоприводами в середовиші Arduino IDE є вбудована бібліотека `Servo.h`, функції якої реалізують основні операції для керування. Для доступу до функцій бібліотеки необхідно створити глобальну змінну типу `Servo` та ініціалізувати сервопривод функцією `ВашаЗмінна.attach(Номер вивода, до якого підключений сервопривод)`. Задання кута повороту здійснюється функцією `ВашаЗмінна.write(кут у градусах)`.

Зміст звіту:

1. Номер, тема та мета роботи.
2. Елементи теорії.
3. Програмний код розробленого додатку.
4. Відповіді на контрольні питання.

Контрольні питання:

1. Назвіть технічні характеристики ультразвукового далекоміра HR-SC04.
2. Основні команди бібліотек `Ultrasonic.h` і `Servo.h` та їх призначення.
3. Назвіть області використання систем ехо-локації.
4. Назвіть області використання сервоприводів, їх переваги та недоліки.

Лабораторна робота №7

Тема: «Робота із засобами радіочастотної ідентифікації в Arduino»

Мета: освоїти роботу з апаратною та програмною складовою прийомо-передатчиків RFID з використанням платформи Arduino. Навчитись реалізовувати контроль доступу за допомогою RFID міток.

Елементи теорії.

Радіочастотна ідентифікація (Radio Frequency Identification — RFID) — загальна назва для радіохвильових технологій автоматичної безконтактної ідентифікації різноманітних об’єктів. Системи р.і. є функціональними модулями сучасних інформаційних систем і поєднують три основні компоненти: радіомітку — засіб маркування об’єктів, який складається з мікросхеми та антени; приймально-передавальний радіопристрій для зчитування інформації з радіомітки; програмно-технічне забезпечення, яке реалізує процес зчитування та опрацювання інформації. Загальним принципом для технологій р.і. є зберігання ідентифікаційної та іншої інформації про об’єкт на мікросхемі, до якої приєднана антена. Для радіоміток використовують і інші назви — RFID-мітки, тег або транспондер.

Для цілей р.і. використовується декілька діапазонів радіохвиль. У кожному з них р.і. має різні характеристики, що зумовлює вибір конкретного діапазону для вирішення різних прикладних завдань. Серед частотних діапазонів Р.і. виділяють такі: низькочастотний діапазон (НЧ, LF) – 100–500 КГц; високочастотний діапазон (ВЧ, HF) — 10–15 МГц; ультрависокочастотний діапазон (УВЧ, UHF) — 860–960 МГц та діапазон мікрохвиль (НДЧ, SHF) — 2,45 ГГц. Радіомітки, які працюють у НЧ діапазоні, мають нижчу вартість і використовуються у системах контролю та управління доступом, ідентифікації тварин, інвентаризації. Діапазон високих та надвисоких частот використовують у системах, де є потреба в підвищеній дальності та високій швидкості ідентифікації.

Відповідно до типу пам’яті радіомітки поділяють на три групи: 1) радіомітки RO, в яких інформація записується один раз, одразу після виготовлення. У такі радіомітки інформацію вносить виробник і в подальшому її змінити, видалити або підробити практично неможливо, вона лише зчитується; 2) радіомітки WORM, що мають незмінний унікальний ідентифікатор і блок пам’яті, в який інформацію можна записати одноразово і в подальшому багатократно зчитувати; 3) радіомітки RW, що мають незмінний унікальний ідентифікатор і

блок пам’яті, в який інформацію можна багаторазово записувати, а потім зчитувати.

Порядок виконання роботи.

1. Підключити до шини SPI контролера Arduino модуль радіочастотної ідентифікації MFRC522, для цього виводи модуля MOSI, MISO і SCK під’єднати до цифрових входів Arduino 11, 12 і 13 відповідно, а RST і SDA до будь-яких інших цифрових портів.
2. Ініціалізувати глобальні змінні р.і. модуля, створивши змінну типу `MFRC522` з параметрами `SDA_PIN` і `RST_PIN` – номерами цифрових портів, до яких підключені виводи р.і. модуля `SDA` і `RST`; а також змінну типу `MFRC522::MIFARE_Key`, попередньо підключивши бібліотеки `SPI.h` та `MFRC522.h`.
3. Інціціалізувати шину SPI та підключити р.і. модуль за допомогою команд `SPI.begin()` та `Ваша_змінна.PCD_Init()` відповідно.
4. Реалізувати зчитування ідентифікатора мітки при піднесенні до зчитувача за допомогою команди `Ваша_змінна.uid.uidByte[i]`, де і–порядковий номер блоку ідентифікатора (RFID мітка містить 4 таких блоки). Результат зчитування вивести і послідовний порт.
5. Реалізувати код, який би при співпадінні ідентифікатора мітки із попередньо збереженим в константі перемикав стан реле, підключеного до одного з виходів Arduino на 2 секунди.

Контрольні питання:

1. Що таке радіочастотна ідентифікація?
2. Назвіть технічні характеристики модуля рідіочастотної ідентифікації MFRC522.
3. Основні команди бібліотеки MFRC522.h