

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ АЕРОНАВІГАЦІЇ, ЕЛЕКТРОНІКИ ТА ТЕЛЕКОМУНІКАЦІЙ
КАФЕДРА ЕЛЕКТРОНІКИ, РОБОТОТЕХНІКИ І ТЕХНОЛОГІЙ
МОНІТОРИНГУ ТА ІНТЕРНЕТУ РЕЧЕЙ

ДОПУСТИТИ ДО ЗАХИСТУ
Завідувач випускової кафедри
_____ Шутко В.М.
« ____ » _____ 2020 р.

ДИПЛОМНА РОБОТА

ЗДОБУВАЧА ОСВІТНЬОГО СТУПЕНЯ МАГІСТРА
ЗІ СПЕЦІАЛЬНОСТІ 153 «МІКРО- ТА НАНОСИСТЕМНА ТЕХНІКА»
ОСВІТНЬО-ПРОФЕСІЙНОЇ ПРОГРАМИ
«ФІЗИЧНА ТА БІОМЕДИЧНА ЕЛЕКТРОНІКА»

Тема: «Мікроконтролерна система управління об'єктом, що рухається»

Виконавець
студент групи МН-206М _____ Шкарівський Владислав Ігорович

Керівник
д.т.н., професор _____ Ліпінський Олександр Юрійович

Консультант розділу
«Охорона праці» _____ Якимець І.В.

Консультант розділу
«Охорона навколишнього середовища» _____ Маджд С.М.

Нормоконтролер _____ Сініцин Р.Б.

КИЇВ 2020

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ АЕРОНАВІГАЦІЇ, ЕЛЕКТРОНІКИ ТА ТЕЛЕКОМУНІКАЦІЙ
КАФЕДРА ЕЛЕКТРОНІКИ, РОБОТОТЕХНІКИ І ТЕХНОЛОГІЙ
МОНІТОРИНГУ ТА ІНТЕРНЕТУ РЕЧЕЙ
НАПРЯМ ПІДГОТОВКИ ФІЗИЧНА ТА БІОМЕДИЧНА ЕЛЕКТРОНІКА

ДОПУСТИТИ ДО ЗАХИСТУ
Завідувач випускової кафедри
_____ Шутко В.М.
« ____ » _____ 2020 р.

ЗАВДАННЯ

на виконання дипломної роботи
Шкарівський Владислав Ігорович

1. Тема дипломної роботи : Мікроконтролерна система управління об'єктом, що рухається.
наказом ректора від «2» жовтня 2020 р. № 1900/СТ
2. Термін виконання роботи: з 05.10.2020 по 27.12.2020.
3. Вихідні дані до роботи : мікроконтролер, Спеціалізоване програмне забезпечення.
4. Зміст пояснювальної записки реферат, список скорочень, вступ, 5 частини, висновки, список використаних джерел.
5. Перелік обов'язкового ілюстративного матеріалу:
Графічне представлення результатів дослідження, таблиці, рисунки.

6. Календарний план-графік

№	Етапи виконання дипломної роботи	Термін виконання етапів	Підпис керівника
1.	Пошук обробка та опрацювання літератури за темою дипломної роботи	05.10.20. – 18.10.20.	
2.	Опис базових компонентів, базової мови програмування за темою дипломної роботи	19.10.20. – 25.10.20	
3.	Розробка алгоритму роботи	26.10.20. – 01.11.20	
4.	Розробка коду	02.11.20. – 15.11.20	
5.	Оптимізація алгоритму	16.11.20. – 22.11.20	
6.	Моделювання переміщення	23.11.20. – 29.11.20	
7.	Вибір компонентів	30.11.20. – 04.12.20	
8.	Аналіз отриманих результатів	05.12.20. – 8.12.20	
9.	Написання розділу «Охорона праці»	9.12.20. – 12.12.20	
10.	Написання розділу «Охорона навколишнього середовища»	13.12.20. – 16.12.20	
11	Подання на кафедру. Усунення недоліків. Оформлення пояснювальної записки.	17.12.20	
12	Електронна версія доповіді, ілюстративний матеріал доповіді	18.12.30	

7. Консультація з окремого(мих) розділу(ів):

Назва розділу	Консультант (посада, П.І.Б.)	Дата, підпис	
		Завдання видав	Завдання прийняв
Охорона праці	Асистент Якимець Ірина Вячеславівна		
Охорона навколишнього середовища	професор, д.т.н., професор Маджд Світлана Михайлівна		

8. Дата видачі завдання: «05» жовтня 2020 р.

Керівник дипломної роботи : _____ Ліпінський Олександр Юрійович

Завдання прийняв до виконання: _____ Шкарівський Владислав Ігорович

РЕФЕРАТ

Пояснювальна записка до дипломної роботи «Мікроконтролерна система управління об'єктом, що рухається»: 110 с., 21 рис., 1 табл., 33 літературних джерела.

Об'єкт дослідження: системи на основі мікроконтролера для управління рухомим об'єктом

Тема роботи: реалізація системи управління рухомим об'єктом

Мета роботи: розробити та реалізувати оптимальну систему управління для рухомого об'єкта

Методи дослідження: обробка літературних джерел.

Результати магістерської роботи рекомендується використовувати для проектування та розробки систем управління для автономних рухомих об'єктів, які використовують в якості рухомого елемента ноги.

Скорочення

3D - 3-dimensional 3D-принтер

ШИМ - Широтно-імпульсна модуляція

USB - Universal Serial Bus універсальна послідовна шина

AC/DC - alternating current/direct current переменный ток/постоянный ток

UART - universal asynchronous receiver/transmitter — універсальний асинхронний приймач/передавач

DFU Device Firmware Update

COM communications port

ICSP In-Circuit Serial Programming

LCD - liquid crystal display

LED - light-emitting diode

3DOF 3 степеня свободи

EEPROM - Electrically Erasable Programmable Read-Only Memory постійний запам'ятовувач

АБС - Акрилонітрилбутадієнстирол

ЗМІСТ

Вступ.....	11
РОЗДІЛ 1. МІКРОКОНТРОЛЕР ARDUINO ТА СЕРЕДОВИЩЕ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.	12
1.1. Контролери сімейства Arduino та плати розширення.....	12
1.1.1. Arduino Uno	12
1.1.1.1. Загальні відомості	12
1.1.1.2. Живлення	15
1.1.1.3. Пам'ять	16
1.1.1.4. Входи та виходи.....	16
1.1.1.5. Зв'язок.....	18
1.1.1.6. Програмування.....	18
1.1.1.7. Автоматичний (програмний) скидання	19
1.1.1.8. Захист USB від перевантажень	21
1.1.1.9. Фізичні характеристики	21
1.1.2. Arduino Nano.....	22
1.1.2.1. Загальні відомості	22
1.1.2.2. Характеристики[2]:.....	22
1.1.2.3. Живлення	23
1.1.2.4. Пам'ять	24
1.1.2.5. Входи і виходи	25
1.1.2.6. Зв'язок.....	26
1.1.2.7. Програмування.....	27
1.1.3. Bluetooth модуль HC.....	28
1.1.3.1. Вступ	28
1.1.3.2. Вибір модуля	30
1.1.4. Arduino Shields.....	34
1.1.4.1. Arduino Shields - плати розширення для Ардуіно	34
1.1.4.2. Навіщо потрібні Shield arduino?	35

1.1.4.3. Програмування Arduino Shield	36
1.1.4.4. Arduino Sensor Shield	36
1.1.4.5. Arduino Motor Shield.....	37
1.1.4.6. Arduino Ethernet Shield	37
1.1.4.7. Плати розширення для прототипування	37
1.1.4.8. Arduino LCD shield і tft shield	37
1.1.4.9. Arduino Data Logger Shield.....	38
1.2. Мова програмування мікроконтролерів Arduino.	38
1.2.1. Arduino, спочатку підтримує мову, яку ми називаємо мовою програмування Arduino, або мовою Arduino.....	38
1.2.2. Підтримка іншої мови	40
1.2.3. Вбудовані константи мови програмування Arduino	40
1.2.4. Життєвий цикл програми	41
1.2.5. Часові функції	43
1.2.6. Робота з буквено-цифровими символами.....	44
1.2.7. Робота з бітами та байтами	45
1.2.8. Переривання	45
РОЗДІЛ 2. АЛГОРИТМ УПРАВЛІННЯ СЕРВОПРИВОДАМИ ОБ'ЄКТА, ЩО РУХАЄТЬСЯ.	46
2.1. Arduino - бібліотека Servo.	46
2.1.1. Використання	46
2.1.2. Як користуватися сервотекою Arduino.....	47
2.2. Arduino - бібліотека SerialCommand	50
2.2.1. Типи команд АТ та відповіді	50
2.2.2. Команди без параметрів:	51
2.2.3. Команди з параметрами:	51
2.3.1.Опис конструкції, та їх особливостей.....	52
2.3.2.Алгоритм руху робота в прямому напрямку.....	53
2.3.3.Алгоритм руху робота в зворотному напрямку.....	58
2.3.4.Алгоритм руху робота при повороті на право.	60
2.3.5.Алгоритм руху робота при повороті на ліво.....	64

2.3.6. Координати в кути.	68
РОЗДІЛ 3. РУХОМИЙ ОБ'ЄКТ НА СЕРВОПРИВОДАХ.	68
3.1. Компоненти.....	68
3.1.1. Коротко про Arduino nano.....	68
3.1.1.1. Основні характеристики Arduino Nano	69
3.1.1.2. Пам'ять в Arduino Nano	70
3.1.2. Коротко про Bluetooth HC-05.	70
3.1.3. Сервоприводи Tower Pro MG90S	71
3.1.4. Система живлення	72
3.1.5 Головна плата.	73
3.1.6. Корпус робота	74
3.2. Висновки до дипломної роботи.....	74
РОЗДІЛ 4. ОХОРОНА ПРАЦІ	75
4.1. Аналіз шкідливих та небезпечних виробничих факторів при виробництві електронного роботизованого пристрою.....	75
4.2. Організаційні та конструктивно-технологічні заходи для зниження впливу шкідливих виробничих факторів.....	77
4.2.1. Для забезпечення безпеки працівника, роботодавець та робітник повинні слідувати вимогам охорони праці під час паяння електрифікованим інструментом.	77
4.2.2. Для уникнення короткого замкнення при роботі з паяльником необхідно слідувати таким нормам:.....	78
4.2.3. Технологічний процес пайки радіоелементів супроводжуються забрудненням повітряного середовища аерозолями припоїв, флюсів, парами рідин, що застосовуються для змивання флюсу або для розчинення лаків, які застосовуються для покриття друкованих плат тощо.....	79
4.2.4. Для уникнення канцерогенних факторів при роботі з 3D принтером необхідно слідувати таким нормам: ДСТУ Б А.3.2-12:2009 ССБП Системи вентиляційні.....	80
4.2.5. Для уникнення проблем психофізіологічного напруження. монотонності праці, перенапруженості зорових аналізаторів, розумової напруженості, незручність і статичність пози. Необхідно виконувати ряд рекомендацій:	82

4.2.6. Для уникнення несприятливого освітлення необхідно слідувати таким нормам:.....	83
4.2.7. Методика розрахунку шарнірно-телескопічних відсмоктувачів повітря із зон пайки.	84
Висновок	87
4.3. Пожежо- та вибухонебезпека.....	87
4.4. Інструкція з охорони праці при виконанні паяльних робіт	89
РОЗДІЛ 5. ОХОРОНА НАВКОЛИШНЬОГО СЕРЕДОВИЩА	90
5.1. Аналіз проблеми впливу роботизованого електронного пристрою на стан довкілля. Розберемо окремі частини проекту та їх вплив на довкілля.	91
5.2. Аналіз основних джерел впливу та їх наслідків на людину та її оточення.	94
5.3. Рекомендації щодо зниження цих негативних чинників.	101
Література	Ошибка! Закладка не определена.

Вступ

Різноманітні роботи та роботизовані системи останні років 20 зазнають суттєвий розвиток в області мініатюризації. З'являється більше автономних роботів та дорнів, які можуть працювати доволі довгий час на одному заряді. Це досягалося за рахунок більш енергоефективних електронних компонентів (процесори, мікроконтролери, енергоефективна пам'ять). Також розробляються більш енергоємкісні джерела автономного живлення.

Існує клас роботів які ходять. Переваги ходьби проявляються при русі по складній місцевості, зокрема по кам'янистій поверхні, або при необхідності підніматися і спускатися по сходах. Такі роботи як правило використовуються в екстремальних умовах де людині буде небезпечно знаходитись, або дуже мало простору для руху. Наприклад в рятувальних місіях при завалі будинку. Невеликий робот павук зможе залізти в отвори завалів та знайти людей.

Тому розробка алгоритмів руху для таких роботів являється важливою задачею. За допомогою мікроконтролера Arduino та деяких бібліотек для нього, такі алгоритми можливо розробити. Завдяки 3D друку па програмам розрахунку навантажень являється можливо побудувати скелето подібну конструкцію, для корпусу робота. Така конструкція буде легкою та витримуватиме розрахункові навантаження.

РОЗДІЛ 1. МІКРОКОНТРОЛЕР ARDUINO ТА СЕРЕДОВИЩЕ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.

1.1. Контролери сімейства Arduino та плати розширення.

1.1.1 Arduino Uno



Рис. 1.1.1. Фотографія Arduino Uno. Вид зверху.

1.1.1.1 Загальні відомості

Arduino Uno - це пристрій на основі мікроконтролера ATmega328 [1]. У його склад входить все необхідне для зручної роботи з мікро контролером: 14 цифрових входів / виходів (з них 6 можуть використовуватися в якості ШІМ-виходів), 6 аналогових входів, кварцовий резонатор на 16 МГц, роз'єм USB, роз'єм живлення, роз'єм для внутрішньосхемне програмування (ICSP) і кнопка скидання. Для початку роботи з пристроєм досить просто подати живлення від AC / DC-адаптера або батарейки, або підключити його до комп'ютера за допомогою USB-кабелю.

На відміну від всіх попередніх плат Ардуіно, Уно в якості перетворювача інтерфейсів USB-UART використовує мікроконтролер АТmega16U2 (АТmega8U2 до версії R2) замість мікросхеми FTDI.

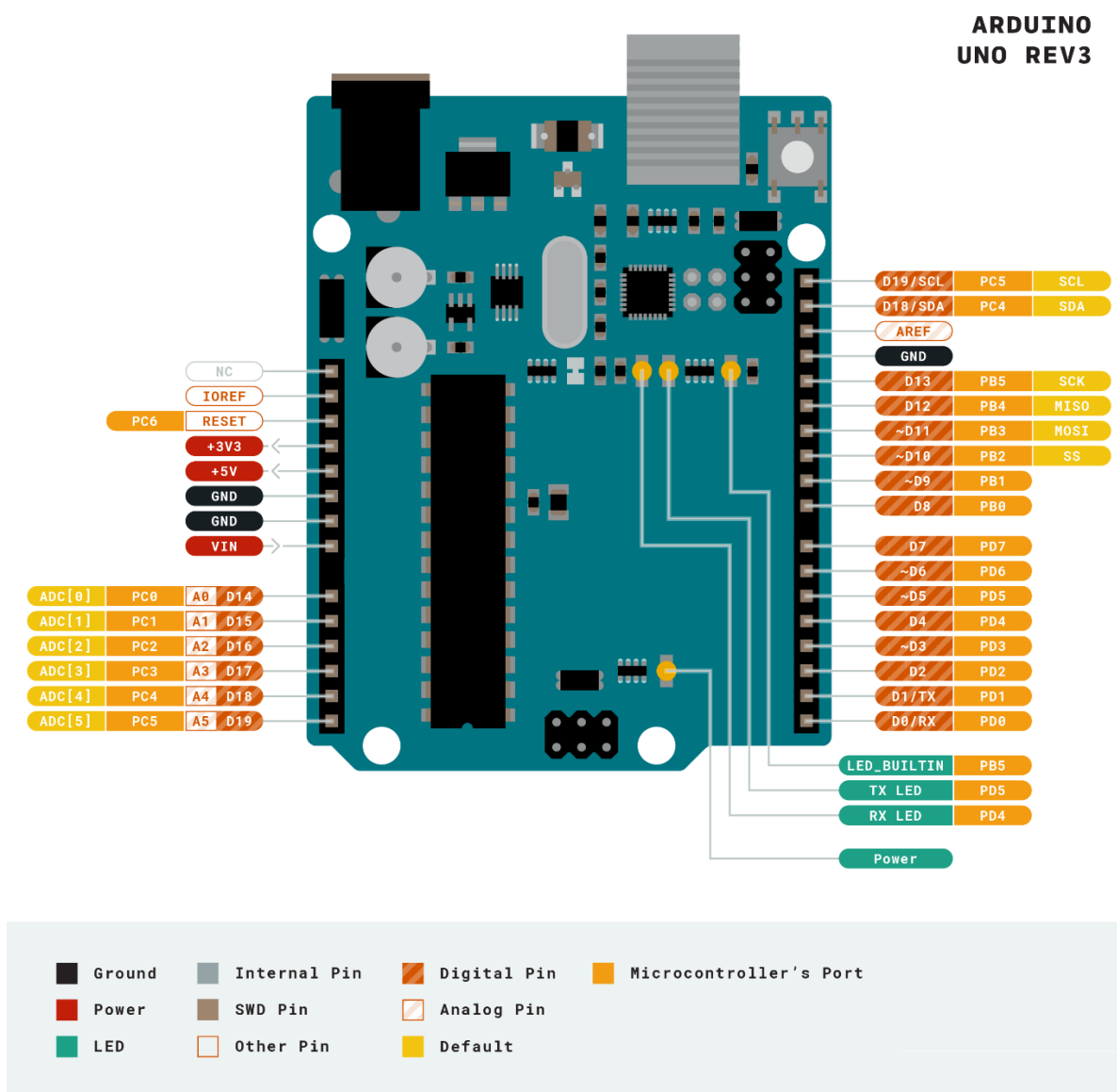


Рис. 1.1.2. Найменування входів та виходів на платі Arduino Uno.

На платі Arduino Uno версії R2 для спрощення процесу оновлення прошивки доданий резистор, що підтягує до землі лінію HWB мікроконтролера 8U2.

Зміни на платі версії R3 перераховані нижче:

Терморегулятори 1.0: додані висновки SDA і SCL (біля виведення AREF), а також два нових виведення, розташованих біля виведення RESET. Перший - IOREF - дозволяє платам розширення підлаштовуватися під робочу напругу Ардуіно. Даний висновок передбачений для сумісності плат розширення як з 5В-Ардуіно на базі мікроконтролерів AVR, так і з 3.3В-платами Arduino Due. Другий висновок ні до чого не приєднаний і зарезервований для майбутніх цілей.

Покращена стійкість ланцюга скидання.

Мікроконтролер ATmega8U2 замінений на ATmega16U2.

"Uno" (в перекладі з італійської - "один") названий з нагоди майбутнього випуску Arduino 1.0. Спільно з Arduino 1.0 дані пристрої будуть базовими версіями Ардуіно. Uno - еталонна модель платформи Arduino і є останньою в серії USB-плат.

Характеристики

мікроконтролер ATmega328

Робоча напруга 5В

Напруга живлення (рекомендований) 7-12В

Напруга живлення (граничне) 6-20В

Цифрові входи / виходи 14 (з них 6 можуть використовуватися в якості ШІМ-виходів)

Аналогові входи 6

Максимальний струм одного виведення 40 мА

Максимальний вихідний струм виводу 3.3V 50 мА

Flash-пам'ять 32 КБ (ATmega328) з яких 0.5 КБ використовуються загрузчиком

SRAM 2 КБ (ATmega328)

EEPROM 1 КБ (ATmega328)

Тактова частота 16 МГц

1.1.1.2 Живлення

Arduino Uno може бути живиться від USB або від зовнішнього джерела живлення - тип джерела вибирається автоматично.

В якості зовнішнього джерела живлення (НЕ USB) може використовуватися мережевий AC / DC-адаптер або акумулятор / батарея. Штекер адаптера (діаметр - 2.1мм, центральний контакт - позитивний) необхідно вставити у відповідний роз'єм живлення на платі. У разі живлення від акумулятора / батареї, її проводу необхідно під'єднати до висновків Gnd і Vin роз'єму POWER.

Напруга зовнішнього джерела живлення може бути в межах від 6 до 20 В. Однак, зменшення напруги живлення нижче 7В призводить до зменшення напруги на виводі 5V, що може стати причиною нестабільної роботи пристрою. Використання напруги більше 12В може призводити до перегріву стабілізатора напруги і виходу плати з ладу. З огляду на це, рекомендується використовувати джерело живлення з напругою в діапазоні від 7 до 12В.

Нижче перераховані висновки живлення, розташовані на платі:

VIN. Напруга, що надходить в Arduino безпосередньо від зовнішнього джерела живлення (не пов'язане з 5В від USB або іншим стабілізованою напругою). Через цей висновок можна як подавати зовнішнє живлення, так і споживати струм, коли пристрій живиться від зовнішнього адаптера.

5V. На висновок надходить напруга 5В від стабілізатора напруги на платі, поза незалежності від того, як живиться пристрій: від адаптера (7 - 12В), від USB (5В) або через висновок VIN (7 - 12В). Живити пристрій через висновки 5V або 3V3 не рекомендується, оскільки в цьому випадку не використовується стабілізатор напруги, що може привести до виходу плати з ладу.

3V3. 3.3В, що надходять від стабілізатора напруги на платі. Максимальний струм, споживаний від цього висновку, становить 50 мА.

GND. Висновки землі.

IOREF. Цей висновок надає платам розширення інформацію про робочій напрузі мікроконтролера Ардуіно. Залежно від напруги, ліченого з виведення IOREF, плата розширення може переключитися на відповідний джерело живлення або задіяти перетворювачі рівнів, що дозволить їй працювати як з 5В, так і з 3.3В-пристроями.

1.1.1.3 Пам'ять

Обсяг флеш-пам'яті ATmega328 становить 32 КБ (з яких 0.5 КБ використовуються загрузчиком). Мікроконтролер також має 2 КБ пам'яті SRAM і 1 КБ EEPROM (з якої можна зчитувати або записувати інформацію за допомогою бібліотеки EEPROM).

1.1.1.4 Входи та виходи

З використанням функцій `pinMode ()`, `digitalWrite ()` і `digitalRead ()` кожен з 14 цифрових висновків може працювати в якості входу або виходу. Рівень напруги на висновках обмежений 5В. Максимальний струм, який може віддавати або споживати один висновок, становить 40 мА. Всі висновки пов'язані з внутрішніми підтягуються резисторами (за замовчуванням відключеними) номіналом 20-50 кОм. Крім цього, деякі висновки Ардуіно можуть виконувати додаткові функції:

Послідовний інтерфейс: висновки 0 (RX) і 1 (TX). Використовуються для отримання (RX) і передачі (TX) даних по послідовному інтерфейсу. Ці висновки з'єднані з відповідними висновками мікросхеми ATmega8U2, яка виконує роль перетворювача USB-UART.

Зовнішні переривання: висновки 2 і 3. Чи можуть служити джерелами переривань, що виникають при фронті, спаді або при низькому рівні сигналу на цих висновках. Для отримання додаткової інформації див. Функцію `attachInterrupt ()`.

ШІМ: висновки 3, 5, 6, 9, 10 і 11. За допомогою функції `analogWrite ()` можуть виводити 8-бітові аналогові значення в вигляді ШІМ-сигналу.

Інтерфейс SPI: висновки 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Із застосуванням бібліотеки SPI дані висновки можуть здійснювати зв'язок по інтерфейсу SPI.

Світлодіод: 13. Вбудований світлодіод, приєднаний до висновку 13. При відправці значення HIGH світлодіод включається, при відправці LOW - вимикається.

В Arduino Uno є 6 аналогових входів (A0 - A5), кожен з яких може уявити аналогову напругу у вигляді 10-бітного числа (1024 різних значення). За замовчуванням, вимір напруги здійснюється щодо діапазону від 0 до 5 В. Проте, верхню межу цього діапазону можна змінити, використовуючи висновок AREF і функцію `analogReference ()`. Крім цього, деякі з аналогових входів мають додаткові функції:

TWI: висновок A4 або SDA і висновок A5 або SCL. З використанням бібліотеки `Wire` дані висновки можуть здійснювати зв'язок по інтерфейсу TWI.

Крім перерахованих на платі існує ще кілька висновків:

AREF. Опорна напруга для аналогових входів. Може бути задіяний функцією `analogReference ()`.

Reset. Формування низького рівня (LOW) на цьому висновку призведе до перезавантаження мікроконтролера. Зазвичай цей висновок служить для функціонування кнопки скидання на платах розширення

1.1.1.5 Зв'язок

Arduino Uno надає ряд можливостей для здійснення зв'язку з комп'ютером, ще одним Ардуіно або іншими мікроконтролерами. У ATmega328 є приймач UART, що дозволяє здійснювати послідовну зв'язок за допомогою цифрових висновків 0 (RX) і 1 (TX). Мікроконтролер ATmega16U2 на платі забезпечує зв'язок цього приймача з USB-портом комп'ютера, і при підключенні до ПК дозволяє Ардуіно визначатися як віртуальний COM-порт. Прошивка мікросхеми 16U2 використовує стандартні драйвера USB-COM, тому установка зовнішніх драйверів не потрібно. На платформі Windows необхідний тільки відповідний .inf-файл. У пакет програмного забезпечення Ардуіно входить спеціальна програма, що дозволяє зчитувати і відправляти на Ардуіно прості текстові дані. При передачі даних через мікросхему-перетворювач USB-UART під час USB-з'єднання з комп'ютером, на платі будуть мигати світлодіоди RX і TX. (При послідовній передачі даних за допомогою висновків 0 і 1, без використання USB-перетворювача, дані світлодіоди задіюються).

Бібліотека SoftwareSerial дозволяє реалізувати послідовну зв'язок на будь-яких цифрових висновках Arduino Uno.

У мікроконтролері ATmega328 також реалізована підтримка послідовних інтерфейсів I2C (TWI) і SPI. У програмне забезпечення Ардуіно входить бібліотека Wire, що дозволяє спростити роботу з шиною I2C. Для роботи з інтерфейсом SPI використовуйте бібліотеку SPI.

1.1.1.6. Програмування

Arduino Uno програмується за допомогою програмного забезпечення Ардуіно (скачати). Для цього з меню Tools> Board необхідно вибрати "Arduino Uno" з мікро контролером, відповідним вашій платі.

ATmega328 в Arduino Uno випускається з прошитим загрузчиком, що дозволяє завантажувати в мікроконтролер нові програми без необхідності використання зовнішнього програматора. Взаємодія з ним здійснюється за оригінальним протоколу STK500 (опис, заголовки C).

Проте, мікроконтролер можна прошити і через роз'єм для внутрішньо схемне програмування ICSP (In-Circuit Serial Programming), не звертаючи уваги на завантажувач; більш докладно про це див. відповідні інструкції.

Вихідний код прошивки мікроконтролера ATmega16U2 (або 8U2 на платах версії R1 і R2) знаходиться у вільному доступі. Прошивка ATmega16U2 / 8U2 включає в себе DFU-завантажувач (Device Firmware Update), що дозволяє оновлювати прошивку мікроконтролера. Для активації режиму DFU необхідно:

На платах версії R1: замкнути перемичку на звороті плати (біля зображення Італії), після чого скинути 8U2.

На платах версій R2 і вище - для спрощення переходу в режим DFU присутній резистор, що підтягує до землі лінію HWB мікроконтролера 8U2 / 16U2.

Після переходу в DFU-режим для завантаження нової прошивки можна використовувати програмне забезпечення Atmel's FLIP (для Windows) або DFU programmer (для Mac OS X і Linux). Альтернативний варіант - прошити мікроконтролер через роз'єм для внутрішньо схемне програмування ISP за допомогою зовнішнього програматора, проте в цьому випадку DFU-завантажувач затреться.

1.1.1.7. Автоматичний (програмний) скидання

Щоб кожен раз перед завантаженням програми не було потрібно натискати кнопку скидання, Arduino Uno спроектований таким чином, що дозволяє здійснювати його скидання програмно з підключеного комп'ютера. Один з висновків ATmega8U2 / 16U2, який бере участь в управлінні потоком даних (DTR), з'єднаний з висновком RESET мікроконтролера ATmega328 через конденсатор номіналом 100 нФ. Коли на лінії DTR з'являється нуль, висновок RESET також переходить в низький рівень на час, достатній для перезавантаження мікроконтролера. Дана особливість використовується для того, щоб можна було прошивати мікроконтролер всього одним натисненням кнопки в середовищі програмування Ардуіно. Така архітектура дозволяє зменшити таймаут завантажувача, оскільки процес прошивки завжди синхронізований зі спадом сигналу на лінії DTR.

Однак ця система може призводити і до інших наслідків. При підключенні Uno до комп'ютерів, що працюють на Mac OS X або Linux, його мікроконтролер буде скидатися при кожному з'єднанні програмного забезпечення з платою. Після скидання на Arduino Uno активізується завантажувач на час близько півсекунди. Незважаючи на те, що завантажувач запрограмований ігнорувати сторонні дані (тобто всі дані, які не стосуються процесу прошивки нової програми), він може перехопити кілька перших байт даних з посилки, що відправляється платі відразу після установки з'єднання. Відповідно, якщо в програмі, що працює на Ардуіно, передбачено отримання від комп'ютера будь-яких налаштувань або інших даних при першому запуску, переконайтеся, що програмне забезпечення, з яким взаємодіє Ардуіно, здійснює відправку через секунду після установки з'єднання.

На платі Uno існує доріжка (зазначена як "RESET-EN"), розімкнувши яку, можна відключити автоматичне скидання мікроконтролера. Для повторного відновлення функції автоматичного скидання необхідно спаяти між собою висновки, розташовані по краях цієї доріжки. Автоматичний скидання також

можна вимкнути, підключивши резистор номіналом 110 Ом між висновком RESET і 5В.

1.1.1.8 Захист USB від перевантажень

В Arduino Uno є відновлювані запобіжники, що захищають USB-порт комп'ютера від коротких замикань і перевантажень. Незважаючи на те, що більшість комп'ютерів мають власний захист, такі запобіжники забезпечують додатковий рівень захисту. Якщо від USB-порту споживається струм більше 500 мА, запобіжник автоматично розірве з'єднання до усунення причин короткого замикання або перевантаження.

1.1.1.9 Фізичні характеристики

Максимальна довжина і ширина друкованої плати Uno становить 6.9 см і 5.4 см відповідно, з урахуванням роз'єму USB і роз'єму живлення, які виступають за межі плати. Чотири кріпильних отвори дозволяють прикріплювати плату до поверхні або корпусу. Зверніть увагу, що відстань між цифровими висновками 7 і 8 не кратне традиційним 2.54 мм і становить 4 мм.

1.1.2 Arduino Nano

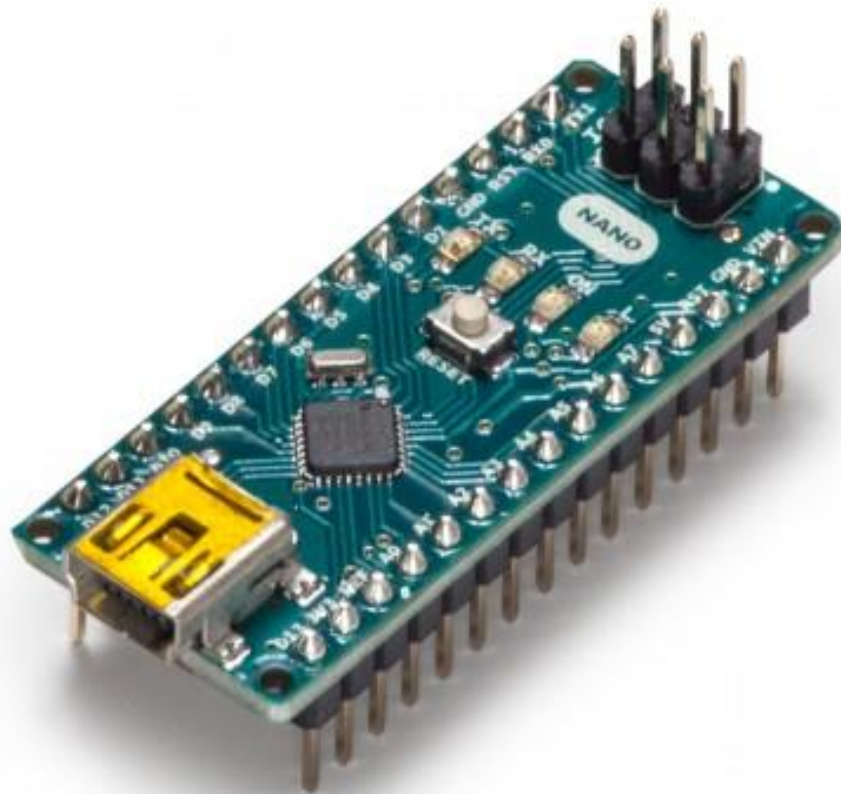


Рис.1.2.1. Фотографія Arduino Nano. Вид зверху.

1.1.2.1. Загальні відомості

Arduino Nano - це повнофункціональний мініатюрний пристрій на базі мікроконтролера ATmega328 (Arduino Nano 3.0) або ATmega168 (Arduino Nano 2.x), адаптоване для використання з макетної платі. За функціональністю пристрій схожий на Arduino Duemilanove, і відрізняється від нього розмірами, відсутністю роз'єму живлення, а також іншим типом (Mini-B) USB-кабелю. Arduino Nano розроблено і випускається фірмою Gravitech.

1.1.2.2. Характеристики[2]:

Мікроконтролер Atmel ATmega168 або ATmega328

Робоча напруга (логічний рівень) 5В

Напруга живлення (рекомендований) 7-12В

Напруга живлення (граничне) 6-20В

Цифрові входи / виходи 14 (з яких 6 можуть використовуватися як ШІМ-виходи)

Аналогові входи 8

Максимальний струм одного виведення 40 мА

Flash-пам'ять 16 КБ (АТmega168) або 32 КБ (АТmega328) з яких 2 КБ використовуються загрузчиком

SRAM 1 КБ (АТmega168) або 2 КБ (АТmega328)

EEPROM 512 байт (АТmega168) або 1 КБ (АТmega328)

Тактова частота 16 МГц

Розміри плати 1.85 см x 4.3 см

1.1.2.3. Живлення

Arduino Nano може живитися через кабель Mini-B USB, від зовнішнього джерела живлення з нестабілізованою напругою 6-20В (через вихід 30) або зі стабілізованою напругою 5В (через вихід 27). Пристрій автоматично вибирає джерело живлення з найбільшим напругою.

Напруга на мікросхему FTDI FT232RL подається тільки в разі живлення Arduino Nano через USB. Тому при живленні пристрою від інших зовнішніх джерел (НЕ USB), вихід 3.3В (формований мікросхемою FTDI) буде неактивний, в результаті чого світлодіоди RX і TX можуть мерехтіти при наявності високого рівня сигналу на висновках 0 і 1.

1.1.2.4. Пам'ять

Обсяг пам'яті програм мікроконтролера ATmega168 становить 16 КБ (з них 2 КБ використовуються загрузчиком); в ATmega328 - цей обсяг становить 32 КБ (з яких 2 КБ також відведені під завантажувач). Крім цього, ATmega168 має 1 КБ оперативної пам'яті SRAM і 512 байт EEPROM (для взаємодії з якої служить бібліотека EEPROM); а мікроконтролер ATmega328 - 2 КБ SRAM і 1 КБ EEPROM.

ARDUINO NANO

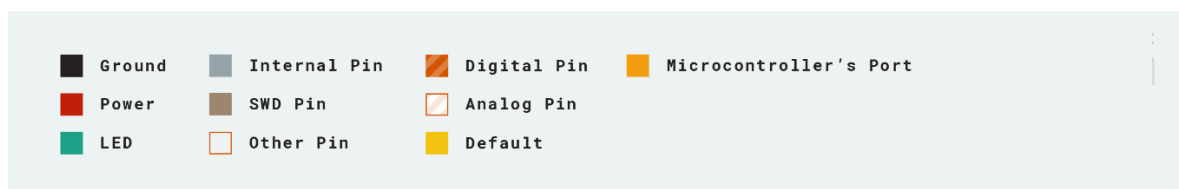
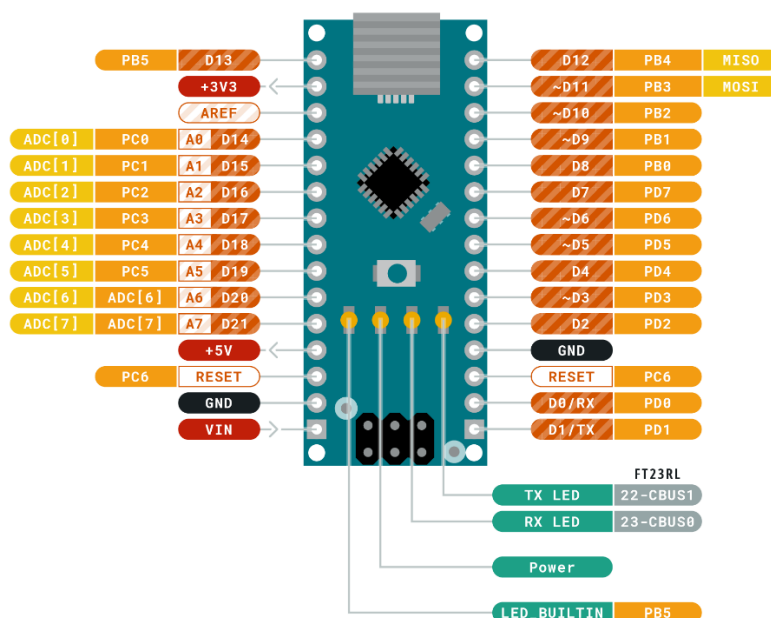


Рис. 1.2.2 Найменування входів та виходів на платі Arduino Nano.

1.1.2.5. Входи і виходи

З використанням функцій `pinMode ()`, `digitalWrite ()` і `digitalRead ()` кожен з 14 цифрових висновків Arduino Nano може працювати в якості входу або виходу. Робоча напруга висновків - 5В. Максимальний струм, який може віддавати або споживати один висновок, становить 40 мА. Всі висновки пов'язані з внутрішніми підтягуються резисторами (за замовчуванням відключеними) номіналом 20-50 кОм. Крім основних, деякі висновки Arduino можуть виконувати додаткові функції:

Послідовний інтерфейс: висновки 0 (RX) і 1 (TX). Використовуються для отримання (RX) і передачі (TX) даних по послідовному інтерфейсу. Ці висновки з'єднані з відповідними висновками мікросхеми-перетворювача USB-UART від FTDI.

Зовнішні переривання: висновки 2 і 3. Дані висновки можуть бути налаштовані в якості джерел переривань, що виникають при різних умовах: при низькому рівні сигналу, по фронту, по спаду або при зміні сигналу.

ШІМ: висновки 3, 5, 6, 9, 10 і 11. За допомогою функції `analogWrite ()` можуть виводити 8-бітові аналогові значення в вигляді ШІМ-сигналу.

Інтерфейс SPI: висновки 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Дані висновки дозволяють здійснювати зв'язок по інтерфейсу SPI. У пристрої реалізована апаратна підтримка SPI, проте на даний момент мова Ардуіно поки її не підтримує.

Світлодіод: висновок 13. Вбудований світлодіод, приєднаний до цифрового висновку 13. При відправці значення HIGH світлодіод включається, при відправці LOW - вимикається.

В Arduino Ethernet є 8 аналогових входів, кожен з яких може уявити аналогову напругу у вигляді 10-бітного числа (1024 різних значення). За замовчуванням, вимір напруги здійснюється щодо діапазону від 0 до 5 В.

Проте, верхню межу цього діапазону можна змінити, використовуючи висновок AREF і функцію `analogReference()`. Крім цього, деякі з висновків мають додаткові функції:

I2C: висновки 4 (SDA) і 5 (SCL). З використанням бібліотеки `Wire` (документація на веб-сайті `Wiring`) дані висновки можуть здійснювати зв'язок по інтерфейсу I2C (TWI).

Крім перерахованих на платі існує ще кілька висновків:

AREF. Опорна напруга для аналогових входів. Може бути задіяний функцією `analogReference()`.

Reset. Формування низького рівня (LOW) на цьому висновку призведе до перезавантаження мікроконтролера. Зазвичай цей висновок служить для функціонування кнопки скидання на платах розширення.

1.1.2.6. Зв'язок

Arduino Nano надає ряд можливостей для здійснення зв'язку з комп'ютером, ще одним Ардуіно або іншими мікроконтролерами. У ATmega168 і ATmega328 є приймач UART, що дозволяє здійснювати зв'язок з послідовним інтерфейсів за допомогою цифрових висновків 0 (RX) і 1 (TX). Мікросхема FTDI FT232RL забезпечує зв'язок приймача з USB-портом комп'ютера, і при підключенні до ПК дозволяє Ардуіно визначатися як віртуальний COM-порт (драйвера FTDI включені в пакет програмного забезпечення Ардуіно). У пакет програмного забезпечення Ардуіно також входить спеціальна програма, що дозволяє зчитувати і відправляти на Ардуіно прості текстові дані. При передачі даних комп'ютера через USB на платі будуть мигати світлодіоди RX і TX. (При послідовній передачі даних за допомогою висновків 0 і 1 дані світлодіоди задіюються).

Бібліотека SoftwareSerial дозволяє реалізувати послідовну зв'язок на будь-яких цифрових висновках Arduino Nano.

У мікроконтролерах ATmega328 і ATmega168 також реалізована підтримка послідовних інтерфейсів I2C (TWI) і SPI. У програмне забезпечення Ардуіно входить бібліотека Wire, що дозволяє спростити роботу з шиною I2C; для отримання більш докладної інформації див. документацію. Для роботи з інтерфейсом SPI см. Даташіта мікроконтролерів ATmega168 і ATmega328.

1.1.2.7. Програмування

Arduino Nano програмується за допомогою програмного забезпечення Ардуіно (скачати). Для цього з меню Tools> Board необхідно вибрати "Arduino Diecimila, Duemilanove, or Nano w / ATmega168" або "Arduino Duemilanove or Nano w / ATmega328" (в залежності від мікроконтролера на вашій платі).

ATmega168 і ATmega328 в Arduino Nano випускається з прошитим загрузчиком, що дозволяє завантажувати в мікроконтролер нові програми без необхідності використання зовнішнього програматора. Взаємодія з ним здійснюється за оригінальним протоколу STK500.

Проте, мікроконтролер можна прошити і через роз'єм для внутрішньо схемне програмування ICSP (In-Circuit Serial Programming), не звертаючи уваги на завантажувач.

1.1.3 Bluetooth модуль HC

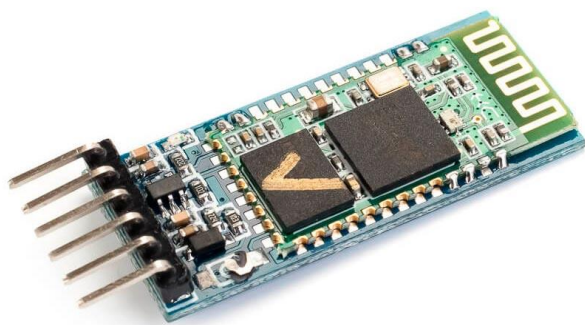


Рис. 1.1.3.1 Bluetooth модуль HC 05

1.1.3.1.Вступ

Продукти послідовного Bluetooth HC [3] складаються з модуля послідовного інтерфейсу Bluetooth та адаптера Bluetooth як:

(1) Модуль послідовного інтерфейсу Bluetooth:

Промисловий рівень: HC-03, HC-04 (HC-04-M, HC-04-S)

Цивільний рівень: HC-05, HC-06 (HC-06-M, HC-06-S)

HC-05-D, HC-06-D (з плінтусом, для випробування та оцінки)

(2) Адаптер Bluetooth:

HC-M4

HC-M6

Послідовний модуль Bluetooth використовується для перетворення послідовного порту в Bluetooth. Ці модулі мають два режими: головний і підлеглий пристрій. Пристрій, названий парним номером, визначається як ведучий або підлеглий, коли він не працює на заводі, і його не можна змінити на інший режим. Але для пристрою, іменованого непарним числом, користувачі

можуть встановити режим роботи (ведучий або робочий) пристрою за допомогою команд АТ. (ведучий або підлеглий) пристрою за командами АТ.

НС-04 зокрема включає:

Головний пристрій: НС-04-М, М = ведучий

Ведений пристрій: НС-04-S, S = раб

Ситуація за замовчуванням для НС-04 - ведений режим. Якщо вам потрібен головний режим, будь-ласка, чітко його заявіть або зробіть замовлення безпосередньо на НС-04-М.

Коли НС-03 та НС-05 не працюють із заводу, встановлюється одна частина параметрів для активації пристрою. Режим роботи не встановлений, оскільки користувач може встановити режим НС-03, НС-05 як завгодно.

Основною функцією послідовного модуля Bluetooth є заміна лінії послідовного порту, наприклад:

1. Є два мікроконтролери, які хочуть спілкуватися між собою. Один підключається до головного пристрою Bluetooth, а інший - до підлеглому пристрою. Їх з'єднання можна побудувати, як тільки пара буде створена.

Це Bluetooth-з'єднання еквівалентно з'єднанню послідовного порту, включаючи RXD, TXD

сигнали. І вони можуть використовувати послідовний модуль Bluetooth для спілкування між собою.

2. Коли MCU має модуль Bluetooth, він може спілкуватися з адаптером Bluetooth

комп'ютери та смартфони. Потім між віртуальним комунікаційним послідовним портом знаходиться лінія між MCU та комп'ютером або смартфоном.

3. Пристрої Bluetooth на ринку переважно є пристроями, такими як Bluetooth-принтер,

Bluetooth GPS. Отже, ми можемо використовувати головний модуль для створення пари та спілкування з ними.

Для роботи послідовного модуля Bluetooth не потрібен диск, і він може обмінюватися даними з іншим

Пристрій Bluetooth, який має серійний номер. Але для зв'язку між двома модулями Bluetooth потрібні принаймні дві умови:

- (1) Зв'язок повинен здійснюватися між ведучим і веденим.
- (2) Пароль повинен бути правильним.

Однак ці дві умови не є достатніми. Існують також деякі інші умови на основі іншої моделі пристрою. Детальна інформація наведена в наступних розділах.

У наступних розділах ми неодноразово посилатимемося на матеріали та фотографії Лінвора (раніше відомого як Гуанчжоу HC Information Technology Co., Ltd.).

1.1.3.2 Вибір модуля

Серійний модуль Bluetooth, названий парним номером, сумісний між собою; Модуль salve також сумісний між собою. Іншими словами, функції HC-04 та HC-06, HC-03 та HC-05 взаємно сумісні між собою. HC-04 та HC-06 - це попередні версії, в яких користувач не може скинути робочий режим (головний або ведений). І лише декілька команд і функцій AT можуть бути використані, як-от скинути ім'я Bluetooth (лише раб), скинути пароль, скинути швидкість передачі та перевірити номер версії. Набори команд HC-03 та HC-05 є більш гнучкими, ніж HC-04 та HC-06. Як правило, користувачеві рекомендується Bluetooth HC-03 / HC-05.

Ось основні заводські параметри HC-05 та HC-06. Зверніть увагу на відмінності:

HC-05	HC-06
Головний і ведений режими можна перемикаєти	Неможливо переключити головний та ведений режими
Назва Bluetooth: HC-05	Назва Bluetooth: linvor
Пароль: 1234	Пароль: 1234
Головна роль: не має функції запам'ятовувати останнє спарене пристрій для розмивання. Його можна зробити в парі з будь-яким підлеглим пристроєм. Іншими словами, просто встановіть AT + CMODE = 1, коли вийшли з заводу. Якщо ви хочете, щоб HC-05 запам'ятав останню пару пристрою-веденого пристрою, наприклад HC-06, можна встановити AT + CMODE = 0 після створення пари з іншим пристроєм. Для отримання детальної інформації зверніться до набору команд HC-05.	Головна роль: мати парну пам'ять для запам'ятовування останнього підпорядкованого пристрою та встановлювати пару з цим пристроєм лише за умови, що КЛЮЧ (PIN26) спрацьовує на високому рівні. За замовчуванням підключений PIN26 - низький рівень.
Сполучення: головний пристрій може не тільки здійснювати з вказаною адресою Bluetooth, наприклад стільниковий телефон, адаптер комп'ютера, підлеглий пристрій, але також може автоматично шукати та встановлювати пару з підлеглим пристроєм. Типовий метод:	Сполучення: головний пошук пристрою та автоматичне встановлення пари з веденим пристроєм. Типовий метод: За певних умов головний і ведений пристрій можуть автоматично з'єднувати пари між собою.

<p>За певних умов головний пристрій і підлеглий пристрій можуть автоматично з'єднувати пари між собою. (Це метод за замовчуванням.)</p>	
<p>Зв'язок із кількома пристроями: для модулів існує лише зв'язок точка-точка, але адаптер може взаємодіяти з мультимодулями.</p>	<p>Зв'язок із кількома пристроями: для модулів існує лише зв'язок точка-точка, але адаптер може взаємодіяти з мультимодулями.</p>
<p>Режим АТ 1: Після включення живлення він може увійти в АТ режимі, запускаючи PIN34 з високим рівнем. Тоді швидкість передачі даних для встановлення команди АТ дорівнює швидкості передачі даних у зв'язку, наприклад: 9600. Режим АТ 2: Спочатку встановіть PIN34 як високий рівень, або під час включення модуля встановіть PIN34 на високий рівень, Baud тут використовується 38400 біт / с. Примітка: Усі команди АТ можуть працювати лише тоді, коли PIN34 знаходиться на високому рівні. Лише частину команд АТ можна використовувати, якщо PIN34 не підтримує високий рівень після переходу в режим АТ. Завдяки такому проектуванню встановлені дозволи для модуля залишаються за зовнішнім</p>	<p>Режим АТ: перед створенням пари він знаходиться в режимі АТ. Після створення пари відбувається прозоре спілкування.</p>

<p>користувацьким ланцюгом управління, що робить застосування HC-05 дуже гнучким.</p>	
<p>Під час процесу зв'язку модуль може перейти в режим АТ, встановивши PIN34 на високий рівень. Випустивши PIN34, модуль може повернутися до режиму зв'язку, в якому користувач може динамічно запитувати деяку інформацію. Наприклад, щоб дізнатись, чи поєднання закінчено чи ні.</p>	<p>Під час режиму зв'язку модуль не може перейти в режим АТ.</p>
<p>Швидкість передачі даних за замовчуванням: 9600, 4800-1,3 Мб.</p>	<p>Швидкість передачі даних за замовчуванням: 9600, 1200-1,3 Мб.</p>
<p>КЛЮЧ: PIN34, для входу в режим АТ</p>	<p>КЛЮЧ: PIN26, для майстра залишає пам'ять.</p>
<p>LED1: PIN31, індикатор режиму Bluetooth. Повільне мерехтіння (1 Гц) являє собою вхід в АТ ode2, тоді як швидке мерехтіння (2 Гц) означає вхід в режим АТ 1 або під час з'єднання в пару. Подвійне мерехтіння в секунду означає, що спарювання закінчено, модуль є комунікативним. LED2: PIN32, перед сполученням на низькому рівні, після сполучення на високому рівні. Метод використання показників господаря та раба однаковий. Примітка: PIN-код LED1</p>	<p>Світлодіод: Частота мерехтіння веденого пристрою становить 102 мс. Якщо головний пристрій вже має пам'ять веденого пристрою, частота мерехтіння під час сполучення становить 110 мс / с. Якщо ні, або майстер спорожнив пам'ять, частота мерехтіння становить 750 м / с. Після сполучення, незалежно від основного або підлеглого пристрою, світлодіодний PIN-код знаходиться на високому рівні.</p>

та LED2 підключений до LED +.	
Витрата: Під час сполучення струм коливається в діапазоні 30-40мА. Середній струм становить близько 25 мА. Після розбору, незалежно від обробки зв'язку чи ні, струм становить 8 мА. Режим сну відсутній. Цей параметр однаковий для всіх модулів Bluetooth.	Витрата: Під час сполучення струм коливається в діапазоні 30-40 м. Середній струм становить близько 25 мА. Після розбору, незалежно від обробки зв'язку чи ні, струм становить 8 мА. Режим сну відсутній. Цей параметр однаковий для всіх модулів Bluetooth.
Скидання: PIN11, активний, якщо він вводить низький рівень. Його можна призупинити у використанні.	Скидання: PIN11, активний, якщо він вводить низький рівень. Його можна призупинити у використанні.
Рівень: Цивільний	Рівень: Цивільний

Таблиця 1.1.3.1. У таблиці вище, яка включає основні параметри двох послідовних модулів.

Рекомендується серійний продукт HC-03 / HC-05.

1.1.4.Arduino Shields

1.1.4.1 .Arduino Shields - плати розширення для Ардуіно

Одним з ключових переваг платформи Arduino є популярність. Популярну платформу активно підтримують виробники електронних пристроїв, що випускають спеціальні версії різних плат, що розширюють базову функціональність контролера. Такі плати, абсолютно логічно звані платами розширення (інша назва: arduino shield, Шилд), служать для виконання найрізноманітніших.

Плата розширення Ардуіно [4] - це закінчений пристрій, призначене для виконання певних функцій і підключається до основного контролера за допомогою стандартних роз'ємів. Інша популярна назва плати розширення - англomовне Arduino shield або просто Шилд. На платі розширення встановлені

всі необхідні електронні компоненти, а взаємодія з мікро контролером і іншими елементами основної плати відбуваються через стандартні Піни Ардуіно. Найчастіше харчування на Шилд теж подається з основної плати arduino, хоча в багатьох випадках є можливість живлення з інших джерел. У будь-якому Шилд залишаються кілька вільних пинов, які ви можете використовувати на свій розсуд, підключивши до них будь-які інші компоненти.

1.1.4.2. Навіщо потрібні Shield arduino?

Навіщо витрачати час, проектуючи, розміщуючи, паяти і налагоджуючи те, що можна взяти вже в зібраному варіанті, відразу почавши використовувати? Добре продумані і зібрані на якісному обладнанні плати розширення, як правило, більш надійні і займають менше місця в кінцевому пристрої. Це не означає, що потрібно повністю відмовлятися від самостійного складання і не потрібно розбиратися в принципі дії тих чи інших елементів. Адже справжній інженер завжди намагається зрозуміти, як працює той, що він використовує. Але ми зможемо робити більш складні пристрої, якщо не будемо щоразу винаходити велосипед, а зосередимо свою увагу на тому, що до нас ще мало хто вирішував.

Природно, за можливості доводиться платити. Практично завжди вартість кінцевого Шилд буде вищою за ціну окремих комплектуючих, завжди можна зробити аналогічний варіант подешевше. Але тут вже вирішувати вам, наскільки критично для вас витрачені час або гроші. З урахуванням посильної допомоги китайської промисловості, вартість плат постійно знижується, тому найчастіше вибір робиться на користь використання готових пристроїв.

Найбільш популярним прикладами Шилд є плати розширення для роботи з датчиками, двигунами, LCD-екранами, SD-картами, мережеві і GPS-Шилд, Шилд з вбудованими реле для підключення до навантаження.

1.1.4.3. Програмування Arduino Shield

Програмування схеми з платою розширення не відрізняється від звичайного програмування Ардуіно, адже з точки зору контролера ми просто підключили наші устрйоства до його звичайним пінам. У скетчі потрібно вказувати ті Піни, які з'єднані в Шілд з відповідними контактами на платі. Як правило, виробник вказує відповідність пінів на самому Шілд або в окремій інструкції з підключення. Якщо ви скачаєте скетчі, рекомендовані самим виробником плати, то навіть це робити не знадобиться.

Читання або запис сигналів Шілд проводиться теж звичайним методом: за допомогою функцій `analogRead ()`, `digitalRead ()`, `digitalWrite ()` та інших команд. У деяких випадках можливі колізії, коли ви звикли до неї схемою з'єднання, а виробник вибрав іншу (наприклад, ви підтягували кнопку до землі, а на Шілд - до живлення).

1.1.4.4. Arduino Sensor Shield

Як правило, ця плата розширення йде в наборах Ардуіно і тому саме з нею зустрічаються найчастіше. Шілд досить простий - його основне завдання надати більш зручні варіанти підключення до плати Arduino. Це здійснюється за рахунок додаткових роз'ємів живлення і землі, виведених на плату до кожного з аналогових і цифрових пинов. Також на платі можна знайти роз'єми для підключення зовнішнього джерела живлення (для перемикання потрібно встановити перемички), світлодіод і кнопка перезапуску.

Існує кілька версій сенсорної плати розширення. Всі вони відрізняються кількістю і видом роз'ємів. Найбільш популярними сьогодні є версії Sensor Shield v4 і v5.

1.1.4.5.Arduino Motor Shield

Даний Шилд Ардуіно дуже важливий в робототехнічних проектах, тому що дозволяє підключати до плати Arduino відразу звичайний і серво двигуни. Основне завдання Шилд - забезпечити управління пристроями споживають досить високий для звичайної плати Ардуіно струм. Додатковим можливостями плати є функція управління потужністю двигуна (з допомогою ШІМ) і зміни напрямку обертання. Існує безліч різновидів плат motor shield. Загальним для всіх них є наявність в схемі потужного транзистора, через який підключається зовнішнє навантаження, тепловідвідних елементів (як правило, радіатора), схеми для підключення зовнішнього живлення, роз'ємів для підключення двигунів і піни для підключення до Ардуіно.

1.1.4.6.Arduino Ethernet Shield

Організація роботи з мережею - одна з найважливіших задач в сучасних проектах. Для підключення до локальної мережі через Ethernet існує відповідна плата розширення.

1.1.4.7.Плати розширення для прототипування

Ці плати досить прості - на них розташовані контактні площадки для монтажу елементів, виведена кнопка скидання і є можливість підключення зовнішнього живлення. Призначення даних Шилд - підвищити компактність пристрою, коли всі необхідні компоненти розташовуються відразу над основною платою.

1.1.4.8.Arduino LCD shield і tft shield

Даний тип Шилд використовується для роботи з LCD-екранами в Ардуіно. Як відомо, підключення навіть самого простого 2-сатиричного текстового екрану далеко не тривіальна задача: потрібно правильно підключити відразу 6 контактів екрану, не рахуючи харчування. Набагато простіше вставити готовий модуль в плату Ардуіно і просто завантажити відповідний

скетч. У популярному LCD Keypad Shield на плату зразу заведені від 4 до 8 кнопок, що дозволяє СРЗА організувати і зовнішній інтерфейс для користувача пристрою. TFT Shield також допомагає.

1.1.4.9.Arduino Data Logger Shield

Ще одне завдання, яке досить важко реалізувати самостійно в своїх виробках - це збереження даних, отриманих з датчиків, з прив'язкою за часом. Готовий Шилд дозволяє не тільки зберегти дані і отримувати час зі вбудованих годин, але і підключити датчики в зручному вигляді шляхом пайки або на монтажній платі.

Мінусом використання плат розширення можна назвати їх вартість і можливий втрати ефективності через універсальність Шилд, що лежить в їх природі. Для вашої вузької задачі або кінцевого пристрою всі функції Шилд можуть бути не потрібні. У такому випадку варто використовувати Шилд тільки на етапі макетування і тестування, а при створенні фінального варіанту свого пристрою задуматися про заміну конструкцією з власної схемою і типом компонування.

1.2. Мова програмування мікроконтролерів Arduino.

1.2.1 Arduino, спочатку підтримує мову, яку ми називаємо мовою програмування Arduino, або мовою Arduino.

Ця мова заснований на Wiring платформи розробки, яка , в свою чергу , заснований на обробці , що , якщо ви не знайомі з, що p5.js ґрунтується. Це довга історія проектів, що базуються на інших проектах, у дуже відкритому коді. IDE Arduino заснована на IDE обробки та IDE підключення, яка будується поверх неї.[5]

Коли ми працюємо з Arduino, ми зазвичай використовуємо Arduino IDE (інтегроване середовище розробки), програмне забезпечення, доступне для всіх

основних настільних платформ (macOS, Linux, Windows), яке дає нам дві речі: редактор програмування з підтримкою інтегрованих бібліотек і спосіб легко скомпілювати та завантажити наші програми Arduino на плату, підключену до комп'ютера.

Мова програмування Arduino - це, в основному, фреймворк, побудований поверх C ++. Ви можете заперечити, що це не справжня мова програмування в традиційному терміні, але я думаю, це допомагає уникнути плутанини для початківців.

Програма, написана мовою програмування Arduino, називається ескізом . Зазвичай ескіз зберігається з .ino розширенням (від Arduino).

Основна відмінність від “звичайного” C або C ++ полягає в тому, що ви обертаєте весь свій код у 2 основні функції. Звичайно, ви можете мати більше двох, але будь-яка програма Arduino повинна містити принаймні ці 2.

Один називається `setup()`, другий називається `loop()`. Перший викликається один раз, коли програма запускається, другий повторно викликається під час роботи вашої програми.

У нас немає такої `main()` функції, як ви звикли в C / C ++, як точки входу для програми. Після того, як ви скомпілюєте свій ескіз, IDE переконається, що кінцевим результатом є правильна програма на C ++, і в основному додасть відсутній клей, попередньо обробивши його.

Все інше є звичайним кодом C ++, і оскільки C ++ є надмножиною C, будь-який дійсний C також є дійсним кодом Arduino.

Одна відмінність, яка може викликати у вас проблеми, полягає в тому, що, хоча ви можете створити програму на декількох файлах, ці файли повинні знаходитися в одній папці. Можливо обмеження угоди, якщо ваша програма стане дуже великою, але в цей момент буде легко перейти до власної установки C ++, що можливо.

Частина мови програмування Arduino - це вбудовані бібліотеки, які дозволяють легко інтегруватися з функціональними можливостями, що надаються платою Arduino.

1.2.2. Підтримка іншої мови

Нагадую, я хочу зазначити, що ви не обмежуєтесь використанням цієї мови та IDE для програмування Arduino. Проекти існують, в зокрема, щоб запуснути Node.js код на ньому з допомогою Johnny Five проекту, Python коду з використанням pyserial і Go код з Gobot , але Arduino Мова програмування, безумовно, один ви будете бачити більшість посібників , заснованих на, так як це рідний і канонічний спосіб роботи з цими пристроями.

1.2.3. Вбудовані константи мови програмування Arduino

Arduino встановлює дві константи, які ми можемо використати

HIGH прирівнює до високого рівня напруги, який може відрізнитися залежно від апаратного забезпечення (> 2 В на платах 3,3 В, таких як Arduino Nano,> 3 В на платах 5 В, як Arduino Uno) LOW прирівнює низький рівень напруги. Знову ж таки, точне значення залежить від використовуваної дошки

Тоді у нас є 3 константи, які ми можемо використовувати в поєднанні з pinMode()функцією:

INPUT встановлює штифт як вхідний штифт

OUTPUT встановлює штифт як вихідний штифт

INPUT_PULLUP встановлює штифт як внутрішній підтягуючий резистор

Інша константа, яку ми маємо, - LED_BUILTIN це вказівка на номер вбудованого штифта, який зазвичай дорівнює числу 13.

На додаток до цього, у нас є константи C / C ++ true та false.

Arduino Math Константи

M_PI константа π (3.14159265358979323846)

M_E константа e

M_LN10 натуральний логарифм числа 10.

M_LN2 натуральний логарифм числа 2.

M_LOG10E логарифм e до основи 10.

M_LOG2E логарифм e до основи 2.

M_SQRT2 квадратний корінь з 2.

NAN константа NAN (не число).

Вбудовані функції мови програмування Arduino

У цьому розділі я збираюся зробити посилання на вбудовані функції, передбачені мовою програмування Arduino.

1.2.4. Життєвий цикл програми

setup() ця функція викликається один раз, коли програма запускається і коли Arduino вимикається і перезапускається.

loop() ця функція неодноразово викликається під час роботи програми Arduino.

Обробка вводу-виводу

Наступні функції допомагають обробляти вхідні та вихідні дані з вашого пристрою Arduino.

Цифровий ввід / вивід

digitalRead()зчитує значення з цифрового штифта. Приймає номер параметра як параметр і повертає HIGH або LOW константу.

digitalWrite()записує значення HIGH або LOW на цифровий вихідний штифт. Ви передасте номер пину та HIGH або LOW як параметри.

`pinMode()` встановлює штифт як вхід або вихід. Ви передаєте номер PIN-коду та значення INPUT або OUTPUT як параметри.

`pulseIn()` зчитує цифровий імпульс від LOW до, HIGH а потім LOW знову, або від HIGH до LOW і HIGH знову на штифті. Програма буде блокувати, поки не буде виявлено імпульс. Ви вказуєте номер виводу та тип імпульсу, який ви хочете виявити (LHL або HLH). Ви можете вказати необов'язковий тайм-аут, щоб припинити очікування цього імпульсу.

`pulseInLong()` те саме `pulseIn()`, що за винятком того, що реалізовано інакше, і його не можна використовувати, якщо переривання вимкнено. Переривання зазвичай вимикають, щоб отримати точніший результат.

`shiftIn()` зчитує байт даних по одному біту з піна.

`shiftOut()` записує байт даних по одному біту за раз на пін.

`tone()` посилає квадратну хвилю на штифті, яка використовується для звукових сигналів / динаміків для відтворення тонів. Ви можете вказати штифт та частоту. Він працює як на цифрових, так і на аналогових контактах.

`noTone()` зупиняє `tone()` генеровану хвилю на штифті.

Аналоговий ввід / вивід

`analogRead()` зчитує значення з аналогового штифта.

`analogReference()` конфігурує значення, яке використовується для верхнього діапазону вхідного сигналу в аналоговому вході, за замовчуванням 5 В на платах 5 В і 3,3 В на платах 3,3 В.

`analogWrite()` записує аналогове значення на штифт

`analogReadResolution()` дозволяє змінити роздільну здатність аналогових бітів `analogRead()` за замовчуванням для , за замовчуванням 10 біт. Працює лише на певних пристроях (Arduino Due, Zero та MKR)

`analogWriteResolution()` дозволяє змінити роздільну здатність аналогових бітів `analogWrite()` за замовчуванням для , за замовчуванням 10 біт. Працює лише на певних пристроях (Arduino Due, Zero та MKR)

1.2.5. Часові функції

`delay()` призупиняє програму на кількість мілісекунд, вказану як параметр

`delayMicroseconds()` призупиняє програму на кількість мікросекунд, вказаних як параметр

`micros()` кількість мікросекунд з моменту запуску програми. Скидається через ~ 70 хвилин через переповнення

`millis()` кількість мілісекунд з моменту запуску програми. Скидається через ~ 50 днів через переповнення

Математичні функції

`abs()` абсолютне значення числа

`constrain()` обмежує число в межах діапазону, див. використання

`map()` змінює число з одного діапазону на інший, див. використання

`max()` максимум два числа

`min()` мінімум два числа

`pow()` значення числа, піднесеного до степеня

`sq()` квадрат числа

`sqrt()` квадратний корінь із числа

`cos()` косинус кута

`sin()` синус кута

`tan()` тангенс кута

Примітка. Існує більше вбудованих математичних функцій, якщо вони вам потрібні, про що задокументовано тут .

1.2.6. Робота з буквено-цифровими символами

`isAlpha()` перевіряє, чи є символ символом альфа (буква)

`isAlphaNumeric()` перевіряє, чи символ є буквено-цифровим (буква або цифра)

`isAscii()` перевіряє, чи є символ символом ASCII

`isControl()` перевіряє, чи є символ символом управління

`isDigit()` перевіряє, чи символом є число

`isGraph()` перевіряє, чи є символ символом ASCII для друку та містить вміст (наприклад, це не пробіл)

`isHexadecimalDigit()` перевіряє, чи є символом шістнадцяткова цифра (AF 0-9)

`isLowerCase()` перевіряє, чи символом є буква з малої літери

`isPrintable()` перевіряє, чи є символ символом ASCII для друку

`isPunct()` перевіряє, чи символом є розділові знаки (кома, крапка з комою, знак оклику тощо)

`isSpace()` перевіряє, чи \fсимволом є пробіл, стрічка форми , новий рядок \n, повернення каретки \r, горизонтальна \tта вертикальна вкладка \v.

`isUpperCase()` перевіряє, чи є символ символом великої літери

`isWhitespace()` перевіряє, чи символом є пробіл або горизонтальна вкладка \t

Генерація випадкових чисел

`random()` генерувати псевдовипадкове число

`randomSeed()` ініціалізувати генератор псевдовипадкових чисел довільним початковим числом

В Arduino, як і в більшості мов, неможливо отримати дійсно випадкові числа, а послідовність завжди однакова, тому ви породжуєте її з поточним часом або (у випадку з Arduino) ви можете читати вхідні дані з аналогового порту[] .

1.2.7. Робота з бітами та байтами

`bit()` обчислює значення біта (0 = 1, 1 = 2, 2 = 4, 3 = 8 ...)

`bitClear()`clear (встановлює 0) біт числової змінної. Приймає число та номер біта, що починається справа

`bitRead()`прочитати трохи числа. Приймає число та номер біта, що починається справа

`bitSet()`встановлює 1 біт числа. Приймає число та номер біта, що починається справа

`bitWrite()` запис 1 або 0 у певний біт числа Приймає число, номер біта, що починається справа, і значення для запису (0 або 1)

`highByte()` отримати старший (лівий) байт змінної слова (який має 2 байти)

`lowByte()` отримати нижчий (крайній правий) байт змінної слова (який має 2 байти)

1.2.8. Переривання

`noInterrupts()` відключає переривання

`interrupts()` знову вмикає переривання після їх відключення

`attachInterrupt()` дозволяють цифровий вхідний штифт бути перериванням. Різні дошки мають різні дозволені шпильки, перевірте офіційні документи .

`detachInterrupt()` вимикає переривання, увімкнене за допомогою `attachInterrupt()`

РОЗДІЛ 2. АЛГОРИТМ УПРАВЛІННЯ СЕРВОПРИВОДАМИ ОБ'ЄКТА, ЩО РУХАЄТЬСЯ.

2.1. Arduino - бібліотека Servo.

Дозволяє платам Arduino / Genuino керувати різноманітними сервомоторами.

Ця бібліотека може контролювати велику кількість сервоприводів. Він обережно використовує таймери: бібліотека може керувати 12 сервоприводами, використовуючи лише 1 таймер. На Arduino Due ви можете контролювати до 60 сервоприводів.

2.1.1. Використання

Ця бібліотека дозволяє платі Arduino керувати сервомоторами RC. Сервоприводи мають вбудовані шестерні та вал, якими можна точно керувати. Стандартні сервоприводи дозволяють розташовувати вал під різними кутами, зазвичай від 0 до 180 градусів. Сервоприводи безперервного обертання дозволяють встановлювати обертання вала на різні швидкості.

Бібліотека Servo підтримує до 12 двигунів на більшості плат Arduino та 48 на Arduino Mega. На платах, відмінних від Mega, використання бібліотеки відключає функціональність `analogWrite()`(ШІМ) на штифтах 9 і 10, незалежно від того, чи є сервопривід на цих штифтах. На Mega можна використовувати до 12 сервоприводів, не заважаючи функціоналу ШІМ; використання 12-23 двигунів вимкне ШІМ на висновках 11 і 12.

Щоб користуватися цією бібліотекою:

```
#include <Servo.h>
```

Сервомотори мають три дроти: живлення, землю та сигнал. Провід живлення, як правило, червоний і повинен бути підключений до 5-контактного виводу на платі Arduino. Провід заземлення, як правило, чорний або коричневий і повинен бути підключений до штиря заземлення на платі Arduino. Сигнальний штифт, як правило, жовтий, оранжевий або білий і повинен бути підключений до цифрового штифта на платі Arduino. Зверніть увагу, що сервоприводи отримують значну потужність, тому, якщо вам потрібно керувати кількома або двома, вам, ймовірно, доведеться жити їх від окремого джерела живлення (тобто не від штифта + 5 В на вашому Arduino). Обов'язково під'єднайте основи Arduino та зовнішнє джерело живлення разом.

Методи

attach()

write()

writeMicroseconds()

read()

attached()

detach()

2.1.2. Як користуватися сервотекою Arduino

Кращий спосіб генерувати сигнали керування сервоприводом - це включити в свій ескіз бібліотеку Arduino Servo, одну зі стандартних бібліотек попередньо записаного коду, що входить до складу програмного забезпечення Arduino.

Щоб переглянути список бібліотек Arduino, клацніть меню Довідка програмного забезпечення Arduino та виберіть Довідка.

Знайдіть і перейдіть за посиланням Бібліотеки.

Ми хочемо ближче поглянути на сервотеку.

Знайдіть і перейдіть за посиланням Servo.

Перейдіть і прочитайте посилання для цих функцій на сторінці сервобібліотеки:

`attach()`

`writeMicroseconds()`

`detach()`

Сервоприводи повинні отримувати високоімпульсні контрольні сигнали через рівні проміжки часу, щоб продовжувати крутитись. Якщо сигнал зупиняється, зупиняється і сервопривід. Як тільки ваш ескіз використовує сервотеку для налаштування сигналу, він може переходити до іншого коду, наприклад, затримки, перевірка датчиків тощо. Тим часом сервопривід продовжує обертатися, оскільки бібліотека сервоприводу продовжує працювати у фоновому режимі. Він регулярно перериває виконання іншого коду, щоб ініціювати ці високі імпульси, роблячи це так швидко, що це практично непомітно.

Використання сервотеки для надсилання сигналів керування сервоприводом займає чотири кроки:

Скажіть редактору Arduino, що ви хочете отримати доступ до функцій сервобібліотеки за допомогою цієї декларації на початку ескізу перед функцією налаштування .

```
#include <Servo.h> // Включити сервотеку
```

Заявіть і назвіть екземпляр бібліотеки сервоприводу для кожного сигналу, який потрібно надіслати, між `#include` і функцією налаштування .


```
Servo servoLeft; // Оголосити лівий сервопривід
```

У функції налаштування використовуйте ім'я, яке ви дали сервосигналу, після чого крапка, а потім виклик функції приєднання, щоб приєднати сигнальний штифт. Цей приклад повідомляє системі, що сервосигнал з назвою `servoLeft` повинен передаватися цифровим штифтом 13.

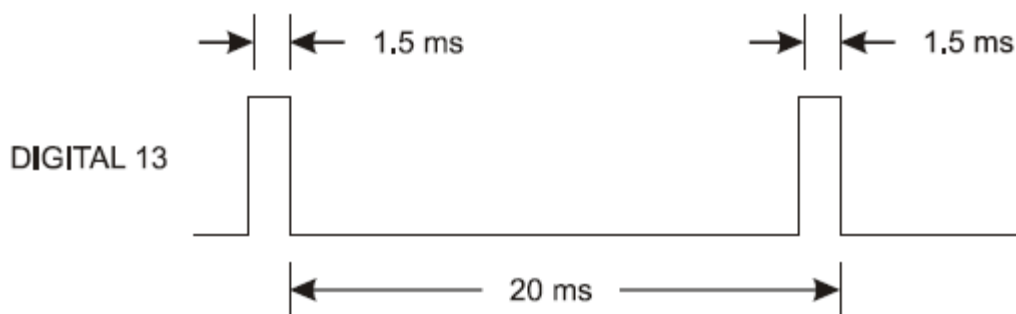
```
servoLeft.attach (13); // Прикріпіть лівий сигнал до виводу 13
```

Використовуйте функцію `writeMicroseconds` для встановлення часу імпульсу. Ви можете зробити це всередині установки або функції циклу :

```
servoLeft.writeMicroseconds (1500); // 1,5 мс сигналу нерухомості.
```

Приклад ескізу: `LeftServoStayStill`

Для калібрування сервоприводів ескізу потрібно буде надсилати сигнали з імпульсами 1,5 мс. Погляньте на часову діаграму нижче. Високі імпульси цього сигналу нерухомості тривають 1,5 мс. Це половина між імпульсами 1,7 мс на повній швидкості проти годинникової стрілки та 1,3 мс на повній швидкості за імпульсами.



/*

Робототехніка з BOE Shield - `LeftServoStayStill`

Генерувати сигнал, щоб сервопривід залишався нерухомим для центрування. */

```

#include <Servo.h>                // Включити сервотеку

Servo servoLeft;                 // Оголосимо лівий серво

void setup()                     // Вбудований блок ініціалізації
{
    servoLeft.attach(13);        // Приєднати лівий сигнал до піна 13

    servoLeft.writeMicroseconds(1500); // 1.5 мс сигнал залишається
нерухомим }

void loop()                      // Головний цикл автоматично
повторюється

{                                // Порожній, нічого не потрібно повторювати

```

2.2 Arduino - бібліотека SerialCommand

Бібліотека Arduino для послідовних команд через послідовний порт

Основним наміром цієї бібліотеки було впровадження пристрою, здатного реагувати на команди AT, наприклад "AT", "AT + RST", таких команд, яких ви часто бачите в модемах або SoC, таких як ESP8266 або SIM900.

2.2.1. Типи команд AT та відповіді

Test Command AT + =? Перевірте існування команди та перевірте її діапазон підпараметрів.

Read Command AT +? Ця команда повертає встановлене на даний момент значення параметрів.

Write Command AT + = <...> Ця команда встановлює визначені користувачем значення параметрів.

Execution Command AT + Ця команда виконує функцію.

2.2.2. Команди без параметрів:

Додавання нової команди вимагає 2 речей, рядка та вказівника на функцію void. Рядок використовується для порівняння вводу користувача з наявними командами. Функція буде викликана, коли команда отримана правильно.

```
mySerialCMD.addCommand ("AT", ping);
```

Дійсні дані користувача:

```
AT <CR>
```

2.2.3. Команди з параметрами:

Оголошення для параметричних команд таке саме, як і оголошення без параметрів

```
mySerialCMD.addCommand ("AT + LED", ledHandler);
```

При використанні ці команди повинні містити '=' char для першого параметра та ',' для кінцевих параметрів.

Дійсні дані користувача:

```
AT + LED = 1 <CR>
```

Для того, щоб отримати параметри, ми використовуємо метод "next", який повертає покажчик на наступний параметр. ПРИМІТКА. Щоб знати, чи параметр дійсний, слід завжди перевіряти, чи має значення NULL.

```
param = mySerialCMD.next ();
```

```
if (param! = NULL) // Зроби щось
```

2.3 Алгоритми переміщення об'єкта що рухається.

2.3.1.Опис конструкції, та їх особливостей.

Конструктивно було обрано мімікрувати рухам павука[6]. Робот має кінцівки з 3 степенями свободи (3DOF), тобто кінець, або ступня такої кінцівки має можливість рухатися в трьох напрямках в просторі. Таких кінцівок робот має 4, і це достатньо для того, щоб мати високу гнучкість, та стабільність при виконанні команд оператора.

Така нога має 3 сервоприводи, по одному в суглобі, які забезпечують 3DOF. Схематично конструкцію кінцівки зображена на Рис 2.3.1.1. .

Кут відведення стегна (θ)

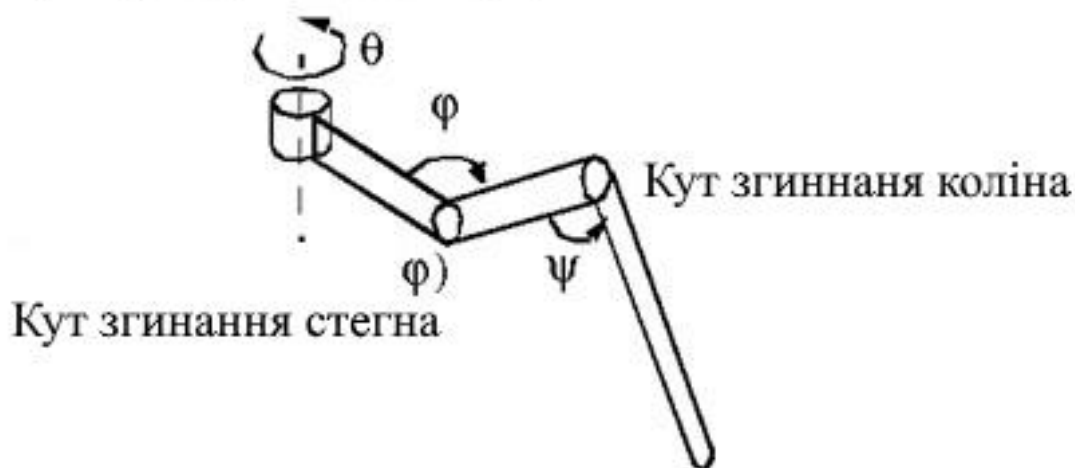


Рис 2.3.1.1. Схема ноги робота з вказанням назв частин, та позначенням кутів.

Сам робот має такий вигляд, 3d рендер.

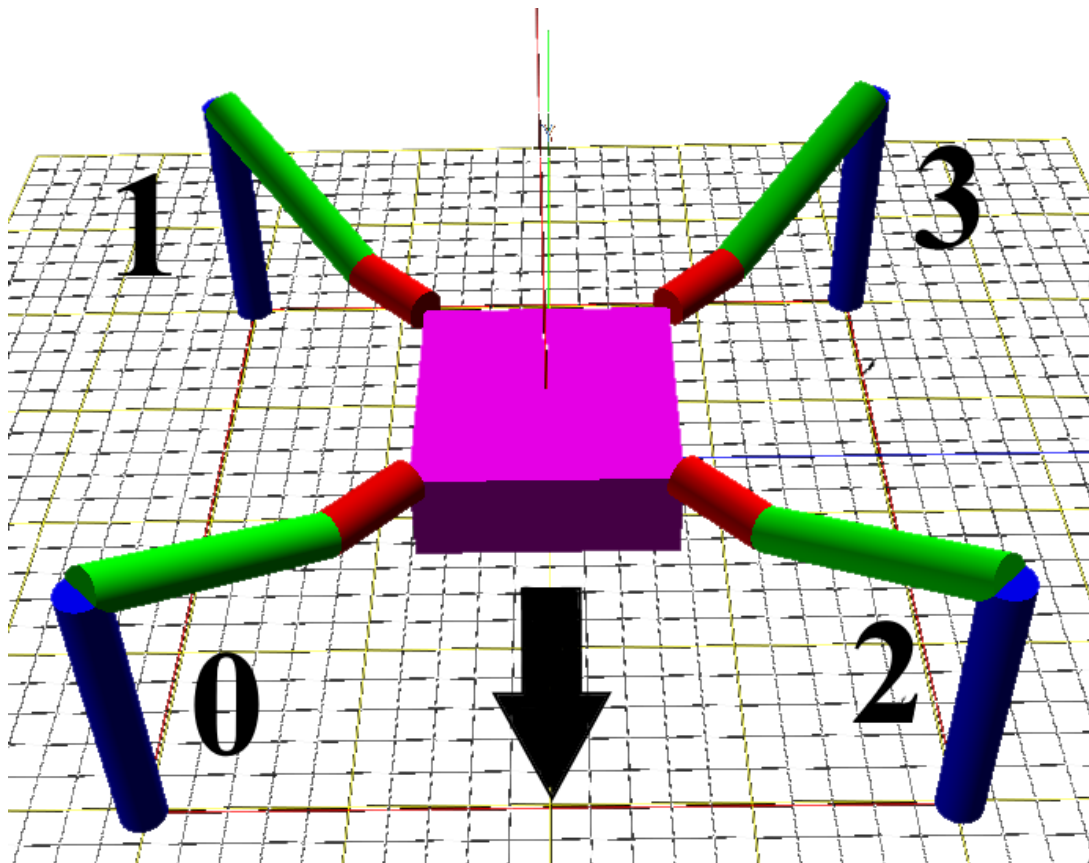


Рис. 2.3.1.2 3d зображення робота прораховане в спеціальній програмі. Нумерація всіх кінцівок.

Рух робота у просторі відбувається завдяки зміні координат ступні ноги в просторі відносно корпусу і далі знаючи ці координати, завдяки спеціальним формулам вираховуються кути для 12 сервоприводів розташованих в 4 кінцівках, по 3 сервоприводи на кінцівку. Кожна стопа має свою вісь x, y, z , та область в якій вона може рухатись. Початкова точка для кожної ноги автоматично задається в конфігурації прошивки, знаючи довжину ніг робота, а також розміри ого бази[7].

2.3.2. Алгоритм руху робота в прямому напрямку.

Для розуміння алгоритму руху необхідно зобразити схему Рис 2.3.2.1. Також необхідно об'явити деякі константи, які будуть використовуватися в рівняннях.

Всі значення в декартовій системі координат $X Y Z$

```

z_default = -50, z_up = -30, z_boot = z_absolute;

x_default = 62, x_offset = 0;

y_start = 0, y_step = 40

length_a = 55 стегно

length_b = 77.5 гомілка

length_c = 27.5 стегно

length_side = 71 довжина бази

z_absolute = -28;

temp_a = sqrt(pow(2 * x_default + length_side, 2) + pow(y_step, 2));

temp_b = 2 * (y_start + y_step) + length_side;

temp_c = sqrt(pow(2 * x_default + length_side, 2) + pow(2 * y_start + y_step
+ length_side, 2));

temp_alpha = acos((pow(temp_a, 2) + pow(temp_b, 2) - pow(temp_c, 2)) / 2 /
temp_a / temp_b);

turn_x1 = (temp_a - length_side) / 2;

turn_y1 = y_start + y_step / 2;

turn_x0 = turn_x1 - temp_b * cos(temp_alpha);

turn_y0 = temp_b * sin(temp_alpha) - turn_y1 - length_side;

```

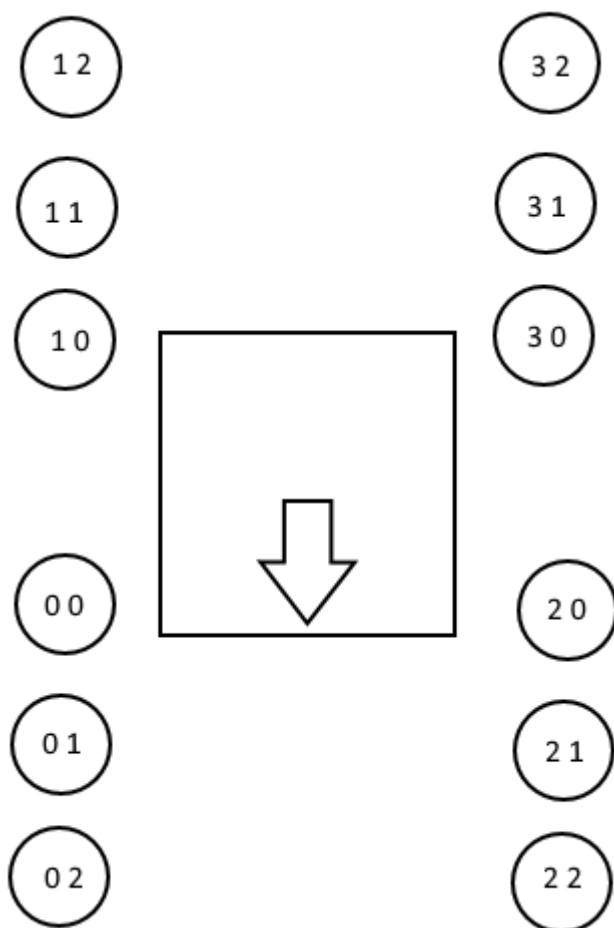


Рис 2.3.2.1. Схематичне зображення робота, а саме розташування позицій дотику кінцівки з площиною. При русі вперед, та назад.

На Рис 2.3.2.1 зображено схематично робот вид зверху. Стрілка вказує напрям руху. Кола вказують на ймовірне находження стопи ноги, кожне коло пронумеровано. Для розуміння, від 0 0 до 3 2. Перша цифра вказує на номер ноги, а друга вказує положення ноги в деякий час $t[8]$.

Робот стартує з позиції 1 1 , 0 1 , 3 0 , 2 0.

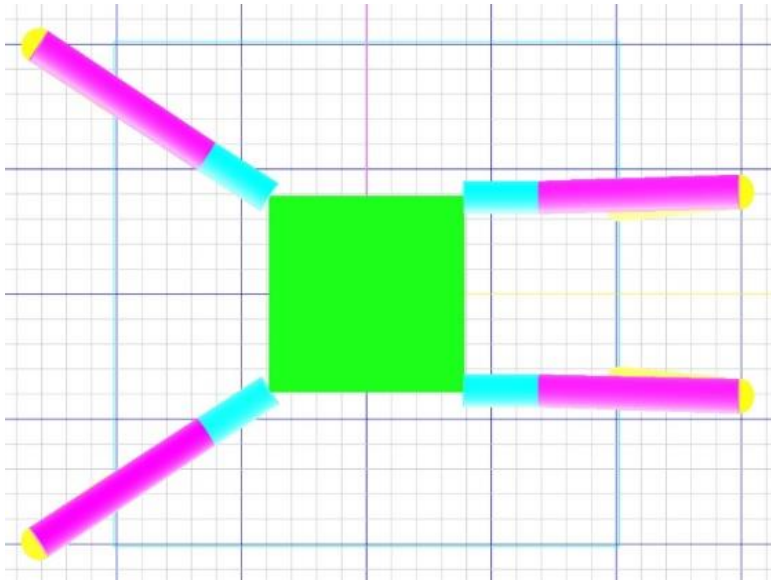


Рис 2.3.2.2. Схематичне зображення стартової позиції 1 1 , 0 1 , 3 0 , 2 0.
Вид зверху.

Кожне піднімання змінює положення кінця ноги робота в площині Z на 20 одиниць. Кожна зміна положення кола, наприклад з 1 1 на 1 0 змінює положення кінця ноги робота в площині Y на 40 одиниць.

1) Нога під номером 2 піднімається та змінює положення з 2 0 на 2 2 і опутається.

```
set_site(2, x_default + x_offset, y_start, z_up);
wait_all_reach();
set_site(2, x_default + x_offset, y_start + 2 * y_step, z_up);
wait_all_reach();
set_site(2, x_default + x_offset, y_start + 2 * y_step, z_default);
wait_all_reach();
```

2) Всі ноги змінюють своє положення на одну сходинку, без піднімання.

0 1 > 0 0 , 1 1 > 1 2 , 3 0 > 3 1 , 2 2 > 2 1.

```
set_site(0, x_default + x_offset, y_start, z_default);
```



```
set_site(1, x_default + x_offset, y_start + 2 * y_step, z_default);  
set_site(2, x_default - x_offset, y_start + y_step, z_default);  
set_site(3, x_default - x_offset, y_start + y_step, z_default);  
wait_all_reach();
```

3) Нога 1 піднімається та змінює положення з 1 2 на 1 0, та опускається

```
set_site(1, x_default + x_offset, y_start + 2 * y_step, z_up);  
  
wait_all_reach();  
  
set_site(1, x_default + x_offset, y_start, z_up);  
  
wait_all_reach();  
  
set_site(1, x_default + x_offset, y_start, z_default);  
  
wait_all_reach();
```

4) Нога під номером 0 піднімається та змінює положення з 0 0 на 0 2 і опускається.

```
set_site(0, x_default + x_offset, y_start, z_up);  
  
wait_all_reach();  
  
set_site(0, x_default + x_offset, y_start + 2 * y_step, z_up);  
  
wait_all_reach();  
  
set_site(0, x_default + x_offset, y_start + 2 * y_step, z_default);  
  
wait_all_reach();
```

5) Всі ноги змінюють своє положення на одну сходинку, без піднімання.

0 0 > 0 1 , 1 0 > 1 1 , 3 1 > 3 2 , 2 1 > 2 0.

```
set_site(0, x_default - x_offset, y_start + y_step, z_default);
```

```
set_site(1, x_default - x_offset, y_start + y_step, z_default);  
set_site(2, x_default + x_offset, y_start, z_default);  
set_site(3, x_default + x_offset, y_start + 2 * y_step, z_default);  
wait_all_reach();
```

6) Нога під номером 3 піднімається та змінює положення з 3 2 на 3 0 і опутається.

```
set_site(3, x_default + x_offset, y_start + 2 * y_step, z_up);  
wait_all_reach();  
set_site(3, x_default + x_offset, y_start, z_up);  
wait_all_reach();  
set_site(3, x_default + x_offset, y_start, z_default);  
wait_all_reach();
```

7) Весь цей цикл повторюється заново.

2.3.3. Алгоритм руху робота в зворотному напрямку.

Для руху в зворотному напрямку алгоритм дуже схожий с рухом в перед.

Початково робот стартує з положення 1 1 , 0 1 , 3 0 , 2 0.

1) Нога під номером 3 піднімається та змінює положення з 3 0 на 3 2 та опутається.

```
set_site(3, x_default + x_offset, y_start, z_up);  
wait_all_reach();  
set_site(3, x_default + x_offset, y_start + 2 * y_step, z_up);  
wait_all_reach();
```

```
set_site(3, x_default + x_offset, y_start + 2 * y_step, z_default);
```

```
wait_all_reach();
```

2) Всі ноги змінюють своє положення на одну сходинку, без піднімання.

0 1 > 0 2 , 1 1 > 1 0 , 3 2 > 3 1 , 2 0 > 2 1.

```
set_site(0, x_default + x_offset, y_start + 2 * y_step, z_default);
```

```
set_site(1, x_default + x_offset, y_start, z_default);
```

```
set_site(2, x_default - x_offset, y_start + y_step, z_default);
```

```
set_site(3, x_default - x_offset, y_start + y_step, z_default);
```

```
wait_all_reach();
```

3) Нога 0 піднімається та змінює положення з 0 2 на 0 0, та опускається.

```
set_site(0, x_default + x_offset, y_start + 2 * y_step, z_up);
```

```
wait_all_reach();
```

```
set_site(0, x_default + x_offset, y_start, z_up);
```

```
wait_all_reach();
```

```
set_site(0, x_default + x_offset, y_start, z_default);
```

```
wait_all_reach();
```

4) Нога 1 піднімається та змінює положення з 1 0 на 1 2, та опускається.

```
set_site(1, x_default + x_offset, y_start, z_up);
```

```
wait_all_reach();
```

```
set_site(1, x_default + x_offset, y_start + 2 * y_step, z_up);
```

```
wait_all_reach();
```

```
set_site(1, x_default + x_offset, y_start + 2 * y_step, z_default);
```

```
wait_all_reach();
```

5) Всі ноги змінюють своє положення на одну сходинку, без піднімання.

0 2 > 0 1 , 1 2 > 1 1 , 3 2 > 3 0 , 2 1 > 2 2.

```
set_site(0, x_default - x_offset, y_start + y_step, z_default);
```

```
set_site(1, x_default - x_offset, y_start + y_step, z_default);
```

```
set_site(2, x_default + x_offset, y_start + 2 * y_step, z_default);
```

```
set_site(3, x_default + x_offset, y_start, z_default);
```

```
wait_all_reach();
```

6) Нога 2 піднімається та змінює положення з 2 2 на 2 0, та опускається.

```
set_site(2, x_default + x_offset, y_start + 2 * y_step, z_up);
```

```
wait_all_reach();
```

```
set_site(2, x_default + x_offset, y_start, z_up);
```

```
wait_all_reach();
```

```
set_site(2, x_default + x_offset, y_start, z_default);
```

```
wait_all_reach();
```

2.3.4. Алгоритм руху робота при повороті на право.

Для повороту на право вводяться нові координати для x та y , а також додається нова точка для становлення стопи. Для розуміння було зображено схему (схема не передає 100% масштабу розташування кінцівок, а лише демонструє принцип роботи)[9].

Хоч і схема подібна до схеми при прямому русі, але вони являються різними.

Прирівняти можна тільки точки. Тобто координати цих точок являються рівнозначними, або різниця між ними незначна. Точки на схемі з прямим рухом позначу як $x \ x^{\wedge}$.

0 0 та 0 0[^], 1 0 та 1 0[^], 2 0 та 2 0[^], 3 0 та 3 0[^].

0 2 та 0 1[^], 1 2 та 1 1[^], 2 2 та 2 1[^], 4 2 та 3 1[^].

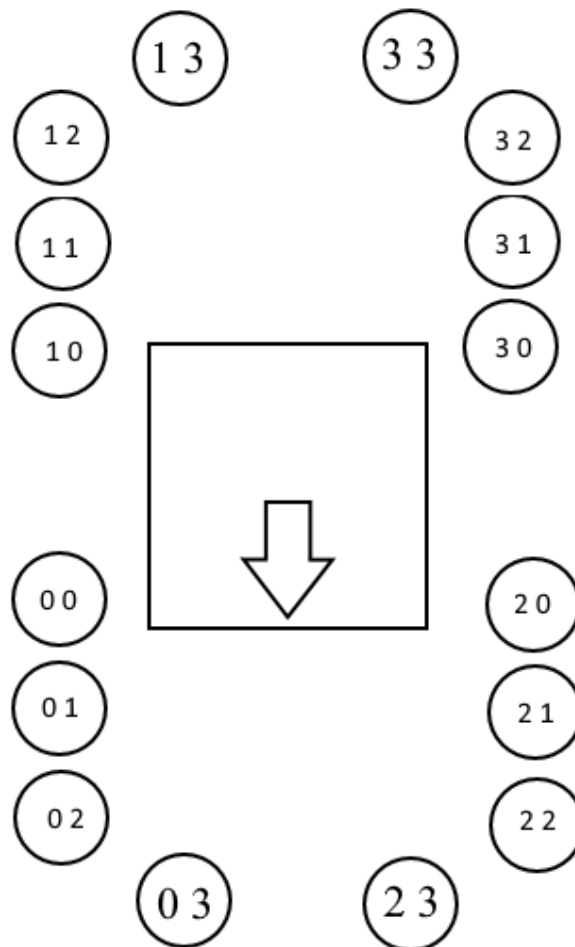


Рис. 2.3.4.1 Схематичне зображення робота, а саме розташування позицій дотику кінцівки з площиною. При русі на ліво, та на право.

Для повороту робот займає за умовчанням положення 0 2, 1 2, 3 0, 2 0,

1) Нога 2 піднімається та залишається в положенні 2 0 до наступного тіка.

`set_site(2, x_default + x_offset, y_start, z_up);`

```
wait_all_reach();
```

2) Всі ноги змінюють положення на деяке значення

```
0 2 > 0 3,
```

2 0 > 2 3 нога номер 2 залишається в піднятому положенні до наступного
тіка

```
1 2 > 1 1,
```

```
3 0 > 3 1
```

```
set_site(0, turn_x0 - x_offset, turn_y0, z_default);
```

```
set_site(1, turn_x1 - x_offset, turn_y1, z_default);
```

```
set_site(2, turn_x0 + x_offset, turn_y0, z_up);
```

```
set_site(3, turn_x1 + x_offset, turn_y1, z_default);
```

```
wait_all_reach();
```

3) Нога 2 опускається та залишається в положенні 2 3.

```
set_site(2, turn_x0 + x_offset, turn_y0, z_default);
```

```
wait_all_reach();
```

4) Нога 0 піднімається та залишається в положенні 0 3.

```
set_site(0, turn_x0 + x_offset, turn_y0, z_up);
```

```
wait_all_reach();
```

5) Всі ноги змінюють положення на деяке значення

0 3 > 0 0 нога номер 0 залишається в піднятому положенні до наступного
тіка

```
2 3 > 2 2,
```

```
1 1 > 1 0,
```

3 1 > 3 2.

```
set_site(0, x_default + x_offset, y_start, z_up);  
  
    set_site(1, x_default + x_offset, y_start, z_default);  
  
    set_site(2, x_default - x_offset, y_start + y_step, z_default); // 2 2  
  
    set_site(3, x_default - x_offset, y_start + y_step, z_default);  
  
wait_all_reach();
```

6) Нога 0 опускається та залишається в положенні 0 0.

```
set_site(0, x_default + x_offset, y_start, z_default);  
  
wait_all_reach();
```

7) Нога 1 піднімається та залишається в положенні 1 0.

```
set_site(1, x_default + x_offset, y_start, z_up);  
  
wait_all_reach();
```

8) Всі ноги змінюють положення на деяке значення

0 0 > 0 1,

2 2 > 2 1,

1 0 > 1 3, нога номер 1 залишається в піднятому положенні до наступного

тіка

3 2 > 3 3.

```
set_site(0, turn_x1 + x_offset, turn_y1, z_default);  
  
    set_site(1, turn_x0 + x_offset, turn_y0, z_up);  
  
    set_site(2, turn_x1 - x_offset, turn_y1, z_default); //2 1  
  
    set_site(3, turn_x0 - x_offset, turn_y0, z_default);  
  
wait_all_reach();
```

9) Нога 1 опускається та залишається в положенні 1 3.
set_site(1, turn_x0 + x_offset, turn_y0, z_default);

wait_all_reach();

10) Нога 3 піднімається та залишається в положенні 3 3.
set_site(3, turn_x0 + x_offset, turn_y0, z_up);

wait_all_reach()

11) Всі ноги змінюють положення на деяке значення
0 1 > 0 2,
2 1 > 2 0,
1 3 > 1 2,
3 3 > 3 0 нога номер 3 залишається в піднятому положенні до наступного

тіка

set_site(0, x_default - x_offset, y_start + y_step, z_default);

set_site(1, x_default - x_offset, y_start + y_step, z_default);

set_site(2, x_default + x_offset, y_start, z_default); // 2 0

set_site(3, x_default + x_offset, y_start, z_up);

wait_all_reach();

12) Нога 3 опускається та залишається в положенні 3 0.
set_site(3, x_default + x_offset, y_start, z_default);

wait_all_reach()

2.3.5. Алгоритм руху робота при повороті на ліво

Алгоритм подібний до руху на право, але всі рухи віддзеркалені.

Для повороту робот займає за умовчанням положення 0 2, 1 2, 3 0, 2 0.

1) Нога 3 піднімається та залишається в положенні 3 0 до наступного тіка.


```
set_site(3, x_default + x_offset, y_start, z_up);
```

```
wait_all_reach();
```

2) Всі ноги змінюють положення на деяке значення

0 2 > 0 1,

2 0 > 2 1

1 2 > 1 3,

3 0 > 3 3 нога номер 3 залишається в піднятому положенні до наступного тіка

```
set_site(0, turn_x1 - x_offset, turn_y1, z_default);
```

```
set_site(1, turn_x0 - x_offset, turn_y0, z_default);
```

```
set_site(2, turn_x1 + x_offset, turn_y1, z_default);
```

```
set_site(3, turn_x0 + x_offset, turn_y0, z_up);
```

```
wait_all_reach();
```

3) Нога 3 опускається та залишається в положенні 3 3.

```
set_site(3, turn_x0 + x_offset, turn_y0, z_default);
```

```
wait_all_reach();
```

4) Нога 1 піднімається та залишається в положенні 1 3.

```
set_site(1, turn_x0 + x_offset, turn_y0, z_up);
```

```
wait_all_reach();
```

5) Всі ноги змінюють положення на деяке значення

0 1 > 0 0,

2 1 > 2 2,

1 3 > 1 0 нога номер 1 залишається в піднятому положенні до наступного тіка

3 0 > 3 2.

```
set_site(0, x_default + x_offset, y_start, z_default);
```

```
    set_site(1, x_default + x_offset, y_start, z_up);
```

```
    set_site(2, x_default - x_offset, y_start + y_step, z_default);
```

```
    set_site(3, x_default - x_offset, y_start + y_step, z_default);
```

```
    wait_all_reach()
```

6) Нога 1 опускається та залишається в положенні 1 0.

```
set_site(1, x_default + x_offset, y_start, z_default);
```

```
    wait_all_reach();
```

7) Нога 0 піднімається та залишається в положенні 0 0.

```
set_site(0, x_default + x_offset, y_start, z_up);
```

```
    wait_all_reach();
```

8) Всі ноги змінюють положення на деяке значення

0 0 > 0 3 нога номер 0 залишається в піднятому положенні до наступного тіка

2 2 > 2 3,

1 0 > 1 1

3 2 > 3 1.

```
set_site(0, turn_x0 + x_offset, turn_y0, z_up);
```

```
    set_site(1, turn_x1 + x_offset, turn_y1, z_default);
```

```
    set_site(2, turn_x0 - x_offset, turn_y0, z_default);
```

```
set_site(3, turn_x1 - x_offset, turn_y1, z_default);
```

```
wait_all_reach();
```

9) Нога 0 опускається та залишається в положенні 0 3.

```
set_site(0, turn_x0 + x_offset, turn_y0, z_default);
```

```
wait_all_reach();
```

10) Нога 2 піднімається та залишається в положенні 2 3.

```
set_site(2, turn_x0 + x_offset, turn_y0, z_up);
```

```
wait_all_reach()
```

11) Всі ноги змінюють положення на деяке значення

0 3 > 0 2,

2 3 > 2 0 нога номер 2 залишається в піднятому положенні до наступного
тіка

1 1 > 1 2,

3 1 > 3 0

```
set_site(0, x_default - x_offset, y_start + y_step, z_default);
```

```
set_site(1, x_default - x_offset, y_start + y_step, z_default);
```

```
set_site(2, x_default + x_offset, y_start, z_up);
```

```
set_site(3, x_default + x_offset, y_start, z_default);
```

```
wait_all_reach();
```

12) Нога 2 опускається та залишається в положенні 2 0.

```
set_site(2, x_default + x_offset, y_start, z_default);
```

```
wait_all_reach();
```

2.3.6. Координати в кути.

Далі знаючи координати x , y , z стопи кінцівки робота знаходяться кути для 3 сервоприводів які знаходяться в кінцівці (α , β , γ)[10]. Кожні кути різні для окремих ніг.

Знаючи параметр set_site в який входять значення координат кінцівки для кожної окремої ноги(в залежності від параметра номер ноги 0-3), вираховуємо значення $length_x$, $length_y$, $length_z$.

Потім за допомогою цих параметрів довжини та координат знаходимо кути за такими формулами.

$$w = (x \geq 0 ? 1 : -1) * (\text{sqrt}(\text{pow}(x, 2) + \text{pow}(y, 2)));$$

$$v = w - length_c;$$

$$\alpha = \text{atan2}(z, v) + \text{acos}((\text{pow}(length_a, 2) - \text{pow}(length_b, 2) + \text{pow}(v, 2) + \text{pow}(z, 2)) / 2 / length_a / \text{sqrt}(\text{pow}(v, 2) + \text{pow}(z, 2)));$$

$$\beta = \text{acos}((\text{pow}(length_a, 2) + \text{pow}(length_b, 2) - \text{pow}(v, 2) - \text{pow}(z, 2)) / 2 / length_a / length_b);$$

$$\gamma = (w \geq 0) ? \text{atan2}(y, x) : \text{atan2}(-y, -x);$$

$$\alpha = \alpha / \pi * 180;$$

$$\beta = \beta / \pi * 180;$$

$$\gamma = \gamma / \pi * 180;$$

Такі значення кутів знаходяться для кожної ноги.

РОЗДІЛ 3. РУХОМИЙ ОБ'ЄКТ НА СЕРВОПРИВОДАХ.

3.1 Компоненти

За основу було обрано Arduino nano як зручна платформа яка має достатню гнучкість для реалізації пристрою.

3.1.1. Коротко про Arduino nano.

Arduino Nano - це невелика, повна і зручна плата на базі ATmega328 (Arduino Nano 3.x). Він має більш-менш однакову функціональність Arduino Duemilanove, але в іншому пакеті. У ньому бракує лише гнізда постійного струму, і він працює із кабелем Mini-B USB замість стандартного.

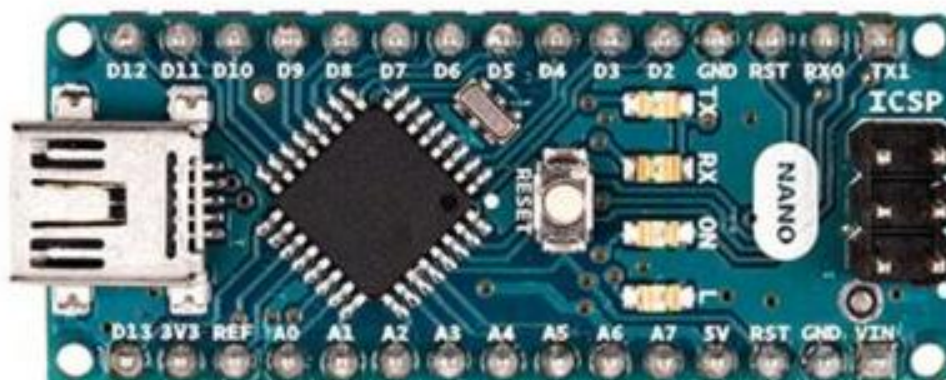


Рис.3.1.1. фотографія Arduino nano.

3.1.1.1 Основні характеристики Arduino Nano

Він має 22 вхідні / вихідні піни.

14 з них є цифровими.

Arduino Nano має 8 аналогових піни.

Він має 6 ШІМ-контактів серед цифрових контактів.

Він має кристальний генератор 16 МГц.

Робоча напруга коливається від 5 до 12 В.

Він також підтримує різні способи спілкування, а саме:

Послідовний протокол.

Протокол I2C.

Протокол SPI.

Він також має міні-USB-пін, який використовується для завантаження коду.

На ньому також є кнопка Reset.

3.1.1.2. Пам'ять в Arduino Nano

Флеш-пам'ять Arduino Nano становить 32 Кб.

На ньому попередньо встановлений завантажувач, який займає флеш-пам'ять 2 кб.

Пам'ять SRAM цієї плати мікроконтролера становить 8 кб.

Він має пам'ять EEPROM 1 кб.

Для бездротового управління використовується модуль Bluetooth HC-05.

За допомогою цього модуля можливо підключитись по блютуз каналу до робота і керувати ним.

3.1.2.Коротко про Bluetooth HC-05.

Контролер Arduino не підтримують бездротовий зв'язок, а часто це просто необхідно. Наприклад, віддалене управління роботом на Arduino, відправка даних з метеостанції на арудіно в Інтернет або на домашній сервер, зв'язок кількох пристроїв між собою. Тут на допомогу розробникам пристроїв безліч зовнішніх модулів для організації різних технологій бездротового зв'язку: модулі WiFi, GSM / GPRS, IR, Bluetooth, радіомодулі для роботи в різних частотних діапазонах.

Технологія Bluetooth використовується для передачі даних між двома пристроями, які знаходяться в безпосередній близькості один з одним, причому необов'язкова пряма видимість. Технологія Bluetooth забезпечує хорошу стійкість до ширококутних перешкод, що дозволяє безлічі пристроїв, що знаходяться в одному місці, одночасно спілкуватися між собою, не заважаючи один одному. Дуже широко дана технологія використовується в телефонах, планшетах, ноутбуках.

Одне з кращих рішень для організації двостороннього зв'язку по Bluetooth вашого Arduino-пристрої з планшетом, ноутбуком або іншим Bluetooth-пристроєм - Bluetooth-модуль HC-05, який може працювати як master (здійснювати пошук Bluetooth-пристроїв і ініціювати установку зв'язку), так і slave (ведене пристрій).

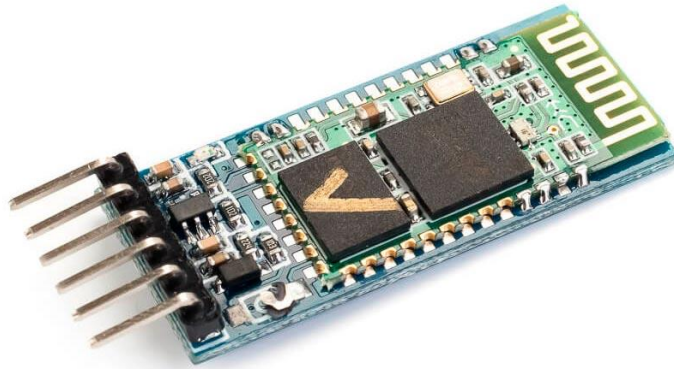


Рис.3.1.2.1 Bluetooth модуль HC-05.

3.1.3 Сервоприводи Tower Pro MG90S

Рух 4 ніг відбувається завдяки 12 сервоприводам Tower Pro MG90S micro servo. Особливістю цих сервоприводів закладається в металевому редукторі, що надає вищу конструкційну твердість на відмінну від пластикових.

Сервоприводи мають розміри 22.8мм x 12.2мм x 28.5мм.

Пусковий момент 1.8 - 2.2 кг / см(4.8 В).

Робоча напруга 4.8-6 вольт.

Швидкість без навантаження 60 градусів/0.1 секунду.(4.8 В).



Рис.3.1.3.1 Сервопривод Tower Pro MG90S

3.1.4. Система живлення

Живлення пристрою відбувається завдяки 2 18650HG2 3,6 В акумуляторам під'єднаними до захисної, балансуєчої плати 2S 8A Li-Ion 7,4 v 8,4 V 18650 BMS PCM 15A піковим током.

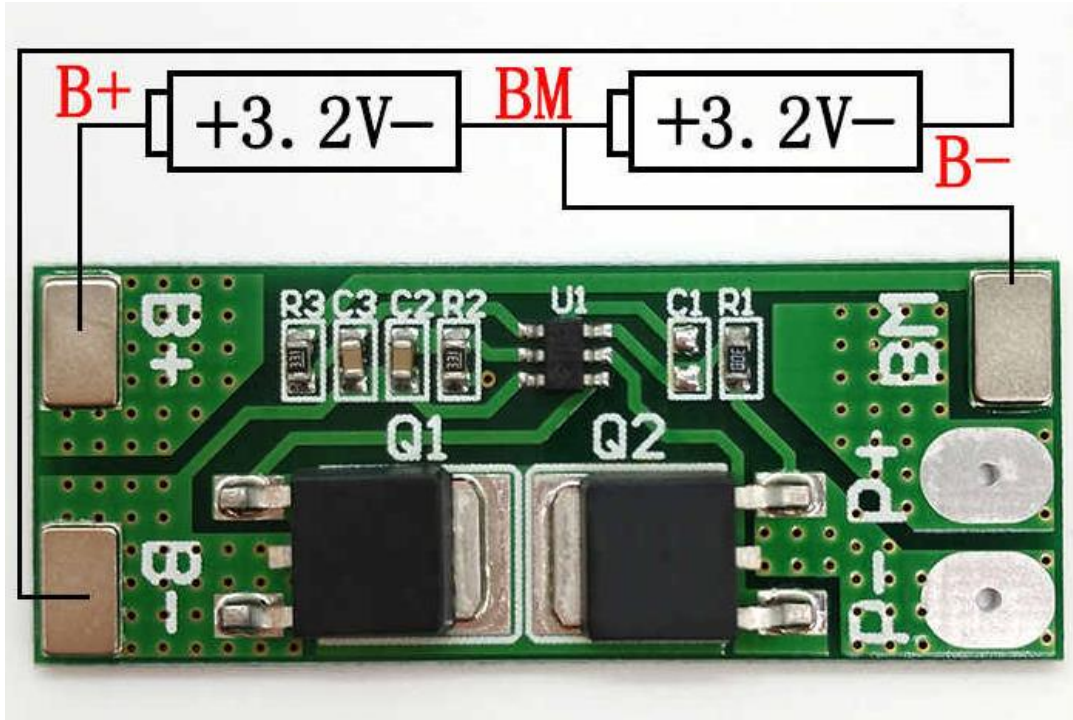


Рис.3.1.4.1 схема під'єднання 2 18650HG2 3,6 В до 2S 8A Li-Ion 7,4 v 8,4 V 18650 BMS.

Далі живлення іде на понижуючу плату LM2596s DC-DC яка з 8 вольт дає 5 (3 ампера макс), які уже ідуть на головну плату робота.



Рис.3.1.4.2. Понижуючий стабілізатор LM2596s DC-DC.

3.1.5 Головна плата.

Взагалі-то можна взяти уже готову плату розширення I/O для ардуіно, але зазвичай живляча частина в таких платах не видає достатньо для управління 12 сервоприводами, тому було обрано варіант виготовити плату самостійно. Плата має таку схему.

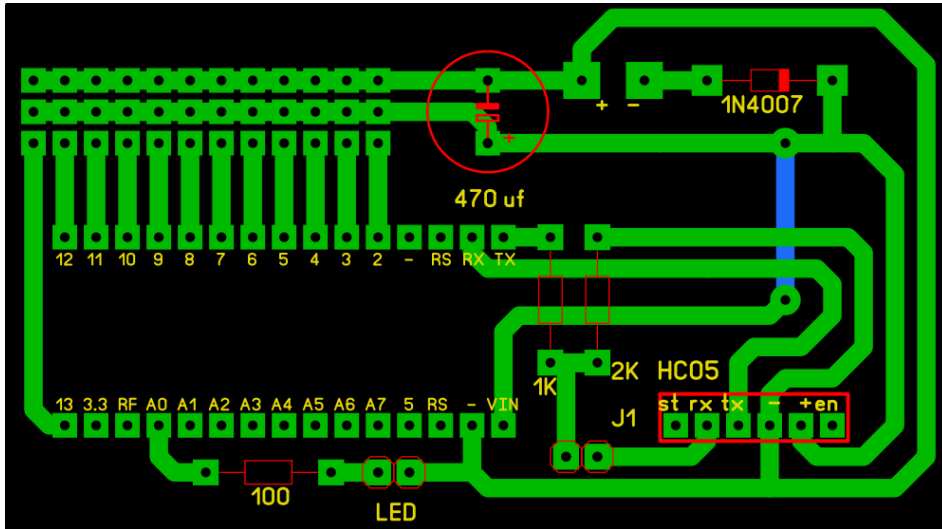


Рис.3.1.5.1 Схема головної плати.

Така плата вміщує всі входи і виходи які необхідні для проекту.

З'єднання компонентів надзвичайно легке лише необхідно вставити плату Arduino Nano до входів які назначені для неї. Зазвичай на платах Arduino нанесені помітки, тому просто з'єднуємо їх з помітками на платі.

Далі під'єднаємо блютуз модуль по аналогічній схемі.

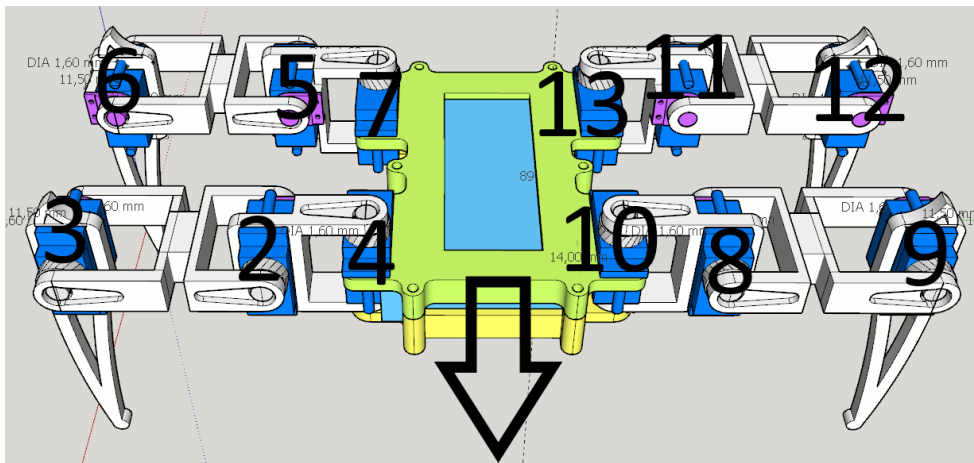


Рис.3.1.5.2. Пронумероване розташування сервоприводів в роботі. 3D візуалізація.

Згідно схеми Рис.3.1.5.1 з'єднуємо сервоприводи та контакти на платі. На малюнку пронумеровані сервоприводи та їх розташування.

3.1.6. Корпус робота

Корпус виготовлено за допомогою 3D принтера основним матеріалом являється АБС пластик. Таке рішення дає можливість швидко відтворювати конструкцію в різноманітних умовах. Також конструктивно АБС пластик оптимально підходить, він забезпечує оптимальну міцність та надійність конструкції. Змодельоване зображення робота можна побачити на Рис.3.1.5.2.

3.2. Висновки до дипломної роботи

Якщо прикріпити камеру, то робот павук потенційно може бути корисним в рятувальній та розвідувальній сфері. Завдяки своїй формі, та мімікруванню рухам павука він може долати нерівності рельєфу, та пролазити в наприклад руїни будинку для знаходження людей або розвідування місцевості.

Конструкція робота надзвичайно легка до 1 кг. Це досягаються використання технологій 3D друку. Весь корпус було надруковано, це також робить його дешевим при виробництві. Для його виготовлення потрібно декілька електронних компонентів та 3D принтер.

Вся обробка програмного коду відбувається завдяки Arduino Nano, вона була обрано за свою компактність. Також було використано декілька бібліотек для рішення задач.

Такі як Servo для забезпечення коректної роботи сервоприводів для плати Arduino. Рух робота відбувається завдяки зміні точки контакту з поверхнею і знаючи цю точку в трьох (X Y Z) мірному просто вираховуються кути для сервоприводів.

Для безпроводного управління була використана бібліотека SerialCommand. Вона необхідна для послідовних команд через послідовний порт. Завдяки спеціального програмного забезпечення являється можливим керувати робот завдяки смартфона по Bluetooth.

РОЗДІЛ 4. ОХОРОНА ПРАЦІ

Робоче місце – місце постійного або тимчасового перебування працівника в процесі трудової діяльності. Робоче місце працівників, які займаються зборкою та виголошенням деталей для роботизованого електронного пристрою. Приміщення достатніх розмірів для розміщення стола для зборки на налаштування розмірами 1 метр на 2 метри, та розташуванням 3D принтера. Таке приміщення має вентиляцію, а також хороший рівень освітлення, для контролю якості при зборці.

4.1. Аналіз шкідливих та небезпечних виробничих факторів при виробництві електронного роботизованого пристрою.

При роботі над роботизованим електронним пристроєм працівник піддається впливу наступним небезпечним та шкідливим факторам:

Підвищена температура поверхні устаткування, а саме паяльна станція. Температури в діапазоні 245 до 300 градусів Цельсія. Такі температури можуть бути потенційно небезпечні при контакті з шкірою, вони можуть викликати тяжкі опіки;

Підвищене значення напруги в електричному колі чи появі їх на струмоведучому обладнанні, внаслідок цього, збільшена імовірність проходження струму через тіло людини. Така ситуація може виникнути при необережному відношенні з паяльною станцією. Розжарене жало паяльника може розплавити ізоляцію на проводі живлення, та викликати коротке замикання. Що також може викликати пожежу.

Наявність токсичних та канцерогенних факторів при роботі з паяльною станцією. Припій — метал, сплав або суміш оксидів, що застосовується для з'єднання металевих, мінерало-керамічних та інших деталей, для лудіння посуду тощо. В склад припою можуть входити такі хімічні елементи як: олово,

свинець, цинк, мідь, нікель, фосфор, срібло, кадмій та ін. Флюс — речовина, що додається до розплавленого металу для видалення його окисів і сторонніх шлаків, чи під час паяння для запобігання окиснення поверхні металу. Кожен з цих елементів в залежності від складу конкретного примою може випаровуючись осідати в легенях, та негативно впливати на організм людини.

Канцерогенних фактором являється випаровування АБС-пластику при роботі 3D принтера. АБС-пластик (Акрилонітрилбутадієнстирол) — термопластичний листовий конструкційний матеріал. Температура склування становить близько 105 градусів Цельсія. АБС-пластик є, як правило, безпечним матеріалом. Реальна небезпека, яку може представляти АБС-пластик для людини, може виникнути в декількох випадках: При нагріванні в процесі виробництва (для формування або екструзії, а також в 3D-друку) утворюються пари акрилонітрилу, які є отруйними. Існує побоювання, що акрилонітрил може бути канцерогеном для людини. Бутадієн є відомим канцерогеном. Стирол теж має канцерогенні властивості. 3D друк використовується для створення корпусу для роботизованого електронного пристрою.

Психофізіологічні напруження. Монотонність праці, перенапруженість зорових аналізаторів, розумова напруженість, незручність і статичність пози. Постійна концентрація на дрібних об'єктах під час зборки та пайки виробу. Робота вимагає хорошої зорової концентрації при контролі якості. Може знижується темп та якість праці, з'являється рання втома, знижується реакція та ослабляється увага, з часом негативно впливати на якість зору, при довгих незмінних навантаженнях. Негативно на стан здоров'я може викликати довге сидіння на робочому місці в одній позі, яка не змінюється протягом часу.

Несприятлива освітленість. Недостатнє штучне і природне освітлення. Недостатня освітленість приміщення цеху, де відбувається виробництво пристрою. Може викликати проблеми з якістю виконання робіт, а також я станом здоров'я робітника. Такі як погану концентрацію, втоми, головні болі,

зниження реакції, також знижуються темп роботи. Збільшуються ризики порушення техніки безпеки з паяльною станцією.

4.2. Організаційні та конструктивно-технологічні заходи для зниження впливу шкідливих виробничих факторів.

4.2.1 Для забезпечення безпеки працівника, роботодавець та робітник повинні слідувати вимогам охорони праці під час паяння електрифікованим інструментом.

Вимоги охорони праці під час паяння електрифікованим інструментом:

1. Паяння електрифікованим інструментом (далі - паяльник) необхідно виконувати відповідно до вимог Інструкції з охорони праці під час виконання монтажних робіт інструментами і пристроями, затвердженої наказом Міністерства праці та соціальної політики України від 05 червня 2001 року № 254, зареєстрованої у Міністерстві юстиції України 20 липня 2001 року за № 616/5807 (НПАОП 0.00-5.24-01), ДСТУ 7237:2011.

2. Електрифікований інструмент для паяння деталей повинен відповідати вимогам ДСТУ ІЕС 60745-1:2010 «Інструмент ручний електромеханічний. Безпека. Частина 1. Загальні вимоги».

3. Паяння великогабаритних виробів необхідно виконувати паяльником з вбудованим відсмоктувачем.

4. Виконувати паяння в замкнутих об'ємах необхідно із застосуванням паяльника напругою не вище 12 В.

5. Паяльник на робочих місцях повинен встановлюватися на вогнестійкій підставці для запобігання його падінню.

6. Паяльник, що знаходиться в робочому стані, повинен постійно перебувати в зоні дії місцевої витяжної вентиляції.

7. У проміжках часу між паяльними операціями нагрівання жала паяльників повинно бути знижене до 150-180 °С, а при тимчасовому припиненні робіт - відключене. Робочі місця повинні бути обладнані регуляторами нагріву паяльників.

8. Паяння малогабаритних виробів у вигляді штепсельних роз'ємів, наконечників, клем та аналогічних виробів необхідно виконувати із закріпленням їх у спеціальних пристроях (затискачі, струбцини).

9. Під час паяння інтегральних мікросхем необхідно використовувати біноклярні стереоскопічні мікроскопи з телевізійними екранами.

Слідуючи всім цим правилам буде досягнуто найоптимальніші умови для працівника

4.2.2 Для уникнення короткого замкнення при роботі з паяльником необхідно слідувати таким нормам:

Ручні електричні машини повинні відповідати вимогам Правил устаткування електроустановок, ДНАОП 0.00-1.21-98 Правила безпечної експлуатації електроустановок споживачів, затвердженого наказом Держнаглядодохоронпраці від 09.01.98 N 4 (з0093-98), зареєстрованого у Міністерстві юстиції 10.02.98 за N 93/2533, та ГОСТ 12.2.013.0-91.

Вимоги безпеки перед початком роботи

1. Необхідно пройти інструктаж на робочому місці.
2. Отримати для виконання робіт спецодяг, засоби індивідуального захисту, інструмент, пристосування і перевірити їх комплектність та цілість.
3. Підготувати робоче місце: прибрати зайві речі, перевірити достатність освітлення робочого місця;

4.2.3 Технологічний процес пайки радіоелементів супроводжуються забрудненням повітряного середовища аерозолями припоїв, флюсів, парами рідин, що застосовуються для змивання флюсу або для розчинення лаків, які застосовуються для покриття друкованих плат тощо.

Також на працюючих може негативно впливати і інфрачервоне (ІЧ) випромінювання від нагрітої поверхні електропаяльника. Клас небезпеки, гранично допустима концентрація в повітрі робочої зони (ГДКрз), характер токсичності та можлива біологічна дія основних компонентів, що входять до складу олов'яно-свинцевих припоїв, а також флюсів і миючих засобів знаходяться в ДСН 3.3.6.042-99. Наявність в повітрі робочої зони аерозолі свинцю, який відноситься до шкідливих речовин 1 класу небезпеки, потребує обов'язкового застосування ефективної системи вентиляції виробничого приміщення.

При проведенні технологічного процесу пайки радіоелементів олов'яносвинцевими припоями концентрація аерозолі свинцю в повітрі робочої зони, як правило, перевищує гранично припустиму концентрацію (ГДКрз), що вимагає застосування місцевої витяжної вентиляції на робочих місцях.

Згідно з існуючою нормативною базою, експлуатація ділянок пайки, не обладнаних місцевою витяжною вентиляцією, забороняється. Вентиляційні установки повинні включатися до початку робіт і виключатися після їхнього закінчення. Місцеві відсмоктувачі від зон пайки повинні обслуговуватися автономною вентиляційною установкою.

Розведення вентиляційної мережі і конструкція місцевих відсосів повинні забезпечувати можливість регулярної очистки повітропроводів. Електропаяльник у робочому стані повинен знаходитися в зоні дії витяжної вентиляції.

У зоні ручної пайки швидкість спрямованого потоку, що створюється місцевими відсмоктувачами, повинна не менш ніж на 0,2 м/с перевищувати рухливість повітря в зоні пайки і бути не меншою ніж 0,5 м/с.

При виконанні паяльних робіт з використанням олов'яно-свинцевих припоїв на робочих місцях необхідно користуватися „Інструкцією з охорони праці при роботах з олов'яно-свинцевими малосурм'янистими припоями” ПІ 1.4.32-423- 2005.

4.2.4 Для уникнення канцерогенних факторів при роботі з 3D принтером необхідно слідувати таким нормам: ДСТУ Б А.3.2-12:2009 ССБП Системи вентиляційні.

4.2.4.1 ЗАГАЛЬНІ ПОЛОЖЕННЯ

4.2.4.1.1 Вентиляційні системи для виробничих приміщень у комплексі з технологічним устаткуванням, що виділяє шкідливі речовини згідно з ГОСТ 12.0.003, надлишкове тепло або вологу, повинні забезпечувати мікрокліматичні умови та чистоту повітря, що відповідають вимогам ГОСТ 12.1.005, ДСН 3.3.6.042 на постійному і тимчасовому робочих місцях у робочій зоні виробничих приміщень.

У зоні адміністративно-побутових приміщень промислових підприємств, що обслуговуються, а також у приміщеннях громадських будинків повинні бути забезпечені мікрокліматичні умови відповідно до вимог ДСН 3.3.6.042.

4.2.4.1.2 Технічні рішення, прийняті при проектуванні вентиляційних систем, а також вимоги, які висуваються до них при спорудженні та експлуатації, повинні відповідати ДБН А.3.2-2 2009, ДБН В.2.5-67:2013, СНиП 2.09.02. 85, ДБН В.2.2-28-2010.

4.2.4.1.3 Випробування вентиляційних систем має виконуватися відповідно до вимог нормативної документації та організації-виробника.

4.2.4.1.3,4 Розташування вентиляційних систем має забезпечувати безпечний і зручний монтаж, експлуатацію та ремонт технологічного устаткування. При розміщенні вентиляційних систем слід дотримуватись норм освітлення приміщень, робочих місць і проходів згідно з ДСТУ Б А.3.2-15:2011, ДБН В.2.5:28-2018.

4.2.4.1.5 Для монтажу, ремонту та обслуговування елементів вентиляційних систем, а також для переходу через них повинні передбачатися стаціонарні площадки, проходи, сходи і містки.

4.2.4.1.7 Розміщення припливних і витяжних вентиляційних агрегатів у приміщеннях для вентиляційного устаткування повинне виконуватися відповідно до ДБН В.2.5-67:2013.

4.2.4.1.8 На випадок виникнення пожежі слід передбачити спеціальні пристрої, що забезпечують відключення вентиляційних систем, а також включення, при необхідності, систем аварійної протидимної вентиляції, відповідно до вимог ДБН В.2.5-67:2013.

4.2.4.1.9 Розміщення і улаштування електроустаткування вентиляційних систем, а також контрольно-вимірювальної апаратури, улаштування струмоведучих частин і заземлень має відповідати вимогам ДСТУ Б В.2.5-82:2016, НПАОП 0.00-1.29, НПАОП 40.1-1.01, НПАОП 40.1-1.21, НПАОП 40.1-1.32, а також діючих стандартів на вибухозахищене і рудничне устаткування.

4.2.4.1.10 Вентиляційні системи, що обслуговують приміщення категорій А, Б згідно із НАПБ Б.03.002 та системи місцевих відсмоктувачів, у яких можливе утворення статичної електрики, повинні бути у вибухобезпечному виконанні і захищені від дії статичної електрики згідно з ГОСТ 12.1.018, ГОСТ 12.4.124, ДБН В.2.5-27, НПАОП 0.00-1.29.

4.2.4.1.11 Виконання вентиляційного устаткування систем, що обслуговує приміщення категорій А, Б згідно із НАПБ Б.03.002 і місцевих

відсмоктувачів вибухопожежобезпечних та пожежобезпечних сумішей, мають відповідати вимогам ДБН В.2.5-67:2013.

4.2.4.1.12 Робітники, які працюють з вентиляційними системами, повинні бути забезпечені захисними засобами згідно з ДСТУ 7239:2011 ССБП Засоби індивідуального захисту. Загальні вимоги та класифікація.

4.2.5 Для уникнення проблем психофізіологічного напруження. монотонності праці, перенапруженості зорових аналізаторів, розумової напруженості, незручність і статичність пози. Необхідно виконувати ряд рекомендацій:

Потрібно час від часу робити відпочинок, наприклад подивитись у вікно на хвилину. Для відпочинку очей можна виконати такі вправи:

Швидко покліпайте очима 20 разів, при цьому заплющуйте і розплющуйте повіки з зусиллям.

З заплющеними очима починаємо водити очними яблуками вліво - вправо (4 рази), тоді вгору - вниз (4 рази).

З заплющеними очима малюємо очними яблуками кола – спершу за годинниковою стрілкою, потім проти (по 5 разів).

Подивіться за вікно і знайдіть вдалині якийсь об'єкт (наприклад, пташку на дереві, або антену на даху сусіднього будинку). Сфокусовуйте на ньому погляд протягом 5 секунд. Тепер переведіть погляд на кінчик власного носа й утримуйте на ньому погляд також протягом 5 секунд.

Міцно замружте очі, а потім різко й широко відкрийте (повторіть 5 разів).

Ви можете виконувати ці вправи в будь-якій зручній для вас послідовності і так часто, як вам буде зручно – але не менше ніж двічі на день. Це не забере у вас багато часу, але може допомогти зберегти зір на довгі роки.

Перебування у стані сидючи протягом тривалого часу не лише погано впливає на ваш опорно-руховий апарат, а й погіршує кровопостачання мозку – це призводить до зниження когнітивних здібностей, а значить і до зменшення ефективності вашої роботи. Тож лікарі радять робити перерви кожних 20 хвилин. Саме цей проміжок часу дозволяє зробити сидіння безпечним. Намагайтеся після 20 хвилин сидіння встати та пройтися офісом, підійти до колеги, піти помити руки, набрати стакан води – будь-що. Якщо у вас дуже багато термінової роботи і навіть прогулянка до вбиральні може їй нашкодити – спробуйте працювати стоячи.

4.2.6 Для уникнення несприятливого освітлення необхідно слідувати таким нормам:

Основні вимоги до виробничого освітлення:

- Створювати на робочій поверхні освітленість, що відповідає характеру зорової роботи і не є нижчою за встановлені норми;
- Забезпечити достатню рівномірність та постійність рівня освітленості у виробничих приміщеннях, щоб уникнути частої переадаптації органів зору;
- Не створювати засліплювальної дії як від самих джерел освітлення, так і від інших предметів, що знаходяться в полі зору;
- Не створювати на робочій поверхні різних та глибоких тіней (особливо рухомих);
- Повинен бути достатній для розрізнення деталей контраст поверхонь, що освітлюються;
- Не створювати небезпечних та шкідливих виробничих чинників (шум, теплові випромінювання, небезпека ураження струмом, пожежо- та вибухонебезпека світильників): вони повинні бути надійним і простими в експлуатації.

Нормативні величини освітленості робочих місць для різних видів робіт та відповідних зорових навантажень визначаються ДБН Б.2.5:28-2018

«Природне і штучне освітлення».

4.2.7. Методика розрахунку шарнірно-телескопічних відсмоктувачів повітря із зон пайки.

Шарнірно-телескопічні відсмоктувачі можуть бути як з прямокутним, так і з круглим отвором. Шарнірно-телескопічні відсмоктувачі прямокутної форми з гострими кромками встановлюються у вертикальній площині столу, що обумовлено тим, що друкована плата має прямокутну форму і, як правило, в процесі пайки радіоелементів (РЕ) розміщується у горизонтальній площині робочої поверхні столу (рис.4.1).

Кількість повітря, що видаляється відсмоктувачем з прямокутним отвором із зони пайки РЕ, визначається за формулою:

$$L_{ВП} = (S + 7.7 \cdot E^{0.63} \cdot X^{1.4}) \cdot v_x \quad 1)$$

де: $L_{ВП}$ - об'єм повітря, що видаляється прямокутним відсмоктувачем,

S ; $= B \cdot E$ - площа усмоктувального отвору, м² ;

E – розмір більшої сторони усмоктувального отвору, м. ($E = 0,14 \dots 0,28$ м);

X - відстань від площини усмоктувального отвору до зони пайки ($X = 0,1 \dots 0,3$ м);

v_x - осьова швидкість повітря в зоні пайки, ($v_x > 0,5$ м/с).

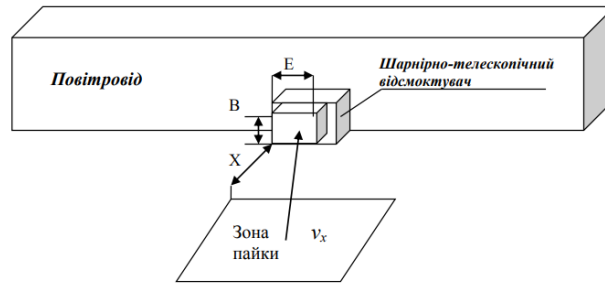


Рис.4.1. Шарнірно-телескопічний відсмоктувач з прямокутним отвором

Примітка: Розмір меншої сторони прямокутного усмоктувального отвору (В) визначається за умови отримання оптимального співвідношення між сторонами усмоктувальної щілини В и Е:

$$\frac{B}{E} = 0,24 \cdot \left(\frac{X}{E} \right)^{0,36}$$

З урахуванням даного співвідношення В визначається за наступною формулою:

$$B = E \cdot 0,24 \cdot \left(\frac{X}{E} \right)^{0,33} \quad 2)$$

Шарнірно-телескопічні відсмоктувачі з круглими отворами (рис.4.2.) також встановлюються у вертикальній площині робочого столу.

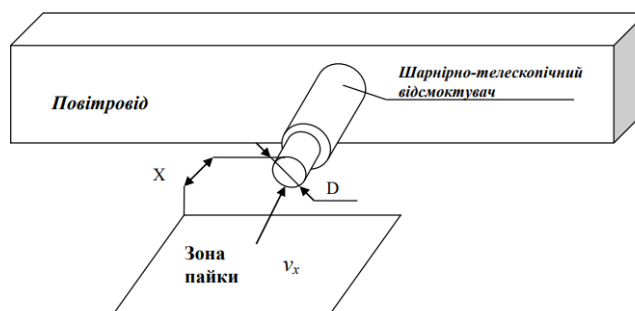


Рис.4.2. Шарнірно-телескопічний відсмоктувач з круглим отвором

Кількість повітря, що відсмоктується визначається з формулою:

$$L_{BK} = \frac{\pi}{4} \cdot (D^2 + 9.1 \cdot D^{0.6} \cdot X^{1.4}) \cdot v_x \quad 3)$$

де: L_{BK} - об'єм повітря, що видаляється круглим відсмоктувачем, м³;

D – діаметр круглого відсмоктувального отвору, м., (0,12...0,24 м.);

X – відстань від площини відсмоктувального отвору до зони пайки, м;

v_x - осьова швидкість повітря в зоні пайки, ($v_x > 0,5$ м/с).

У разі застосування на робочих місцях шарнірно-телескопічних відсмоктувачів повітря із зон пайки, концентрація в повітрі робочої зони аерозолю свинцю при проведенні технологічного процесу пайки РЕ з використанням олов'яно-свинцевих припоїв визначається за наступною формулою:

$$C_{PЗ.} = 0,6 \cdot \frac{y \cdot n \cdot t \cdot N}{V + L_B \cdot N \cdot t} \quad 4)$$

де: L_B - об'єм повітря, що видаляється місцевим відсмоктувачем на кожному робочому місці, м³;

y - питома утворення аерозолю свинцю, мг/100 пайок;

n - кількість пайок у хвилину, шт.;

t - тривалість робочої зміни, год.;

N - кількість робочих місць, на яких ведеться пайка, шт.;

V - загальний об'єм повітря у виробничому приміщенні, м³.

*Примітка: Питоме утворення аерозолю свинцю (y ,) при лудженні і пайці олов'яно-свинцевими припоями за допомогою електропаяльника потужністю 20-60 Вт, складає 0,02- 0,04 мг на 100 пайок.

Розрахункова частина.

Умови:

$$E := 0.14 \quad X := 0.1 \quad \nu_x := 0.6$$

Розмір меншої сторони прямокутного усмоктувального отвору (В)

$$B := E \cdot 0.24 \left(\frac{X}{E} \right)^{0.33} = 0.03$$

визначається (м):

$$\text{Площа усмоктувального отвору (м}^2\text{): } S := B \cdot E = 0.004$$

ЛВП - об'єм повітря, що видаляється прямокутним відсмоктувачем (м³):

$$L_{ВП} := (S + 7.7 \cdot E^{0.63} \cdot X^{1.4}) \cdot \nu_x = 0.056$$

Висновок

Проводячи розрахунок було розраховано яку площу повітря буде видаляти місцевим відсмоктувачем з заданими в умові розмірами. Це значення дорівнює 0.056 м³. Таким чином при зміні початкових параметрів можна регулювати потрібний розмір місцевого відсмоктувача.

4.3 Пожежо- та вибухонебезпека.

Виробниче переміщення можна віднести до Г згідно НАПБ Б.03.002-2007 Норми визначення категорій приміщень, будинків та зовнішніх установок за вибухопожежною та пожежною небезпекою. (помірно пожежонебезпечна). Негорючі речовини і/або матеріали у гарячому, розпеченому і/або розплавленому стані, процес обробки яких супроводжується виділенням променистого тепла, утворенням іскор і/або полум'я; горючі гази, рідини і/або тверді речовини, що спалюються або утилізуються як паливо. Оскільки в виробничому переміщенні відбуваються пайка електронних компонентів.

Згідно до ПРАВИЛА експлуатації та типові норми належності вогнегасників. ДСТУ 3855-99 Пожежна безпека. Визначення пожежної небезпеки матеріалів та конструкцій.

Для приміщення які підпадають під такі параметри(Додаток 4 до Правил експлуатації та типових норм належності вогнегасників):

Виробничі, сільськогосподарські, складські та лабораторні будинки і приміщення, адміністративні та побутові будинки і приміщення та споруди промислових підприємств, громадські будинки та споруди, гаражі та автомайстерні.

Рекомендовано використовувати такі вогнегасники як:

Порошковий (ВП-5,ВП-6,ВП-9,ВП-12). В кількості (Додаток 1 до Правил експлуатації та типових норм належності вогнегасників) для приміщення прощею до 50 м² , та клас приміщення категорії Г слід використовувати 1 вогнегасник вагою від 8 до 12 кг.

Для сповіщення людей о пожежі встановлюються комплекс технічних засобів, за допомогою якого забезпечується відповідно до розроблених планів евакуації передавання сигналів оповіщення одночасно на всьому захищуваному об'єкті, а у разі необхідності - послідовно або вибірково до окремих його частин (поверхи, секції тощо).

Шлях евакуації з виробничого приміщення.

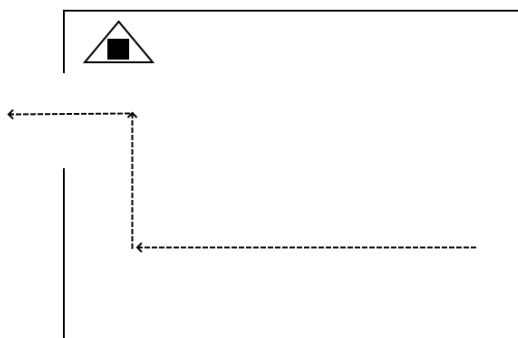


Рис 4.3 Стрілкою вказано напрям евакуації з приміщення. Чорний квадрат в трикутнику – вогнегасник.

4.4 Інструкція з охорони праці при виконанні паяльних робіт

Згідно до Положення про розробку інструкцій з охорони праці (ДНАОП 0.00-4.15-98) розробимо типову інструкцію.

ЗАГАЛЬНІ ВИМОГИ

1) До роботи з приладом допускається інженерно - технічний склад, що вивчив об'єкт, інструкцію з технічної експлуатації, діючу інструкцію, а також склав залік з технічної безпеки та пожежної безпеки;

2) Робоче місце або ділянка має бути устаткована засобами захисту від пожежі – вогнегасниками порошкового або іншого типу;

ВИМОГИ БЕЗПЕКИ ПЕРЕД ПОЧАТКОМ РОБОТИ

Перед початком роботи слід пересвідчитись, що:

- прилад правильно підключений і має заземлення;
- усі з'єднувальні кабелі та місця рознімання справні.

ВИМОГИ БЕЗПЕКИ ПІД ЧАС РОБОТИ

Під час виконання роботи необхідно:

- використовувати тільки справний інструмент і за призначенням;
- слідкувати, щоб на робочому місці не було зайвих предметів, що відволікають увагу і можуть привести до його травмування;
- при появі іскріння, короткого замикання, запаху гару, диму прилад негайно відключити та виявити причини можливого виникнення пожежі.

ВИМОГИ БЕЗПЕКИ ПІСЛЯ ЗАКІНЧЕННЯ РОБОТИ

Після закінчення роботи необхідно:

- вимкнути прилад, коли спеціаліст залишає своє місце;
- прибрати своє робоче місце;
- перевірити наявність всього інструменту згідно опису;
- повідомити керівника робіт про виявлені недоліки в роботі приладу.

ВИМОГИ БЕЗПЕКИ В АВАРІЙНИХ СИТУАЦІЯХ:

- у випадку виникнення пожежі негайно викликати пожежну команду. До її приїзду приступити до тушіння пожежі підручними засобами, а також спасінню людей та надання їм допомоги;
- у випадку ураження електричним струмом відключити живлення, прийняти необхідні міри по наданню першої медичної допомоги;
- в робочому приміщенні працівники мають бути ознайомлені з планом та порядком евакуації з приміщення, що має бути повішеним на видному місці.

РОЗДІЛ 5. ОХОРОНА НАВКОЛИШНЬОГО СЕРЕДОВИЩА

Завданням законодавства про охорону навколишнього природного середовища є регулювання відносин у галузі охорони, використання і відтворення природних ресурсів, забезпечення екологічної безпеки, запобігання і ліквідації негативного впливу господарської та іншої діяльності на навколишнє природне середовище, збереження природних ресурсів, генетичного фонду живої природи, ландшафтів та інших природних комплексів, унікальних територій та природних об'єктів, пов'язаних з історико-культурною спадщиною. Згідно закону України «Про підприємства в Україні» усі роботодавці повинні турбуватись про дотримання у своїй діяльності вимог законів України стосовно охорони праці та навколишнього природного середовища.

В дипломній роботі основним предметом дослідження була розробка систем управління рухомим об'єктом. А саме в роботі розроблено та протестовано алгоритм роботи на роботі з 3DOF(3 степені свободи) ніг. Також можливість реалізації всієї конструкції на практиці. Весь корпус для проекту друкується на 3D принтері при використанні АБС пластику, для забезпечення максимальної міцності, та надійності конструкції. Основні матеріали які використовуються при виробництві: різновид пластику, текстоліт для друкованих плат, різноманітні електронні компоненти(резистори транзистори котушки і тп.), акумулятори(які містять хімічні реагенти), мідь для проводів, припій, флюс для пайки. Багато з цих компонентів являються доволі токсичними при виробництві та неправильній утилізації.

5.1 Аналіз проблеми впливу роботизованого електронного пристрою на стан довкілля. Розберемо окремі частини проекту та їх вплив на довкілля.

Почну з компоненту корпусу робота який виготовляється на 3D принтері при використанні АБС пластику.

АБС (Acrylonitrile butadiene styrene) - це терполімер, виготовлений полімеризацією стиролу та акрилонітрилу в присутності полібутадієну. Пропорції можуть варіюватися від 15% до 35% акрилонітрилу, 5% до 30% бутадієну та 40% до 60% стиролу. В результаті виходить довгий ланцюг полібутадієну, перехрещений з коротшими ланцюгами полі (стирол-коакрилонітрил). Групи нітрилів із сусідніх ланцюгів, будучи полярними, притягують одна одну і зв'язують ланцюги, роблячи АБС міцнішим за чистий полістирол. Стирол надає пластику блискучу, непроникну поверхню. Полібутадієн, гумова речовина, забезпечує в'язкість навіть при низьких температурах.

При виробництві АБС необхідно застосовувати двох роторні вакуумні насоси Рутса і рідинно кільцеві вакуумні насоси, при виробництві АБС - для видалення речовини «повітря + N₂ + етилу бензол, стирол і акрилонітрил».

Через виділень стиролу і акрилонітрилу і інших речовин можливе перевищення норм по викидах в навколишнє середовище.

АБС пластик являється не біорозкладаним матеріалом, якщо просто викидати непотрібний пластик, то він буде накопичуватися в природі. Відоме велика тихоокеанська сміттєва пляма, така природна катастрофа негативно впливає на всіх живих істот, які проживають в океані а також на узбережжі. Наприклад в шлунках деяких морських жителів було знайдено велику кількість сміття, в тому числі пластикові вироби. Також під час нагрівання утворюються пари акрилонітрилу, які є отруйними.[23]

Інший компонент який несе небезпеку це Літій-іонна акумуляторна батарея. Основні проблеми для навколишнього середовища, це видобуток літію та кобальту, а також утилізація акумуляторів.

Вплив видобутку літію

Літій міститься в солончаках. В солончаках всвердляться отвори, і розсіл закачується на поверхню, залишаючи його випаровуватися у ставках. Це дозволяє екстрагувати карбонат літію за допомогою хімічного процесу.

Видобуток літію має значні екологічні та соціальні наслідки, особливо через забруднення води та виснаження. Крім того, для переробки літію необхідні токсичні хімічні речовини. Викид таких хімічних речовин через вимивання, розливи або викиди в атмосферу може завдати шкоди громадам, екосистемам та виробництву їжі. Більше того, видобуток літію неминуче шкодить ґрунті, а також спричинює забруднення повітря[24].

Соляні площі, де міститься літій, знаходяться на посушливих територіях. У цих місцях доступ до води є ключовим для місцевих громад та їх засобів до існування, а також місцевої флори та фауни. У солончаках Чилі в Атакамі гірничодобувна промисловість споживає, забруднює та відволікає дефіцитні водні ресурси від місцевих громад[25]. Видобуток літію спричинив конфлікти,

пов'язані з водою, з різними громадами, такими як громада Токонао на півночі Чилі[4]. В Аргентині Салар де Хомбре Муерто місцеві громади стверджують, що операції з літієм забруднили потоки, що використовуються для зрошення людей, худоби та сільськогосподарських культур[26].

Широко поширені припущення про те, чи може Болівія стати наддержавою літію, можливо, випередивши Чилі, розблокувавши свої значні ресурси, які можуть перевищувати 100 мільйонів тонн у її солончаках.

Розвідка та інвестування літію також відбувається за межами Андського регіону. Наприклад, американська гірнична корпорація Nova рухається вперед із придбанням ліцензійних угод на майно для видобутку літію в Монголії у відповідь на поточний бум продажів електронних товарів.

Наразі Болівія чинила опір широкомасштабному промислому видобутку літію, хоча вона планує побудувати пілотний проект як попередник можливого розвитку гірничодобувної галузі літію в майбутньому. Однак багатий літієм Салар де Уюні знаходиться недалеко від шахти Сан-Крістобаль, яка з моменту відкриття у 2007 році спричинила "екологічну та соціальну катастрофу, яка вражає всі південно-західні Потосі", в тому числі завдяки використанню 50 000 літрів води на день.

Загалом ЄС виробляє близько 24 кг електричних та електронних відходів на одного громадянина на рік, що включає літій, який використовується у високотехнологічних галузях[28]. Що стосується батарей, ЄС має правила щодо їх збору, переробки, обробки та утилізації, [29] який вимагав коефіцієнту збору батареї щонайменше 25% до кінця вересня 2012 року, а вимагатиме 45% до кінця вересня 2016 року. Це законодавство спеціально не стосується збору літєвих батарей.

Кількість літій-іонних акумуляторів, зібраних в ЄС у 2010 році, оцінювалась у 1289 тонн разом із 297 тоннами первинних літєвих батарей. За даними бельгійських переробників Umicore, це лише близько 5% літій-іонних

акумуляторів, що надходять на ринок. Німеччина, Франція, Бельгія та Нідерланди мають найкращі результати збору акумуляторів, включаючи первинні та вторинні літій-іонні акумулятори. Проте навіть у цих країнах рівень збору дуже низький.

Акумулятори не тільки ризикують виділити токсичні гази при пошкодженні, але й основні інгредієнти, такі як літій та кобальт, обмежені, і видобуток може призвести до забруднення води та виснаження серед інших екологічних наслідків.

5.2 Аналіз основних джерел впливу та їх наслідків на людину та її оточення.

При виготовленні електронного пристрою завжди потрібно використовувати пайку. Але цей процес не являється корисним для людина, а навпаки може викликає деякі хронічні хвороби.

При пайці людина постійно розігріває припій та каніфоль до температури в районі 300 градусів. А що таке припій Припій — метал, сплав або суміш оксидів, що застосовується для з'єднання металевих, мінералокерамічних та інших деталей, для лудіння посуду тощо. Припої на основі олов'яно-свинцевих сплавів дуже широко використовувалися в минулому і ще й досі виробляються. Вони особливо зручні при ручному паянні, але свинець що входить до їхнього складу може негативно впливати на довкілля. З цієї причини здійснюється поступовий перехід на без свинцеві припої, які, однак, з різних причин є менш придатними для ручного застосування. [30]

Свинець може спричинити серйозні хронічні наслідки для здоров'я. Вплив в основному відбувається через випадкове потрапляння на шкіру, одягніть рукавички, якщо безпосередньо обробляєте припій. Обмежені випари можуть утворюватися при пайці. Свинець впливає на кровотворну і нервову системи, шлунково-кишковий тракт і нирки. Спричиняє анемію (оскільки включається до ланцюгу біосинтезу гема і скорочує період життя еритроцитів),

а також енцефалопатію, зниження розумових властивостей, викликає шлунково-кишкові розлади, диспепсію, коліки, нефропатію.

Як і більшість подібних сполук, свинець може спричинити гостре та хронічне отруєння. Симптомами гострого отруєння є: розлади травного каналу (збільшене слиновиділення, присмак металу в роті, нудота, блювання, свинцеві кольки в животі, діарея), зниження температури тіла і кров'яного тиску, брадикардія або тахікардія, холодний піт, порушення дихання, колапс. Потрапляння до організму 250 мг найбільш токсичної сполуки тетраетилсвинцю $(C_2H_5)_4Pb$ закінчується смертю. У тих осіб, які з огляду на особливості своєї професії змушені контактувати зі сполуками свинцю або перебувають у забрудненому середовищі, спостерігаються хронічні отруєння. Хронічне отруєння виявляється у вигляді анемії, яка є наслідком блокування ензимних систем, що ведуть до синтезу гемоглобіну; ушкодженнями як центральної, так і периферійної нервової системи. Симптомами таких ушкоджень є погіршення розумової діяльності (у дітей, які відстають у розвитку, встановлено вищий рівень свинцю у крові), надмірна рухливість, агресивність, порушення сну, порушення функцій нирок. Також свинець надзвичайно небезпечний для вагітних жінок, оскільки має здатність проникати через плаценту та призводить до важких необоротних неврологічних порушень у плода.

Каніфоль (колофонія, ерсин) - це смола, що міститься в флюсі припою. Потік утворює видимі випари, помітні під час пайки. Вплив каніфолі може спричинити подразнення очей, горла та легенів, кровотечу з носа та головний біль. Повторний вплив може спричинити сенсibiliзацію дихання та шкіри, викликаючи та посилюючи астму. Каніфоль є серйозною загрозою для здоров'я праці .

Вплив паяльних парів смолистого або колоніального потоку на основі колонії може призвести до безлічі небезпек для здоров'я[31][32]:

Професійна астма - один із багатьох ризиків, викликаних потоками газів; кашель, задишка, хрипи та біль у грудях є одними із симптомів астми.

Алергічна гіперчутливість - ще одна загальна небезпека для здоров'я, алергічна гіперчутливість розвивається з перших кількох місяців впливу, і вона може продовжувати наростати протягом багатьох років - викликаючи хрипи та важке дихання.

Подразнення - в результаті прямого або непрямого контакту з припоєм на основі смоли симптоми можуть варіювати від простого подразнення очей або носа до більш важких захворювань шкіри, що потрапляють у повітря.

Дим припою може також викликати інші захворювання, такі як хронічний бронхіт, хімічна гіперчутливість, біль у грудях, головні болі та запаморочення .

Для виготовлення корпусу пристарою необхідно використовувати 3D друк. Під час друку використовуються пластик, який під час нагрівання може випаровувати деякі токсичні пари. Не тільки АБС, а й ПЛА можуть виділяти токсичні випари, відомі як ЛОР (Леткі органічні речовини). Не всі ЛОР насправді токсичні, але деякі можуть бути, особливо для молодих користувачів. Перш ніж це стане серйозною проблемою для здоров'я, нове дослідження, проведене 3Dsafety.org у співпраці з італійським виробником 3D-принтерів WASP, проаналізувало точну кількість токсичних ЛОР, а також потенційно небезпечних наночастинок, що виділяються під час екструзії ниток, з метою оцінити потенційні ризики для здоров'я.

Нове дослідження, представлене д-ром Фабріціо Мерло та доктором інж. Стефано Маццоні відштовхується від інших попередніх досліджень, проведених на початку 90-х років, які продемонстрували, що під час плавлення та обробки пластикових матеріалів у вигляді газів виділяється кілька токсичних частинок, серед яких аміак, ціанідринова кислота, фенол та бензол.

Лабораторні випробування показали, що АБС значно токсичніший, ніж, але полімер на основі кукурудзи не звільняється від небезпечних викидів, особливо якщо екструдується при температурі вище 200 ° С. Крім того (як і слід було очікувати), однакові матеріальні котушки, придбані у різних торгових посередників, випускають дуже різну кількість ЛОР , навіть якщо вони використовуються в одному 3D-принтері та мають однакові параметри швидкості та температури.

Другий критичний аспект - це викид наночастинок, тобто частинок діаметром менше 0,1 мкм, які можуть поглинатися безпосередньо легеневою альвеолою та епідермісом. У цьому випадку викиди при використанні АБС варіюються від 3 до 30 разів від тих, що виникають при використанні нитки PLA. Тест також продемонстрував, що час, необхідний для того, щоб концентрація наночастинок у повітрі повернулася до стандартних рівнів, становив від 10 до 30 хвилин після припинення процесів екструзії. Завдяки технології фотоіонізації дослідження (яке було опубліковане на сайті 3Dsafety.org і буде поступово оновлюватися додатковою інформацією) також проводилось на нейлоні, полістиролі, ПЕТ та інших матеріалах.

Проблема АБС При нагріванні в процесі виробництва (для формування або екструзії, а також в 3D-друку) утворюються пари акрилонітрилу, які є отруйними. Існує побоювання, що акрилонітрил може бути канцерогеном для людини. Бутадієн є відомим канцерогеном. Стирол теж має канцерогенні властивості.

Стирол легко окислюється, приєднує галогени, полімеризується (утворюючи тверду склоподібну масу - полістирол) і сополімеризується з різними мономерами. Полімеризація відбувається вже при кімнатній температурі (іноді з вибухом), тому при зберіганні стирол стабілізують антиоксидантами (наприклад, третбутилгідрокатехіном, гідроксиноном).

Вдихання парів стиролу загрожує численними гострими і хронічними захворюваннями. Ця речовина негативно впливає на функцію печінки і нирок, на кровоносну і нервову системи. Тривале потрапляння стиролу в організм людини загрожує катарам дихальних шляхів, подразненням шкіри і слизових оболонок, зміною складу крові, порушеннями функцій вегетативної системи.

Акрилонитрил за ступенем впливу на організм людини відноситься до 2-го класу небезпеки (речовини високонебезпечні класу). АН необоротно зв'язується з білками, РНК і ДНК різних тканин. Небезпечний при вдиханні, отруйний при прийомі всередину - аж до летального результату . Пари викликають роздратування слизових оболонок і шкіри. Діє через неушкоджену шкіру. При горінні утворюються отруйні гази.

Симптоми ураження: Головний біль, запаморочення, слабкість, нудота, блювота, задишка , пітливість, серцебиття, зниження температури тіла, ослаблення пульсу , судоми , втрата свідомості, почервоніння і печіння шкіри.

1,3-Бутадиєн був призначений Група 1 канцероген («канцерогенності для людини») по IARC , , а також був включений в список канцерогенів Агентства з реєстрації токсичних речовин і захворювань охорони навколишнього середовища США. Американська конференція урядових гігієністів промисловості (ACGIH) перелічує цю хімічну речовину як підозрюваний канцероген. Рада з питань захисту природних ресурсів (NRDC) перелічує деякі кластери хвороб, які, як підозрюється, пов'язані з цією хімічною речовиною. Деякі дослідники прийшли до висновку, що це найпотужніший канцероген у сигаретному димі , вдвічі потужніший, ніж акрилонітрил, що витікає.

1,3-бутадиєн також є підозрою на тератоген людини . Тривалий і надмірний вплив може впливати на багато областей людського тіла; Встановлено, що кров, мозок, око, серце, нирки, легені, ніс та горло реагують на наявність надлишку 1,3-бутадиєну. Дані на тваринах свідчать, що жінки мають вищу чутливість до можливих канцерогенних ефектів бутадиєну на

чоловіків під впливом хімічних речовин. Це може бути пов'язано з впливом рецепторів естрогену. Хоча ці дані виявляють важливі наслідки для ризиків впливу людиною бутадієну, для отримання остаточної оцінки ризику необхідно більше даних. Також бракує людських даних щодо впливу бутадієну на репродуктивну функцію та розвиток, що виявляється у мишей, але дослідження на тваринах показали, що дихання бутадієном під час вагітності може збільшити кількість вроджених вад розвитку, а люди мають ті ж гормональні системи, що і тварини.

1,3-бутадієн визнаний високореактивною летючою органічною сполукою (HRVOC) завдяки своїй здатності легко утворювати озон, і як такий, викиди хімічної речовини суттєво регулюються TCEQ в частинах озону Х'юстон-Бразорія-Галвестон. Зона досягнення.

Під час проведення пайки потрібно обережно поводитись з літій іонними акумуляторами. Літій-іонні акумулятори складаються з анода і катода, розділених пористим полімерним сепаратором. Активним матеріалом катода найчастіше є оксиди перехідних металів з вбудованими в кристал іонами літію. У аноді зазвичай використовується графіт. Електроліт, яким залита електрохімічна комірка, являє собою органічний розчин солей літію. При першій зарядці, виробленої фірмою-виробником, при встановленні літію в анод на електродах (особливо на аноді) утворюється захисний іон-провідний шар (SEI), що складається з розкласа електроліту. Цей шар захищає електроди від паразитичних реакцій з електролітом.

Найчастіше причиною самозагорання акумуляторів є коротке замикання всередині електрохімічної комірки. Електричний контакт між анодом і катодом може виникнути з багатьох причин. Це може бути, наприклад, механічне пошкодження осередки. Ще внутрішнє коротке замикання виникає через порушення технології виробництва при нерівній нарізці електродів або попаданні металевих частинок між анодом і катодом, що веде до пошкодження

пористого сепаратора. Також причиною внутрішнього короткого замикання може бути «проростання» ланцюжків металевого літію (дендритів) через сепаратор. Такий ефект виникає, якщо іони літію не встигають вбудуватися в кристал анода при занадто швидкій зарядці або низькій температурі, а також якщо ємність активного матеріалу катода перевищує ємність анода,

Отже, після того, як сталося коротке замикання, акумулятор починає нагріватися. Коли температура досягає 70-90 ° С, іон-який проводить захисний шар на аноді починає розкладатися. А далі літій, вбудований в анод, вступає в реакцію з електролітом, виділяючи леткі вуглеводні: етан, метан, етилен і т.д. Але, не дивлячись на наявність такої вибухонебезпечної суміші, загоряння не відбувається, так як в системі поки немає кисню.

Так як реакції з електролітом екзотермічні, температура і тиск усередині акумулятора продовжують підвищуватися. Коли температура досягає 180-200 ° С, матеріал катода, зазвичай представляє з себе оксид перехідних металів з вбудованим в кристал літієм, вступає в реакцію диспропорціонування і виділяє кисень. Ось тут-то і відбувається самозаймання і ще більш різкий стрибок температури. Паралельно йде термічний розклад електроліту (200-300 ° С), також виділяє тепло. І, врешті-решт, в реакцію з електролітом (якщо він ще залишився) вступає графіт, а коли температура досягає 660 ° С, плавиться алюмінієвий струмоприймач. Вище 900 ° С температура зазвичай не піднімається, так як розкладатися вже нічому.

Крім внутрішнього короткого замикання існують і інші причини самозаймання: перегрів акумулятора, неправильна зарядка / розрядка (перевищення максимально допустимого напруги, зарядка на високих токах, занадто глибока розрядка), і т.д. Але всі ці причини призводять до одного результату: термічному розгону і розкладання електроліту при взаємодії з електродами. Розрізняються лише порядки вищеписаних реакцій і їх швидкість.

Такі вибухи можуть завдати значну фізичну шкоду людині.

5.3 Рекомендації щодо зниження цих негативних чинників.

Відходи АБС пластику можна перероблювати. Для переробки відходів АБС застосовують різні термічні способи.

Пластик може пройти термічну переробку без втрати вихідної якості або з малою часткою таких втрат, але іноді може набути жовтуватого відтінку (що легко виправляється забарвленням в необхідний колір).

Технології та обладнання

Оскільки АБС є термопластичних, то його можна регенерувати за рахунок високих температур. Однак важливо пам'ятати, що в його складі є токсичні елементи. Наприклад, в процесі нагрівання виділяється токсичний газоподібний стирол. Його необхідно відводити з цеху, а краще використовувати автоматизоване управління на етапі плавлення.

Також варто врахувати, що різні марки пластика несумісні, тому відходи АБС пластика попередньо потрібно впорядкувати. Визначити до якого типу пластику відноситься полімер можна за маркуванням знаками утилізації .

Матеріал переробляється методом лиття або екструзії. Лиття передбачає нагрівання пластику до вязкотекучого стану і подальшу подачу матеріалу в закриту форму, де матеріал твердне і набуває вигляду готового виробу. Маса виробів, отриманих таким шляхом, може варіюватися від декількох грамів до декількох кілограмів, товщина стінок - від 1 мм до 20 мм.

При екструзії АБС пластик плавиться в екструдері, потім «продавлюється» через формуючий пристрій з подальшим охолодженням і калібруванням. Стандартні екструзійні марки відрізняються гарною стійкістю до ударів, деякі мають антистатичні властивості. Матеріал застосовують у виготовленні деталей холодильних установок.

Важливо зауважити, що для екструзії АБС потрібен водокільцевий гранулюючий екструдер, так як цей матеріал негативно реагує на присутність кисню в місці нагріву. Від нього він окислюється і втрачає товарні властивості, а вода захищає гранулу від впливу повітря.

Підсумкова вторинна гранула вимагає ретельного сушіння. Її щільність нижче, ніж у вихідного матеріалу з наявністю пір. Через це вона має властивості вбирати воду і довго утримувати її в собі. Якщо матеріал буде недостатньо сухим, відлити з нього якісний виріб не вийде.

Стандартні ливарні марки відрізняються високою стійкістю до горіння, антистатичними властивостями. Зі стандартних марок виготовляють великогабаритні, тонкостінні вироби, корпуси приладів, товари народного споживання.

Що знову виробляють з отриманого вторсировини?

Класичною сферою застосування вторсировини є зниження собівартості вихідних виробів, тому що вдруге перероблений вид сировини обходиться дешевше, ніж первинний. Повторно перероблені відходи АБС можуть використовуватися без будь-яких обмежень, тому при дотриманні всіх умов переробки їх структура не змінюється.

Простий, енергоефективний процес переробки літій-іонних батарей.

Простий новий процес переробки відновлює старі катоди літієвих акумуляторів до стану монетного двора, використовуючи половину енергії поточних процесів. На відміну від сучасних методів утилізації, які розбивають катоди на окремі елементи, які доводиться складати знову, нова техніка випльовує сполуки, які готові потрапити в нову батарею[33].

У методі робота на батареях оксиду літію кобальту, що використовується в ноутбуках і смартфонах, а також на складних батареях літій-нікель-марганець-кобальт знайдена в електричних машинах.

Літієві батареї мають аноди з графіту та катоди з оксидів металевих літіїв, де метал являє собою деяку комбінацію кобальту, нікелю, марганцю та заліза . Сьогодні менше п'яти відсотків старих літієвих акумуляторів переробляється. Оскільки мільйони великих акумуляторів електромобілів підуть у відставку протягом наступного десятиліття, ми збираємось відправляти ще більші гори легкозаймистих, токсичних відходів акумуляторів на звалища. Крім того, ці відходи містять цінні метали. Існує серйозна занепокоєність тим, що поставки таких важливих металів, як кобальт та літій , скорочуються . Переробка буде ключовою, якщо ми хочемо не відставати від попиту на акумулятор.

Кілька компаній, переважно в Китаї, вже переробляють батареї. Стандартна процедура вимагає подрібнення батареї, а потім або їх плавлення, або розчинення в кислоті. В кінці виходять окремі метали, такі як кобальт, літій, нікель та марганець. На додаток до використання великої кількості енергії, методи знищують найцінніше в катодних батареях, каже Чжен Чен , професор наноінженерії в Каліфорнійському університеті Сан-Дієго.

"Матеріал має форму красивих, добре розроблених частинок зі специфічною мікроскопічною структурою, що визначає ефективність роботи акумулятора", - говорить він. "Багато техніки, енергії та часу йде на виготовлення цих конструкцій".

Простий метод, який розробили Чен та його колеги, зберігає цю мікроструктуру. Спочатку дослідники крутили на комерційних літієвих елементах, поки вони не втратили половину своєї ємності для зберігання енергії. Вони видалили катодний матеріал з підкладки з алюмінієвої фольги і замочили його у гарячій солі з літієвою ванною. Потім сушили розчин, отримуючи порошок, який швидко нагрівали до 800 градусів С, а потім дуже повільно охолоджували.

Процес відновлює атомну структуру катодного матеріалу і повторно вводить в нього іони літію. І він використовує половину енергії звичайних

процесів. Дослідники виготовили нові акумуляторні елементи з регенованого катодного матеріалу. Нові катоди показали таку ж ємність накопичення енергії, час зарядки та термін служби, що і оригінали. Результати повідомляються в журналі Green Chemistry .

Дві інші компанії застосовують подібну технологію "прямої переробки", яка регенує весь структурований катодний матеріал. Заснована в Сан-Франциско акумуляторна компанія Farasis Energy та Bend, штат Орегон, стартап OnTo Technologies одночасно розробляє технологію та намагається розширити її. Всі процеси дещо відрізняються один від одного.

Для рішення проблеми вибуху літій іонних акумуляторів при паянні необхідно слідувати робочим інструкціям. Або перейти на інший тип акумуляторів.

Рішення проблеми отруєння парами при пайці.

Безпосередні виробничі приміщення, в яких знаходяться ділянки пайки, потрібно обладнати постійно діючою місцевою вентиляцією. Місцеві відсмоктувачі, які видаляють шкідливі речовини від виробничого устаткування, слід блокувати з включенням цього обладнання для виключення його роботи при вимкненому вентиляції.

Забезпечення безпеки людям, які працюють з 3Д принтером реалізуються шляхом відокремлення принтера в окремий герметичний копус, та проведенням системи відводу газів.

Робочі місця при пайку олов'яно-свинцевими припоями необхідно обладнати місцевими витяжними пристроями, що забезпечують швидкість руху повітря безпосередньо на місці пайки не менше 0,6 м / с, незалежно від конструкції повітря приймачів. Повітря приймачі повинні легко переміщатися з надійною фіксацією положення в процесі монтажних робіт для максимального наближення до місця пайки.

Розрахунок необхідного повітрообміну при пайку припоєм ПОС-61

$$G = K \cdot B / (q_2 - q_1), \text{ де:}$$

$B = 0,45 \text{ мг/год}$ – кількість свинцю, що виділяється на робочому місці за 1 годину;;

$$K = 1,7;$$

$q_2 = 0,01 \text{ мг/м}^3$ – концентрація свинцю в повітрі, що видаляється, приймається рівною гранично допустимої;

$q_1 = 0,003 \text{ мг/м}^3$ – концентрація свинцю в припливно повітрі, становить 30% від ГДК

$$G_{св} = 0,45 \cdot 1,7 / (0,01 - 0,003) = 79 \text{ м}^3/\text{год}.$$

В якості місцевої вентиляції виберемо відсмоктувальну панель.

Витрата повітря через панель обчислюється за формулою:

$$G_n = C \cdot Q_K^{1/3} \cdot (H + B)^{5/3}, \text{ де:}$$

C – коефіцієнт, що залежить від конструкції панелі і її розташування відносно джерел тепла;

$Q_K = 0,25 Q_{чел} + Q_n = 0,25 \cdot 77 + 9 = 28,25 \text{ Вт}$ – конвективна складова джерела тепла;

$H = 0,35 \text{ м}$ – відстань від верху площині джерела до центру всмоктуючих отворів панелі;

$B = 0,3 \text{ м}$ – ширина джерела тепла.

Коефіцієнт C застосовується рівним:

$$C = 228 \cdot F \cdot [l / (H + B)]^{2/3}, \text{ де:}$$

$l = 0,5 \text{ м}$ – максимальне видалення джерела від панелі;

$F = 0,5 \text{ м}^2$ – площа джерела тепловиділення.

$$C = 228 \cdot 0,5 \cdot [0,8 / (0,35 + 0,3)]^{2/3} = 130,9$$

Отже:

$$G_n = 130,5 \cdot 3,04 \cdot (0,35 + 0,3)^{5/3} = 193,6 \text{ м}^3/\text{год}.$$

Висновки до розділу охорона навколишнього середовища .

При виробництві електроніки існує деякі проблемні фактори, які можуть негативно впливати, як на екологію цілої планети, так і на людину. Наприклад добуток літію негативно впливає на цілі регіони, та рішення цієї проблеми дуже складне. Але у всякому разі утилізувати, та навіть використовувати повторно, шляхом відновлення, літій іонні акумулятори являється можливою задачею.

Також на екологію цілих регіонів, океанів, річок впливає АБС пластик, який використовую в 3Д друку. Також було знайдено рішення цієї проблеми. Рішенням було повторне використання пластику шляхом переплавки, якщо коротко.

Для забезпечення безпеки людини при виконанні паяльних робіт. Оскільки без спеціальних мір, таких як витяжки, робітники могли зазнавати дуже серйозних виробничих хвороб. Корпус 3Д принтера відділено герметичним кожухом від основного переміщення, та проведено вентиляцію всередину.

Загалом такі рішення сприяють покращенню екологічного стану на виробництві та в цілому, на протязі всього життєвого циклу для виробу.

Література

1. Arduino Uno R3 Technical Details; веб-сайт. URL:
<http://static6.arrow.com/aropdfconversion/97b354c3cd11851042ebd1fd252fbd6b6def5a73/adafruit2877arduinouno.pdf>.
2. Arduino Nano. User Manual веб-сайт. URL:
<https://www.arduino.cc/en/uploads/Main/ArduinoNanoManual23.pdf>
3. HC-05 Bluetooth Module; веб-сайт. URL:
<https://www.gme.cz/data/attachments/dsh.772-148.1.pdf>
4. Arduino Shield Manual: веб-сайт. URL;
<https://www.robotshop.com/media/files/PDF/dfrobot-arduino-shields-manual.pdf>
5. Arduino programming notebook .
веб-сайт. URL; <http://engineering.nyu.edu/gk12/amps-cbri/pdf/ArduinoBooks/Arduino%20Programming%20Notebook.pdf>
6. Y. Fukuoka, H. Kimura, A.H. Cohen Adaptive dynamic walking of a quadruped robot on irregular terrain based on biological concepts Int. J. Robot Res., 22 (3–4) (2003), pp. 187-202
- 7 . S. Kitano, S. Hirose, A. Horigome, G. Endo TITAN-XIII: sprawling-type quadruped robot with ability of fast and energy-efficient walking ROBOMECH J., 3 (1) (2016), p. 1
- 8 . H. Kimura, Y. Fukuoka, A.H. Cohen Adaptive dynamic walking of a quadruped robot on natural ground based on biological concepts Int. J. Robot Res., 26 (5) (2007), pp. 475-490
- 9 C. Semini, N.G. Tsagarakis, E. Guglielmino, M. Focchi, F. Cannella, D.G. Caldwell Design of HyQ—a hydraulically and electrically actuated quadruped robot Proc. IME J. Syst. Contr. Eng. (2011) 0959651811402275

10 P.T. Doan, H.D. Vo, H.K. Kim, S.B. Kim A new approach for development of quadruped robot based on biological concepts Int. J. Precis. Eng. Manuf., 11 (4) (August 2010), pp. 559-568

11 Інструкції з охорони праці під час виконання монтажних робіт інструментами і пристроями, затвердженої наказом Міністерства праці та соціальної політики України від 05 червня 2001 року № 254, зареєстрованої у Міністерстві юстиції України 20 липня 2001 року за № 616/5807 (НПАОП 0.00-5.24-01), ДСТУ 7237:2011 та ГОСТ 12.1.030-81.

12 ДСТУ ІЕС 60745-1:2010 «Інструмент ручний електромеханічний. Безпека. Частина 1. Загальні вимоги».

13 ДНАОП 0.00-1.21-98 Правила безпечної експлуатації електроустановок споживачів, затвердженого наказом

Держнаглядохоронпраці від 09.01.98 N 4 (z0093-98), зареєстрованого у Міністерстві юстиції 10.02.98 за N 93/2533, та ГОСТ 12.2.013.0-91.

14 ДСН 3.3.6.042-99

15 „Інструкцією з охорони праці при роботах з олов'яно-свинцевими малосурм'янистими припоями” ПІ 1.4.32-423- 2005.

16 ДСТУ Б А.3.2-12:2009 ССБП Системи вентиляційні

17 ГОСТ 12.0.003

18 ГОСТ 12.1.005, ДСН 3.3.6.042

19 ДСН 3.3.6.042

20 ДБН А.3.2-2 2009, ДБН В.2.5-67:2013, СНиП 2.09.02. 85, ДБН В.2.2-28-2010.

21 ДСТУ Б А.3.2-15:2011, ДБН В.2.5:28-2018

22 ДБН В.2.5-67:2013

23 ДСТУ Б В.2.5-82:2016, НПАОП 0.00-1.29, НПАОП 40.1-1.01, НПАОП 40.1-1.21, НПАОП 40.1-1.32

24 НАПБ Б.03.002

25 ГОСТ 12.1.018, ГОСТ 12.4.124, ДБН В.2.5-27, НПАОП 0.00-1.29

26 НАПБ Б.03.002

27 ДБН В.2.5-67:2013

28 ДСТУ 7239:2011 ССБП Засоби індивідуального захисту. Загальні вимоги та класифікація.

19 ДБН Б.2.5:28-2018

20 ПРАВИЛА експлуатації та типові норми належності вогнегасників.

21 НАПБ Б.03.002-2007

22 ДСТУ 3855-99 Пожежна безпека. Визначення пожежної небезпеки матеріалів та конструкцій.

23 Will Dunham (12 February 2019). "[World's Oceans Clogged by Millions of Tons of Plastic Trash](#)". *Scientific American*. Retrieved 31 July 2019. China was responsible for the most ocean plastic pollution per year with an estimated 2.4 million tons, about 30 percent of the global total, followed by Indonesia, the Philippines, Vietnam, Sri Lanka, Thailand, Egypt, Malaysia, Nigeria and Bangladesh.

24 Democracy Center Special Report, Bolivia and its Lithium: Can the “Gold of the 21st Century” Help Lift a Nation out of Poverty? May 2010. <http://www.ifg.org/pdf/DClithiumfullreportenglish.pdf>

25 CODEFF, REdUSE Chile: Litio en el Salar de Atacama, May 2011.

26 CODEFF Data research on lithium within the REdUSE Project Partners Countries, April 2011. See summary here: <http://www.reduse.org/en/blog/lithium-extraction-chilean-north>

27 Democracy Center Special Report, Bolivia and its Lithium: Can the “Gold of the 21st Century” Help Lift a Nation out of Poverty? May 2010. <http://www.ifg.org/pdf/DClithiumfullreportenglish.pdf>

28 European Commission, Commission Staff Working Paper: Analysis associated with the Roadmap to a Resource Efficient Europe Part II, 20 September 2011. http://ec.europa.eu/environment/resource_efficiency/pdf/working_paper_part2.pdf

29 It stipulates that collection rates of at least 25% must be met by the end of September 2012 and 45% by the end of September 2016. Directive (2006/66/EC) of the European Parliament and of the Council, 6 September 2006. <http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2006:266:0001:0014:EN:PDF>

30 Lead-free solders: Issues of toxicity, availability and impacts of extraction Proceedings - Electronic Components and Technology Conference DOI: 10.1109/ECTC.2003.1216255

31 Humpston, Giles; Jacobson, David M. (2004). Principles of soldering - Google Books. ISBN 9781615031702. Archived from the original on 2013-06-20. Retrieved 2011-08-19.

32 Controlling health risks from rosin (colophony) based solder fluxes, IND(G)249L, United Kingdom Health and Safety Executive, 1997 (online PDF) Archived 2011-01-12 at the Wayback Machine

33 Effective regeneration of LiCoO₂ from spent lithium-ion batteries: a direct approach towards high-performance active particles Department of Chemical and Biomolecular Engineering, University of California Los Angeles, Los Angeles, CA 90095, USA rogram of Materials Science and Engineering, University of California San Diego, La Jolla, CA 92093, USA