**Міністерство освіти і науки України**  
**Національний технічний університет України**  
**“Київський політехнічний інститут”**

Інститут Прикладного системного аналізу  
Кафедра Системного проектування

**РОЗРАХУНКОВО-ГРАФІЧНА РОБОТА**

**з дисципліни  
«Архітектура обчислювальних систем»**

**на тему:**«БЕЗКОНТАКТНИЙ ДОЗАТОР АНТИСЕПТИКА ДЛЯ ОБРОБКИ РУК НА БАЗІ МІКРОКОНТРОЛЛЕРА ARDUINO»

Студента 2 курсу

групи ДА-92

Насікана Д. Ю.

Керівник:

Артюхов В. Г.

Київ 2021

**ЗМІСТ**

1. **ВСТУП...........................................................................................................3**
2. **ОБГРУНТУВАННЯ ВИБОРУ КОМПОНЕНТІВ ПРИСТРОЮ........4**
   1. ***Список компонентів..........................................................................*4**
   2. ***Обгрунтування вибору мікроконтроллера....................................*4**
   3. ***Обгрунтування виботу інших компонентів.................................*4**
3. **БУДОВА, ХАРАКТЕРИСТИКИ ТА ФУНКЦІОНУВАННЯ МІКРОКОНТРОЛЛЕРА ADRUINO NANO V3.....................................5**
   1. ***Архітектура мікроконтроллера.....................................................*5**
   2. ***Типи даних мікроконтроллера......................................................*10**
   3. ***Система команд...............................................................................*11**
   4. ***Програмна модель ATmega328P*.....................................................12**
   5. ***Організація пам’яті*.........................................................................13**
   6. ***Адресація даних................................................................................*14**
   7. ***Адресація команд.............................................................................*15**
   8. ***Система переривань та таблиця переревань.............................*16**
   9. ***Характеристики плати arduino nano v3.....................................*18**
   10. ***Розпіновка плати та схеми плати...............................................*19**
4. **БУДОВА ТА ХАРАКТЕРИСТИКИ ДОПОМІЖНИХ ЕЛЕКТРОННИХ ПРИСТРОЇВ..............................................................21**
   1. ***Ультразвуковий далекомір HC-SR04............................................*21**
   2. ***DC сервопривод SG90.......................................................................*22**
5. **ПРОЄКТУВАННЯ ПРИСТРОЮ...........................................................26**
   1. ***Ідея роботи.......................................................................................*26**
   2. ***Під’єднання елементів...................................................................*27**
   3. ***Програмування мікроконтроллера...............................................*28**
6. **ВИСНОВКИ...............................................................................................30**
7. **СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ..............................................31**
8. **ВСТУП**

Метою цієї розрахунково-графічної роботи є розробка пристрою, що здатен безконтактно розприскувати антисептичну рідину на руки людини, використовуючи мікроконтроллер Arduino архітектури AVR та ряд інших допоміжних електронних пристроїв. У рамках цієї роботи буде поглиблено, стуктуровано та використано знання з курсів архітектури обчислювальних систем, цифрової схемотехніки, електротехніки та програмування.

Arduino - це електронна платформа з відкритим кодом, заснована на простому у використанні апаратному та програмному забезпеченні. Мікроконтроллер arduino - невелика плата з власним процесором і пам'яттю, на якій також є пара десятків контактів, до яких можна підключати компоненти: світлодіоди, датчики, мотори, роутери, магнітні дверні замки і взагалі все, що працює від електрики. У процесор Arduino можна завантажити програму, яка буде керувати всіма цими пристроями за заданим алгоритмом. Зазвичай, Arduino використовують при проектуванні та прототипуванні цифрових гаджетів, таким чином, спочатку збирають прототип розробленого гаджета використовуючи Arduino та тестують його, а потім запускають масове виробництво з використанням плат своєї розробки. Крім цього, через свою відносну простоту, платформу активно використовують й любителі електронних пристроїв.

1. **ОБГРУНТУВАННЯ ВИБОРУ КОМПОНЕНТІВ ПРИСТРОЮ**

У цьому розділі буде наведений перелік електронних компонентів та пристроїв, що потрібні для реалізації мети роботи, та обгрунтований саме їх вибір.

* 1. ***Список компонентів***
* Мікроконтроллер архітектури AVR - Arduino Nano v3
* Ультразвуковий дальномір - HC-SR04
* DC сервопривод SG90
* Джампери
  1. ***Обгрунтування виботу мікроконтроллера***

Одною із вимог для створення пристрою є компактність, простота конструкції й низьке енергоспоживання, тому було вирішено обрати Arduino Nano v3 – одну з найбільш мініатюрних та енергоекономних плат Arduino. Ця плата порівняно дешева, використовує чіп на базі архітектури AVR. Arduino Nano v3 – не надто потужний мікроконтроллер з малою кількістю пінів, але, так як проєкт дозатора антисептичної рідини для рук не є ресурсомістким та не містить велику кількість пристроїв, потреби даного проекту він цілком задовольняє. Більше того, на відміну від багатьох аналогів, мікрокнтроллер має вбудований програматор та зрозуміле й не важке в освоєнні програмне забезпечення для програмування пристрою, що теж є великим плюсом і дозволяє швидко й без зайвих клопотів приступити до розробки пристрою.

* 1. ***Обгрунтування вибору інших компонентів***

Інші компоненти, такі як ультразвуковий дальномір HC-SR04 та DC сервопривод SG90 були вибрані саме такими для сумісності з обраним мікроконтроллером та через доступність й невелику ринкову ціну. Вони також цілком задовольняють потреби роботи й їх застосування у роботі пристрою буде наведено у розділі «Ідея роботи пристрою».

1. **БУДОВА, ХАРАКТЕРИСТИКИ ТА ФУНКЦІОНУВАННЯ МІКРОКОНТРОЛЛЕРА ADRUINO NANO V3**

Цей розділ містить теоретичний матеріал з архітектури обраного мікроконтроллера, його характеристики, внутрішню будову та схеми.

* 1. ***Архітектура мікроконтроллера***

Обрана плата – Arduino nano v3 побудована на базі мікроконтроллера ATmega328P.

Atmel ATmega328P - це 8-розрядний мікроконтролер CMOS логіки із низькою потужністю, заснований на вдосконаленій архітектурі RISC AVR. RISC (Reduced Instruction Set Computer) - архітектура з ретельно підібраним набором команд, які як правило виконуються за один такт роботи процесора. Сучасні AVR мікроконтролери, , як і ATmega328P, містять близько 130 команд, які дуже швидко виконуються і не вимагають як великих витрат ресурсів, так і значної споживчої потужності. [1]

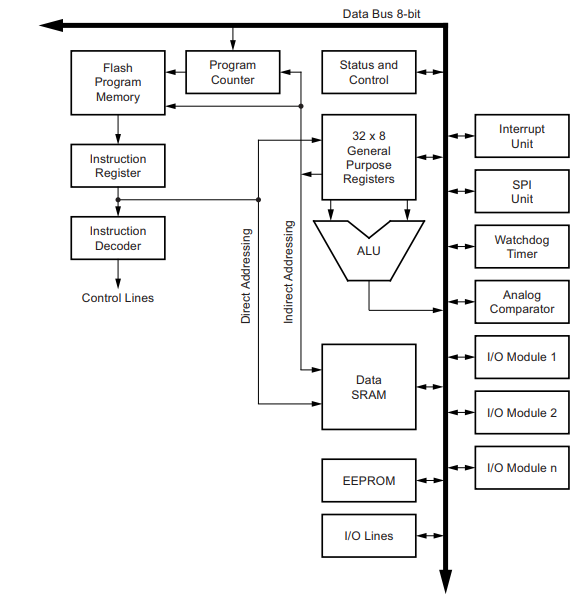


Рис.1. Схема мікроконтроллера ATmega328P [1]

Опишемо основні складові мікроконтроллера, які присутні на даній схемі.

* + 1. ***Flash program memory***

Незалежний ПЗУ (постійний запам'ятовуючий пристрій) що побудований за технологією FLASH. Тут зберігається програма, яка буде виконуватися блоком ALU мікроконтролера. Флеш-пам'ять чіпа можна багаторазово перезаписувати, тим самим змінюючи або доповнюючи програмний код для виконання. Даний тип пам'яті може зберігати записані в неї дані протягом 40 років, а кількість можливих циклів стирання / запису може досягати 10000. У даній моделі мікроконтроллера об’єм флеш-пам’яті складає 32 КБ пам’яті. [6]

* + 1. ***EEPROM***

EEPROM - незалежна пам'ять даних в якій дані будуть зберігатися навіть при відключенні живлення мікроконтролера. У даній пам'яті можна зберігати налаштування виконання програми, зібрані дані для статистики роботи пристрою і іншу корисну інформацію. Наприклад, зібравши невелику метеостанцію на мікроконтролері, в EEPROM на кожен день можна зберігати дані про температуру повітря, тиск, силу вітру, а потім в будь-який момент використати ці зібрані дані і провести статистичні дослідження. Для EEPROM виділено окремий адресний простір, який відрізняється від адресного простору RAM і FLASH. Пам'ять EEPROM мікроконтролера - дуже цінний ресурс, оскільки її як правило дуже мало - від 0,5 до декількох кілобайт на чіп. Кількість перезаписів для даного типу пам'яті становить близько 100000 що в 10 разів більше ніж ресурс FLASH пам'яті. [6]

* + 1. ***ALU***

ALU - Арифметико-логічний пристрій, який синхронно за тактовим сигналом і спираючись на стан лічильника команд (Program Counter) вибирає з пам'яті програми (FLASH) чергову команду й виконує її.

* + 1. ***Регістри загального використання***

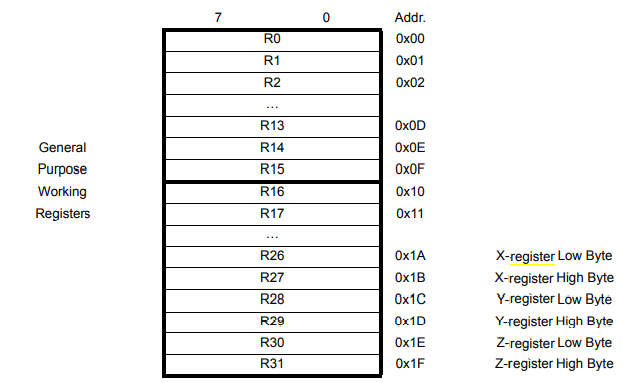


Рис.2. Регістри загального призначення [1]

Регістри швидкого доступу включають в себе 32 8-бітових робочих регістрів загального призначення, доступ до яких здійснюється за 1 такт. Це дозволяє одноциклові операції для ALU. У типовій операції ALU два операнди надходять до регістрів, операція виконується, а результат зберігається назад у регістрах або пам’яті - за один тактовий цикл. 6 з 32 регістрів можна використовувати як три 16-бітові непрямі вказівники адрес для адресації простору даних - що дозволяє розраховувати ефективні адреси. Один із цих покажчиків адрес також може бути використаний як покажчик адреси для пошуку таблиць у пам'ять флеш програми. [3]

Крім описаних вище регістрів процесор містить Instruction Pointer та Program Counter регістри, які містять адресу наступної комади, та відповідають за нормальне виконання програми. Безпосередня зміна вмісту цих регістрів заборонена.

Також, мікроконтроллер має Status and Control Register, що є регістром флагів. Схема регістра наведена нижче.

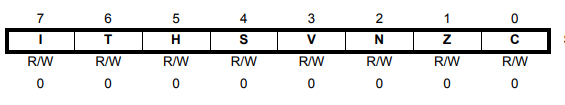


Рис.3. будова регістру флагів [1]

* Флаг І – дозвіл на глобальні переривання
* Флаг Н – половинний флаг переносу. Використовується при операціях з двійково-десятковими числами.
* Флаг S – флаг знаку.
* Флаг V – флаг переповнення для операцій у доповненому коді.
* Флаг N – флаг негативного знаку, що показує негативний результат арифметичної або логічної операції.
* Флаг Z – флаг нуля, позначає нульовий результат арифметичної або логічної операції.
* Флаг C – флаг переносу.
  + 1. ***SPI***

Послідовний периферійний інтерфейс (SPI), який застосовується для обміну даними між декількома мікроконтроллерами із швидкістю до декількох МГц (кілька мільйонів тактів у секунді). Пристрої з SPI-інтерфейсом діляться на два типи: Master та Slave. Якщо до інтерфейсу підключено кілька пристроїв то для обміну даними між ними потрібні додаткові лінії зв'язку (провідники) щоб майстру можна було вибрати Slave пристрій і зробити запит до нього. Також, по цьому інтерфейсу до мікроконтролеру підключається програматор.

* + 1. ***Analog Comparator***

Даний блок порівнює між собою два рівня сигналу і запам'ятовує результат порівняння в певному регістрі, після чого зданий результат можна проаналізувати і виконати необхідні дії. Для прикладу: можна використовувати цей блок як АЦП (Аналогово-Цифровий Перетворювач) і вимірювати напругу батареї живлення, в разі якщо якщо напруга батареї досягла низького рівня - зробити деякі дії. Також даний модуль можна застосовувати для вимірювання тривалості аналогових сигналів, зчитування встановлених режимів роботи пристрою за допомогою потенціометра і т.п.

* + 1. ***Таймери та лічильники***

Більшість мікроконтроллерів AVR, як і ця модель, мають у своєму складі кілька лічильників з розрядністю 8 або 16 біт, які можуть працювати і як таймери від внутрішнього джерела тактової частоти, і як лічильники зовнішніх подій.

Їх можна використовувати для точного формування тимчасових інтервалів, підрахунку імпульсів на виходах мікроконтролера, формування послідовності імпульсів, тактування приймача послідовного каналу зв'язку. У режимі ШІМ лічильник може являти собою широтно-імпульсний модулятор і використовуватися для генерування сигналу з програмованими частотою. Також, таймери здатні виробляти запити переривань, перемикаючи процесор на їх обслуговування з подій і звільняючи його від необхідності періодичного зчитування стану таймерів.

* + 1. ***Watchdog timer***

Сторожовий таймер (WatchDog Timer) призначений для запобігання катастрофічним наслідкам від випадкових збоїв програми. Він має свій власний RC-генератор, що працює на частоті 1 МГц.

* + 1. ***I/O Port модулі***

I/O модулі - це набір блоків портів введення/виведення до пінів яких можна підключити різноманітні датчики, виконуючі пристрої й електричні кола. Кількість пінів вхід/вихід, що йдуть від портів в мікроконтролері, може бути від 3 до 86. Вихідні драйвери в портах AVR мікроконтролера дозволяють безпосередньо підключати навантаження зі струмом 20 мА (максимум 40 мА) при напрузі живлення 5В. Загальне навантаження струму для одного порту не повинене перевищувати значення в 80 мА.

* 1. ***Типи даних мікроконтроллера***
* char - символ, тобто число, що займає один байт (тобто 8-бітове число)
* int - ціле число, що займає 2 байта (тобто 16-бітове число)
* long - довге ціле число, що займає 4 байта (тобто 32-бітове число)
* float - дійсне число, тобто число з плаваючою точкою
* double - дійсне число подвоєною точності

За допомогою додаткових службових слів визначаються різновиди типів цілих чисел:

* signed - позначає, що ціле число має знак (тобто може приймати і негативні значення)
* unsigned - позначає, що ціле число не має знака, тобто може бути тільки нулем або більше
* short - позначає, що фактичний розмір цілого числа вдвічі менше

Таким чином, виходять такі типи:

* unsigned char - число від 0 до 255
* signed char - число від -127 до 127
* unsigned int - число від 0 до 65535
* signed int - число від -32767 до 32764
* unsigned long - число від 0 до 4294967295
* signed long - число від -2147483648 до 2147483648

Варто зазначити, що так як даний мікроконтроллер має регістри розміром у 8 бітів, тобто 1 байт, то бажано використовувати найменші типи даних, які підходять для завдання, так як зберігання одного числа у різних регістрах сповілнює швидкіть виконання операцій з ним.

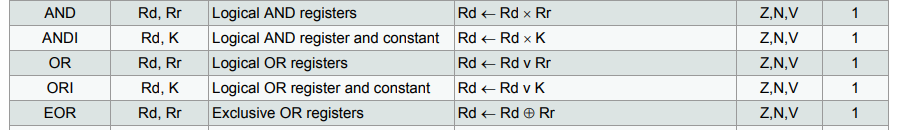
* 1. ***Система команд***

Система команд даного мікроконтроллера дуже розвинена і налічує більше 100 різних інструкцій. Більшість команд займає лише 2 байта пам'яті (16 біт).Більшість команд виконується за 1 такт. Команди можна розбити на кілька груп:

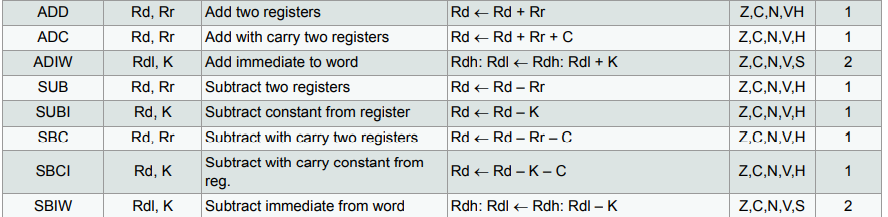
* команди логічних операцій
* команди арифметичних операцій і команди зсуву
* команди пересилання даних
* команди передачі управління

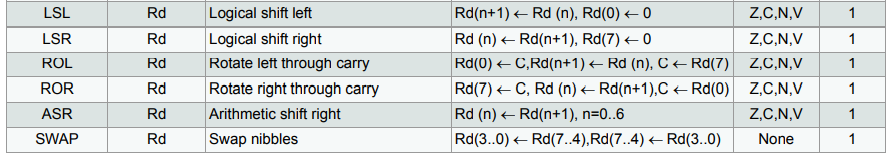
Наведемо деякі команди з кожної групи:

* + 1. ***Команди логічних операцій [2]***

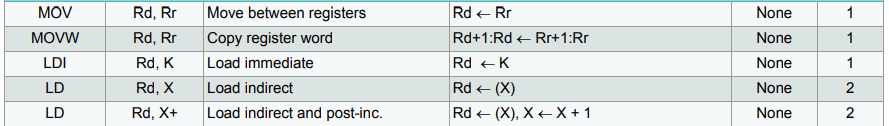


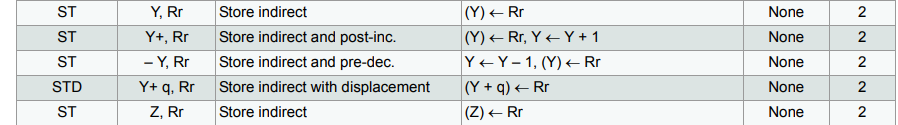
* + 1. ***Комадни арифметичних операцій та зсуву [2]***



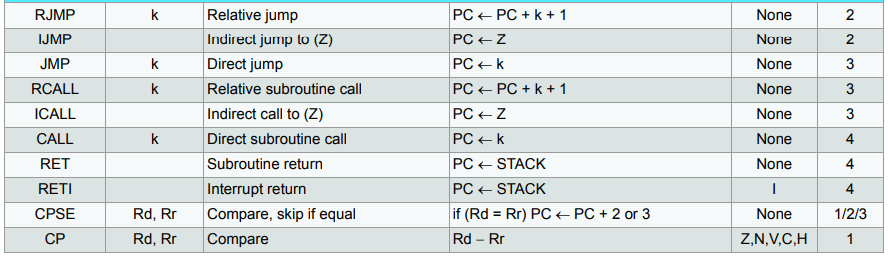


* + 1. ***Команди роботи з пам’ятю [2]***



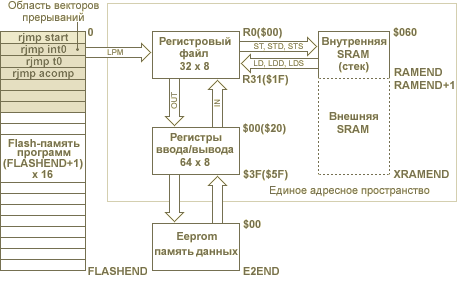


* + 1. ***Комадни управління виконанням програми [2]***



* 1. ***Програмна модель ATmega328P***

Програмна модель AVR-мікроконтролерів - це сукупність програмно доступних ресурсів мікроконтроллера. Програмну модель мікроконтроллера ATmega328P наведено на діаграмі.



Центральним блоком на цій діаграмі є регістровий файл, що складається з 32 оперативних регістрів (R0-R31) або як їх зазвичай називають регістри загального призначення. Всі ці регістри безпосередньо доступні АЛУ, та можуть використовуватися при написанні програмного забезпечення. Старші регістри об'єднані парами й утворюють три 16-розрядних регістра, призначених для непрямої адресації осередків пам'яті (AVR без SRAM мають тільки один 16-бітний регістр Z). [7]

Крім регістрів, до програмної моделі входить система АЛУ, також флеш пам’ять програм, регістри введення/виведення, EEPROM та стек.

Регістровий файл, блок регістрів введення/вивдення і оперативна пам'ять, як показано на рис, утворюють єдиний адресний простір, що дає можливість при програмуванні звертатися до 32 оперативним регістрів і до регістрів введення /виведення як до комірок пам'яті, використовуючи команди доступу до SRAM (в тому числі і з непрямою адресацією).

Всі арифметичні і логічні операції, а також частина операцій роботи з бітами виконуються в АЛУ тільки над вмістом регістрів загального призначення. Слід звернути увагу, що команди, які в якості другого операнда мають константу (SUBI, SBCI, ANDI, ORI, SBR, CBR), можуть використовувати в якості першого операнда лише регістри з другої половини регістрів (R16-R31). Команди 16-розрядного складання з константою ADIW і віднімання константи SBIW в якості першого операнда використовують тільки регістри R24, R26, R28, R30. [7]

* 1. ***Організація пам’яті***

У цьому підрозділі описано різні типи пам'яті мікроконтроллера. Архітектура AVR має дві основні адресні простори: пам’ять даних та пам’ять програми. Крім того, пристрій має пам'ять EEPROM для зберігання даних.

* + 1. ***Вбудована пам’ять програми***

ATmega328P містить вбудовану в систему флеш-пам’ять для зберігання програми розміром 32 КБ. Program Counter має ширину 14 бітів, таким чином адресуючи 16K пам’яті.

* + 1. ***SRAM пам’ять для даних***

Пристрій являє собою складний мікроконтролер з більшою кількістю периферійних блоків, ніж можна адресувати в 64КБ пам’яті, зарезервовані в для інструкцій IN і OUT. Для розширеного простору вводу-виводу від 0x60

- 0xFF у SRAM, можна використовувати лише інструкції ST / STS / STD та LD / LDS / LDD.

Доступні такі способи адресації даних [1]:

* Прямий - покриває весь адресний простір
* Непрямий зі зміщенням – база адреси знаходиться в одному з 32 регістрів.
* Непрямий – адреса має знаходитися у регістра R26 або R31.
* Непрямий з інкрементацією – адреса у регістрах інкрементується після виконання команди.
* Непрямий з декрементацією – адреса у регістрах зменшується на 1 після виконання команди.
  1. ***Адресація даних***

Мікроконтролери AVR застосовують різноманітні способи адресації даних. Можливо отримати доступ до будь-якої області пам'яті даних (регістрів загального призначення, регістрів вводу / виводу, пам'яті SRAM), а також Flash-пам'яті програм та енергонезалежної пам'яті даних EEPROM. При цьому часто до одного і того ж місця у пам’яті можна обрати різними способами, використовуючи відповідну адресу. [8]

**Пряма регістрова адресація**. При цьому способі адресації дані містяться в одному з регістрів R загального призначення, адреса якого наведена безпосередньо в команді. Прикладом команди, що використовує цей метод адресації, є команди для роботи з стеком (PUSH, POP), обміну SWAP, ряд команд арифметичних та логічних операцій.

**Пряма адресація регістру введення/виведення**. Даний вид адресації використовується для виконання обмінів між регістром введення/виведення, розташованому в адресному просторі введення/виведення та одним із регістрів загального призначення за командами IN та OUT.

**Пряма адресація пам'яті даних**. Даний спосіб адресації застосовується при зверненні до будь-якої комірки адресного простору SRAM. Є всього дві команди: LDS і STS, кожна довжиною в два слова (32 розряди). Перше слово містить код операції та адресу регістра загального призначення, друге - 16-розрядну адресу комірки пам’яті SRAM, до якої направлено звернення.

**Непряма адресація пам'яті даних**. При непрямій адресації звернення направлено до комірки пам'яті, адреса якої знаходиться в 16-розрядному індексному регістрі X, Y або Z. В ролі цих регістрів виступають пари регістрів: R26, R27 (регісрт X), R28, R29 (регісрт Y) і R30, R31 (регісрт Z).

**Непряма адресація пам'яті даних зі зміщенням**. При цьому способі адресації, комірка пам'яті визначається шляхом додавання вмісту індексного регістра Y або Z (бази) до 6-розрядного зміщення, що задається в команді. Цей спосіб адресації використовують команди LDD (пересилання байта з комірки пам'яті SRAM в регістр загального призначення) і STD (пересилання байта з регістра загального призначення в комірку SRAM).

**Непряма адресація пам'яті даних з попереднім декрементом.** При цьому способі адресації вміст індексного регістра X, Y або Z спочатку зменшується на 1, а потім проводиться звернення до пам'яті за отриманим адресою. Цей спосіб адресації використовують команди LD (пересилання байта даних з пам'яті в регістр регістр загального призначення) і ST (пересилання байта даних з регістра регістр загального призначення в пам'ять.

**Непряма адресація пам'яті даних з постінкрементом**. При цьому способі адресації вміст індексного регістра X, Y або Z спочатку використовується в якості адреси звернення до пам'яті даних, а потім збільшується на 1. Використовується тоді ж, коли й попередній тип.

* 1. ***Адресація команд***

**Непряма адресація пам'яті програм**. Мікроконтролери AVR дозволяють звернутися до осередків пам'яті програм для зчитування констант та інструкцій. Даний вид адресації використовують команди зчитування з комірки пам'яті в регістр R0 (LPM) і записи в пам'ять з регістрів R1: R0 (SPM).

**Відносна адресація пам'яті програм**. При цьому способі, адреса обчислюється шляхом складання вмісту Program Counter і константи k, що задається безпосередньо в команді. Відносну адресацію використовують команди відносного переходу (RJMP) і відносного виклику підпрограми (RCALL), численна група команд умовних переходів.

**Програмний стек та адресація при виклику підпрограм**

Стек являє собою область пам'яті, яку мікроконтроллер використовує для збереження і відновлення адрес повернення з підпрограм. Практично у всіх мікроконтролерів AVR стек розміщується в SRAM. Для адресації поточного елемента (вершини стека) використовується покажчик стека SP (Stack Pointer).

Коли мікропроцесор зустрічає одну з інструкцій викликів rcall / call / ecall / icall / eicall, то адреса наступного за ними слова в пам'яті програм апаратно копіюється в стек. У момент виходу з підпрограми по команді ret адресаповернення відновлюється з стека в програмний лічильник. У моделях з об'ємом пам'яті програм 128 і 256 к / слів для збереження PC в стеці буде потрібно 3 байта, для всіх інших - 2 байта. При збереженні кожного байта вміст SP зменшується на одиницю, а при відновленні, відповідно збільшується.

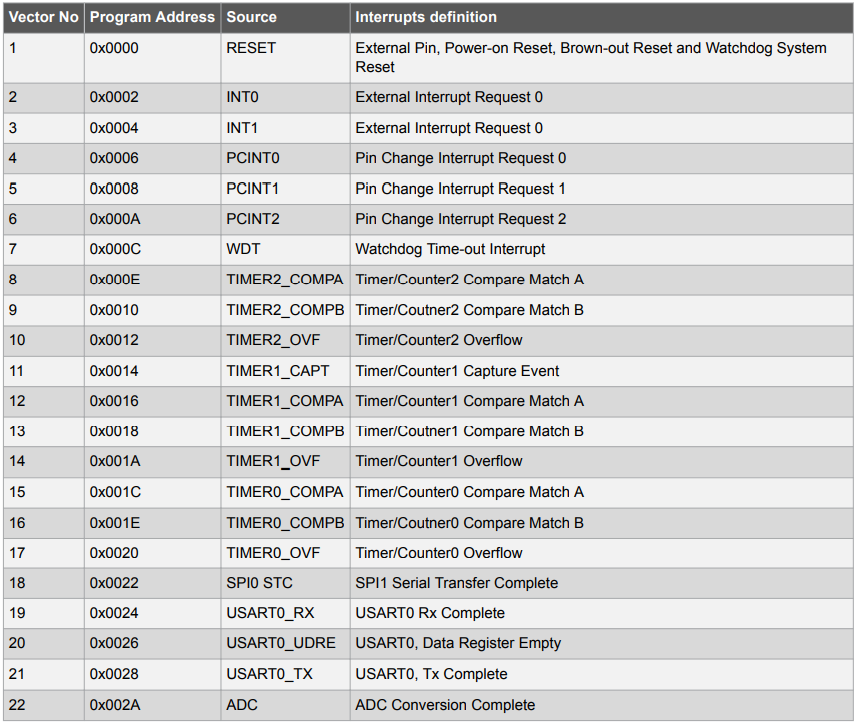
* 1. ***Система переривань та таблиця переревань***

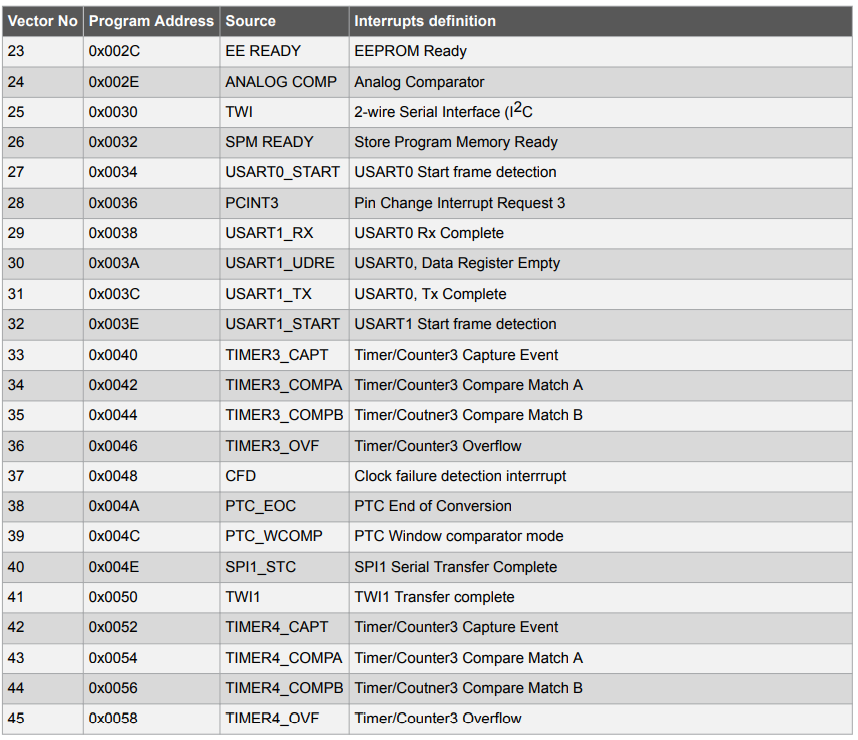
В основному є два типи переривань:

Перший тип ініціюється подією, яка встановлює прапор переривання. Для цих переривань Program Counter векторується до фактичного вектора переривань, щоб виконати процедуру обробки переривань, і обладнання очищає відповідний прапор переривання. Прапори переривань також можна очистити, записавши логічну одиницю у регістр станів на відповідну позицію прапора. [1]

Другий тип переривань буде спрацьовувати, доки існує умова переривання. Цим перериванням не обов’язково мати прапори переривань у регітрі станів. Якщо умова переривання зникає до ввімкнення переривання, то переривання не спрацьовує. Коли мікроконтроллер виходить із переривання, він завжди повертається до основної програми та виконує ще одну інструкцію до того, як наступне переривання буде виконано. [1]

Таблиця переривань для мікроконтроллера ATmega328P [1]:





* 1. ***Характеристики плати arduino nano v3 [5]***
* Мікроконтроллер - ATmega328P
* Архітектура - AVR
* Споживча потужність - 5 В
* Флеш-пам’ять - 32 KБ, 2 KБ з якої використовується бутлоадером
* SRAM - 2 KB
* Тактова частота - 16 MГц
* Аналогових IN пінів - 8
* EEPROM - 1 KБ
* DC струм на I/O пінах - 40 mA
* Вхідна напруга - 7-12 В
* Цифрові I/O піни - 22
* ШІМ(Широтно імпульсна модуляція) виходів - 6
* Споживання струму - 19 mA
* Розмір - 18 x 45 mm
* Маса - 7 g
* Код продукту - A000005
  1. ***Розпіновка плати та схеми плати***

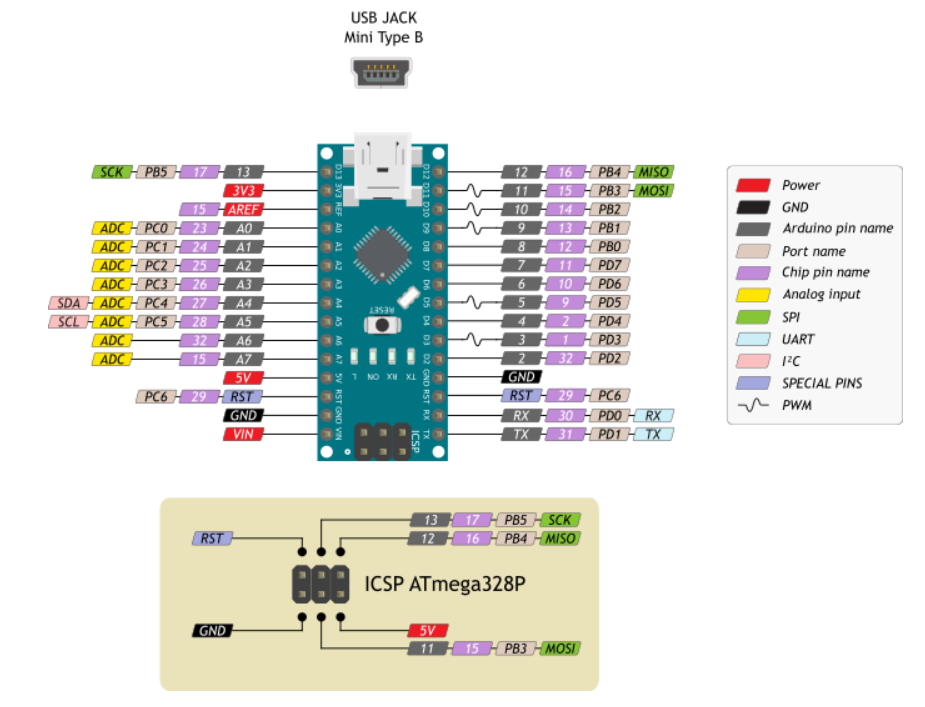


Рис. 4. Розпіновка плати

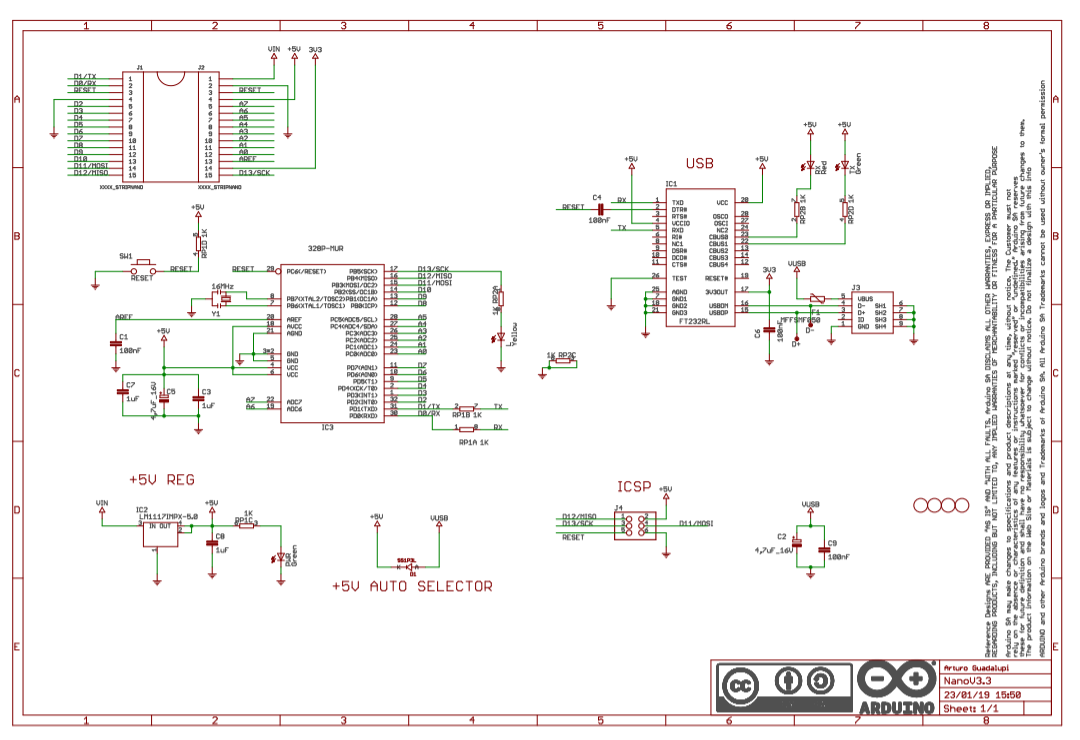
**Піни подачі живлення:**

* VIN: Вхідний пін для підключення зовнішнього джерела живлення з напругою в діапазоні від 7 до 12 В.
* 5V: Вихідний пін від регулятора напруги на платі з виходом 5 В і максимальних струмом 800 мА.
* 3.3V: Вихідний пін від стабілізатора мікросхеми FT232R з виходом 3,3 вольта і максимальних струмом 50 мА.
* AREF: Пін для підключення зовнішнього опорного напруги АЦП щодо якого відбуваються аналогові вимірювання при використанні функції analogReference() з параметром «EXTERNAL».
* GND: вихід до землі.

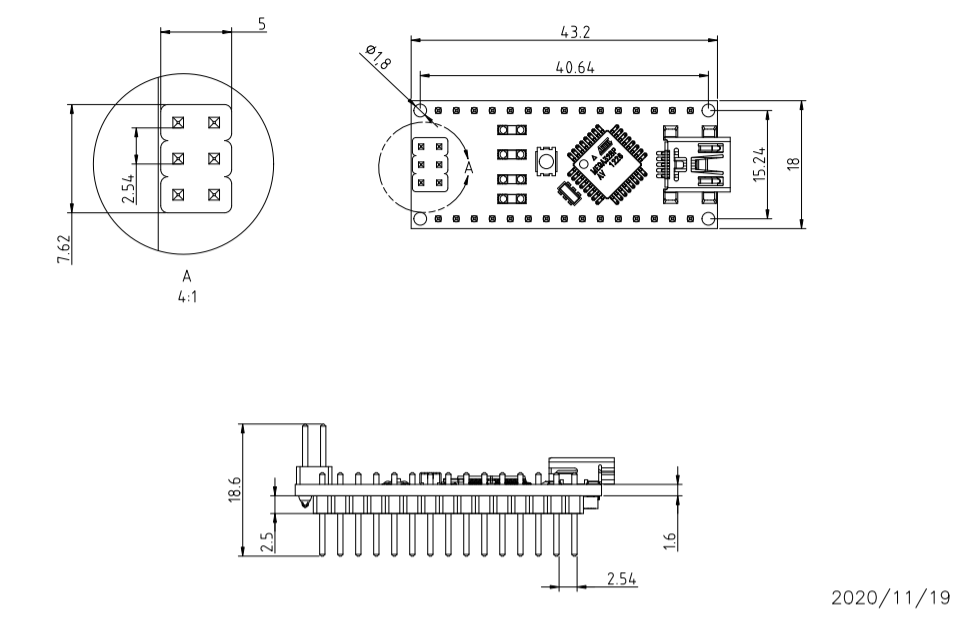
**Піни введення/виведення:**

* Цифрові входи / виходи (Піни 0-13): Логічний рівень одиниці - 5 В, нуля - 0 В. Максимальний струм виходу - 40 мА. До контактів підключені резистори, які за замовчуванням вимкнені, але можуть бути включені програмно.
* ШІМ (Піни 3,5,6,9,10 і 11): Дозволяють виводити аналогові значення в вигляді ШІМ-сигналу. Розрядність ШІМ не змінюється і встановлена в 8 біт.
* АЦП (Піни A0-A7): Дозволяють представити аналогову напругу в цифровому вигляді. Розрядність АЦП не змінюється і встановлена в 10 біт. Діапазон вхідної напруги від 0 до 5 В.
* TWI/I²C (піни A4(SDA) і A5(SCL)): Для спілкування з переферією по інтерфейсу I²C.
* SPI (піни 11 (MOSI), 12 (MISO), 13 (SCK) і 10 (SS)): Для спілкування з периферією по інтерфейсу SPI.
* UART (Піни 0 (RX) і 1 (TX)): Використовується для комунікації плати Arduino з комп'ютером або іншими пристроями по послідовному інтерфейсу.

Схема [4]:



Монтажна схема та розміри [4]:



1. **БУДОВА ТА ХАРАКТЕРИСТИКИ ДОПОМІЖНИХ ЕЛЕКТРОННИХ ПРИСТРОЇВ**

У цьому розділі будуть стисло наведені характеристики, будова та принципи роботи інших електронних пристроїв, що будуть використані під час розробки автоматичного безконтактного дозатора антисептика для рук.

* 1. ***Ультразвуковий далекомір HC-SR04***

Ультразвуковий далекомір розрахований на визначення відстані до об'єктів в радіусі чотирьох метрів. Робота модуля заснована на принципі ехолокації. Модуль посилає ультразвуковий сигнал і приймає його відображення від об'єкта. Вимірявши час між відправленням і отриманням імпульсу, неважко вирахувати відстань до перешкоди. [10]

Модуль підключається чотирма проводами. Контакти VCC і GND служать для підключення живлення, а Trig і Echo - для відправки і прийому сигналів далекоміра.

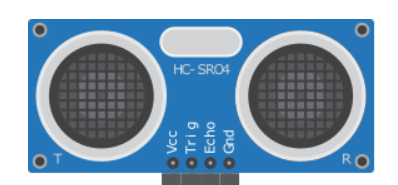


Рис.5. Розпіновка далекоміра

Для того щоб ініціювати відправку сигналу далекоміром, необхідно подати високий сигнал тривалістю 10 μs на пін Trig. Після отримання високого сигналу тривалістю 10 μs на пін Trig, модуль генерує пучок з восьми сигналів частотою 40 кГц і встановлює високий рівень на піні Echo. Після отримання відбитого сигналу модуль встановлює на піні Echo має низький рівень. Знаючи тривалість високого сигналу на піні Echo можемо знайти відстань, помноживши час, який витратив звуковий імпульс перш ніж повернувся до модуля, на швидкість поширення звуку в повітрі (340 м / с).

**Характеристики [10]:**

* Напруга живлення - 5 В
* Споживання в режимі тиші - 2 мА
* Споживання при роботі - 15 мА
* Діапазон відстаней - 2-400 см
* Ефективний кут спостереження - 15 °
* Робочий кут спостереження - 30 °

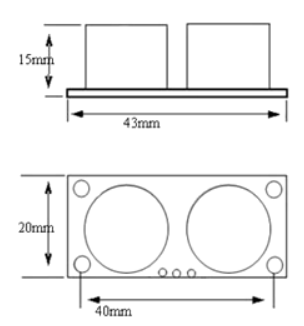
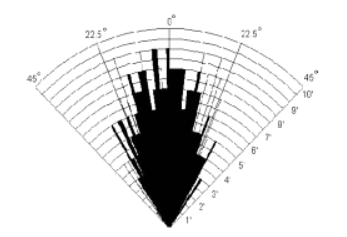
 

Рис. 6. Габарити та діаграма сигналу

* 1. ***DC сервопривод SG90***

SG90 - це аналоговий серводвигун з обертовим моментом 1.8 кг / см і діапазоном обертання 180°. Серводвигун вважається аналоговим, так як кут положення вала визначається кутом повороту потенціометра (змінного резистора). Незамінний для проектів, в яких потрібно управляти положенням вала за допомогою двигуна. Але не просто управляти, а ставити точний кут положення вала. [9]

Усередині корпусу знаходиться невеликий модуль управління, який під дією вхідного сигналу подає живлення відповідної полярності на електродвигун. Вхідний сигнал управління містить дані про необхідне положення вала. Для визначення поточного положення вала редуктор з'єднаний з двигуном змінного резистора. Електроніка Tower Pro SG90 обчислює різницю між поточним становищем редуктора і необхідним. Модуль управління орієнтуючись на опір змінного резистора подає живлення необхідної полярності на двигун для повороту редуктора приводить у відповідність положення тому, що було передане сигналом управніння. [9]



Рис.7 Сервопривод

Підключення:

* Коричневий провід - до землі (GND)
* Червоний провід - 5V
* Помаранчевий провід - до цифрового піна Arduino

Керування:

* 1 мс Позиція «-90» - рухається наліво
* 1,5 мс Позиція «0» - Постає в центр
* 2 мс Позиція «90» - рухається вправо

Характеристики:

* Діапазон обертання: 180°
* Напруга живлення: 4,8-5 В
* Обертовий моменг момент: 1.8 кг/см (при 4.8 В)
* Швидкість обертання: 60 ° за 0.1 сек (при 4.8 В)
* Внутрішній інтерфейс: аналоговий
* Габарити: 22,2 x 11,8 x 31 мм
* Матеріал зубчастих коліс: нейлон

1. **ПРОЄКТУВАННЯ ПРИСТРОЮ**
   1. ***Ідея роботи***

Необхідно розробити механізм, який буде створювати силу для натискання виходу у верхній частині дозатора вниз і дозування рідини, оскільки ми використовуємо сервомотор, який забезпечує круговий рух, щоб створити тиск на витік баночки, можна, наприклад, використати дріт. Якщо закріпити один кінець дроту, середину прикріпити до верхньої частини дозатора, а інший кінець – до вала сервопривода, то при виконанні обертового руху сервоприводом, дріт буде створювати тиск на кнопку дозатора, і рідина буде виливатися. Ідею роботи ілюстровано на ескізі.

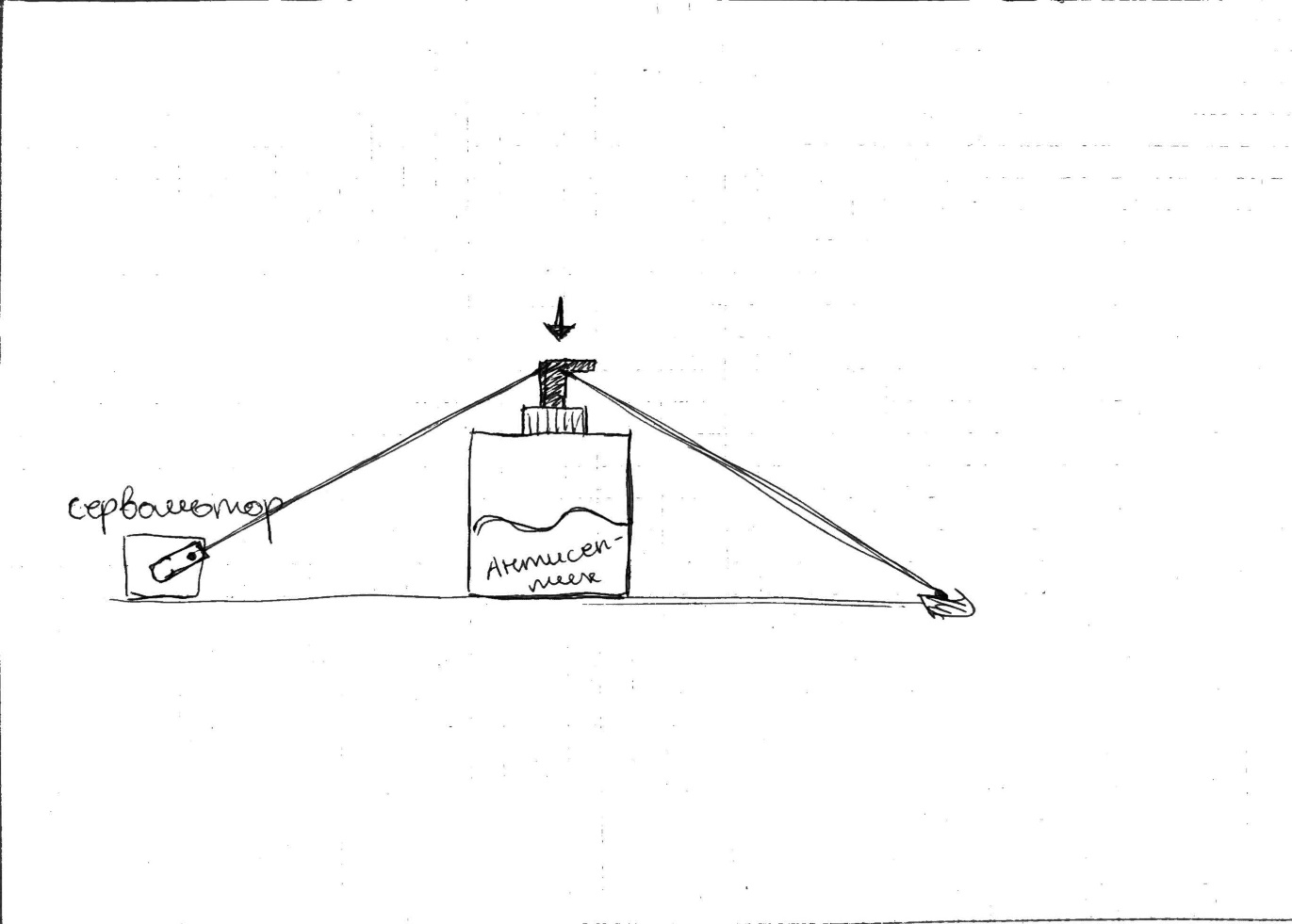


Рис. 8. Ескіз роботи пристрою.

Тепер потрібно спроєктувати механізм безконтактного включення сервопривода. Для цього і знадобиться далекомір та сам мікроконтроллер. Система дуже проста: потрібно під’єднати далекомір та сервомотор до мікроконтроллера та запрограмувати його таким чином, що коли далекомір помічає якийсь предмет на відстані, яка нам підходить (візьмемо десь 15 сантиметрів), то він вмикає сервомотор, що і здійснює розбризкання антисептичної рідини.

* 1. ***Під’єднання елементів***

Під’єднуємо пристрої до мікроконтроллера враховуючи їх розпіновку, що була наведена в попередньому розділі.

**Далекомір:**

* Trig до D10
* Echo до D11
* Vcc до Vcc
* Gnd до Gnd

**Сервомотор:**

* Signal до D9
* Vcc до Vin
* Gnd до Gnd

З’єднання ілюстровано на схемі.

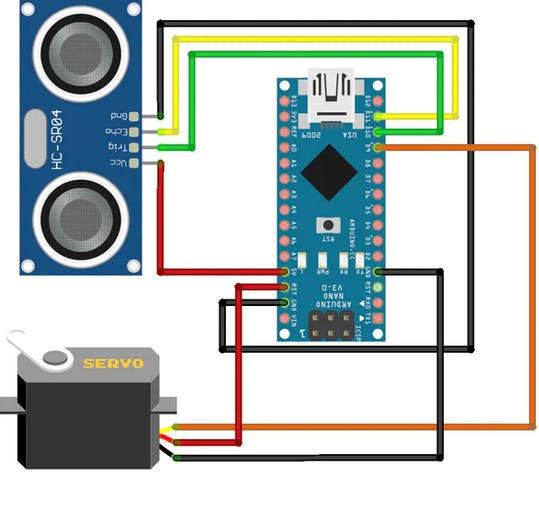


Рис. 9. Схема з’єднань пристроїв з мікроконтроллером.

* 1. ***Програмування мікроконтроллера***

Для програмування мікроконтроллера Arduino nano v3 будемо використовувати програмне забезпечення Arduino IDE, яке пропонують завантажити з офіційного сайту Arduino.

Скористаємося готовою бібліотекою для вза’ємодії з сервомотором.

#include <Servo.h>

const int echo = 8;

const int trigger = 7;

const int servo = 9;

int distance;

long duration;

int pos;

Servo myservo;

// налаштовуємо піни

void setup()

{

pinMode(trigger,OUTPUT);

pinMode(echo,INPUT);

pinMode(trigger,OUTPUT);

myservo.attach(servo);

myservo.write(0);

Serial.begin(9600);

}

// функція, яка буде викликатися для виприскування рідини

void Sanitizer(){

myservo.write(45);

delay(100);

myservo.write(90);

delay(100);

myservo.write(135);

delay(100);

myservo.write(120);

delay(1000);

myservo.write(00);

delay(3000);

}

void loop()

{// розраховуємо відстань до предмета

digitalWrite(trigger,LOW);

delay(500);

digitalWrite(trigger,HIGH);

delay(100);

digitalWrite(trigger,LOW);

duration = pulseIn(echo, HIGH);

distance = duration\*0.034/2;

if(distance<15)

{

Sanitizer();

}

}

Завантажуємо написаний код в пам’ять мікроконтроллера за допомогою Arduino IDE через USB інтерфейс. Пристрій готовий до використання.

1. **ВИСНОВКИ**

У ході даної розрахунково-графічної роботи мною було вивчено будову мікроконтроллерів архітектури AVR, зокрема, ATmega328P. Також, було проаналізовано та детально описано програмну модель даного мікрокнтроллера, систему його команд, будову пам’яті, способи адресація даних та команд й систему переривань.

Після вивчення функціонування мікроконтроллера, був спроєктований та побудований пристрій, що за допомогою ультразвукового дальноміра помічає об’єкт перед собою (руки людини), та користуючить серводвигуном, розприскує на руки антисептичну рідину.

1. **СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ**
2. Atmel corp. Arduino ATmega328P datasheet, 1600 Technology Drive, San Jose, CA 95110 USA, 2015 – 294 сторінки.
3. Atmel corp. AVR Instruction Set Manual, 1600 Technology Drive, San Jose, CA 95110 USA, 2016 – 191 сторінка.
4. Microchip technology inc. ATmega328PB datasheet, 2355 West Chandler Blvd. Chandler, USA, 2017 – 406 сторінок.
5. Arduino nano documentation [Електронний ресурс], Режим доступу: https://store.arduino.cc/arduino-nano
6. Arduino nano characteristics [Електронний ресурс], Режим доступу: https://store.arduino.cc/arduino-nano
7. Устройство микроконтроллеров AVR [Електронний ресурс], Режим доступу: https://myrobot.ru/stepbystep/mc\_architecture.php
8. Програмная модель микроконтроллеров AVR [Електронний ресурс], Режим доступу: https://studfile.net/preview/9023199/page:2/
9. Режимы адресации памяти программ и данных [Електронний ресурс], Режим доступу: http://www.gaw.ru/html.cgi/txt/doc/micros/avr/arh/mega103\_7.htm
10. Сервопривод, Вікіпедія [Електронний ресурс], Режим доступу: https://uk.wikipedia.org/wiki/%D0%A1%D0%B5%D1%80%D0%B2%D0%BE%D0%BF%D1%80%D0%B8%D0%B2%D0%BE%D0%B4
11. Ultrasonic Ranging Module HC - SR04 datasheet, [Електронний ресурс], Режим доступу: https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf