

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ГОРЯ СІКОРСЬКОГО»
МЕХАНІКО-МАШИНОБУДІВНИЙ ІНСТИТУТ
КАФЕДРА КОНСТРУЮВАННЯ ВЕРСТАТІВ ТА МАШИН

«На правах рукопису»

УДК _ 621.865.8 _

До захисту допущено

Завідувач кафедри

_____ В.Б.Струтинський

(підпис) (ініціали, прізвище)

“ _____ ” _____ 2018 р.

Магістерська дисертація
на здобуття ступеня магістра
зі спеціальності 133 Галузеве машинобудування

на тему Цифрова система керування пневматичного промислового робота

Виконав (-ла): студент (-ка) _____ ІІ курсу , групи МВ – 71мп

_____ Прянічніков Артем Дмитрович

(прізвище ім'я по батькові)

(підпис)

Науковий керівник _____ Вакуленко С.В., ст. викл.

(посада, науковий ступінь тачене звання, прізвище, ініціали)

(підпис)

Консультант з розділу _____

(посада, науковий ступінь тачене звання, прізвище, ініціали)

(підпис)

Рецензент:

_____ (посада, наукова ступінь, вчене звання, прізвище та ініціали)

(підпис)

Засвідчую, що у цій магістерській дисертації
немає запозичень з праць інших авторів без
відповідних посилань.

Студент _____
(підпис)

Київ - 2018

розрахунок пневматичного приводу руху руки 5 Модеренізація системи керування 6 Патентний пошук багатопозиційного приводу 7Модеренізація конструкції робота 8 Модеренізація схеми цифрової системи керування

6. Орієнтовний перелік графічного (ілюстративного) матеріалу _____

Схема цифрового приводу. Патентний пошук. Схеми алгоритму цифрової системи керування. Пневматична схема. Принципова схема керування

6. Консультанти розділів проекту __

7. Орієнтовний перелік публікацій 1. Vakulenko S., Prianychnykov A. The mathematical models creation based on state-space representation of the carriage elastic system with installed tool holder with oriented center of rigidity // Zbiór artykułów naukowych. Konferencji Międzynarodowej Naukowo-Praktycznej "Inżynieria i technologia. Badania podstawowe i stosowane: wyzwania i wyniki" (30.05.2017 - 31.05.2017) Z 40 - Warszawa: Wydawca: Sp. z o.o. «Diamond trading tour», 2017. - 92 str. - p. 34-37. ISBN: 978-83-65608-61-1. Сборник внесен в базу РИИЦ.

2. Vakulenko S., Prianychnykov A. Developing a discrete-time control system of the industrial robot «BRIG 10 ZAZ» // Zbiór artykułów naukowych. Konferencji Międzynarodowej Naukowo-Praktycznej "Science, Research, Development Technics and Technology. #5" (30.05.2018 - 31.05.2018) Z 40 - Warszawa: Wydawca: Sp. z o.o. «Diamond trading tour», 2018. - 112 str. - p. 60-65. ISBN: 978-83-66030-30-5. Сборник внесен в базу РИИЦ.

3. Vakulenko S., Prianychnykov A. Development of algorithm and software of an industrial robot control system // Zbiór artykułów naukowych. Konferencji Międzynarodowej Naukowo-Praktycznej "Science, Research, Development Technics and Technology. #5" (30.05.2018 - 31.05.2018) Z 40 - Warszawa: Wydawca: Sp. z o.o. «Diamond trading tour», 2018. - 112 str. - p. 65-68. ISBN: 978-83-66030-30-5. Сборник внесен в базу РИИЦ.

8. Консультанти розділів проекту

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

9. Дата видачі завдання 10.10.2018

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання дипломного проекту	Строк виконання етапів проекту	Примітка
1	Технічні характеристики, огляд конструкції та призначення промислового робота	10.10.2018- 20.10.2018-	виконано
2	Обґрунтування вимог до дискретної системи керування промислового робота	10.10.2018- 20.10.2018-	виконано
3	Розробка програмного середовища системи керування робота	20.10.2018- 10.11.2018-	виконано
4	Проектування та розрахунок пневматичного приводу руху руки	20.10.2018- 10.11.2018-	виконано
5	Модернізація системи керування	10.11.2018- 5.12.2018-	виконано
6	Патентний пошук багатопозиційного привода	10.11.2018- 5.12.2018-	виконано
7	Модернізація конструкції робота	10.11.2018- 5.12.2018-	виконано
8	Модернізація схеми цифрової системи керування	10.11.2018- 5.12.2018-	виконано

Студент _____ Прянічников А.Д.
(підпис) (прізвище та ініціали)

Науковий керівник дисертації _____ Вакуленко С.В.

РЕФЕРАТ

Магістерську дисертацію виконано на 80 аркушах, вона містить 3 додатка та перелік посилань на використані джерела з 2 найменувань. У роботі наведено 10 рисунків та 3 таблиці.

Метою даної магістерської дисертації є створення цифрової системи керування

промислового робота БРИГ-10 ЗАЗ та її модернізація.

У роботі проведено аналіз існуючих рішень указаної задачі а саме, існуюча аналогова система керування призначення для заданої моделі промислового робота. Виконано розрахунок основний керуючий механізмів робота, а саме – схвату. Було запропоновано для використання цифровий багатопозиційний дискретний привід. Для розв'язання задачі в роботі обрано існуючу апаратну платформу на базі її розроблено алгоритм роботи програмного забезпечення, принципову електричну схему та написано код програми.

Ключові слова: робот, система керування, алгоритм програмного забезпечення, програмний код, апаратна платформа.

ABSTRACT

The master's dissertation is executed on 80 sheets, it contains 3 appendices and a list of references on the used sources from 2 names. In the work there are 10 figures and 3 tables.

The purpose of this master's thesis is to create a digital control system BRIG-10 ZAZ industrial robot and its modernization.

The paper analyzes the existing solutions of this problem, namely, the existing analogue destination management system for a given model of industrial robot. Calculation of the main manager of the mechanisms of work, namely - the grip. It was suggested to use a digital multipoint discrete drive. To solve the problem, an existing hardware platform has been selected. An algorithm for software work, a basic electrical circuit and a program code are written on the basis of it.

Keywords: robot, control system, software algorithm, software code, hardware platform.

РЕФЕРАТ

Магистерскую диссертацию выполнено на 80 листах, содержит 3 приложения и список ссылок на использованные источники из 2 наименований. В работе приведены 10 рисунков и 3 таблицы.

Целью данной магистерской диссертации является создание цифровой системы управления промышленного робота БРИГ-10 ЗАЗ и его модернизация.

В работе проведен анализ существующих решений указанной задачи а именно, существующая аналоговая система управления назначение для заданной модели промышленного робота. Выполнен расчет основной управляющий работа механизмов, а именно – схват. Было предложено для использования цифровой дискретный многопозиционный привод. Для решения задачи в работе выбрана существующая аппаратная платформа на базе ее разработан алгоритм работы программного обеспечения, принципиальная электрическая схема и написан код программы.

Ключевые слова: робот, система управления, алгоритм программного обеспечения, программный код, аппаратная платформа.

Зміст

Вступ.....	5
1 Технічні характеристики, огляд конструкції та призначення промислового робота. 7	
1.1 Вивчення пневматичної системи робота, елементів приводу рухомих органів... 10	
1.2 Вивчення особливостей будови та роботи аналогової системи керування..... 12	
1.3 Теоретичний аналіз систем керування промислових роботів..... 15	
2 Обґрунтування вимог до дискретної системи керування промислового робота.. 18	
2.1 Огляд існуючих апаратних платформ для створення систем керування..... 19	
2.2 Розробка електричної схеми прототипа дискретної системи керування..... 21	
3 Розробка програмного середовища системи керування робота..... 25	
3.1 Обґрунтування основних вимог до головних експлуатаційних особливостей роботи циклової системи керування..... 25	
3.2 Розробка алгоритма роботи програмного забезпечення системи керування... 26	
3.3 Опис стандартних бібліотек, що використовуються в програмі..... 29	
3.4 Реалізація програмного забезпечення, опис її функціональних можливостей та узагальнення можливих напрямків її вдосконалення..... 30	
4 Проектування та розрахунок пневматичного приводу руху руки..... 36	
4.1 Проектування та розрахунок схвату робота..... 37	
5 Модернізація системи керування..... 41	
5.1 Технічний опис завдання..... 41	
5.2 Модернізація керуючої плати..... 42	
6 Патентний пошук багатопозиційного приводу..... 44	
7 Модернізація конструкції робота..... 46	
7.1 Опис конструкції багатопозиційного приводу..... 46	
8 Модернізація схеми цифрової системи керування..... 55	
9 ВИСНОВКИ..... 66	
10 ПЕРЕЛІК ПОСИЛАНЬ..... 67	
ДОДАТОК А..... 68	
ДОДАТОК Б..... 70	
ДОДАТОК В..... 71	
ДОДАТОК Г..... 72	

ВСТУП

Одним з основних завдань конструювання верстатів є забезпечення комплексної автоматизації виробничих процесів з метою створення високоавтоматизованих цехів і заводів. Рішення цієї задачі пов'язане з рішенням цілого ряду складних завдань, без яких неможлива робота верстата в умовах автоматизованого виробництва. Неможливо уявити собі не лише завод-автомат, але навіть автоматизована виробнича ділянка, без участі промислових роботів.

Промисловий робот - це автоматична машина, стаціонарна або пересувна, яка складається з виконавчого механізму у вигляді маніпулятора з декількома ступенями свободи і програмного управління, яке дає можливість перепрограмуватися для виконання у виробничих процесах функцій переміщення і управління.

Існує класифікація роботів за різними параметрами, наприклад по кількості координат, типу установки, кількості рук, вантажопідйомності і тд.

Одним з найбільш важливих чинників є тип системи керування або так звана програма. Промислові роботи за типом програми можна поділити на такі категорії: жорсткопрограмовані, гнучкопрограмовані, адаптивні та інтелектуальні. По конструкції можна виділити 3 базові типи маніпуляторів : портальний, вмонтований і підлоговий.

Мета даного дипломного проекту – розробити систему керування робота "БРИГ 10-ЗА3" на базі апаратної-обчислювальної платформи Arduino. Для досягнення цієї мети в роботі необхідно вирішити такі завдання:

1. Детально вивчити конструкцію робота(електронний і паперовий варіанти).
2. Детально вивчити апаратну платформу Arduino, всі її максимальні можливості та обмеження.
3. Зібрати макетну плату з можливістю підключення до робота.
4. Розробити алгоритм роботи програмної частини, а також взаємодію з користувачем.

1. ТЕХНІЧНІ ХАРАКТЕРИСТИКИ, ОГЛЯД КОНСТРУКЦІЇ ТА ПРИЗНАЧЕННЯ ПРОМИСЛОВОГО РОБОТА.

Робот промисловий "БРИГ-10-ЗА3" призначений для автоматизації і механізації допоміжних технологічних операцій : завантаження, вивантаження, установки, зняття деталей і заготовок з обслуговуваного технологічного устаткування при механообробці, пересуванні і так далі, в цехових умовах при температурі довкілля від $+10^{\circ}$ до $+40^{\circ}$ С і відносній вологості 80% при $+20^{\circ}$ С.

ТЕХНІЧНА ХАРАКТЕРИСТИКА РОБОТА.

№ п\п	Найменування	Одиниця виміру	Параметри
2.1	Кількість степенів свободи (без привода схвата)	шт.	5
2.2	Кількість точок зупину		
	При повороті руки	—	3
	При всіх останніх переміщеннях	—	2
2.3	Вантажопідйомність, що складається з		10
	Приводу схвата	кг	7
	Приводу схвата и	кг	5
	механізму повороту кисті	кг	3
	Приводу схвата та повзуна	кг	

	Приводу схвата, механізму повороту кисті и повзуна		
2.4	Точність позиціонування	мм	± 0,3
2.5	Лінійне переміщення руки: По вертикалі По горизонталі	мм мм	0...100 510...600
2.6	Поперечний зсув кисті щодо руки	мм	0...70
2.7	Межа установочного вертикального переміщення руки при ході пневмоциліндра механізму підйому: 100мм 0мм	мм мм	230 330
2.8	Висота осі руки від підлоги	мм	850...1180
2.9	Найбільший кут повороту руки навколо вертикальної осі	град.	210
2.10	Найбільший кут повороту кисті	град.	180
2.11	Найбільша лінійна швидкість Переміщення руки по вертикалі по горизонталі	м/сек м/сек	0,3 0,6

2.12	Найбільша швидкість повороту руки	град/сек	90
2.13	Зусилля, що розвивається приводом схвата	Н	900
2.14	Тип приводу	Пневматичний	
2.15	Тиск повітря	МПа	0,4
2.16	Витрата повітря	м ³ /мин	0,2
2.17	Система управління	Циклова, електронна	
2.18	Елементна база системи управління	інтегральні мікросхеми	
2.19	Програмоносій	Штекерна панель	
2.20	Напруга	В	220
2.21	споживана потужність	кВт	0,3
2.22	Габаритні розміри:		
	Довжина	мм	1530
	Ширина	мм	690
	Висота	мм	975
2.23	Маса	кг	340
2.24	Монтажні розміри	-	-

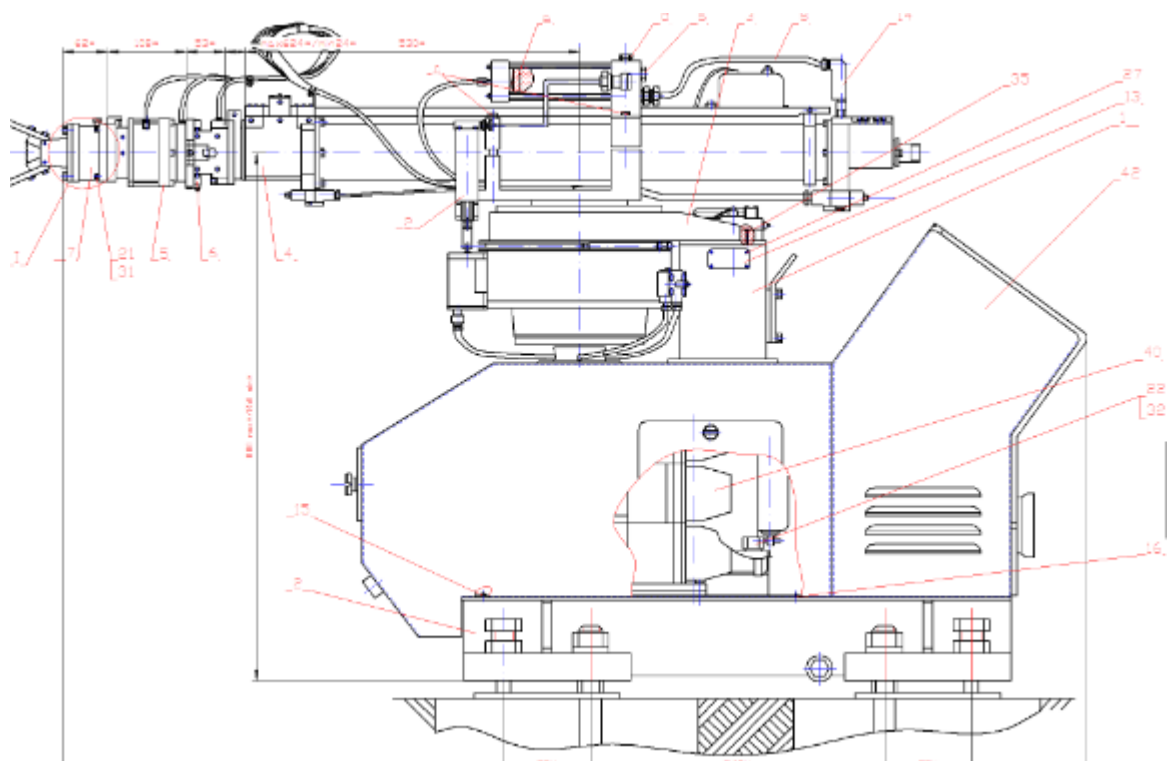


Рисунок 1.1 - Загальний вигляд робота

Робот промисловий "БРИГ-10-ЗАЗ" (див. рисунок 1.1) складається з механізму підйому, встановленого на основі, і повороту, що несе механізм з рукою.

До руки кріпиться кисть, що складається з механізму повороту кисті, повзуна і приводу зхвату. механізм повороту кисті призначений для повороту встановлюваної деталі на заданий кут, необхідний для правильної орієнтації деталі при виготовленні, повзун - для поперечних зрушень деталі в горизонтальній або вертикальній площині при укладанні деталі в пристосування. Привід зхвату служить для приведення в дію захватного пристрою (кліщів, цанг і так далі) для закріплення деталі при транспортуванні в певному положенні. Усі виконавчі органи робота приводяться до руху від пневмоциліндрів або пневмодвигунів, що спрацьовують в заданій послідовності від пневморозподільників, керованих електромагнітами по командах

1.1 ВИВЧЕННЯ ПНЕВМАТИЧНОЇ СИСТЕМИ РОБОТА, ЕЛЕМЕНТІВ ПРИВОДУ РУХОМИХ ОРГАНІВ

Пневмоустаткування призначене для підведення підготовленого стислого повітря до пневмоциліндрів та пневмодвігунів механізмів робота, реулювання його витрати і зміни каналів подання стислого повітря і скидання за заданою програмою.

Пневмоустаткування складається з вузлів згідно таблиці 1; пркріплених до основи за допомогою болтів, рукавів для зв'язку вузлів між собою і ресівером, і повітропроводів з поліетиленових труб круглого перерізу з умовними проходами для підключення руки, для підключення механізмів підйому і повороту, для підключення інших механізмів до пневмоустаткування.

Пневмогідралічна схема робота яка зображена на плакаті забезпечує нормальну роботу пневмоустаткування робота за будь-якою заданою програмою згідно з об'ємом пам'яті програмоносія.

Стисле повітря з цехової пневмосистеми подається до фільтру-вологівідділювачу ФВД блоку підготовки стислого повітря (БПСВ) і пройшовши редукування на пневморозподільнику МР через реле тиску РД подається на пневморозподільник Р8, слугуючий вимикачем пневможивлення робота.

Реле тиску РД служить для відключення електроживлення робота при падінні тиску стислого повітря в цеховій магістралі нижче заданого мінімального значення (0,36 МПа), на яке його налаштовують. При включенні електромагніту пневморозподільника Р8 стисле підготовлене повітря подається до пневморозподільників Р1...Р7, керуючих механізмами робота згідно з вигравійованими написами в лапках на плиті (плитах) блоку (блоків) розподільників повітря БВР (БВР1 і БВР2).

Для регулювання швидкості переміщення штока руки РК слугують дроселі ДР1 і ДР2, для регулювання швидкості механізму повороту МПВ - дроселі ДР3 і ДР4, для регулювання швидкості механізму підйому МПД - дросель ДГ4.1, опускання - дросель БВР.4, для регулювання швидкості повороту кисті МПК - дроселі БВР.2 і БВР.3 . БВР.1.БВР.4 які є елементами блоку (блоків) розподільників повітря, дросель ДГ4.1 - елементом пневмоциліндра ПЦ механізму підйому МПД.

Повітря з неробочих порожнин пневмоциліндрів і пневмодвигунів механізмів робота йде по каналах пневморозподільників Р1.Р7 і блоку (блоків) розподільників повітря БВР (БВР1 і БВР2) через глушник (глушники) Г (Г1 і Г2).

Для підживлення стислим повітрям пневмосистеми в моменти його найбільшої витрати служить ресівер РС, що знаходиться на підставі ОСН.

Рухомими приводами руху робота є рука, механізм повороту кисті, повзун, схват та механізм підйому.

Усі виконавчі органи робота приводяться до руху від пневмоциліндрів або пневмодвигунів, що спрацьовують в заданій послідовності від пневморозподільників, керованих електромагнітами по командах, запрограмованих на штекерній панелі контроллера, розташованого в пульті управління.

1.2 ВИВЧЕННЯ ОСОБЛИВОСТЕЙ БУДОВИ ТА РОБОТИ АНАЛОГОВОЇ СИСТЕМИ КЕРУВАННЯ.

Задана модель робота БРИГ-10 ЗАЗ має циклову, електронну систему управління. Система управління робота складається з керуючого контролера "УПКОН-ПР-01", змонтованого в пульті управління, що кріпиться на основі за пневмоустаткуванням, блоку датчика положення, набору затисків малогабаритних і сполучних дротів різних перерізів, поміщених в запобіжні трубки, сполучаючих контролер з блоком датчика положення, пневмоустаткуванням і зовнішнім живленням через набори затисків. Живлення електроустаткування здійснюється від мережі змінного струму і частотою 50 ± 1 Гц. Аналогова система управління забезпечує в ручному або автоматичному режимах управління усіма механізмами робота, установку і зняття проміжного упору механізмів повороту, зупинку зчитування програми, при пониженні тиску повітря в пневмосистемі нижче 0,36 Мпа – зупинку програми. Контролер "УПКОН-ПР-01" призначений для видачі команд на виконавчі органи робота і зовнішнього устаткування.

Конструктивно контролер виконаний у вигляді вставного приладу, який своїми напрямними вставляється в касету, розташовану в пульті управління.

Контролер побудований за принципом програмного автомата з жорстким циклом управління.

Технічна характеристика аналогової системи на базі контролера "УПКОН-ПР-01":

Кількість програмованих команд.....	193
у тому числі:	
управління рухами робота.....	84
управління зовнішнім устаткуванням.....	60
запитів стану датчиків.....	48
допоміжних (організація неповного циклу).....	1
Кількість кроків програми.....	178

напруга змінного струму частотою 50Гц, В.....110
 струм, А не більше.....2
 Час безперервної роботи, год.8
 Алгоритм роботи системи приведений на Рисунку 1

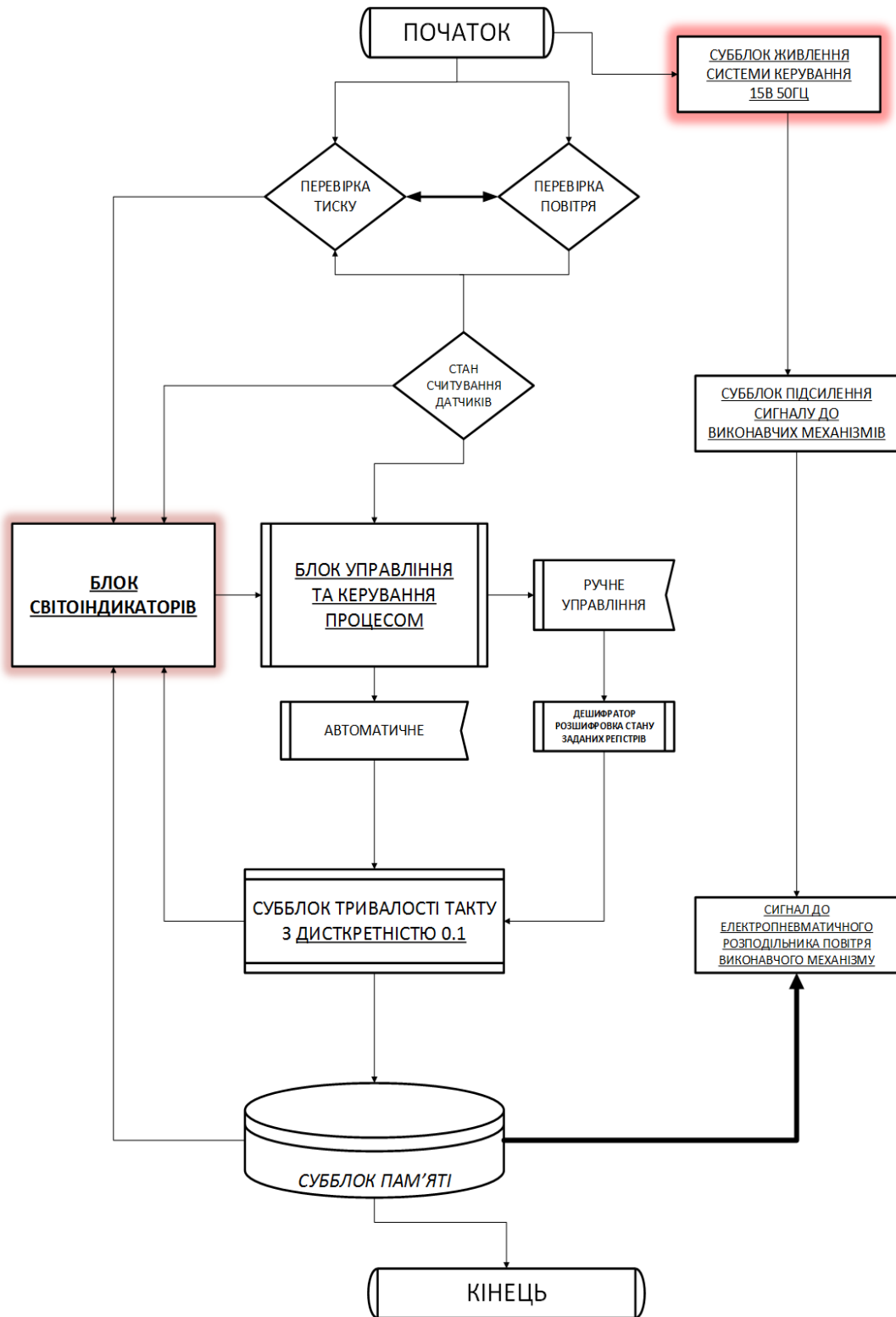


Рисунок1.1 - Алгоритм аналогової системи керування

В склад контролера "УПКОН-ПР-01" входять наступні пристрої:

а) субблок живлення, б) субблок управління, в) субблоки дешифратора, г) субблоки вихідної пам'яті, е) блок світлоіндикаторів, ж) субблок вихідних підсилювачів .

а) субблок живлення виробляє напругу $15 \pm 5\%$ В для живлення мікросхем і пульсуючу напругу 15В для запуску формувача, встановленого в субблоку управління. Крім того, в нім встановлений симістор , через який живиться електромагніт розподільника повітря пневможивлення системи.

б) субблок управління призначений для управління зчитування програми і вибору режиму роботи робота. Необхідна тривалість такту задається за допомогою перемикача "ТРИВАЛІСТЬ", встановленого на передній панелі контролера. З виходу таймера за відсутності сигналу заборони на вході мікросхеми після натиснення кнопки "ПУСК" у віконці пульта управління знімаються тактові імпульси, що забезпечують розгортку регістрів та подальший їх здвиг при поступанні тактового імпульса з вихода таймеру.

Вибір режиму роботи здійснюється перемиканням тумблера "РЕЖИМ".

Дана система управління має два режиму роботи ручний та автоматичний.

При ручному режимі логічні сигнали які задані користувачем на штекерній панелі поступають на вхід і встановлює регістри у початкове положення до перемикання на режим "Автоматичне".

в) субблок дешифратора здійснює подання сигналів управління (логічного нуля) послідовно в гніздах поля програмування з частотою такту.

Дешифратор включає 90 двоходових схем збігу, що виробляють розшифровку стану регістрів. Виходи схем збігу сполучені з відповідними гніздами поля програмування. Субблоків дешифратора в контролері два і вони дозволяють отримати поле програмування з 180 гніздами (10x18).

г) субблоків вихідної пам'яті забезпечує включення і виключення відповідних рухів робота і технологічного устаткування сигналами з поля програмування.

е) блок світлоіндикаторів приймає в себе сигнали з усіх субблоків контролера та є сигналізацією для взаємодії з користувачем.

ж) субблок вихідних підсилювачів призначений для включення та виключення виконавчих механізмів робота. При поданні сигналу управління на керуючий електрод симісторів вони відкриваються і це забезпечує проходження струму через обмотку електромагнітів клапана електропневматичних розподільників повітря.

1.3 ТЕОРЕТИЧНИЙ АНАЛІЗ СИСТЕМ КЕРУВАННЯ ПРОМИСЛОВИХ РОБОТІВ

Управління роботами здійснює його пристрій управління.

У сукупності з сенсорними і виконавчими системами воно утворює систему автоматичного управління робота.

Крім того, через пристрій управління роботом може керувати людина-оператор.

Відповідно до даної раніше класифікації в роботах застосовуються три способу управління - програмне, адаптивне і інтелектуальне.

Практично тільки програмне управління знайшло застосування в чистому вигляді, та й до нього часто додають елементи адаптації.

В цілому ж всі ці три способи управління застосовуються комплексно.

Адаптивне управління зазвичай будується на базі програмного як наступний рівень управління.

Назви систем управління конкретних роботів зазвичай визначається основним використаним в ній способом управління.

За ступенем участі людини-оператора системи розрізняють :

- автоматичного,
- автоматизованого і
- ручного управління.

За типом руху виконавчих систем існують системи управління:

- безперервні (контурні),
- дискретні позиційні (кроками «від точки до точки») та
- дискретні циклові (по упорам, як правило, з одним кроком по кожній координаті).

За керованим змінним розрізняють системи управління

- положенням (позицією),
- швидкістю,
- силою (моментом).

Часто ці способи управління застосовують в комбінації або різні способи по різних координатах, або з послідовним переходом від одного до іншого, або, нарешті, у вигляді функціональної залежності керованої змінної від іншої (наприклад, управління по силі, величина якої задається як функція від положення).

Циклову аналогову систему управління , як уже згадувалося, мають практично всі пневматичні роботи .

Процес управління окремими приводами зводиться до однократному розгону, руху з постійною швидкістю і гальмування при досягненні упору.

Програмування робота полягає в установці на кожному приводі цих упорів, які визначають величину переміщення по відповідній мірі рухливості , швидкості цих переміщень, послідовності включень приводів і можливих затримок часу між цими включеннями.

Всі ці операції крім установки упорів проводяться за допомогою перемикачів або інших органів на пульті пристрою управління.

Внаслідок простоти циклового управління для роботів з таким управлінням, як правило, застосовують пристрої групового управління.

Виходячи з дослідженої аналогової системи керування можна виділити переваги та недоліки. Одним із переваг даної системи є її закінченість та сумісність з заданим промисловим роботом БРИГ-10 ЗАЗ, але в той же час впливає умова гнучко – налаштованості системи при можливій її модернізації. Проблема модернізації полягає в устарілій конструкції системи управління. Сучасний світ здійснив перехід на цифрове керування. При готовій аналоговій конструкції частковий перехід неможливий. Всі вищевказані недоліки підштовхують на створення основи прототипу повністю цифрової дискретної системи керування, яка в свою чергу значно розширює функціональні можливості попередньої.

2. ОБГРУНТУВАННЯ ВИМОГ ДО ДИСКРЕТНОЇ СИСТЕМИ КЕРУВАННЯ ПРОМИСЛОВОГО РОБОТА

Дискретними називаються системи, в яких перетворення сигналів у деяких точках здійснюються тільки у дискретні моменти часу.

Типові роботи з таким управлінням - це промислові роботи для точкової зварки, збірки та обслуговування різного технологічного обладнання. Ці роботи мають велике число (десятки) точок позиціонування робочого органу маніпулятора. На відміну від систем аналогового управління тут точність позиціонування забезпечується не упорами, а точністю відпрацювання приводами зі зворотним зв'язком по положенню заданих керуючої програмою точок позиціонування(тактів).

Швидкість дії цифрових систем керування процесами прискорює обчислення та обробку інформації, дозволяє здійснювати оптимальне керування процесом у цілому, сприяє накопиченню даних, забезпечує можливість використовувати реєстрацію та проводити обробку даних, здійснювати діагностування процесу та прийняття рішень.

Основні вимоги до систем з дискретним цифровим керуванням промислового робота виглядають наступним чином:

1)В пристрої управління носієм програми виступає магнітний носій або вмонтована в контролер системи управління пам'ять типу EPROM.

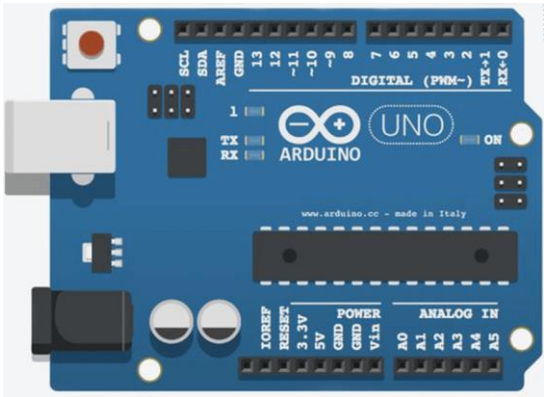
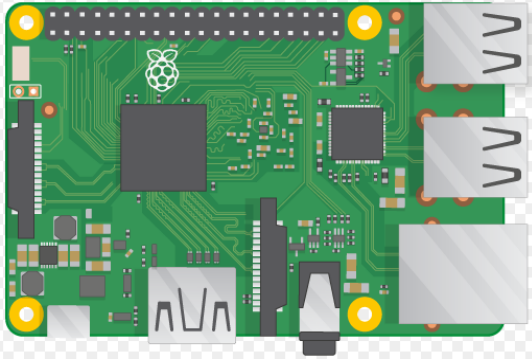
2)Задані програми мають вигляд послідовності числовий значень кроків або тактів позиціонування керуючих механізмів.

3)Відпрацювання програми полягає в одночасній подачі керуючих сигналів(в випадку промислового робота) на керуючі розподільники виконавчих механізмів для задання в початкове положення. Після цього керуюча програма видає команду на виконання наступного такту (шагу).

4)У випадку системи з оберненим зв'язком керуюча програма сприймає сигнали які надходять з датчиків, та записує їх значення для подальшого самонавчання.

2.1 ОГЛЯД ІСНУЮЧИХ АПАРАТНИХ ПЛАТФОРМ ДЛЯ СТВОРЕННЯ СИСТЕМ КЕРУВАННЯ

Одним із головних етапів створення дискретної системи керування став вибір апаратної платформи(бази).Усі плати для розробки можна розділити на дві категорії:

<p>Плати на мікроконтролері (MCU, MicroController Unit)</p>	<p>Одноплатні комп'ютери (SoC, System on a Chip)</p>
<p>Типовий представник – Arduino UNO</p> 	<p>Типовий представник – Raspberry PI</p> 

Мікроконтролери можуть одночасно виконувати всього одну задачу і відмінно з цим справляються. А одноплатні комп'ютери виконують програми в рамках операційної системи (найчастіше Linux), мають більшу продуктивність і оснащені широкими мультимедійними можливостями.

В виконуваній дипломній роботі для створення прототипу дискретної системи керування було обрано апаратну платформу на базі мікроконтролера.За основу було взято існуючу плату Arduino NANO.

Таблиця 2 . Технічна характеристика

Arduino nano:

Мікроконтролер	Atmega328
Робоча напруга	5В
Цифрові виходи/входи	16
Аналогові входи	8
Постійний струм	40мА
Флеш-памт'ять	6кБ
ОЗУ	2кБ
Габаритні розміри	18x42 мм

Дана апаратна платформа має власну середу для розробки програмного забезпечення – Arduino IDE. Мова програмування яка викривується – С++.

Плата має 16 цифрових виходів-входів (пінів) 6 з яких мають широку-імпульсну модуляцію сигналу. На даному етапі роботи платформа задовольняє усі вимоги щодо реалізації робочого прототипу дискретної системи керування.

2. РОЗРОБКА ЕЛЕКТРИЧНОЇ СХЕМИ ПРОТОТИПА ДИСКРЕТНОЇ СИСТЕМИ КЕРУВАННЯ

Розроблювана дискретна система має забезпечувати взаємодію з користувачем а також мати можливість управління керуючих елементів промислового робота.

На Рисунку 2.2.1 схематичну зображено прототип електричної схеми системи керування.

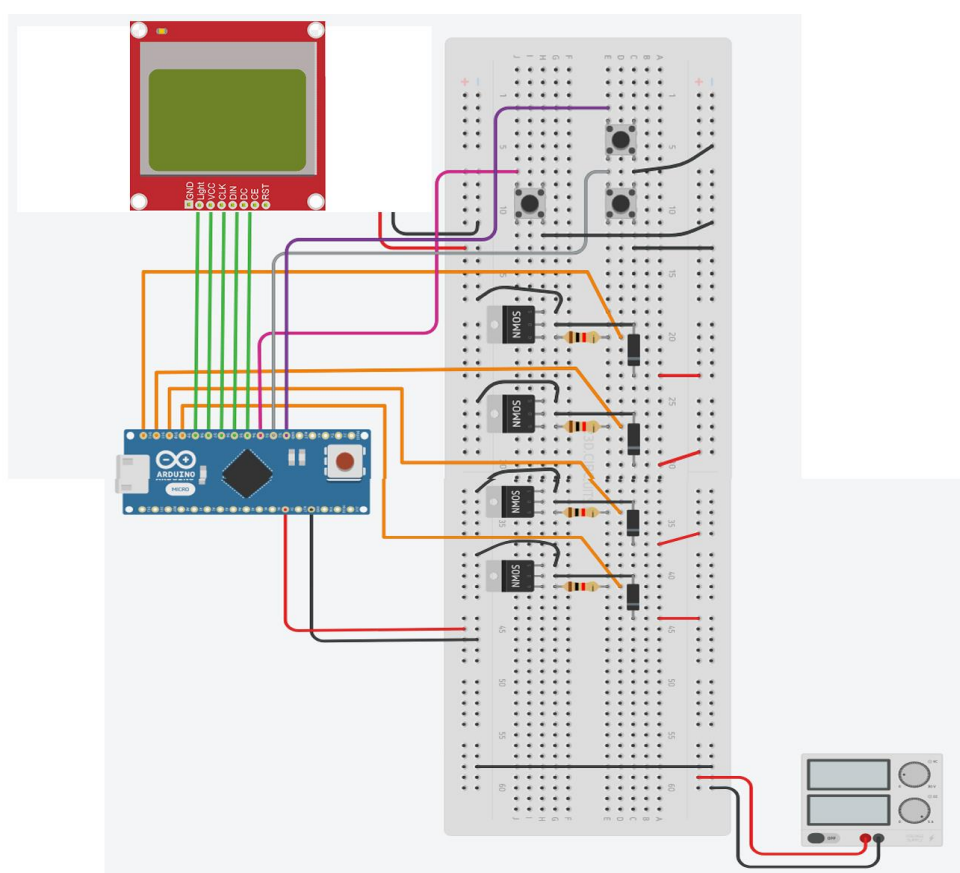


Рисунок.2.2.1 - Прототип електричної схеми цифрової системи керування

Дана схема складеться з:

- 1)Макетної плати з вертикальним напрямком руху сигналу.
- 2)Плати Arduino nano з під'єднаним кабелем протоколу USB та живленням 5В.
- 3)Модульного монохромного дисплею.
- 4)Трьох керуючих кнопок для взаємодії з програмним забезпеченням.
- 5)Чотирьох МДН транзисторів як слугують симісторами для керування пневморозподільниками.
- 6)Чотирьох діодів для усунення ЕДС самоіндукції високої напруги котушки пневморозподільника.

Усі зв'язки між компонентами схеми реалізовані двосторонніми штекерами типу ПАПА-ПАПА.

Для реалізації зв'язку з керуючими приводами використовується МДН транзистор IRL 2203n (Mosfet).Такий вибір зумовлений неможливістю апаратної платформи Arduino видавати на виході напругу 24В яка потрібна для вмикання пневморозподільника. Також, потужні транзистори MOSFET добре відомі своєю винятковою швидкістю перемикання при досить малій потужності управління, яку потрібно прикладати до затвору. Основна причина в тому, що затвор ізолюваний, тому потрібно потужність тільки на перезаряд ємності затвор-витік, і в статичному режимі ланцюг затвора практично не споживає струму.

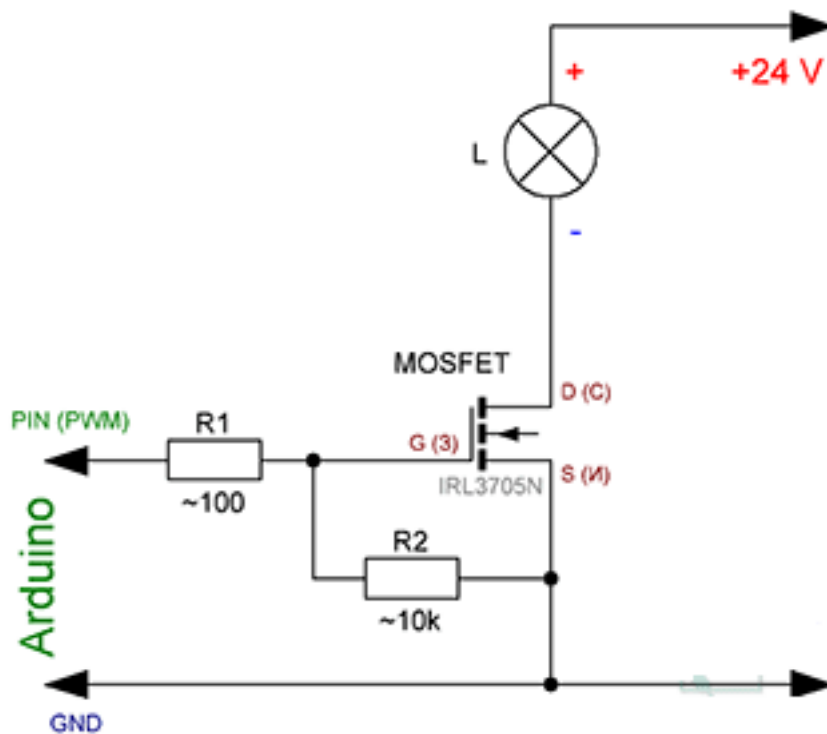
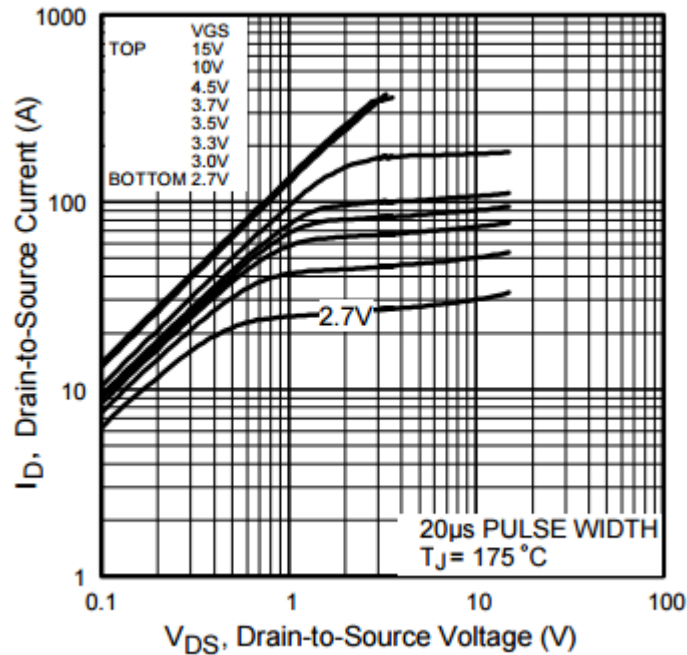


Рисунок.2.2.2 - Принципова схема підключення МДН транзистора

На малюнку 3 зображено принципову схему підключення МДН транзистора який керується сигналом з плати Arduino до пневморозподільника одного з виконавчих механізмів.

Принцип роботи польового МДН транзистора полягає в можливості управління протікаючим через великим малим струмом за допомогою малої потужності току управління. Живлення електромагнітної котушки пневморозподільника підключеною до стоків транзистора. При поданні сигналу з плати Arduino (5в 40мА) затвор транзистора відкривається і це дає змогу струму пройти далі до пневморозподільника. Тобто, МДН транзистор в даній схемі є своєрідним вмикачем який може вмикатися дуже малою величиною напруги.



Мал.2.2.3 Діаграма струму вмикання та вимикання

Діаграми струму вмикання та вимикання транзистора (мал4) відображають величину регульованих струмів та величини напруги, необхідної для відкриття затвору. На один пневморозподільник на електричній принциповій схемі виходить два МДН транзистора. Один з яких дає сигнал, другий при умові вмикання першого вимикає відкривання пневморозподільника. Така умова повина виконуватися тому що пневморозподільник має дві позиції.

Три керуючі кнопки під'єднані до пінів плати без втягуючих резисторів, так як виходи та входи плати мають власні. Кожна кнопка займає один керуючий пін і один загальний пін – заземлення сигналу.

Монохромний дисплей займає 5 керуючих виходів на платі і 2 загальних піна живлення і заземлення сигналу. Дисплей має розмірність 84 на 48 для відображених знаків.

3. РОЗРОБКА ПРОГРАМНОГО СЕРЕДОВИЩА СИСТЕМИ КЕРУВАННЯ РОБОТОЮ

Розробка програмного забезпечення є одним з найважливіших етапів роботи та функціонування дискретної системи керування. Апаратна платформа Arduino підтримує програмне забезпечення написане на мові програмування C++. Ця мова є потужним гнучко налаштованим та швидким інструментом. На заданому етапі проектування прототипу вона забезпечує всі умови реалізації програмного забезпечення.

3.1. ОБГРУНТУВАННЯ ОСНОВНИХ ВИМОГ ДО ГОЛОВНИХ ЕКСПЛУАТАЦІЙНИХ ОСОБЛИВОСТЕЙ РОБОТИ ЦИКЛОВОЇ СИСТЕМИ КЕРУВАННЯ

Як згадувалося раніше циклова дискретна система управління роботою має задовольнятися такими чинниками як:

1) Дискретне управління керуючими приводами в заданий момент часу.

Тобто, на програмно-носії знаходиться програма, яка задана користувачем і є відображеною циклограмою відпрацювання заданих керуючих приводів.

Апаратна платформа реалізує виконання послідовних інструкцій в заданий момент часу (посил керуючих сигналів до приводів), а також забезпечення оберненого зв'язку, при наявності аналогових датчиків (кінцевиків).

2) Повна взаємодія з користувачем (ведення та виведення даних).

Для забезпечення запису веденої циклограми на програмно-носії, апаратна платформа має включати в себе елементи взаємодії запису виводу. На даному етапі такими елементами є – монохромний дисплей та кнопки керування.

3)Для реалізації та написання програмного коду, апаратна платформа повинна мати власну середу розробки з власною програмування.

4)Апаратна платформа має бути забезпечена протоколами вводу виводу програми в вмонтований програмо носій.

На етапі проектування та створення прототипу дискретної системи керування мною було поставлені та виконані все вище поставлені задачі.

3.2. РОЗРОБКА АЛГОРИТМА РОБОТИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ СИСТЕМИ КЕРУВАННЯ.

Алгоритм програмного забезпечення є послідовним описом та набором інструкцій для написання програмного забезпечення що реалізує поставлену задачу. На даному етапі дипломної роботи було обрано використовувати метод комбінованого створення алгоритму. Комбінований алгоритм програмного забезпечення включає в себе лінійні, циклічні та розгалужені алгоритми спрацювання програмного коду.

Розробка алгоритму проводилась на основі готового алгоритму аналогової системи керування промислового робота з подальшим її спрощенням та оптимізацією.

Алгоритм Роботи Системи Керування

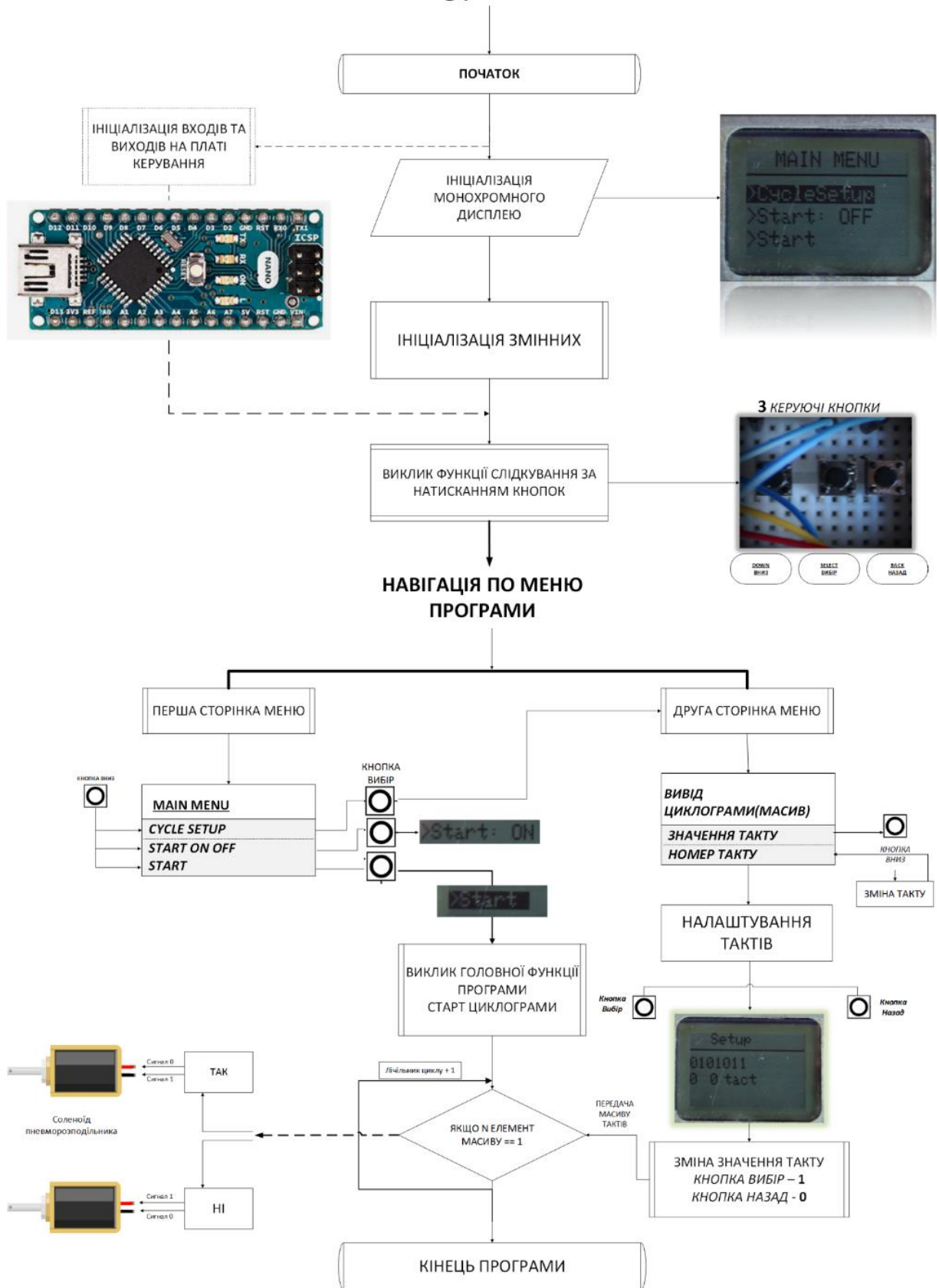


Рисунок.3.2.1 - Алгоритм цифрової системи керування

На малюнку 3.2.1 зображено загальний алгоритм роботи аналогової системи керування.

Загальна структура програмного забезпечення повинна відповідати вимогам щодо структури програмного коду апаратної платформи.

Алгоритм виконання програмного коду складається з двох етапів:

1) Однократне налаштування та ініціалізація програми. Налаштування входів та виходів (пінів) на платі Arduino. Ініціалізація монохромного дисплею. Всі налаштування описуються в програмному коді.

2) Вихід програми на зовнішній нескінчений цикл при якому починає виконання усі основні функції програми. Нескінченний цикл реалізований для виключення примусової зупинки виконання основного коду. Основний нескінченний цикл програми є зовнішнім, тому що в програмному коді можуть знаходитися функції або цикли які мають власні кінцеві умови зупинки та виходу з них.

На основі вищевказаних вимог та умов розглянемо поетапно алгоритм роботи цифрової системи керування. Першим етапом є ініціалізація всіх виходів та входів які використовуються на принциповій електричній схемі.

Нумерація та номер піна задається в програмному коді. Після цієї процедури виконується ініціалізація пінів та самого монохромного дисплею. Монохромний дисплей, в свою чергу призначений для взаємодії з користувачем виводом даних програми.

Після всіх цих процедур програма входить в зовнішній нескінчений цикл, в якому в першу чергу викликається функція слідкування за натисканням та розпізнаванням керуючих кнопок. Три керуючі кнопки, в свою чергу призначені для вводу даних, та навігації по меню програми. Після неперервного слідкування програма виводить на монохромний дисплей меню програми. Кожна керуюча кнопка призначена для різного виконання дій. В загальному алгоритмі на малюнку 5 показано відповідність виконання окремих керуючих кнопок, навігацію по меню та старт основного циклу.

Для реалізації основної дії програми – старту виконання набору інструкцій для включення виключення керуючих приводів – ЦИКЛОГРАМИ в алгоритмі виступає окрема функція(старт циклу). Вона приймає в себе числовий масив с нулів та одиниць що відповідає керуючим сигналам на платі Arduino. Числовий масив в свою чергу вводиться користувачем в відповідному меню програми. Старт циклограми має відбуватися при умові вмикання у другому пункті запобіжної умови для виключення випадкового старту циклу, сам основний цикл стартує при вмиканні третього пункту меню с попередньою перевіркою вищевказаної умови. Основна дія програми – цикл виконується по заданій кількості тактів які задані в змінних в програмному кодї. При наступані крайньої умови програма виходить з основного циклу. Це дає змогу змінювати з кожним новим разом змінювати умову головної функції – циклограми без перезапуску апаратної платформи.

3.3. ОПИС СТАНДАРТНИХ БІБЛІОТЕК, ЩО ВИКОРИСТОВУЮТЬСЯ В ПРОГРАМІ

Апаратна платформа Arduino передбачає використання зовнішніх окремих бібліотек, які в свою чергу розширюють функціонал програмного забезпечення. В даному програмному забезпеченні використання зовнішній бібліотек є необхідною умовою для виконання задуманих функцій.

Першою та невід’ємною зовнішньою бібліотекою виступає вбудована в середу розробки апаратної платформи бібліотека з набором готових функцій та структури головного програмного коду. Дана бібліотека дає можливість перетворювати написаний на мові програмування C++ програмний код в машинний код який є зрозумілим для мікроконтролера плати.

Друга зовнішня бібліотека реалізує підключення та налаштування монохромного дисплею. Бібліотека має набір готових функцій виводу на екран розмірністю 84 на 48 числових та буквених символів.

3.4 РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ, ОПИС ЇЇ ФУНКЦІОНАЛЬНИХ МОЖЛИВОСТЕЙ ТА УЗАГАЛЬНЕННЯ МОЖЛИВИХ НАПРЯМКІВ ЇЇ ВДОСКОНАЛЕННЯ

Для реалізації програмного забезпечення апаратна платформа повинна мати підтримку мови програмування, інтегровану середу розробки та протокол зв'язку з мікроконтролером. Обрана плата Arduino nano повністю задовольняє вищевказані умови.

Інтегрованою середою розробки апаратної платформи виступає Arduino IDE. Середовище розробки Arduino складається з вбудованого текстового редактора програмного коду, області повідомлень, вікна виведення тексту (консолі), панелі інструментів з кнопками часто використовуваних команд і декількох меню. Для завантаження програм і зв'язку середовище розробки підключається до апаратної частини Arduino за допомогою протоколу USB.

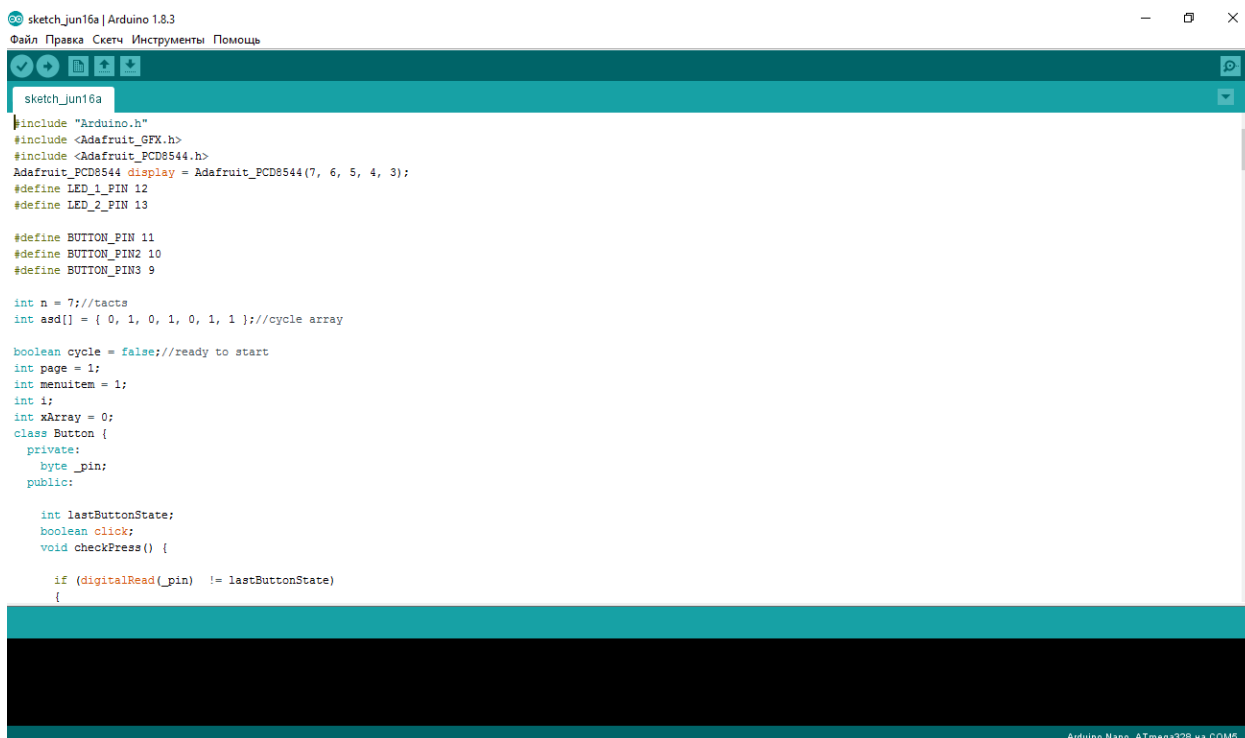


Рисунок.3.4.1 - Загальний вигляд Arduino IDE

Основну реалізацію програмного забезпечення можна описати фрагментами та структурою основних моментів програмного коду.

Повний код програми доступний в додатках.

Структура коду складається з трьох основних етапів:

1) Ініціалізація зовнішніх бібліотек, пінів (директив входів і виходів на платі) та оголошених змінних. Приклад фрагментів коду приведений до кожного етапу нижче.

```
#include "Arduino.h" - ініціалізація головної бібліотеки
#include <Adafruit_GFX.h>
#include <Adafruit_PCD8544.h> - ініціалізація бібліотеки дисплею
Adafruit_PCD8544 display = Adafruit_PCD8544(7, 6, 5, 4, 3);
#define LED_1_PIN 12 - ініціалізація директив виводу та входу
#define LED_2_PIN 13

#define BUTTON_PIN 11
#define BUTTON_PIN2 10
#define BUTTON_PIN3 9
int page = 1; - ініціалізація змінних
int menuitem = 1;
int i;
int xArray = 0;
```

Фрагмент коду 1. Ініціалізації зовнішніх бібліотек та директив

2) Одноразове налаштування, присвоєння директивам змінних для вводу відводу та ініціалізація зовнішніх пристроїв, в даному випадку – монохромного дисплею.

```
void setup() { - функція одноразового налаштування

  pinMode(LED_1_PIN, OUTPUT); - присвоєння директив
  pinMode(LED_2_PIN, OUTPUT);

  display.begin(); } - ініціалізація дисплею
```

Фрагмент коду 2. Одноразове налаштування директив та ініціалізація зовнішніх пристроїв

3)Третій етап виконання коду є основним. В ньому виконуються всі основні функції, та умови. Як зазначалося раніше даний етап реалізований зовнішнім нескінченним циклом.

```
void loop() { - Виклик нескінченного циклу
  down.checkPress(); -Приклад виконання коду в тілі основного циклу
  select.checkPress();
  back.checkPress();
  drawMenu();
```

Фрагмент коду 3.Виклик основного нескінченного циклу

Фрагменти програмного коду відображають приблизну реалізацію коду. Для більш детального ознайомлення з принципом роботи програмного забезпечення нижче приведені та описані принципи роботи основних функцій програми з таблицею змінних.

Таблиця 3.4.1.Основні зміні програмного коду

Змінна	Опис
int n = 7	Змінна кількості тактів циклограми
int asd[] = { 0, 1, 0, 1, 0, 1, 1 };	Одномірний масив чисел 0 -1,передається в функція старту циклограми, кожен елемент є ключем для перевірки умови функції старту.
boolean cycle = false;	Булева змінна для підготовки старту циклограми.
int page = 1;	Змінна яка зберігає в себе поточну сторінку меню
int menuitem = 1;	Змінна яка зберігає в себе поточний елемент меню
Button	Конструктор класу зі змінною, який оголошує керуючі кнопки на макетній платі.

<code>int xArray = 0;</code>	Змінна для нумерації елементу масиву в меню.
------------------------------	--

Основні функції програми можуть бути записані окремо від основної частини виконання – безкінечного циклу, та викликатися по необхідності окремо. Основною виносною функцією програми є функція яка виконує основну мету даної роботи – відпрацювання циклограми.

Дана функція *void startcycle()* виконана окремо від тілу основного циклу і викликається при досягненні умови, в даному випадку, вибором конкретного пункту меню на екрані монохромного дисплею. Для забезпечення роботи цієї функції, вона повинна приймати в себе одномірний числовий масив з нулів та одиниць, який користувач задає в конкретному меню програми.

Для відображення меню програми задана окрема функція *void drawMenu()*. Функція викликається при старті основного безкінечного циклу.

Для навігації по меню програми в програмному коді написані окремі умови переходу від елементів меню до інших сторінок. Умова включає в себе перевірку поточного меню та яку керуюча кнопка була нажата.

Перехід від першої до другої сторінки меню реалізована окрема функція *void setupCycle()*, яка викликається при виконанні умови (натиск кнопки в конкретному елементі меню першої сторінки).

На даному етапі розроблено прототип з написаним програмним забезпеченням який виконує всі поставлені перед ним функції а саме:

- 1)Робота з користувачем (ввід та вивід даних)
- 2)Відпрацювання заданої циклограми.

Але як і всі власно створені прототипи системи управління, ця система не позбавлена недоліків.

Одним із недоліків прототипу створеної системи цифрового управління є програмні та функціональні обмеження системи. Одним із функціональних недоліків є обмеження по кількості входів та виходів плати мікроконтролера.

Програмним недоліком системи можна вважати відсутність заданих переривань роботи програми, приклад – аварійна зупинка відпрацювання

циклограми. Іншим програмним недоліком є малий об'єм пам'яті програмоносія мікроконтролера що не дає в повному обсязі реалізувати потенціал алгоритму дискретної системи керування.

Вищевказані недоліки є логічною умовою для вдосконалення та використання апаратної платформи с вищими характеристиками.

Апаратні платформи Arduino на базі мікроконтролерів Atmega мають у своїй лінійці багато різновидностей плат з різними функціональними та апаратними можливостями. Тому, було вирішено взяти за основу для кінцевого варіанту реалізації дискретної системи управління плату Arduino MEGA на базі мікроконтролера останнього покоління.

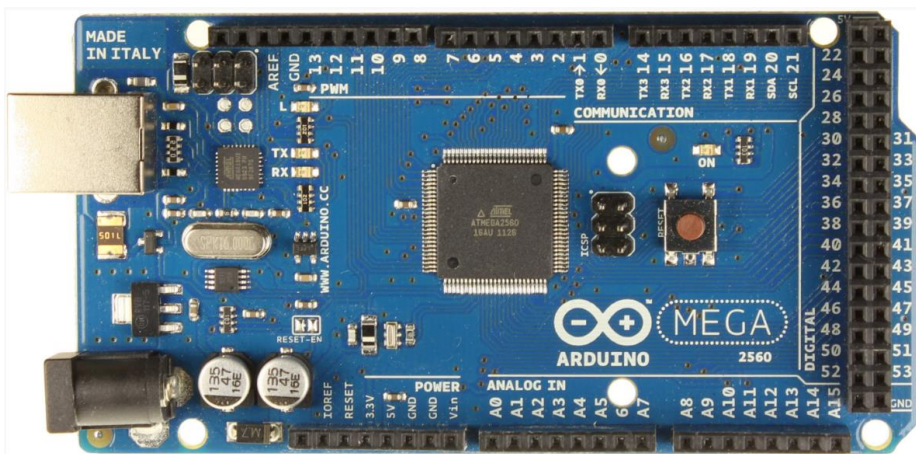


Рисунок.3.4.2 - Апаратна платформа Arduino MEGA

Таблиця 3.4.2 . Технічна характеристика
Arduino MEGA

Мікроконтролер	Atmega2560
Робоча напруга	5В
Цифрові виходи/входи	54(15 ШИМ)
Аналогові входи	16
Постійний струм	40мА
Флеш-памт'ять	256кБ
ОЗУ	8кБ
Габаритні розміри	102x53 мм

Технічна характеристика апаратної платформи Arduino MEGA наведена у таблиці 3.

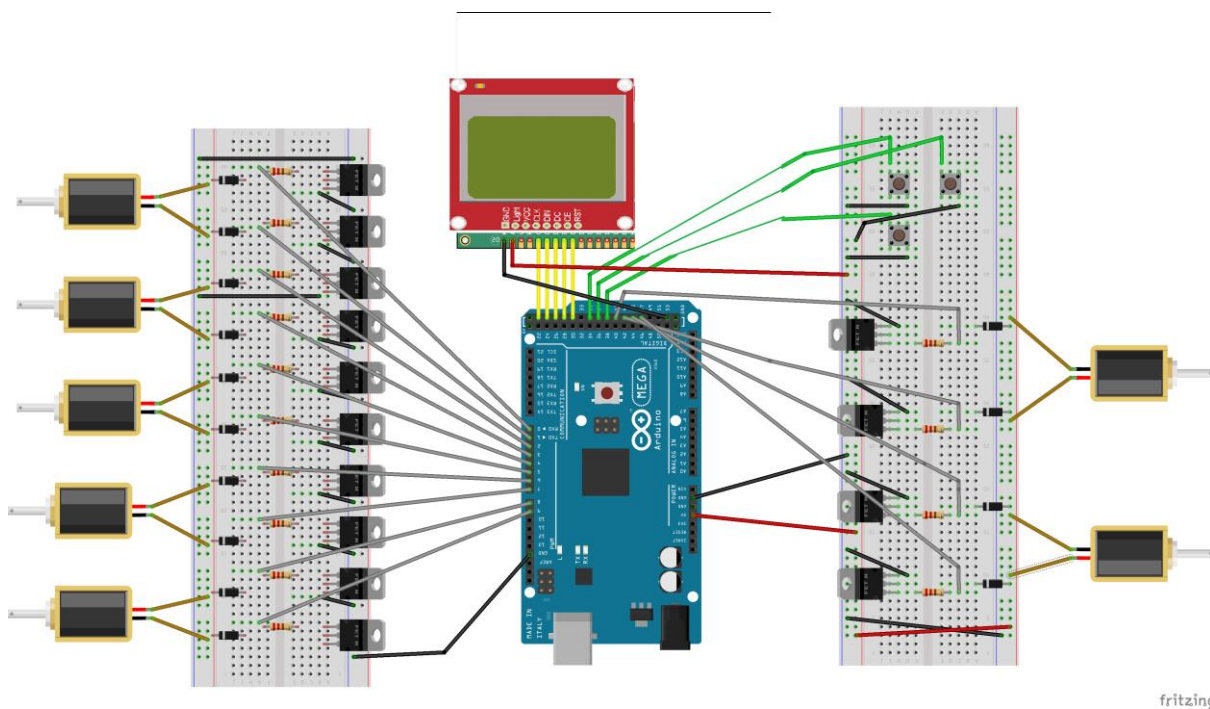


Рисунок.3.4.3 - Принципова електрична схема на базі Arduino MEGA

Виходячи с технічних характеристик апаратної платформи було створено принципову електричну схему з підключенням до всіх керуючих механізмів робота приведених на малюнку 3.4.3.

4. ПРОЕКТУВАННЯ ТА РОЗРАХУНОК ПНЕВМАТИЧНОГО ПРИВОДУ РУХУ РУКИ.

Комплексна автоматизація велико серійного виробництва дозволяє будувати спеціалізовані і вбудовані автоматичні лінії. Але ці методи засоби не придатні для серійного виробництва. У серійному виробництві потрібні універсальні засоби механізації і автоматизації основних і допоміжних, технологічних і транспортних операцій в умовах великої номенклатури та частою змінюваності виробів.

Одним із основних механізмів, який грає важливу роль в спрощені автоматизації промислових робіт є привід руки та схват робота.

Загальний вигляд руки промислового робота наведений на Рисунку 4.1

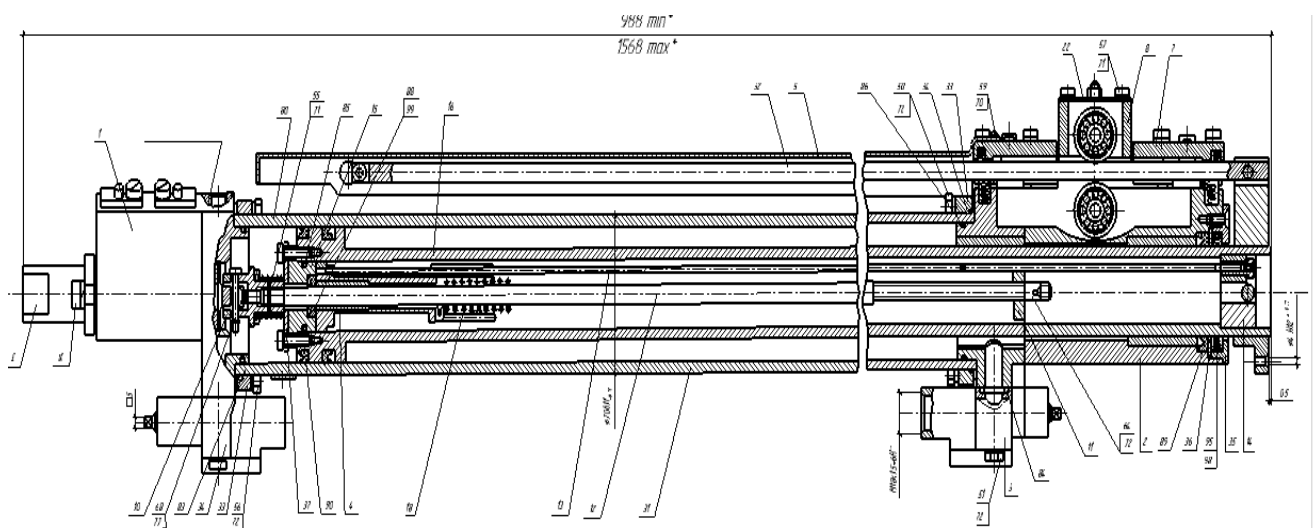


Рисунок 4.1 - Загальний вигляд руки робота

Рука робота виконана у вигляді двоголового пневмоцилиндру і складається з гільзи 1 (см.рисунок. 4.1), корпуса 2, демпфера гідравлічного 3, штока 4.

Рука працює наступним чином: при висуненні шток 4 у кінці ходу взаємодіє через пружину 26 з упором 21, примушуючи переміщуватися шток 15 гіродемпфера вправо. Масло витісняється з правої порожнини гіродемпфера в ліву через ряд каліброваних отворів і канал, що перекривається дросельною голкою 17. В результаті спрацювання демпфера висунення

штока 4 у кінці ходу плавно сповільнюється. При втягуванні шток 4 взаємодіє через пружину 20 з втулкою 19 і шток 15 гіродемпфера переміщається вліво.

Підведення стислого повітря в циліндр здійснюється через дроселі 29, обертання штока, що служать для регулювання швидкості, 4. Дроселі мають клапани, тому впускання повітря вільне, вихлоп регулюється дросельною голкою. Швидкість висунення штока руки регулювати правим дроселем, втягування - лівим.

4.1 ПРОЕКТУВАННЯ ТА РОЗРАХУНОК СХВАТУ РОБОТА

Проектування та розрахунок схвату промислового робота є показником його здатності виконувати поставлені перед ним задачі. Захватні пристрої (ЗП) промислових роботів (ПР) і маніпуляторів (М) служать для захоплення і утримання в певному положенні об'єктів маніпулювання. Ці об'єкти можуть мати особисті розміри, форму, масу і володіти різноманітними фізичними властивостями, тому ЗП відносяться до числа змінних елементів ПР.

Механізм затиску схвату робота працює від пневмоциліндра. Необхідно розрахувати силу і тиск, який необхідно створити в пневмосистемі для того, щоб деталь (заготовка) не випала зі схвату. Загальний вигляд схвату приведений на Рисунку 4.1.1

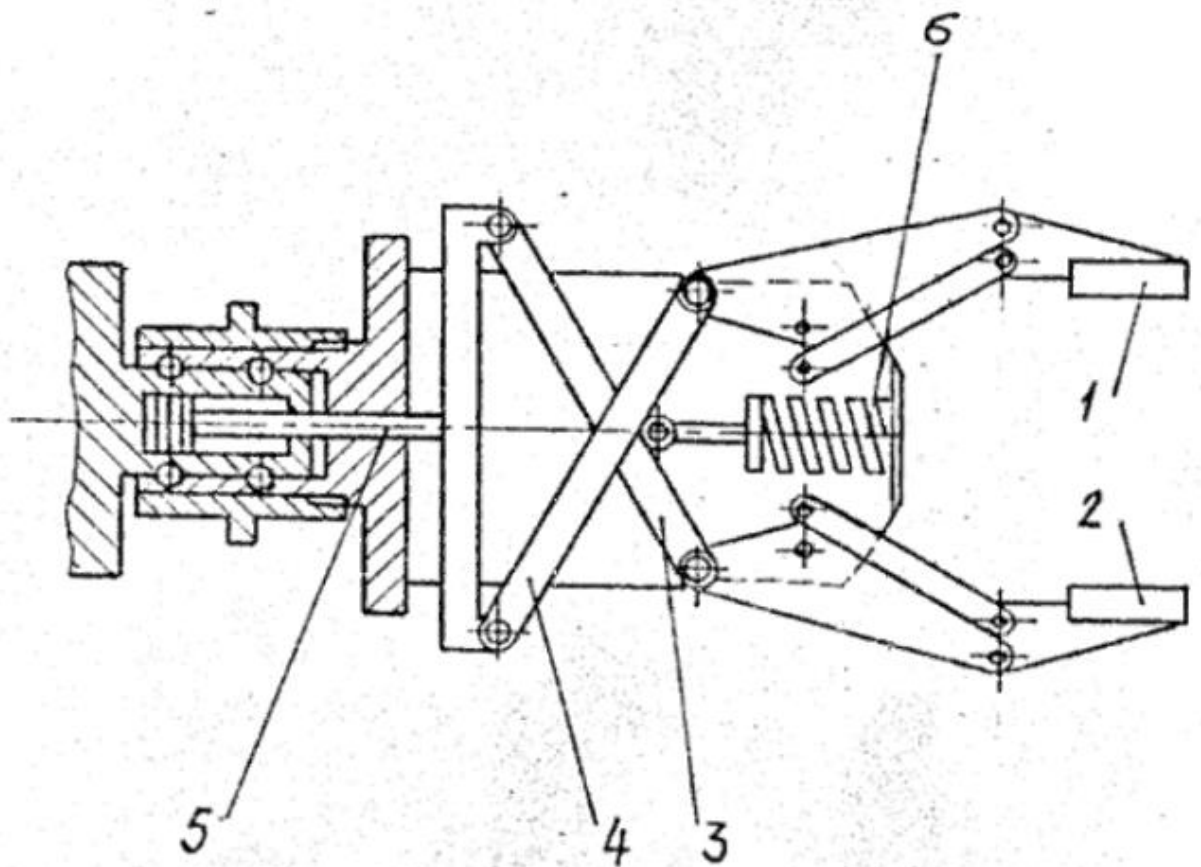


Рисунок.4.1.1 Схват промислового робота

Даний розрахунок можна поділити на декілька етапів:

- 1) Вибір заготовки
- 2) Розрахунок вантажопідйомності робота з обраним коефіцієнтом запасу
- 3) Розробити схему РТК і розміщення промислового робота відносно верстату
- 4) Обрати тип затискних елементів
- 5) Обчислити величини зусилля F та визначити діаметр поршня циліндру.

1) Для вибору заготовки приймаємо крайнє значення вантажопідйомності робота БРИГ-10 ЗАЗ та обираємо максимальне можливу заготовку масою $m_{заг} = 9$ кг (Вал 55x500 мм Сталь45).

2) Величина вантажопідйомності робота задана в паспорті $Q_{min} = 10$ кг .

3) На Рисунку 4.1.2 показано схему типового робото технічного комплексу з промисловим роботом підлогового розташування та горизонтальною рукою.

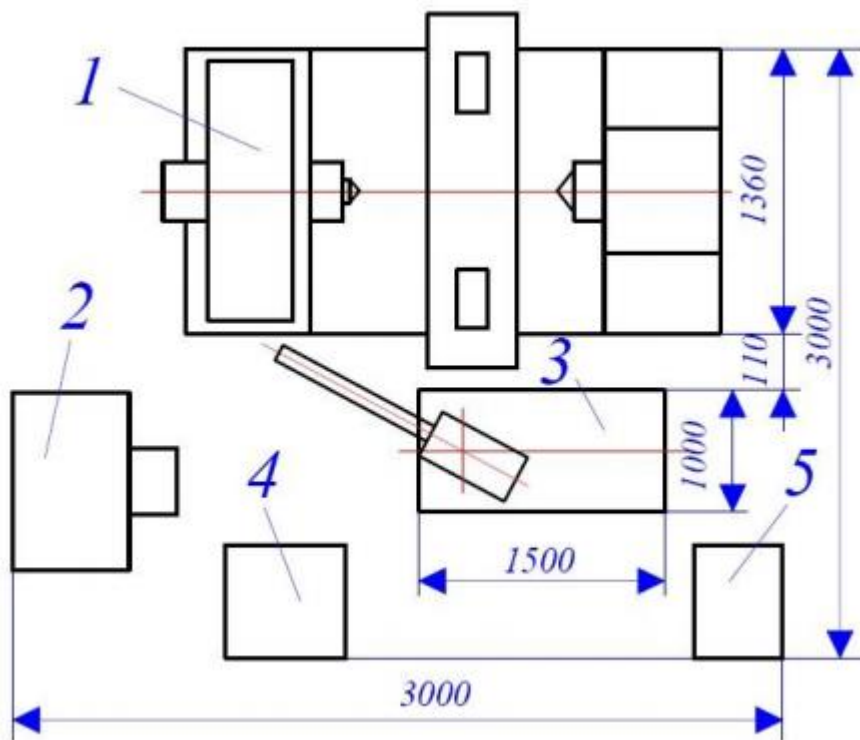


Рисунок. 4.1.2 - Схема РТК

4) Для вибору схеми схвату (затискних елементів) звернемося до паспорту та загальній схемі промислового робота. На малюнку 4.1.3 зображено принципову схему дії сил схвату на заготовку

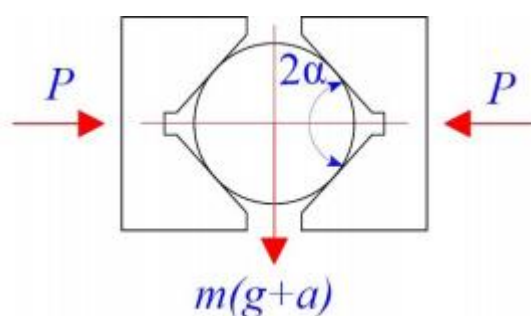


Рисунок.4.1.3 - Принципова схема схвату робота

5) Тип затискних елементів промислового робота БРИГ-10 ЗАЗ беремо для найбільш несприятливих умов, задля перевірки його на працездатність в поганих умовах

Тип затискних елементів - Загартована сталь з насічкою.

Коефіцієнт тертя - 0.25...0.30.

б) Зусилля затискного приводу шукаємо за формулою:

$$F = m(g + a)$$

$$F = 9(10 + 120) = 1170 \text{ Н}$$

Формула для визначення діаметру поршня циліндру

$$D = 1.13 \sqrt{\frac{F + F_n + c \cdot s}{p_n}};$$

$F = 1170 \text{ Н}$ зусилля яке діє на схват

$F_n = 30 \text{ Н}$ зусилля зворотної пружини

$c = 2,5 \text{ кН/м}$ жорсткість пружини

$s = 4 \text{ мм}$ довжина ходу поршня

$p_m = 0,3 \text{ МПа}$ тиск в робочій системі

$$D = 1.13 \sqrt{\frac{1170 + 2,5 * 4}{0.3}} = 71$$

Знайдену величину поршня D округлюємо згідно ДСТУ ISO до 70 мм.

У ході розрахунку було знайдене крайнє значення діаметру у несприятливих умовах циліндра який приводить у рух схват робота.

5. МОДЕРЕНІЗАЦІЯ СИСТЕМИ КЕРУВАННЯ

Одним із основних завдань у створенні системи керування для промислового робота БРИГ10-ЗА3 було перетворення старої системи керування, підтримка якої не можлива. Так, за основу магістерської дисертації за основу було взято завдання розширити діючу систему керування та запропонувати модернізацію одного с діючих приводів робота. В основі діючої системи керування діє мікроконтролер Arduino Nano(At mega 328), який керує розподільні клапани пневматичної системи робота, шляхом сигналів 1 або 0.

5.1 ТЕХНІЧНИЙ ОПИС ЗАВДАННЯ

Діюча дискретна система керування програмується на основі циклограми яка вводиться користувачем за допомогою елементів керування – монохромного дисплею та 3'ох фізичних кнопок для взаємодії с інтерфейсом. Приклад циклограми промислового робота наведений на рис(5.1.1)

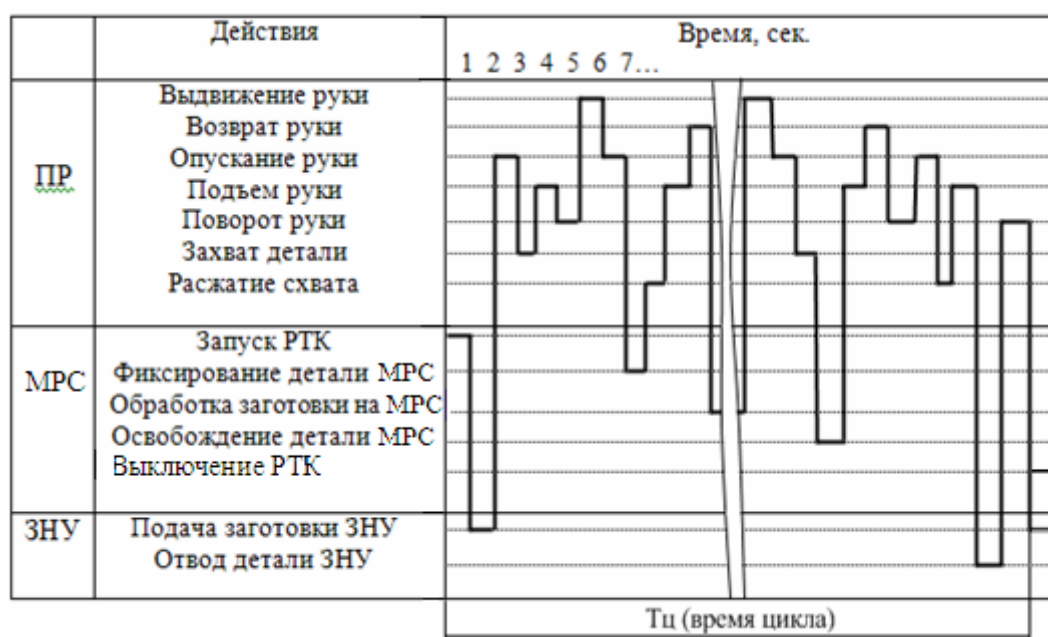


Рисунок.5.1.1 – Приклад циклограми промислового робота

При виконанні магістерської дисертації була поставлена задача модернізації діючої системи з метою покращення таких характеристик як – взаємодія з програмним інтерфейсом, розширення кількості вихідних сигналів для керування пневмоклапанами. Для подальшої модернізації за цими параметрами було запропоновано проаналізувати конструкцію робота БРИГ10-ЗА3 та варіанти розширення цифрових виводів керуючої плати.

5.2 МОДЕРЕНІЗАЦІЯ КЕРУЮЧОЇ ПЛАТИ

Виконавши аналіз готових рішень, щодо розширення кількості цифрових виводів було запропоновано варіант використання Вихідного Зсувного Регістру 74НС595. Приклад показаний на малюнку 5.2.1



Рисунок – 5.2.1 Приклад Зсувного Регістру

Регістри - найпоширеніші вузли цифрових пристроїв. Вони оперують з безліччю пов'язаних змінних, складових сигналів. Над сигналами виконується ряд операцій: прийом, видача, зберігання, зрушення у розрядній сітці, порозрядне логічні операції.

На вхід регістру подається масив сигналів (приклад-[1,1,0,0,1,0]) які на виході посилаються з одного з виходів регістру.

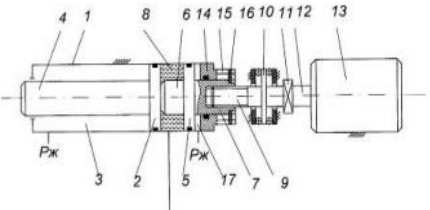
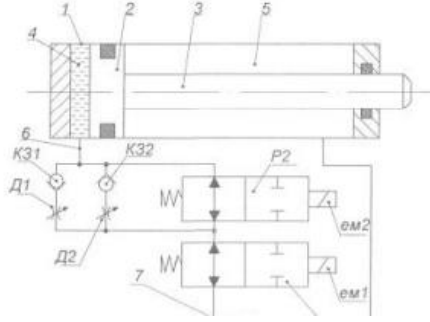
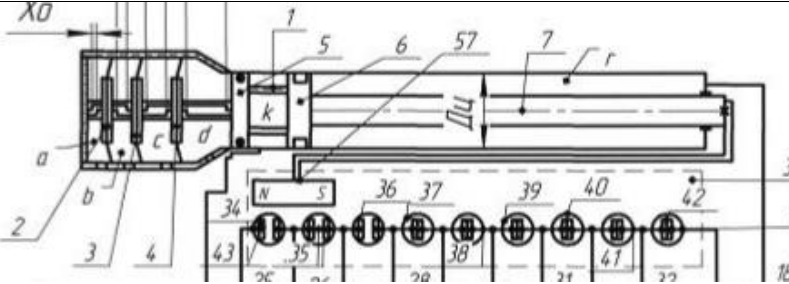
Вибір зсувного регістру було обґрунтовано малою кількістю вихідних сигналів з керуючої плати. Таким чином Зсувний Регістр розширив кількість керуючих сигналів до 19 замість 12 стандартних, використовуючи для цього тільки 1 вихідний сигнал з плати розширення.

6 ПАТЕНТНИЙ ПОШУК БАГАТОПОЗИЦІЙНОГО ПРИВОДА

Опираючись на нову кількість вихідних сигналів було запропоновано використання багатопозиційного дискретного привода. Для рішення цієї задачі був проведений аналіз існуючих цифрових приводів промислових робіт.

Патентний пошук був проведений за такими параметрами – кількість вхідних керуючих сигналів, максимальна довжина переміщення, тип привода.

РЕЗУЛЬТАТИ ПАТЕНТНО ІНФОРМАЦІЙНИХ ДОСЛІДЖЕНЬ

Країна походження	Параметри	Схема
Україна	Кількість сигналів – 4 Позицій - 4 Довжина переміщення – 64 мм	
Україна	Кількість сигналів – 10 Позицій - 10 Довжина переміщення – 1024 мм	
Україна	Кількість сигналів – 7 Позицій - 7 Довжина переміщення – 128 мм	

В результаті патентного пошуку було обрано КОМБІНОВАНИЙ БАГАТОПОЗИЦІЙНИЙ ПРИВІД З ЦИФРОВИМ КЕРУВАННЯМ. Привід має 7 вхідних керуючих сигналів, максимальний вихід штоку складає 127 мм.

Привід має 7 розрядів та дискрету 1 мм. Комбінований багатопозиційний привод з цифровим керуванням працює наступним чином. У вихідному положенні електромагніти всіх розподільників знеструмлені, що відповідає опрацьованій комбінації керуючих сигналів 0000000.

При опрацюванні, наприклад, кодової комбінації керуючих сигналів 0000001 (тиск живлення подається в 1 камеру) вихідний шток переміщується праворуч (за кресленням) на величину 1мм при дискретності 1мм.

Вибір цього багатопозиційного привода був зумовлений новій кількості вільних керуючих сигналів. Таким чином проаналізувавши вибір приводів обрана модель задовольняє всі параметри для пошуку.

7 МОДЕРЕНІЗАЦІЯ КОНСТРУКЦІЇ РОБОТА

На основі обраного багатопозиційного привода була запропонована заміна одного з приводів промислового робота БРИГ10-3А3. Проаналізувавши конструкцію робота було обрано замінити головний привід руки робота. Лінійне переміщення руки за умови старого пневматичного привода складає 90 мм. За запропонованим варіантом багатопозиційного привода переміщення приводу складатиме від 1 до 127 мм. Привод руки пневматичний приводиться в рух за допомогою пневмоклапанів, які спрацьовують від електророзподільників, які в свою чергу отримують сигнал від цифрової системи керування 1 або 0(+ та -).

7.1 ОПИС КОСТРУКЦІЇ БАГАТОПОЗИЦІЙНОГО ПРИВОДУ

Комбінований багатопозиційний привод має виконавчий циліндр з розміщеними в ньому мембранами з обмежувачами відносного переміщення і вихідний шток з поршнем, які утворюють штокову камеру, яка сполучена з гідрокамерами розрядних гідравлічних дозаторів, пневматичні камери яких сполучені з вихідними каналами пневматичних двопозиційних розподільників з одностороннім електричним керуванням, контакти електромагнітів яких примкнені до керуючих контактів електромагнітного дешифратора, вихідні контакти якого сполучені з вхідними контактами датчика лінійного переміщення, виконаного у вигляді ланцюга нормально замкнених герметичних контактів (герконів) і вихідних контактів та закріпленого на штоці розмикача, виконаного у вигляді магніту довжиною, більшою за величину кроку розміщення герметичних контактів. Штокова камера зворотного руху виконавчого циліндра сполучена через зворотні клапани і дроселі з вихідними каналами трьох двопозиційних дволінійних з одностороннім електричним керуванням гідравлічних розподільників, вхідні канали яких з'єднані з гідравлічною камерою пневмогідравлічного

акумулятора. Електричні контакти магнітів двох розподільників підключені до вихідних контактів датчика лінійного переміщення і до вхідних контактів логічного елемента АБО, вихідний контакт якого підключений до котушки електромагніта третього гідравлічного розподільника.

Комбінований багатопозиційний привод з цифровим керуванням працює наступним чином. У вихідному положенні електромагніти ем1...ем6 розподільників 15, 16, 17 і 45, 46, 47 знеструмлені. При цьому розрядні камери а, б, с, d, е, m і n з'єднані з атмосферою, що відповідає опрацьованій комбінації керуючих сигналів 0000000. Вихідний шток 7 під дією тиску, 10

що підводиться до штокової камери г через розподільник 47, дросель 51 і зворотний клапан 56 від акумулятора 48 знаходиться в лівому крайньому положенні. При цьому магніт 57 датчика переміщення 33 розмикає контакти герконів 34 і 35. При опрацюванні, наприклад, кодової комбінації керуючих сигналів 0000001 (тиск живлення подається тільки в камеру а) вихідний шток 7 переміщується праворуч (за кресленням) на величину $0 X$. При цьому рідина із камери г 15

витісняється в акумулятор 48 через дросель 52, зворотний клапан 55 і розподільник 47. Швидкість руху штока 7 при цьому регулюється дроселем 52. При опрацюванні кодової комбінації керуючих сигналів, наприклад 0110011 (тиск живлення одночасно подається до камер а, б і до камер е, m гідравлічних дозаторів від розподільників 15, 16, на електромагніти яких подається напруга) одночасно напруга, що подається на електромагніти ем1 і ем2 20

розподільників від контактів 22 і 23 вона підводиться і до керуючих контактів 18, 19 дешифратора 21, який перемикається і його вхідний контакт 58 замикається з вихідним контактом 28. При цьому напруга від контакту 28 по ланцюгу нормально замкнутих контактів герконів 38...42, контакт 44 передається на електромагніт ем2 розподільника 46, який перемикається ліворуч (за кресленням) і з'єднує штокову камеру г через зворотний клапан 54 і 25

дросель 50 з гідрокамерою акумулятора 48. Одночасно з цим напруга від контакту 44 подається на вхідний контакт елемента АБО, який перемикається і напруга від його вихідного контакту подається на електромагніт ем3, розподільника 47, який перемикається вверх (за кресленням) і запирає канали. Після перемикання розподільників 15, 16 і подачі тиску живлення в мембранні камери а, б поршень 6 з вихідним штоком 7 переміщується праворуч на величину 30Д - діаметр поршня 6. В кінці переміщення штока 7 магнітним полем магніту 57, розмикаються контакти геркона 38, що призводить до знеструмлення електромагнітів ем2 і ем3 розподільників 46 і 47, при цьому розподільник 46 запирається, а розподільник 47 відкривається і рідина із камери г надходить в 35

камеру акумулятора 48 через дросель 52, яким регулюється швидкість гальмування. Швидкість швидкого руху штока 7 праворуч регулюється дроселем 50. При з'єднанні камер а, б, е і т з атмосферою, що відповідає кодовій комбінації керуючих сигналів 0000000 одночасно з цим перемикається і дешифратор 21, при цьому від його вхідного контакту 58 напруга подається на контакт 25, від якого напруга через нормально замкнений геркон 34 і контакт 43 передається на 40

електромагніт ем1 розподільника 45 і на вхідний контакт елемента АБО, який перемикається і подає напругу на електромагніт ем3 розподільника 47. Розподільник 47 перемикається і замикає вхідний і вихідний канали. При перемиканні розподільника 45 рідина під тиском Рж із акумулятора 48 через послідовно увімкнені розподільник 45, дросель 49 і зворотний клапан 53 надходить в камеру г циліндра 1. Під дією тиска в камері г шток 7 з поршнем 6 рухаються 45

ліворуч, витискаючи рідину із камери к в розрядні камери е і т гідродозаторів 9 і 10. При цьому швидкість зворотного руху штока регулюється дроселем 49. В кінці переміщення штока 7 ліворуч магнітним полем магніту 57 розмикається контакт геркона 34, що веде до знеструмлення електромагнітів ем1 і ем3 розподільників 45 і 47, які перемикаються і рідина в

кінці руху штока 7 із камери гідроаккумулятора надходить в камеру г циліндра 1 через дросель 50

51, яким регулюється швидкість гальмування в кінці ходу. Аналогічно працює привод і при опрацюванні інших кодових комбінацій керуючих сигналів. В загальному випадку величина переміщення вихідного штока при перемиканні кодових комбінацій керуючих сигналів визначається по залежності: де X - дискретність цифрового приводу; i - порядковий номер розрядної камери; i В значуща цифра, яка приймає значення "1", коли тиск живлення підводиться до i -тої розрядної камери і значення - "0", коли i -та камера з'єднується з атмосферою. Максимальне число позицій вихідного штока $7 \cdot 12822N \cdot 7n \cdot \square \cdot \square \cdot \square$ (позицій), де n - число розрядних камер 5

Максимальна величина переміщення вихідного штоку)мм(127X). Завдяки включенню в структуру привода акумулятора, від якого рідина через двопозиційні розподільники і дроселі зі зворотними клапанами надходить в штокову камеру виконавчого циліндра, виникає можливість підвищити точність, так як втрата рідини, що надходить в штокову 10

камеру від гідравлічних розподільників не впливає на точність позиціонування вихідного штока, а заміна трипозиційного шестилінійного гідравлічного розподільника на двопозиційні дволінійні розподільники обумовлює спрощення конструкції, так як двопозиційні гідророзподільники масово виготовляються підприємствами, а трипозиційні шестилінійні розподільники даної схеми перемикання каналів підприємствами не випускаються, вони вимагають оригінальної розробки. 15

Запропонована схема керування двопозиційними гідравлічними розподільниками, датчиком лінійного переміщення і дешифратором дає можливість запобігати виникненню "викидів" і "провалів" вихідного штока і регулювання як шляхової швидкості руху, так і гальмування швидкості при підході штока в задану позицію. Крім того, розміщення у виконавчому циліндрі додаткових розрядних камер з мембранами розширює діапазон позиціонування вихідного 20 штока. Комбінований багатопозиційний привод

з цифровим керуванням працює наступним чином. У вихідному положенні електромагніти ем1...ем6 розподільників 15, 16, 17 і 45, 46, 47 знеструмлені. При цьому розрядні камери a, b, c, d, e, m і n з'єднані з атмосферою, що відповідає опрацьованій комбінації керуючих сигналів 0000000. Вихідний шток 7 під дією тиску, 10

що підводиться до штокової камери r через розподільник 47, дросель 51 і зворотний клапан 56 від акумулятора 48 знаходиться в лівому крайньому положенні. При цьому магніт 57 датчика переміщення 33 розмикає контакти герконів 34 і 35. При опрацюванні, наприклад, кодової комбінації керуючих сигналів 0000001 (тиск живлення подається тільки в камеру a) вихідний шток 7 переміщується праворуч (за кресленням) на величину $0 X$. При цьому рідина із камери r 15

витісняється в акумулятор 48 через дросель 52, зворотний клапан 55 і розподільник 47. Швидкість руху штока 7 при цьому регулюється дроселем 52. При опрацюванні кодової комбінації керуючих сигналів, наприклад 0110011 (тиск живлення одночасно подається до камер a, b і до камер e, m гідравлічних дозаторів від розподільників 15, 16, на електромагніти яких подається напруга) одночасно напруга, що подається на електромагніти ем1 і ем2 20

розподільників від контактів 22 і 23 вона підводиться і до керуючих контактів 18, 19 дешифратора 21, який перемикається і його вхідний контакт 58 замикається з вихідним контактом 28. При цьому напруга від контакту 28 по ланцюгу нормально замкнутих контактів герконів 38...42, контакт 44 передається на електромагніт ем2 розподільника 46, який перемикається ліворуч (за кресленням) і з'єднує штокову камеру r через зворотний клапан 54 і 25

дросель 50 з гідрокамерою акумулятора 48. Одночасно з цим напруга від контакту 44 подається на вхідний контакт елемента АБО, який перемикається і напруга від його вихідного контакту подається на електромагніт ем3, розподільника 47, який перемикається вверх (за

кресленням) і запирає канали. Після перемикання розподільників 15, 16 і подачі тиску живлення в мембранні камери а, б поршень 6 з вихідним штоком 7 переміщується праворуч на величину 30Д - діаметр поршня 6. В кінці переміщення штока 7 магнітним полем магніту 57, розмикаються контакти геркона 38, що призводить до знеструмлення електромагнітів ем2 і ем3 розподільників 46 і 47, при цьому розподільник 46 запирається, а розподільник 47 відкривається і рідина із камери г надходить в 35

камеру акумулятора 48 через дросель 52, яким регулюється швидкість гальмування. Швидкість швидкого руху штока 7 праворуч регулюється дроселем 50. При з'єднанні камер а, б, е і т з атмосферою, що відповідає кодовій комбінації керуючих сигналів 0000000 одночасно з цим перемикається і дешифратор 21, при цьому від його вхідного контакту 58 напруга подається на контакт 25, від якого напруга через нормально замкнений геркон 34 і контакт 43 передається на 40

електромагніт ем1 розподільника 45 і на вхідний контакт елемента АБО, який перемикається і подає напругу на електромагніт ем3 розподільника 47. Розподільник 47 перемикається і замикає вхідний і вихідний канали. При перемиканні розподільника 45 рідина під тиском Рж із акумулятора 48 через послідовно увімкнені розподільник 45, дросель 49 і зворотний клапан 53 надходить в камеру г циліндра 1. Під дією тиску в камері г шток 7 з поршнем 6 рухаються 45

ліворуч, витискаючи рідину із камери к в розрядні камери е і т гідродозаторів 9 і 10. При цьому швидкість зворотного руху штока регулюється дроселем 49. В кінці переміщення штока 7 ліворуч магнітним полем магніту 57 розмикається контакт геркона 34, що веде до знеструмлення електромагнітів ем1 і ем3 розподільників 45 і 47, які перемикаються і рідина в кінці руху штока 7 із камери гідроакумулятора надходить в камеру г циліндра 1 через дросель 50

51, яким регулюється швидкість гальмування в кінці ходу. Аналогічно працює привод і при опрацюванні інших кодових комбінацій керуючих

сигналів. В загальному випадку величина переміщення вихідного штока при перемиканні кодових комбінацій керуючих сигналів визначається по залежності: де 0 X - дискретність цифрового приводу; i - порядковий номер розрядної камери; i B значуща цифра, яка приймає значення "1", коли тиск живлення підводиться до i -тої розрядної камери і значення - "0", коли i -та камера з'єднується з атмосферою. Максимальне число позицій вихідного штока 7 $12822N$ $7n$ \square \square (позицій), де n - число розрядних камер 5

Максимальна величина переміщення вихідного штоку $)\text{мм}(127X)12(XX$ 0 n 0max \square \square \square . Завдяки включенню в структуру привода акумулятора, від якого рідина через двопозиційні розподільники і дроселі зі зворотними клапанами надходить в штокову камеру виконавчого циліндра, виникає можливість підвищити точність, так як втрата рідини, що надходить в штокову 10

камеру від гідравлічних розподільників не впливає на точність позиціонування вихідного штока, а заміна трипозиційного шестилінійного гідравлічного розподільника на двопозиційні дволінійні розподільники обумовлює спрощення конструкції, так як двопозиційні гідророзподільники масово виготовляються підприємствами, а трипозиційні шестилінійні розподільники даної схеми перемикання каналів підприємствами не випускаються, вони вимагають оригінальної розробки. 15

Запропонована схема керування двопозиційними гідравлічними розподільниками, датчиком лінійного переміщення і дешифратором дає можливість запобігати виникненню "викидів" і "провалів" вихідного штока і регулювання як шляхової швидкості руху, так і гальмування швидкості при підході штока в задану позицію. Крім того, розміщення у виконавчому циліндрі додаткових розрядних камер з мембранами розширює діапазон позиціонування вихідного 20 штока.

Схема приводу наведена на рисунку 7.1.1.

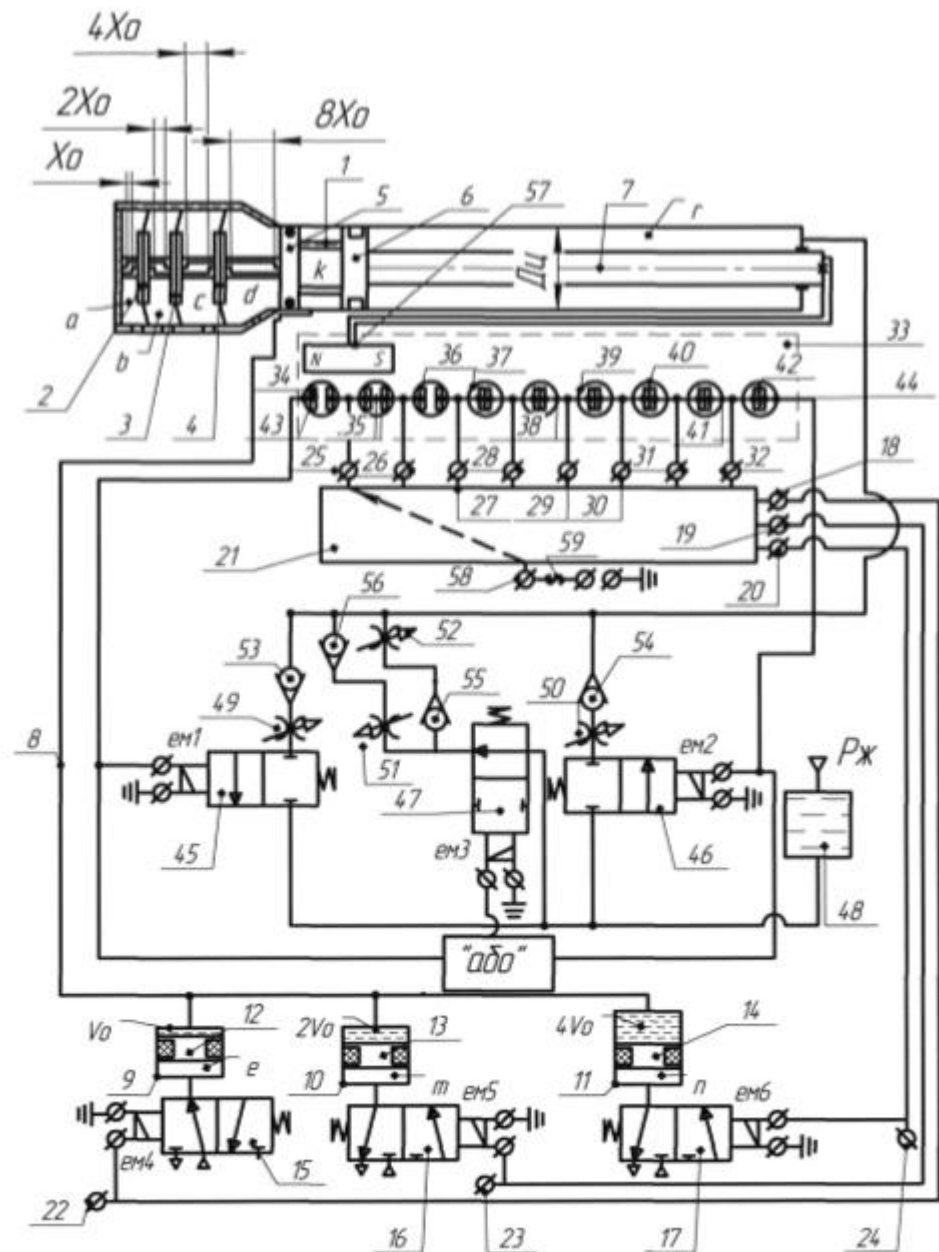


Рисунок - 7.1.1 Схема багатопозиційного комбінованого приводу

Комбінований багатопозиційний привод з цифровим керуванням працює наступним чином. У вихідному положенні електромагніти ем1...ем6 розподільників 15, 16, 17 і 45, 46, 47 знеструмлені. При цьому розрядні камери а, b, c, d, e, m і n з'єднані з атмосферою, що відповідає опрацьованій комбінації керуючих сигналів 0000000.

Даний комбінований багатопозиційний привод був обраний за критерієм кількості вхідних сигналів які не мають перевищувати закладені в оновленій цифровій системі та мають бути не більше 8.

Запропонований варіант заміни головного привода руки є прикладом інтеграції та спільного застосування можливостей цифрової системи керування та комбінованого привода. Маючи можливість дискретно керувати сигналами, спектр використання та можливостей привода та системи керування може бути набагато більший.

8. МОДЕРЕНІЗАЦІЯ СХЕМИ ЦИФРОВОЇ СИСТЕМИ КЕРУВАННЯ

Задана цифрова система керування базується на цифровому контролері Arduino nano на базі мікропроцесора Atmega 328 має напругу живлення 5v та кількість вихідних пінів сигналів – 13. Попередньо створена система мала лінійний алгоритм вмикання та вимикання керуючих приводів робота. В якості запропонованого варіанту з дискретним багатопозиційним приводом була поставлена задача з розробки нової принципової схеми підключення системи керування до промислового робота та алгоритму її дії. Опираючись на вхідні характеристики дискретного привода нова аналогова схема підключення системи керування до робота була модернізована з використанням Зсувного регістру 74НС595. Принцип його роботи полягає у тому, що на 1 пін ми посилаємо дані (1 чи 0), які згодом виводяться на 8 виходів, а подача імпульсів на другий пін “посуває” попереднє записане значення. Його можна використовувати як і для послідовного виводу інформації, так і у ролі “розширювача пінів” (для керування потрібно мінімум 2 піна, тобто можна отримати +6 пінів). Схему зсувного регістру показано на рисунку 8.1

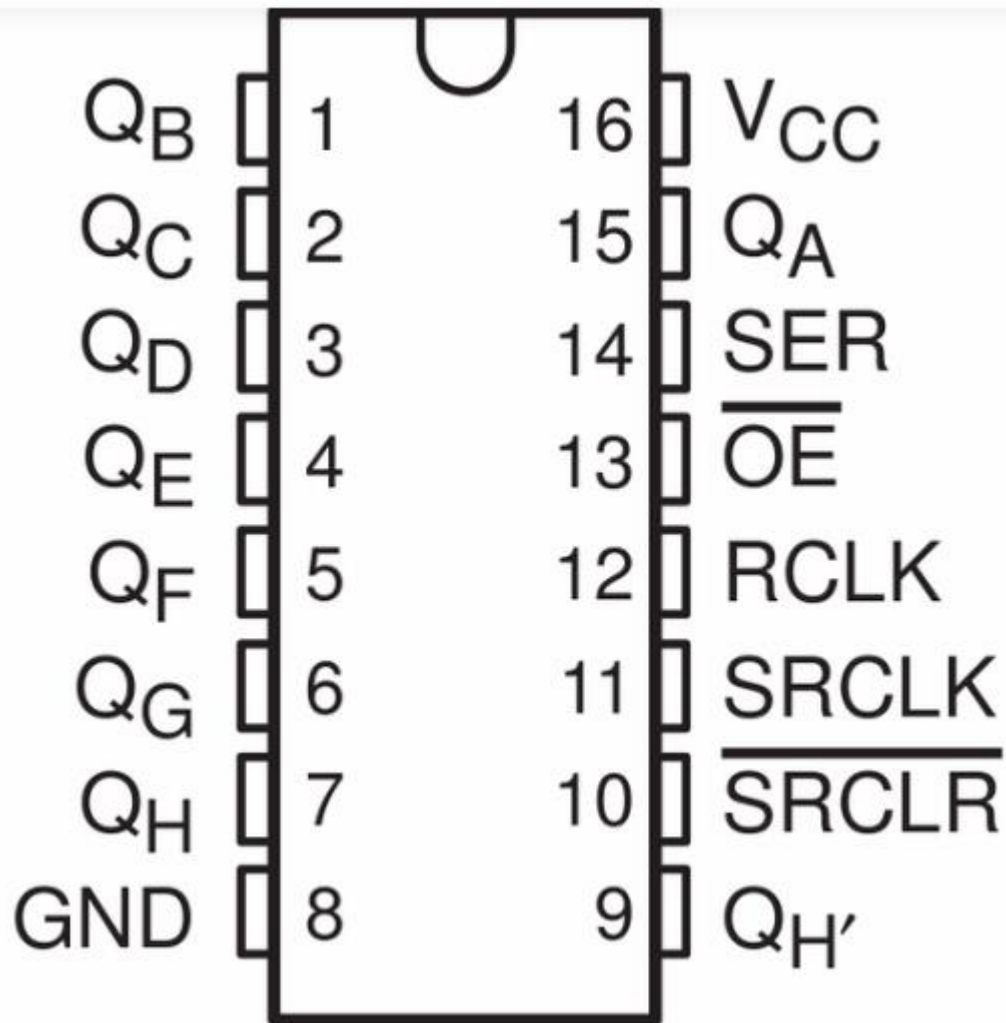


Рисунок 8.1 – Схема зсувного регістру 74HC595

Ми будемо використовувати зсувний регістр 74HC595 так як він є одним з найпоширеніших готових рішень у питанні розширення кількості виводів. Перед тим, як починати роботу з пристроєм, була прочитана документація, оскільки там описані всі вимоги та нюанси роботи. Нижче приведена інформація про піни та їх розшифровка.

VCC—живлення (стандартно 5В)

GND—земля

Qa:Qh—піни виходу

Qh`—пін для передачі даних виходу на наступний регістр (якщо їх кілька)

SER—пін вводу (на нього ми посилатимемо дані, які хочемо записати)

~OE—пін дозволу виводу (дозволяє або забороняє вивід), інвертований (для того, щоб працював, потрібно подавати логічний нуль)

RCLK—пін заціпки, дозволяє або забороняє заціпку.

SRCLK—пін зсуву (посуває дані на один пін вгору (зсув відбувається від Qa до Qh))

~SRCLR—пін очищення (очищує дані), інвертований (для очищення потрібно подати логічний нуль).

Для більшого розуміння інформації нижче також приведені терміни **Інвертований пін**—це пін, який має протилежний принцип роботи. Наприклад, у нас є інвертований пін, і ми до нього підключили привод робота. Так , для того, щоб привід виконав переміщення, ми повинні подати на пін логічний нуль, а для того, щоб повернути його на вихідну позицію—логічну одиницю.

Защіпка (latch)—це такий блок пам'яті, куди спочатку записуються дані, а звідти вони попадають на вивід. Вона служить для того, щоб надати можливість виводити певний об'єм інформації за раз (тобто ми можемо одночасно вивести від 1 до 8 бітів). Це досить зручно, оскільки при використанні регістра у ролі розширювача бітів нам потрібно посилати логічну одиницю лише на один біт і добитися цього без защіпки неможливо.

Отже, принцип роботи зсувного регістру наступний: спочатку потрібно дозволити/заборонити вивід. Для цього, потрібно подати відповідний логічний рівень на біт OE. Оскільки він інвертований, подавати потрібно протилежне значення (тобто, для того, щоб дозволити вивід, потрібно встановити рівень в нуль, щоб заборонити—в одиницю). Після потрібно закрити защіпку. Тепер можна записувати дані. Процес запису відбувається так: подаємо відповідний логічний рівень на біт SER, а потім посилаємо короткий імпульс (1 -> коротка затримка ~50мс -> 0) на біт SRCLK. Коли ми записали всі дані, які потрібні—відкриваємо защіпку, подавши логічну одиницю на біт RCLK. Впринципі, защіпку можна завжди тримати відкритою, однак для зручності бажано робити спочатку запис у пам'ять, а після на вивід. Очищаємо вивід так: оскільки біт SRCLR інвертований, посилаємо наступний імпульс (0 -> коротка затримка -> 1).

На практиці була складена принципова схема підключення Arduino з зсувним регістром, приклад наведений на рисунку 8.2. Також важливо зазначити що принципова схема взаємодії з користувачем не зазнала змін, окрім програмного коду та оновленого алгоритму .

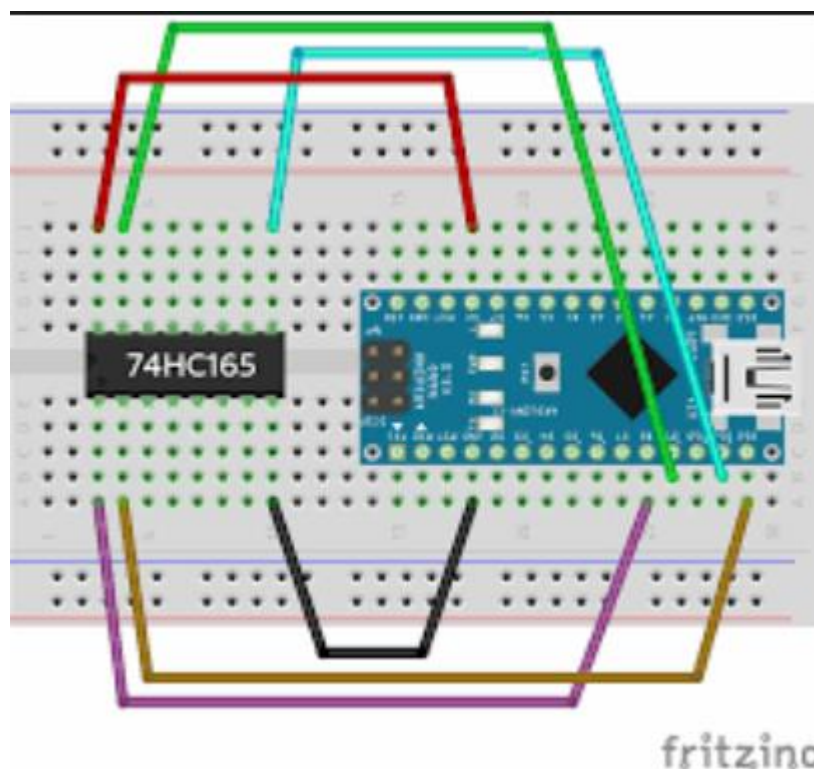


Рисунок 8.2 – Схема підключення Arduino з зсувним регістром

На принциповій схемі усі піни плати з'єднані з зсувним регістром згідно наведеної документації щодо зсувного регістру. Для подальшої спільної роботи системи керування та регістру було доповнено та оновлено алгоритм роботи керуючої програми, програмний код системи керування та алгоритм взаємодії з користувачем.

Згідно опису роботи багатопозиційного приводу була складена принципова схема підключення плати системи керування до пневматичних перемикачів.

Багатопозиційний дискретний привод має 7 вхідних сигналів для управління пересуванням штока приводу. Кожен з вхідних сигналів має керуватися логічними сигналами 1 або 0, загально вхідний сигнал має вигляд

послідовності значень 1 та 0 що відповідає за конкретну камеру пневматичного приводу – [0000000].

Нижче наведений приклад відповідності розряду або сигналу у конкретній позиції багатопозиційного приводу

1	2	3	4	5	6	7
1мм	2мм	4мм	8мм	16мм	32мм	64мм

Таким чином при посиланні системою керування коду 1000000 цифровий привід перемістить шток на величину 1мм, команда 1010011 перемістить шток на 101 мм. Максимальна величина переміщення 127 мм.

Цей сигнал відповідно посилається с системи керування послідовно на зсувний регістр, він в свою чергу відсилає сигнали до реле, які відкривають сигнал малою напругою, далі реле замикає контакти пневморозподільника конкретної камери у багатопозиційному приводі. На рисунку 8.3 наведена принципова схема з'єднання системи керування з електричним реле.

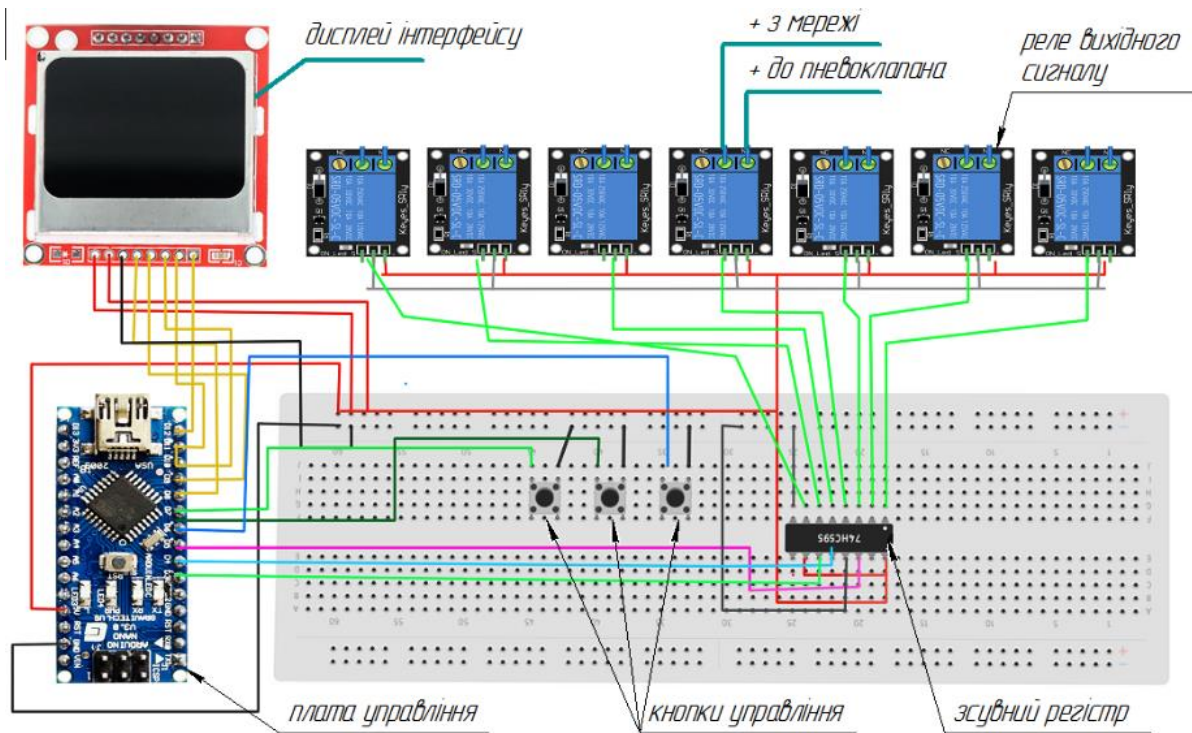


Рисунок 8.3 – Схема системи керування з'єднаним з реле

8.1 ОНОВЛЕННЯ АЛГОРИТМУ ТА ПРОГРАМНОГО КОДУ

Програмний код та алгоритм роботи системи керування був розроблений для роботи з лінійним сигналом 0 або 1 та задовольняв усі потреби у використанні. Згідно з поставленою задачею код та алгоритм зазнав змін та доопрацювання.

Взаємодія з користувачем відбувається за допомогою 3х фізичних кнопок та монохромного дисплею. Інтерфейс програми складається з 3 пунктів меню

- 1) налаштування циклу
- 2) кількість повторень
- 3) старт циклограми

Так як багатопозиційний привід має розширений тип вводу даних було обрано замінити пункт меню кількості повторень на ввід значення переміщення штоку багатопозиційного привода у мм.

Оновлений алгоритм коду роботи та навігації користувача з програмою наведений на рисунку 8.1.1

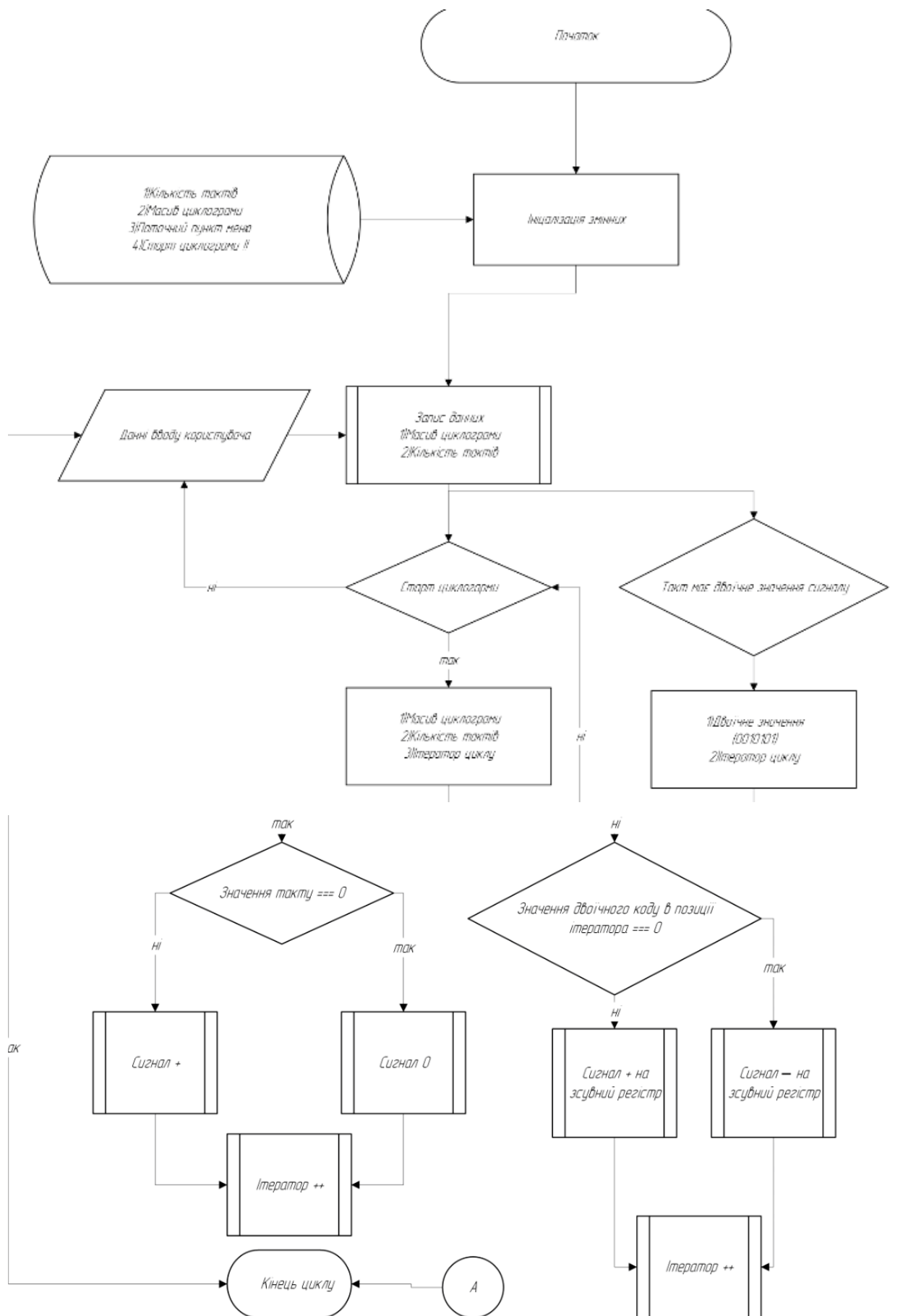


Рисунок 8.1.1 – Алгоритм коду взаємодії з користувачем

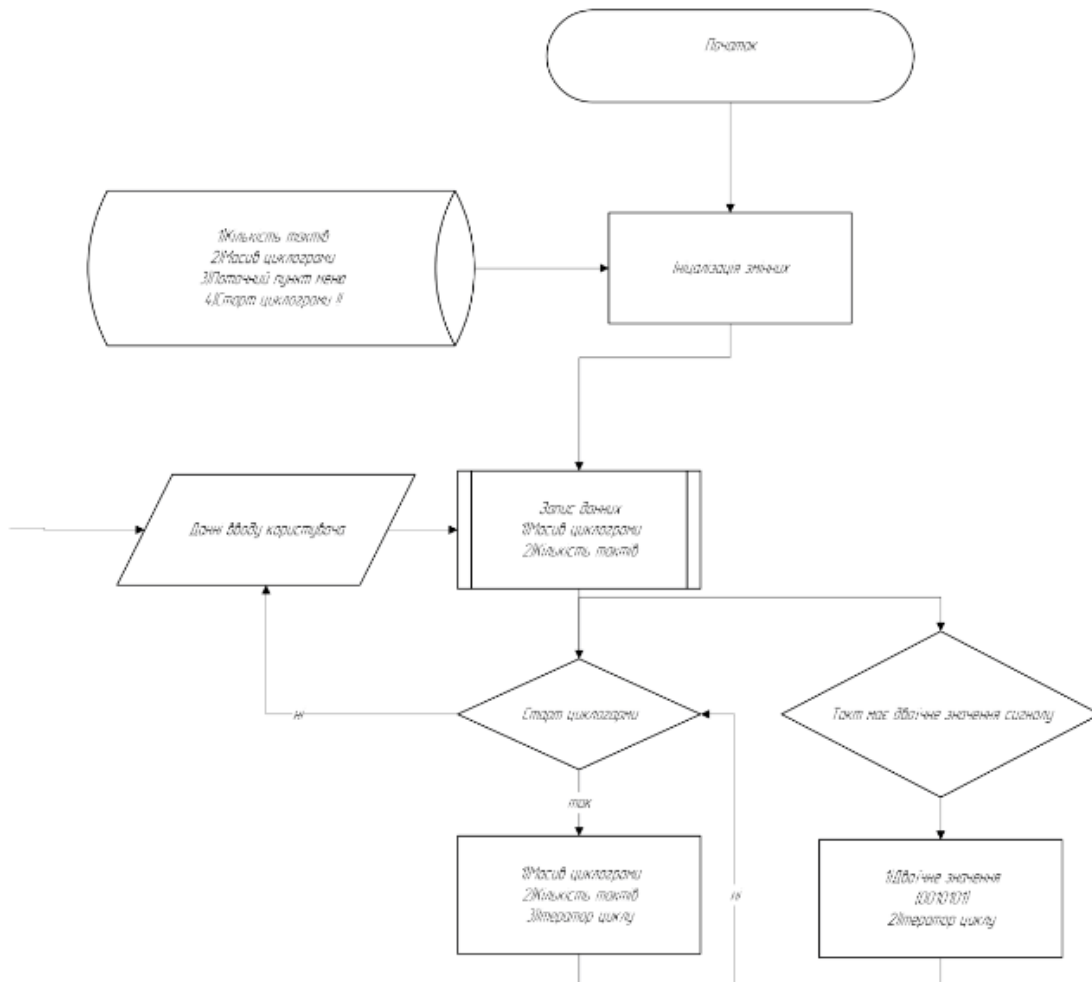
Програмний код роботи системи керування також зазнав змін. Була поставлена така задача як перетворення вхідних значень користувача, а саме, величина у мм переміщення штоку багатопозиційного приводу на зрозумілий для машини двоїчний код.

Першим етапом обробки вхідних значень є запис їх до пам'яті контролера. Значення у мм є десятковим типом даних. При вводі користувачем з меню пункту меню старт циклу, значення переміщення за допомогою додаткової функції перетворюється у 7значний двоїчний код та записується до пам'яті контролера.

Проблема таких перетворень була у не відповідності отриманих даних зі значеннями сигналів камер багатопозиційного приводу. На виході отримане перетворене значення було реверсивно записано до пам'яті та набувало східних значень з сигналами камер приводу.

На старті відпрацювання циклограми отримане перетворене значення (приклад – [0,1,0,1,0,0,0]) передавалось у цикл з 0 ітератором та кінцевою умовою – ітератор не має бути більше 7, так як привід має 7 вхідних значень. Таким чином якщо ітератор = 1 перетворене значення у цій позиції = 0, що є також логічним сигналом 0 до зсувного регістру у 1 контакт. Згідно з наростаючим ітератором цикл перебирає все перетворене значення у кожній позиції та посилає логічні сигнали до зсувного регістру на контакт який дорівнює ітератору циклу.

Загальний алгоритм відпрацювання програми наведений на рисунку 8.1.2



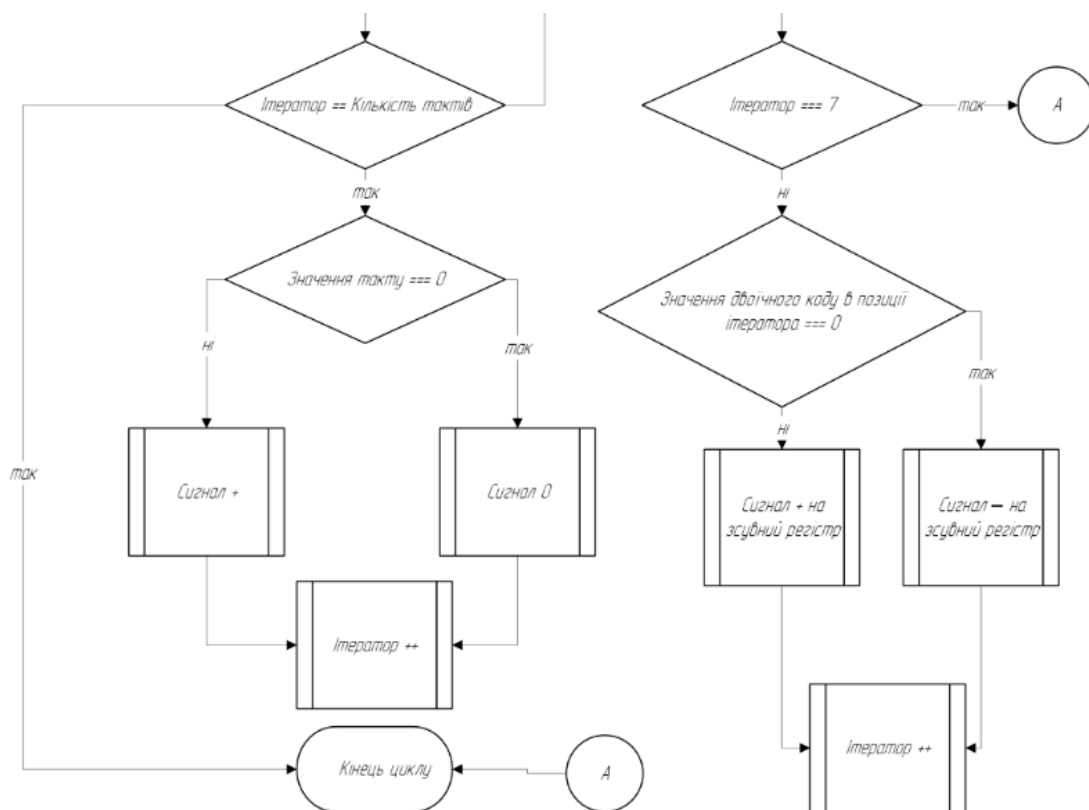


Рисунок 8.1.2 – Загальний алгоритм відпрацювання програми

Для більш детального розуміння роботи перетворення та посилення сигналів приведено приклад зіставлення сигналів регістру та перетвореного значення з мм. Ввід користувача = 5 , перетворений сигнал = 1001000.

ВИХІД ШТОКА НА 5 ММ						
1	2	3	4	5	6	7
1мм	2мм	4мм	8мм	16мм	32мм	64мм
1	0	0	1	0	0	0

9 ВИСНОВКИ

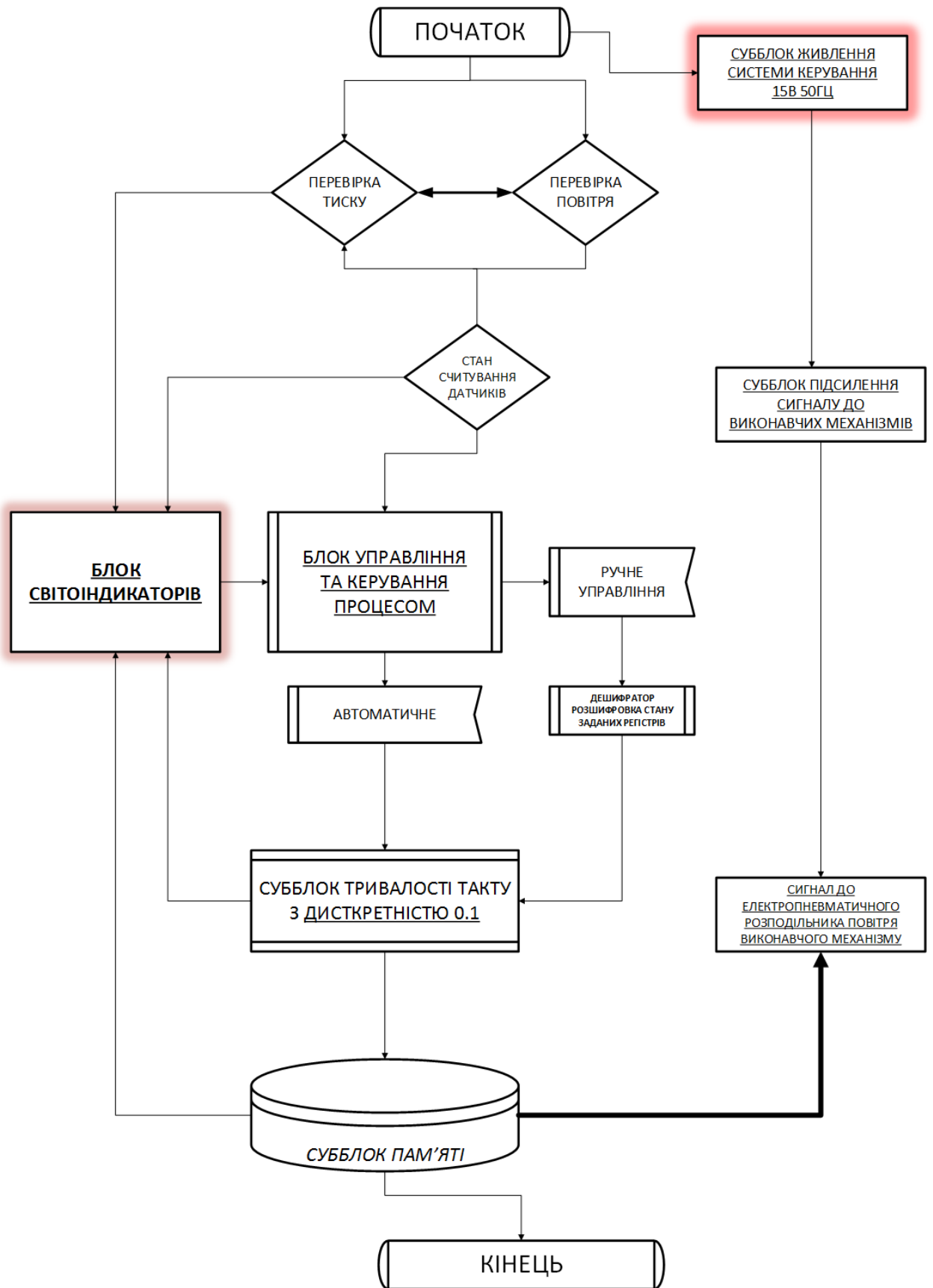
У даній магістерській роботі було розроблено оновлений варіант системи керування на базі платформи Arduino. Оновлена система керування має можливість керування лінійними пневматичним приводом робота а також багатопозиційним дискретним приводом. На основі цього було запропонований варіант використання багатопозиційного приводу у промисловому роботі БРИГ10-ЗА3 в якості приводу руки робота. Цей варіант оновлення розширив технологічні можливості більш точного позиціонування привода руки та дискретної величини переміщення з 1 до 127 мм. Розширення кодової бази також дає подальші можливості для доповнення або переходу на іншу платформу з використанням розробленого алгоритму програми. Було складена нова принципова схема з'єднання та взаємодії привода з системою керування. Розширена схема дає можливість використання більшої кількості багатопозиційних приводів а також більш точної відпрацюванні циклограми додаванням оберненого зв'язку у вигляді датчиків.

ПЕРЕЛІК ПОСИЛАНЬ

1. Е.И. Юревич .Основы робототехники. БХВ-Петербург, 2005. — 252 с.
2. Brian W. Evans .Arduino блокнот програмиста. Creative-Commons, 2007. — 252 с
3. А.И. Корендясев, С.В. Левин. Принципы построения захватных устройств с рекуперацией энергии. Проблемы машиностроения и автоматизации, 1990, № 3.
4. П.Д. Крутько. Управление исполнительными системами роботов. М., Наука, 1991.
5. Е.С. Пятницкий. Принципы декомпозиции в управлении механическими и электромеханическими системами. Сборник трудов “Синтез систем управления манипуляционными роботами на принципе декомпозиции”. М., ИПУ, 1987.
6. Е.И. Юревич. Перспективы развития модульного принципа построения средств робототехники. Стандарты и качество. М., № 5, 1989.
7. Бокселл Д. Изучаем Arduino (Ардуино). Стандарты и качество. М., № 5, 2017.
8. Том Иго. Arduino, датчики и сети для связи устройств. БХВ-Петербург., № 3, 2015.

ДОДАТОК А

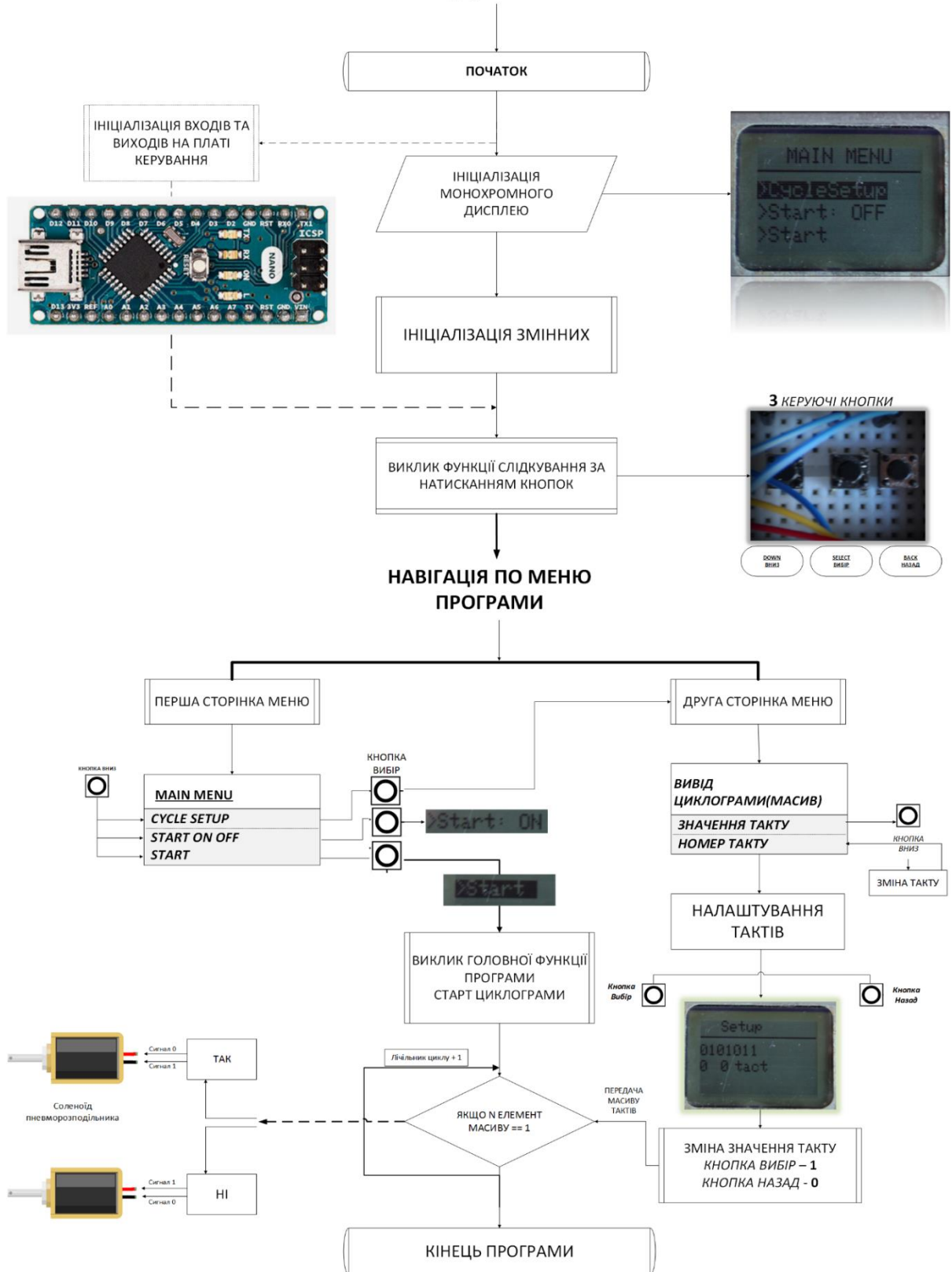
Алгоритм роботи аналогової системи керування



ДОДАТОК Б

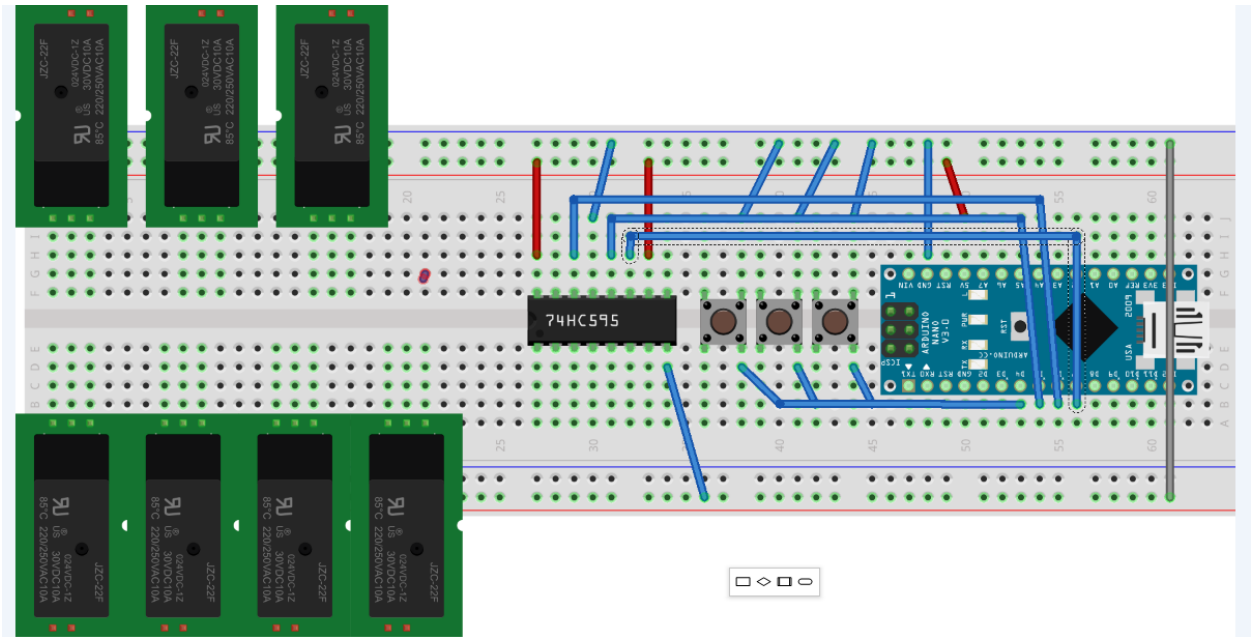
Алгоритм роботи цифрової системи керування

Алгоритм Роботи Системи Керування



ДОДАТОК В

Схема підключення системи керування до зсувного реєстру



ДОДАТОК Г

Лістинг програмного забезпечення

```
#include <Adafruit_GFX.h>
#include <Adafruit_PCD8544.h>

#include "Arduino.h"
// Software SPI (slower updates, more flexible pin options):
// pin 3 - Serial clock out (SCLK)
// pin 4 - Serial data out (DIN)
// pin 5 - Data/Command select (D/C)
// pin 6 - LCD chip select (CS)
// pin 7 - LCD reset (RST)
Adafruit_PCD8544 display = Adafruit_PCD8544(3, 4, 5, 6, 7);
# define LED_1_PIN 10
# define LED_2_PIN 11
# define LED_3_PIN 12
# define LED_4_PIN 13
# define LED_5_PIN 14
# define LED_6_PIN 15
# define LED_7_PIN 16
# define BUTTON_PIN1 2
# define BUTTON_PIN2 8
# define BUTTON_PIN3 9
#define clock 12
#define data 11
#define latch 10
int n = 10; //tacts
int selectdrive = 0;
int selectrepeat = 1;

int asd[] = {1, 1, 1, 1, 1, 1, 1, 1, 1, 1}; //cycle array

int asd1[] = {0, 1, 0, 0, 0, 0, 0, 0, 0, 0}; //cycle array
int asd2[] = {0, 1, 0, 0, 0, 0, 0, 0, 0, 0}; //cycle array
boolean cycle = true; //ready to start
int page = 1;
int menuitem = 1;
int i;
int xArray = 0;
```

```

boolean states[8];

class Button {
  private:
    byte _pin;
  public:

    int lastButtonState;
    boolean click;
    void checkPress() {

      if (digitalRead(_pin) != lastButtonState) {

        if (digitalRead(_pin) == 0) {
          click = true;
        }
        delay(50);
      }
      lastButtonState = digitalRead(_pin);
    }
    void setPin(byte pin) {
      _pin = pin;
      pinMode(_pin, INPUT_PULLUP);
    }

}

; Button down
; Button select
; Button back;

void setup() {
  Serial.begin(9600);
  pinMode(LED_1_PIN, OUTPUT);
  pinMode(LED_2_PIN, OUTPUT);
  pinMode(LED_3_PIN, OUTPUT);
  pinMode(LED_4_PIN, OUTPUT);
  pinMode(LED_5_PIN, OUTPUT);
  pinMode(LED_6_PIN, OUTPUT);
  pinMode(LED_7_PIN, OUTPUT);
  display.begin();
}

```

```

display.setContrast(40); //Set contrast to 40
display.clearDisplay();
display.print(" Loading");
delay(1000);
display.display();

// pins for register
pinMode(clock, OUTPUT);
pinMode(data, OUTPUT);
pinMode(latch, OUTPUT);
digitalWrite(latch, HIGH);
// close register lath to get the signal from arduino
cleanreg();
// fill 0 byte to prevent fail signal

down.setPin(BUTTON_PIN1);
select.setPin(BUTTON_PIN2);
back.setPin(BUTTON_PIN3);

}

void loop() {
  down.checkPress();
  select.checkPress();
  back.checkPress();

  digitalWrite(latch, LOW);
  shiftOut(data, clock, LSBFIRST, 0b10000000);
  digitalWrite(latch, HIGH);
  drawMenu(); //Drive select
  if (back.click && page == 1 && menuitem == 1) {
    back.click = false;
    selectdrive++;
    if (selectdrive == 2) {
      selectdrive = 0;
    }
  }
  if (select.click && page == 1 && menuitem == 1) {
    select.click = false;
    page = 2;
  }
  //углубление

```

```

if (down.click && page == 1) {
    down.click = false;
    menuitem++;
    if (menuitem == 4) {
        menuitem = 1;
    }
}
// переключение ON вывод в исходную позицию
if (select.click && back.click && menuitem == 2) {
    if (cycle == false) {
        cycle = true;
    }
    select.click = false;
    back.click = false;
    delay(500);
    digitalWrite(LED_1_PIN, LOW);
    delay(1000);
    digitalWrite(LED_2_PIN, LOW);
    delay(1000);
    digitalWrite(LED_3_PIN, LOW);
    delay(1000);
    digitalWrite(LED_4_PIN, LOW);
    delay(1000);
    digitalWrite(LED_5_PIN, LOW);
    delay(1000);
    digitalWrite(LED_6_PIN, LOW);
    delay(1000);
    digitalWrite(LED_7_PIN, LOW);
    delay(1000);

}
// выбор числа повторений
if (select.click && page == 1 && menuitem == 2) {
    select.click = false;
    selectrepeat++;
    if (selectrepeat == 128) {
        selectrepeat = 1;
    }
}
if (back.click && page == 1 && menuitem == 2) {
    back.click = false;
    selectrepeat--;
    if (selectrepeat == 0) {

```

```

        selectrepeat = 127;
    }
}

if (select.click && back.click && menuitem == 2) {
    if (cycle == true) {
        cycle = false;
    }
    select.click = false;
    back.click = false;

    delay(500);
}
if (cycle && select.click && menuitem == 3) {
    select.click = false;
    startcycle();
}

//2ая страница
if (page == 2 && down.click) {
    down.click = false;
    xArray++;
    if (xArray == n) {
        xArray = 0;
        page = 1;
    }
}

} // void loop

void drawMenu() {

    if (page == 1) {
        display.setTextSize(1);
        display.clearDisplay();
        display.setTextColor(BLACK, WHITE);
        display.setCursor(15, 0);
        display.print("MAIN MENU");
        display.drawFastHLine(0, 10, 83, BLACK);
        display.setCursor(0, 15);

        if (menuitem == 1) {

```



```

        display.setTextColor(WHITE, BLACK);
    } else {
        display.setTextColor(BLACK, WHITE);
    }
    display.print(">Setup Drive");//Drivesetup
    display.setCursor(75, 15); // print Drive select
    display.print(selectdrive);

    display.setCursor(0, 25);

    if (menuitem == 2) {
        display.setTextColor(WHITE, BLACK);
    } else {
        display.setTextColor(BLACK, WHITE);
    }
    display.print(">Drive");

    display.setCursor(35, 25); // print Drive select
    display.print(" mm=");
    display.print(selectrepeat);
    if (menuitem == 3) {
        display.setTextColor(WHITE, BLACK);
    } else {
        display.setTextColor(BLACK, WHITE);
    }
    display.setCursor(0, 35);
    display.print(">Start ");
}
if (page == 2) {
    setupCycle();
} //page 2
//page 2
display.display();
} //drawmenu

void setupCycle() {
    display.setTextSize(1);
    display.clearDisplay();
    display.setTextColor(BLACK, WHITE);
    display.setCursor(10, 0);
    display.print("SetupDrive");
    display.setCursor(75, 0);

```

```

display.print(selectdrive);//selectdrive
display.drawFastHLine(0, 10, 83, BLACK);
display.setCursor(0, 15);

switch (selectdrive) {
  case 0:
    for (int i = 0; i < n; i++) {
      asd[i] = asd1[i];

    }

    break;
  case 1:
    for (int i = 0; i < n; i++) {
      asd[i] = asd2[i];
    }
    break;
}

for (i = 0; i < n; i++) {
  display.print(asd[i]);
}
display.setCursor(0, 35);
switch (xArray) {
  case 0:
    if (select.click) {
      asd[xArray] = 1;
      select.click = false;
    }
    if (asd[xArray] == 1 && back.click) {
      back.click = false;
      asd[xArray] = 0;
    }
    display.print(asd[xArray]);
    display.setCursor(35, 35);
    display.print("tact");
    display.setCursor(15, 35);
    display.print(xArray + 1);
    break;
  case 1:
    if (select.click) {
      asd[xArray] = 1;

```

```

        select.click = false;
    }
    if (asd[xArray] == 1 && back.click) {
        back.click = false;
        asd[xArray] = 0;
    }
    display.print(asd[xArray]);
    display.setCursor(35, 35);
    display.print("tact");
    display.setCursor(15, 35);
    display.print(xArray + 1);
    break;
case 2:
    if (select.click) {
        asd[xArray] = 1;
        select.click = false;
    }
    if (asd[xArray] == 1 && back.click) {
        back.click = false;
        asd[xArray] = 0;
    }
    display.print(asd[xArray]);
    display.setCursor(35, 35);
    display.print("tact");
    display.setCursor(15, 35);
    display.print(xArray + 1);
    break;
case 3:
    if (select.click) {
        asd[xArray] = 1;
        select.click = false;
    }
    if (asd[xArray] == 1 && back.click) {
        back.click = false;
        asd[xArray] = 0;
    }
    display.print(asd[xArray]);
    display.setCursor(35, 35);
    display.print("tact");
    display.setCursor(15, 35);
    display.print(xArray + 1);
    break;
case 4:

```

```

if (select.click) {
    asd[xArray] = 1;
    select.click = false;
}
if (asd[xArray] == 1 && back.click) {
    back.click = false;
    asd[xArray] = 0;
}
display.print(asd[xArray]);
display.setCursor(35, 35);
display.print("tact");
display.setCursor(15, 35);
display.print(xArray + 1);
break;
case 5:
    if (select.click) {
        asd[xArray] = 1;
        select.click = false;
    }
    if (asd[xArray] == 1 && back.click) {
        back.click = false;
        asd[xArray] = 0;
    }
    display.print(asd[xArray]);
    display.setCursor(35, 35);
    display.print("tact");
    display.setCursor(15, 35);
    display.print(xArray + 1);
    break;
case 6:
    if (select.click) {
        asd[xArray] = 1;
        select.click = false;
    }
    if (asd[xArray] == 1 && back.click) {
        back.click = false;
        asd[xArray] = 0;
    }
    display.print(asd[xArray]);
    display.setCursor(35, 35);
    display.print("tact");
    display.setCursor(15, 35);
    display.print(xArray + 1);

```

```

        break;
case 7:
    if (select.click) {
        asd[xArray] = 1;
        select.click = false;
    }
    if (asd[xArray] == 1 && back.click) {
        back.click = false;
        asd[xArray] = 0;
    }
    display.print(asd[xArray]);
    display.setCursor(35, 35);
    display.print("tact");
    display.setCursor(15, 35);
    display.print(xArray + 1);
    break;
case 8:
    if (select.click) {
        asd[xArray] = 1;
        select.click = false;
    }
    if (asd[xArray] == 1 && back.click) {
        back.click = false;
        asd[xArray] = 0;
    }
    display.print(asd[xArray]);
    display.setCursor(35, 35);
    display.print("tact");
    display.setCursor(15, 35);
    display.print(xArray + 1);
    break;
case 9:
    if (select.click) {
        asd[xArray] = 1;
        select.click = false;
    }
    if (asd[xArray] == 1 && back.click) {
        back.click = false;
        asd[xArray] = 0;
    }
    display.print(asd[xArray]);
    display.setCursor(35, 35);
    display.print("tact");

```

```

        display.setCursor(15, 35);
        display.print(xArray + 1);
        break;
case 10:
    if (select.click) {
        asd[xArray] = 1;
        select.click = false;
    }
    if (asd[xArray] == 1 && back.click) {
        back.click = false;
        asd[xArray] = 0;
    }
    display.print(asd[xArray]);
    display.setCursor(35, 35);
    display.print("tact");
    display.setCursor(15, 35);
    display.print(xArray + 1);
    break;

case 11:
    display.setCursor(25, 35);
    display.print("return");
    if (asd[xArray] == 1 && back.click) {
        back.click = false;
        page = 1;
        xArray = 0;
    }
    break;
}
switch (selectdrive) {
    case 0:
        for (int i = 0; i < n; i++) {
            asd1[i] = asd[i];
        }

        break;
    case 1:
        for (int i = 0; i < n; i++) {
            asd2[i] = asd[i];
        }
        break;
}
}

```

```

}
void startcycle() // функция старт цикла
{
    if (cycle == false) return;
    Serial.println("start cycle");
    byte someValue = selectrepeat;
    char binaryTest[7] = {0};
    someValue += 128;
    itoa(someValue, binaryTest, 2);
    char* string = binaryTest + 1;
    Serial.println(strrev(string));
    char* readyByte = string;

    for (i = 0; i < n; i++) {
        for (int k = 0; k < 7; k++) {
            char someVal = readyByte[k];
            // Serial.println(atoi(someVal));
            if (int(someVal) == 49) {
                sendpin(k, HIGH);
                Serial.println("send high");
            } ;
            if (int(someVal) == 48) {
                sendpin(k, LOW);
                Serial.println("send low");
            };
            delay(50);
        };

        down.checkPress();
        select.checkPress();
        back.checkPress();
        if (down.click || select.click || back.click) {
            cycle = false;
            page = 1;
            return;
        }

        if (asdl[i] == 1) {
            digitalWrite(LED_1_PIN, HIGH);

```

```

    } else {
        digitalWrite(LED_1_PIN, LOW);

    }
    if (asd2[i] == 1) {
        digitalWrite(LED_2_PIN, HIGH);

    } else {
        digitalWrite(LED_2_PIN, LOW);

    }

    delay(2000);

}

return cycle = false;
}

// functions to send signal to register
void sendbyte(byte value) {
    if (Serial.available() > 0) {
        // read the incoming byte:

        // say what you got:
        Serial.print("I received: ");
        Serial.println("send to reg");

    };
    digitalWrite(latch, LOW);
    shiftOut(data, clock, LSBFIRST, value);
    digitalWrite(latch, HIGH);
}

void sendpin(int pin, boolean state) {
    pin--;
    states[pin] = state;

    byte value = 0;
    byte add = 1;

```



```
for (int i = 0; i < 8; i++) {
    if (states[i] == HIGH) value += add;
    add *= 2;
}
digitalWrite(latch, LOW);
shiftOut(data, clock, LSBFIRST, value);
digitalWrite(latch, HIGH);
}

void cleanreg() {
    for (int i = 0; i < 8; i++) states[i] = LOW;
    digitalWrite(latch, LOW);
    shiftOut(data, clock, LSBFIRST, 0);
    digitalWrite(latch, HIGH);
}
```