Національний університет водного господарства та природокористування

(повне найменування вищого навчального закладу)

Навчально-науковий інститут автоматики, кібернетики та обчислювальної техніки

(повна назва інституту)

Кафедра комп’ютерних наук та прикладної математики

(повна назва кафедри, циклової комісії)

Курсова робота

з дисципліни

“Мікроконтролери та їх програмування”

(назва дисципліни)

на тему: Розробка пристрою для відслідковування положення сонця на базі мікроконтролерів Arduino

Студента II курсу ІПЗ-21інт групи

напряму підготовки 121

“Інженерія програмного забезпечення”

Коваля Ігоря Леонідовича

(прізвище та ініціали)

Науковий керівник доцент Ярощак С. В.

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Національна шкала:

Кількість балів:

ЗМІСТ

[ВСТУП 3](#_Toc74741257)

[Розділ 1. Аналіз аналогів та їх недоліків 5](#_Toc74741258)

[1.1. Arduino Solar Tracker за авторством Pak World 5](#_Toc74741259)

[1.2. Arduino Solar Tracker за авторством Aboubakr El Hammoumi та Saad Motahhir 6](#_Toc74741260)

[Розділ 2. Вибір платформи розробки 8](#_Toc74741261)

[Розділ 3. Опис обраної плати 10](#_Toc74741262)

[Розділ 4. Розробка алгоритму 13](#_Toc74741263)

[Розділ 5. Розробка схеми підключення 22](#_Toc74741264)

[5.1. Схема програмного підключення 22](#_Toc74741265)

[5.2. Апаратна схема підключення 25](#_Toc74741266)

[Розділ 6. Вибір компонентів пристрою 27](#_Toc74741267)

[6.1. Вибір плати Arduino 27](#_Toc74741268)

[6.2. Вибір сервопривода 27](#_Toc74741269)

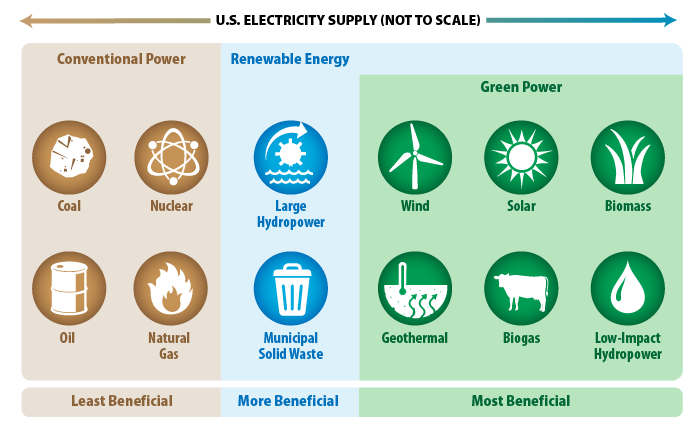
[6.3. Вибір сонячної панелі 28](#_Toc74741270)

[6.4. Вибір резистора 29](#_Toc74741271)

[ВИСНОВОК 30](#_Toc74741272)

[СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ 31](#_Toc74741273)

# **ВСТУП**

В сучасному суспільстві незалежно ми можемо спостерігати значний ріст зацікавленості в «зеленій енергії». Зелена енергія, або потужність (від англ. “Green power”) – підмножина відновлюваної енергії та представляє ті відновлювані енергетичні ресурси та технології, які забезпечують найвищі екологічні переваги. Американський добровільний ринок визначає зелену енергію як електроенергію, вироблену із сонячної енергії, вітру, геотермальної енергії, біогазу, прийнятної біомаси та малих гідроелектричних джерел з невеликим впливом.

Відновлювана енергія включає ресурси, які покладаються на джерела палива, які відновлюються протягом коротких періодів часу і не зменшуються. До таких джерел палива належать сонце, вітер, вода, що рухається, органічні рослини та відходи (біомаса, що відповідає вимогам), і земне тепло (геотермальна)[1].

Зусередимося на сонячній енергії. Вона видобувається за допомогою фотоелеменетів, в якості яких зазвичай використовуються кремнієві сонячні батареї. Ці батареї отримують максимум енергії сонця лише тоді, коли сонячні промені потрапляють на фотоелемент під прямим кутом. Звісно, статично встановлені сонячні батареї більшість часу не зможуть потрапляти під прямі сонячні промені, так як сонце рухається. Отже, потрібен спосіб відслідковувати позицію сонця на небі для повороту сонячних батарей та максималізації отримуваної електроенергії.

**Актуальність теми** розробки пристрою для відслідковування сонця полягає швидкому рості попиту на зелену електроенергію та необхідності максимізації її видобутку.

**Метою роботи** є вивчення способів та методів розробки програмного забезпечення для мікроконтролерів Arduino.

**Предметом роботи** є розробка схеми та алгоритму роботи пристрою для відслідковування сонця на базі Arduino.

**Об’єктом роботи** є мікроконтролери апаратної платформи Arduino, методи та способи розробки апаратних продуктів на їх основі.

**Основними завданнями** відповідно до мети є:

* Ознайомлення з особливостями розробки апаратних продуктів на базі Arduino
* Розробка алгоритму відслідковування сонця
* Розробка схеми підключення елементів до плати Arduino

# **Розділ 1. Аналіз аналогів та їх недоліків**

## 1.1. Arduino Solar Tracker за авторством Pak World

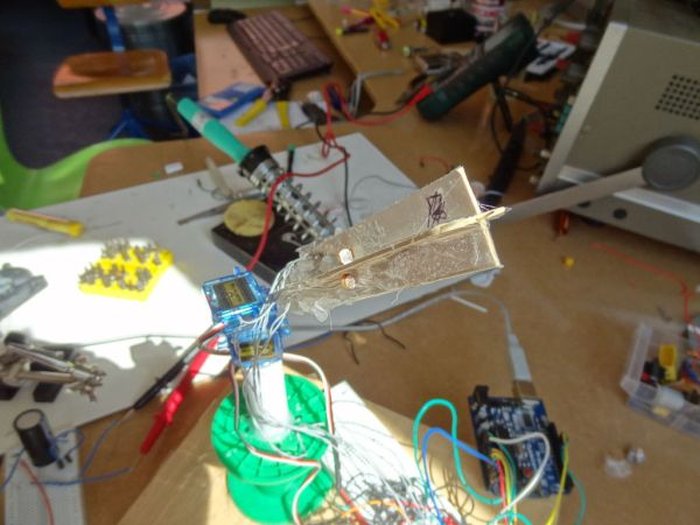
Пристрій відслідковує найсвітліше джерело світла, яким зазвичай є сонце. Між чотирма сенсорами освітленості розміщено дерев’яну перегородку, тінь від якої створює необхідну для розрахунку розташування джерела світла різницю в показаннях давачів. Пристрій задає двом сервоприводам необхідні кути нахилу в залежності від сенсора, який показує найбільше значення[2].

Рис. 1.1. Arduino Solar Tracker за авторством Pak World

Переваги:

* Дешевизна.

Недоліки:

* Висока чутливість до джерел штучного освітлення;
* Використання великої кількості сенсорів.

Використання звичайних сенсорів освітленості дозволила здешевити конструкцію, але збільшила ймовірність некоректної поведінки при наявності джерел штучного освітлення.

## 1.2. Arduino Solar Tracker за авторством Aboubakr El Hammoumi та Saad Motahhir

Пристрій, який може автоматично обертатися для відстеження сонця за допомогою чотирьох датчиків LDR та двох сервомоторів (SM1 та SM2),

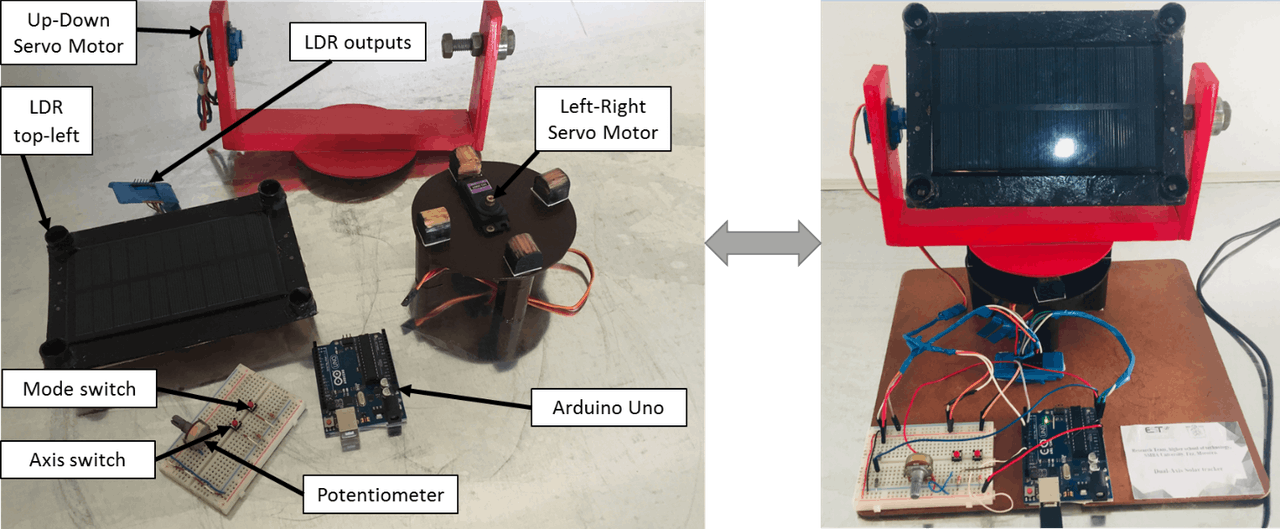
Для горизонтальної осі підшипник закріплений паралельно сервомотору вгору-вниз для кращої гнучкості. Сонячний трекер розроблений таким чином, щоб мати два ступені свободи: зі сходу на захід ліворуч-праворучним сервомотором і з півдня на північ сервомотором вгору-вниз. Датчики LDR розміщені в чотирьох кутах фотоелектричної панелі і розміщені в темних трубках з невеликим отвором на вершині для виявлення освітлення сонця. Ці темні трубки також вважаються концентратором випромінювання і використовуються для підвищення стійкості сонячного трекера[3].

Рис. 1.2. Arduino Solar Tracker за авторством Aboubakr El Hammoumi та Saad Motahhir

Переваги:

* Можливість прямого керування

Недоліки:

* Необхідність підключення до ПК;
* Чутливість до джерел штучного освітлення;
* Використання великої кількості сенсорів.

Можна побачити, що більшість аналогів використовують чотири сенсори освітленості. Ці сенсори є досить дешевими, але вони не бачать різниці між натуральним та штучним освітленням, що може призвести до небажаної поведінки.

Для покращення технології необхідно зменшити чутливість до штучного освітлення. Чутливість до джерел штучного освітлення може бути зменшена при використанні сонячної батареї замість сенсорів освітленості. Сонячна батарея створена для поглинання енергії сонячних променів, тому ключовим фактором є не яскравість освітлення, а саме його джерело. Використання сонячних батарей замість сенсорів освітлення призведе до суттєвого виросту ціни приладу. Для запобігання цьому необхідно зменшити кількість сенсорів.

Постає задача розробити алгоритм відслідковування сонця за допомогою лише одного сенсору.

# **Розділ 2. Вибір платформи розробки**

Зусередимо вибір середовища розробки на трьох варіантах:

**Arduino IDE –** Офіційне середовище розробки Arduino з відкритим кодом. Підтримує всі плати Arduino та має велику кількість вбудованих бібліотек. Основна мова програмування – C++.

**Tinkercad** – це безкоштовна онлайн-колекція програмних засобів, які допомагають людям у всьому світі думати та творити.

Tinkercad дає можливість створювати та візуалізувати електричні кола, тестувати підключення пристроїв та елементів схеми до плати Arduino та писати програмний код для розгортання на платі Arduino. Має ряд готових схем, які можна використовувати в якості прикладів, але кількість доступних для підключення елементів досить низька і середовище програмування не зручне.

**Simulink** – це середовище для проектування блок-схем на основі моделей. Simulink підтримує моделювання, автоматичну генерацію коду та постійне тестування вбудованих систем[4].

Simulink має можливість підключення зовнішніх блоків, один з яких(Simulink Support Package for Arduino) дає можливість додавати до блок-схеми елементи входу-виходу Arduino та іншу периферію, що додає до загального функціоналу досить непогану підтримку програмування для мікроконтролерів. Simulink Support Package for Arduino не підтримує всі плати Arduino і має досить вузький вибір елементів схеми, проте має можливість перевірити роботу коду без розгортання на платі та можливість комбінування візуального програмування з класичним програмуванням за допомогою C, C++ та Matlab.

Табл. 2.1. Порівняння середовищ розробки програмного забезпечення для мікроконтролерів Arduino

|  |  |  |  |
| --- | --- | --- | --- |
| Середовище розробки | Arduino IDE | Tinkercad | Simulink |
| Підтримка плат Arduino | Всі | Лише частина | Лише частина |
| Підтримка підключення елементів | Повна | Лише частина | Лише частина |
| Зручність розробки | Середня | Середня | Висока |
| Можливість перевірки роботи до розгортання на платі | Ні | Так | Так |

Критичними моментами при виборі середовища розробки для мене є можливість перевірки роботи без розгортання на платі та зручність розробки, тому в якості середовища програмування я обрав Simulink. Також я використаю Tinkercad для створення схеми підключення елементів до плати Arduino.

# **Розділ 3. Опис обраної плати**

При виборі плати необхідно враховувати багато факторів, серед яких можна виділити:

* Кількість аналогових входів та виходів;
* Кількість дискретних входів та виходів;
* Робоча напруга;
* Мікроконтролер;
* Тактова частота;
* Підтримка IDE.

Для даного проекту не є важливими тактова частота та модель мікроконтролера, адже він не вимагає високої точності та швидкості виконання заданого алгоритму. Затримки в роботі помітні не будуть і не позначаться на якості фінального пристрою.

Робоча напруга вже важливіший фактор, від нього залежить схема підключення компоненті. Даний проект не вимагає високого рівня живлення для компонентів плати, сервоприводи працюють при напрузі 5V, тому на практиці до будь-якої широко розповсюджена плата Arduino може підійти, а тому питання робочої напруги можна викреслити з факторів підбору плати.

Серед дійсно важливих факторів залишаються кількість аналогових і дискретних входів/виходів на платі та підтримка цієї плати обраним середовищем розробки.

Даний проект буде включати в себе сонячну панель в якості давача(сенсора) наявності сонячних променів(більша напруга на виході панелі означає більшу кількість фотонів на поверхні сонячної батареї), тому плата Arduino повинна мати мінімум один аналоговий вхід для приймання даних з сенсора.

Для вмикання пристрою необхідно мати дискретний перемикач, підключення якого буде реалізовано через дискретний вхід.

Для високого рівня свободи в ротації сенсора необхідно буде підключити два Servo пристрої, які будуть шукати оптимальний кут з максимальною напругою на вході. Для підключення цих пристроїв потрібні два дискретні виходи з підтримкою ШІМ(ШІМ – Широтно-імпульсна модуляція – процес керування шириною (тривалістю) високочастотних імпульсів[5]). Додатково необхідно використати ще два дискретні виходи з підтримкою ШІМ для виводу оптимального кута. Від перших двох виходів вони будуть відрізнятися тим, що на вивід будуть подаватися лише ті дані, що на даний момент вважаються оптимальними, ігноруючи теперішній фактичний кут пристрою, який може часто змінюватися при пошуку оптимального.

Отже, даний проект має наступні вимоги до плати Arduino:

* 1 дискретний вхід;
* 4 дискретні виходи з підтримкою ШІМ;
* 1 аналоговий вхід;
* Плата повинна підтримуватися Simulink’ом.

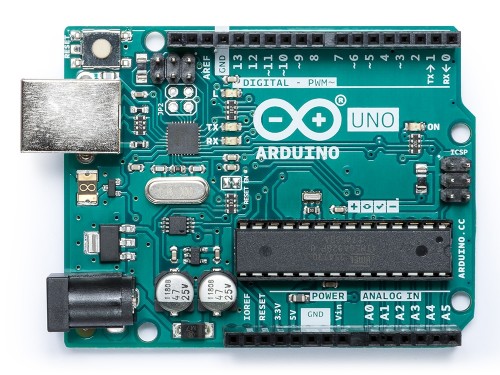
Серед популярних плат під ці вимоги підходить одна з найпоширеніших та найдоступніших плат на ринку мікроконтролерів – Arduino Uno.

Рис. 3.1. Плата Arduino Uno (вид спереду)

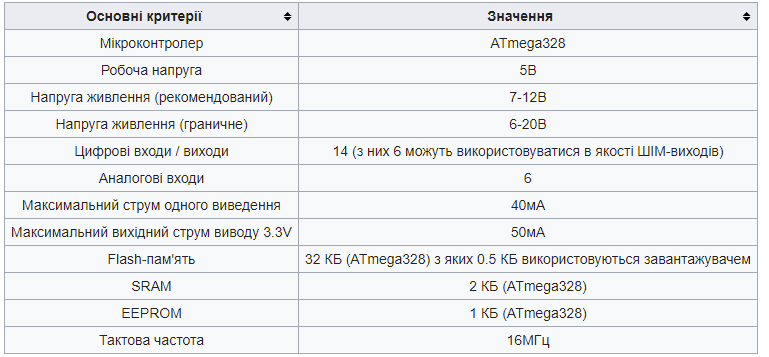
Arduino Uno – мікроконтролер побудований на ATmega328. Платформа має 14 цифрових входів/виходів (6 з яких можуть використовуватися як виходи ШІМ), 6 аналогових входів, кварцовий генератор 16 МГц, роз'єм USB, силовий роз'єм, роз'єм ICSP і кнопку перезавантаження. Для роботи необхідно підключити платформу до комп'ютера за допомогою кабелю USB, або подати живлення за допомогою адаптера AC / DC або батареї.

Рис. 3.2. Основні характеристики плати Arduino Uno[7]

"Uno" означає «один» італійською мовою. Ця назва була обрана в честь випуску Arduino Software (IDE) 1.0. Плата Uno та Arduino Software (IDE) версії 1.0 були еталонними версіями Arduino, які тепер перейшли до нових версій. Плата Uno - перша в серії USB-плат Arduino та еталонна модель для платформи Arduino[6].

# **Розділ 4. Розробка алгоритму**

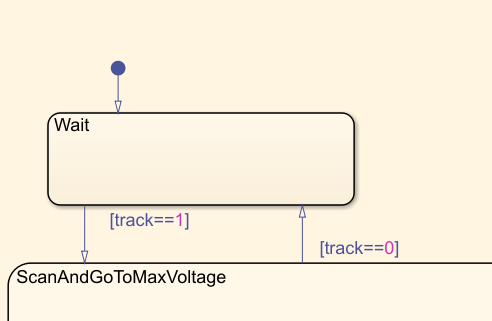
Перш за все необхідно отримати дані з перемикача. В залежності від положення буде виконуватися код одного з режимів. При вимкненому положенні пристрій не нічого робити, а при увімкненні почне роботу. У Simulink це можна реалізувати за допомогою блоків коду. Якщо на момент переходу між двома блоками коду додати умову, перехід відбудеться лише тоді, коли умова виконається. Створимо пустий блок та позначимо його як точку входу в програму, а на момент переходу до наступного блоку додамо умову [track==1], де track – дані з перемикача, що підключений до дискретного входу плати. Таким чином при зміні положення перемикача будуть виконуватися різні блоки коду, один з яких – пустий.

Рис. 4.1. Реалізація режиму простою в Simulink

Перейдемо до алгоритму відслідковування сонця, він буде складатися з наступних пунктів:

1. Вихід на стартову позицію.
2. Початкове сканування всього спектру кутів для пошуку оптимальної початкової позиції.
3. Позиціонування щодо сонця.
4. При зменшенні напруги на вході від сенсору перевірити сусідні кути на факт переміщення сонця. Перейти до пункту №3.

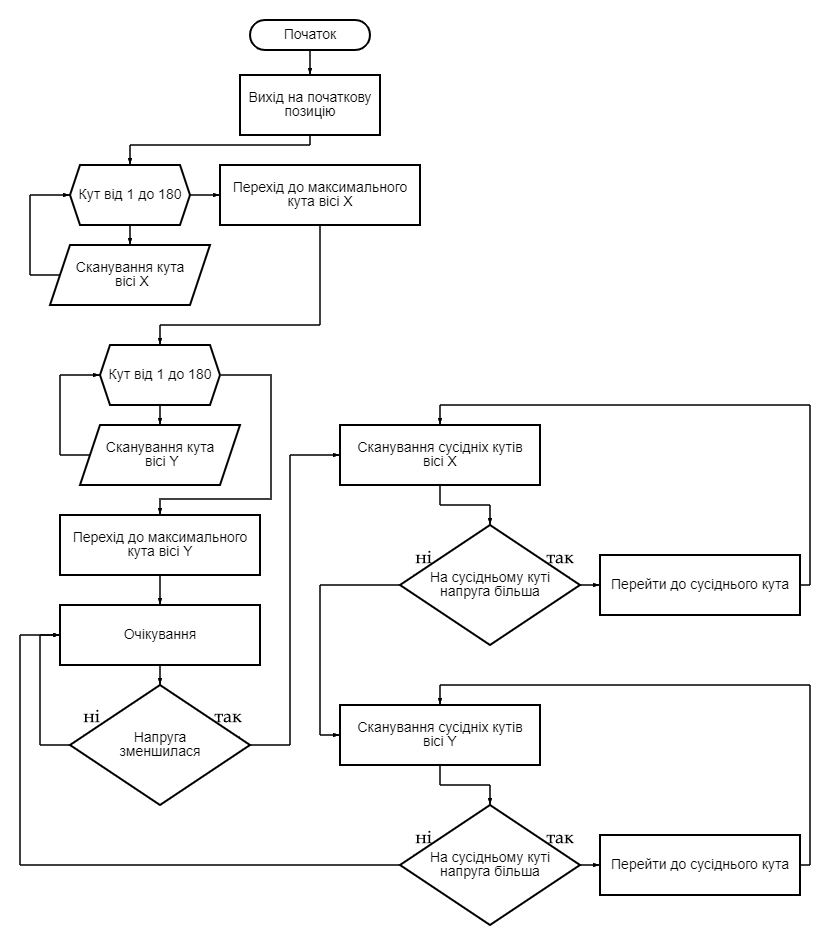
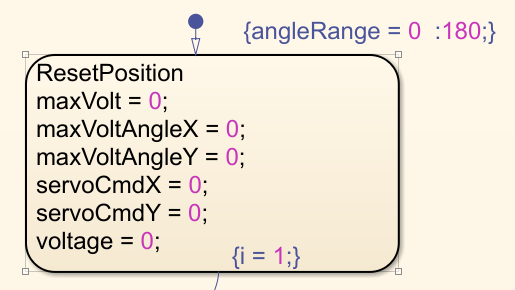
Розробимо блок-схему алгоритму. При розробці загальної блок-схеми не будемо враховувати необхідність виводу результуючих даних.

Рис. 4.2. Блок-схема алгоритму відслідковування сонця

Перейдемо до реалізації. Почнемо з початку, тобто з виходу на початкову позицію.

Рис. 4.3. Блок ResetPosition

Блок ResetPosition оголошує кілька важливих змінних, які знадобляться в майбутньому та масив кутів, що буде використовуватися при початковому пошуку стартової позиції з максимальною напругою на вході. Також цей блок виставляє обидва сервоприводи на початкову позицію, подаючи на вихід дві змінні рівні нулю.

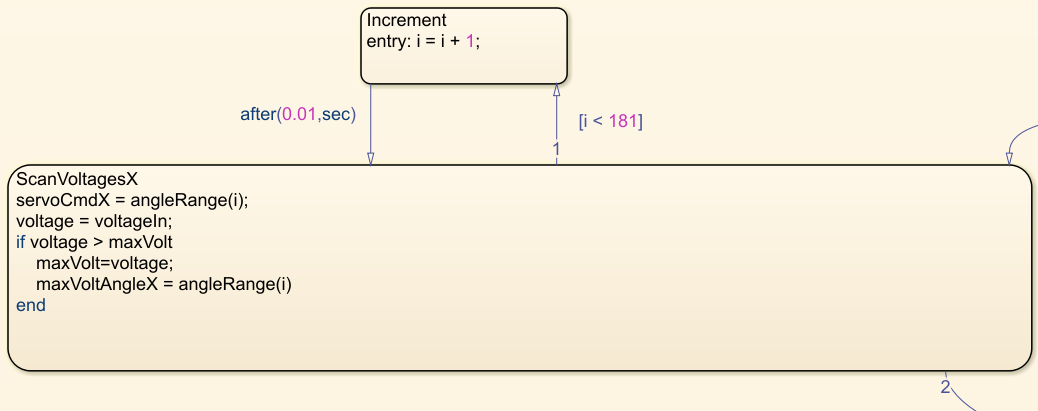
Після того, як обидва сервоприводи були виставлені в початкову позицію, необхідно почати початкове сканування. Для цього використовуються блоки ScanVoltagesX та ScanVoltagesY.

Рис. 4.4. Блок ScanVoltagesX

Блок ScanVoltagesX проходить по кожному зі 180 кутів, зчитує напругу з сенсора і запам’ятовує максимальне значення напруги та положення сервопривода, при якому цього значення було досягнуто.

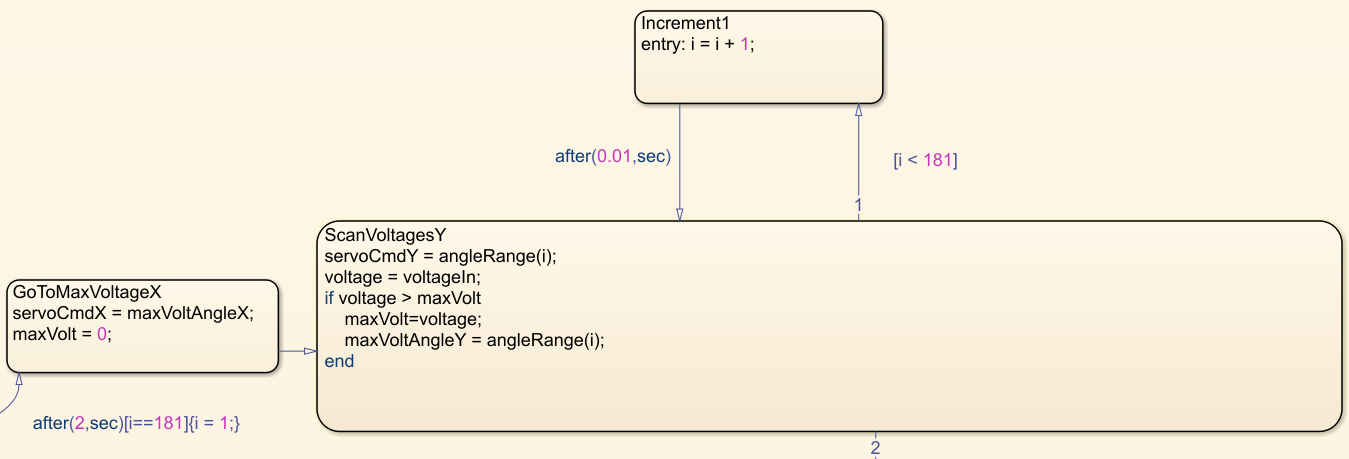
Після того, як всі 180 кутів було проскановано необхідно повернутися до максимального із знайдених значень та почати сканувати положення другого сервопривода. За це відповідають блоки GoToMaxVoltageX та ScanVoltagesY.

Рис. 4.5. Блоки GoToMaxVoltageX та ScanVoltagesY

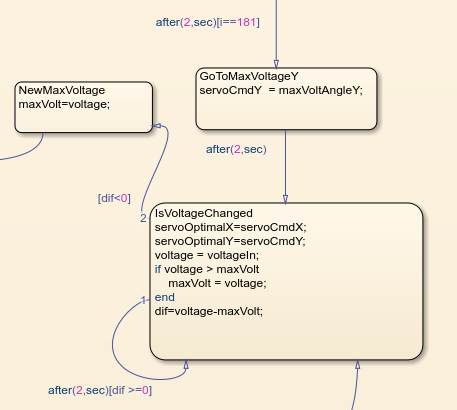
Після невеликої затримки прилад повертає перший сервопривод у положення з максимальною напругою на вході(див. блок GoToMaxVoltageX). Сканування положень другого сервопривода відбувається в блоці ScanVoltagesY аналогічно до блоку ScanVoltagesX. Після сканування всіх позицій другого сервопривода, подібно до блоку GoToMaxVoltageX, за допомогою блоку GoToMaxVoltageY встановлюється положення другого сервопривода. Таким чином обидва сервоприводи налаштовані і разом вказують на точку з найбільшою напругою, іншими словами прилад вказує на сонце.

Рис. 4.6. Блоки GoToMaxVoltageY, IsVoltageChanged та NewMaxVoltage

На цьому процес початкового налагодження закінчений і можна перейти до циклічної частини алгоритму, яка відповідає за перевірку положення сонця та коригування стану пристрою у відповідності до нових даних.

Серцем алгоритму слугує блок IsVoltageChanged(див. рис. 4.6). Спершу цей блок подає на вихід дані двох сервоприводів та налаштовує їх на позицію, яка на даний момент вважається оптимальною. Для цього використовуються вихідні змінні ServoOptimalX та ServoOptimalY, які пізніше підключаться до двох дискретних виходів плати з підтримкою ШІМ(докладніше про ШІМ можна прочитати в розділі 3). Від сусідніх змінних ServoCmdX та ServoCmdY, які теж подають дані на вихід до сервоприводів, оптимальні змінні відрізняються тим, що на них подаються лише дані, які вважаються оптимальними, ігноруючи рухи сервоприводів, що використовуються для перевірки зміни положення сонця. Іншими словами, якщо прилад запідозрить зміну положення сонця та почне регулярно змінювати положення власних сервоприводів для пошуку нової оптимальної позиції, два інших сервовиходи все ще будуть показувати старе оптимальне положення, доки не буде доведено його зміну. Це дозволить зробити загальну конструкцію стабільнішою, не рухаючи зайвий раз підключений пристрій. Таким пристроєм, наприклад, може бути більш масивна та потужна сонячна батарея, яку не бажано рухати без зайвої необхідності.

Після оновлення поточної оптимальної позиції блок IsVoltageChanged знову зчитує дані сенсора і порівнює ці нові дані з останніми, що були записані до цього. Якщо прилад побачить зменшення напруги на вході, алгоритм перейде до перевірки сусідніх кутів на наявність більшої напруги, попередньо записавши поточне значення напруги. В протилежному випадку, прилад зробить паузу на кілька секунд та почне роботу блоку IsVoltageChanged заново.

Перевірка оптимальності положення першого сервопривода відбувається у блоці CheckNearAnglesX.

Рис. 4.7. Блок CheckNearAnglesX

Завдання цього блоку – перевірити два сусідні кути вісі, за яку відповідає перший сервопривід. Алгоритм роботи цього блоку наступний:

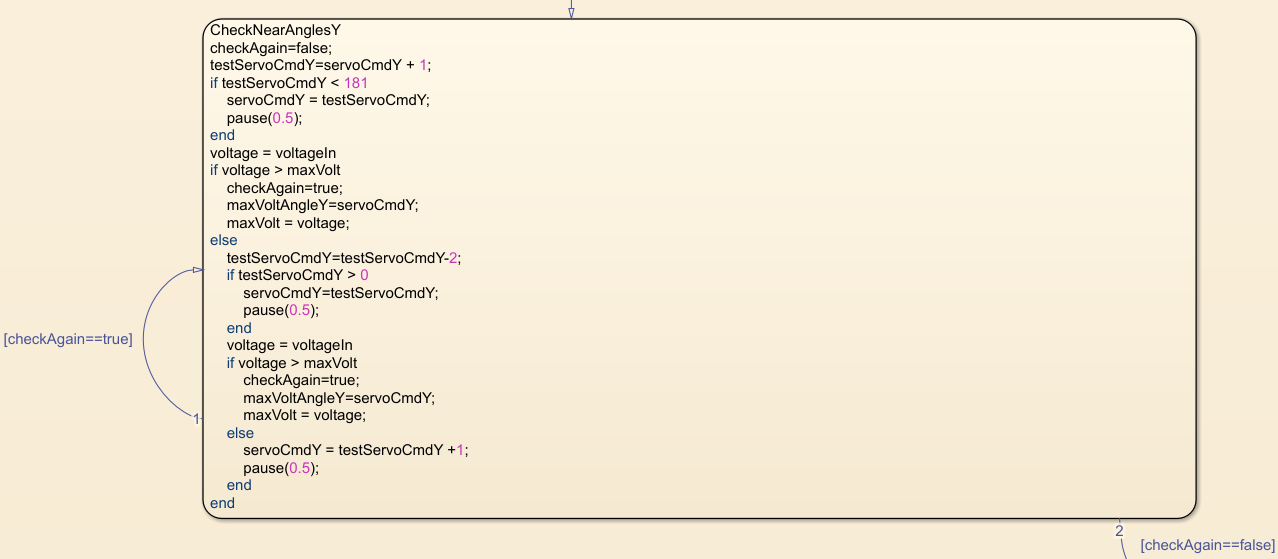
1. Перейти до сусіднього більшого кута.
2. Провести сканування напруги. Якщо напруга більша тої, що вважається максимальною, запам’ятати нову максимальну напругу та положення сервопривода, перейти до пункту 1. Інакше перейти до пункту 3.
3. Перейти до кута на 2 меншого за теперішній(враховуючи збільшення кута при виконанні пункту 1, це означає встановлення кута на 1 менше поточного оптимального).
4. Провести сканування напруги. Якщо напруга більша тої, що вважається максимальною, запам’ятати нову максимальну напругу та положення сервопривода, перейти до пункту 1. Інакше перейти до пункту 5.
5. Перейти до сусіднього більшого кута, тобто до початкового значення на момент початку виконання блоку. Перейти до наступного блоку.

Результатом виконання може бути наступним:

* Кут положення першого сервопривода збільшився;
* Кут положення першого сервопривода зменшився;
* Кут положення першого сервопривода не змінився.

Можна також помітити кілька додаткових перевірок, які використовують локальну змінну testServoCmdX. Вони перевіряють, чи нове значення кута сервопривода не виходить за межі [1:180].

Не можна також забувати за необхідність очікування нового значення напруги на вході після зміни положення сервопривода, адже як процес повертання сервопривода, так і процес збирання сонячної енергії сенсором, в ролі якого виступає сонячна батарея, потребує часу. Через це необхідно регулярно ставити алгоритм на паузу. Для паузи між блоками використовується функція after(), а для паузи всередині блоку було використану функцію pause().

Рис. 4.8. Блок CheckNearAnglesY

Блок CheckNearAnglesY виконує аналогічну до блоку CheckNearAnglesX задачу за таким же алгоритмом. Єдина різниця – на цей раз скануються сусідні позиції другого сервопривода. Відповідно, на заміну змінним testServoCmdX, servoCmdX та maxVoltAngleX приходять змінні testServoCmdY, servoCmdY та maxVoltAngleY.

В обох блоках, CheckNearAnglesY та CheckNearAnglesX, повторна перевірка сусідніх кутів використовується за допомогою булевої змінної checkAgain, яка отримує значення false на початку блоку та змінює його на true, якщо в результаті виконання блоку сервопривід змінив своє положення. Це робиться через те, що теоретично оптимальний кут міг переміститися більше ніж на однин крок сервопривода. В залежності від значення цієї змінної, виконання блоку може початися заново або алгоритм перейде до наступного блоку в черзі.

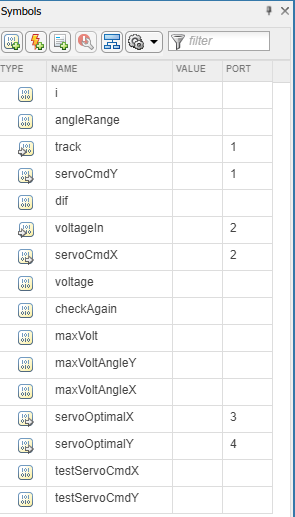
Після виконання блоку CheckNearAnglesY алгоритм повертається до блоку IsVoltageChanged та прдовжує очікувати зменшення напруги на вході, тим самим замикаючи циклічну частину алгоритму, яка має вигляд IsVoltageChanged 🡪 NewMaxVoltage 🡪 CheckNearAnglesX 🡪 CheckNearAnglesY 🡪 IsVoltageChanged.

Рис. 4.9. Список використаних в реалізації алгоритму змінних

Можемо побачити, що всі змінні у Simulink діляться на три типи:

* Локальні змінні;
* Вхідні змінні;
* Вихідні змінні.

Біля змінної кожного типу можна побачити знак типу, де відсутність «стрілочки» означає, що змінна локальна, а її наявність зліва чи справа означає відповідно вхідну та вихідну змінну.

Змінні track та voltageIn необхідно позначити як вхідні, їх значення буде залежати від даних на портах входу та будуть напряму впливати на поведінку пристрою.

Змінні servoCmdX, servoCmdY, servoOptimalX та servoOptimalY необхідно позначити як вихідні. Це дозволить використовувати їх не лише всередині алгоритму, а і вивести їх за межі програмного блоку. Таким чином ми зможемо постійно зчитувати з них дані та передавати до портів на платі.

Можна помітити, що в списку змінних у графі «порт» деякі значення повторюються. Це відбувається через те, що нумерування вхідних та вихідних змінних починається заново з одиниці. Ці порти ніяк не пов’язані з портами входу/виходу на платі Arduino Uno і використовуються лише для нумерації змінних в програмному середовищі. Вибір відповідних до вхідних та вихідних змінних алгоритму портів на платі Arduino Uno відбудеться на етапі розробки програмної схеми підключення описаного в цьому розділі алгоритму до плати Arduino. Для цього буде використаний *Simulink Support Package for Arduino Hardware*.

# **Розділ 5. Розробка схеми підключення**

## 5.1. Схема програмного підключення

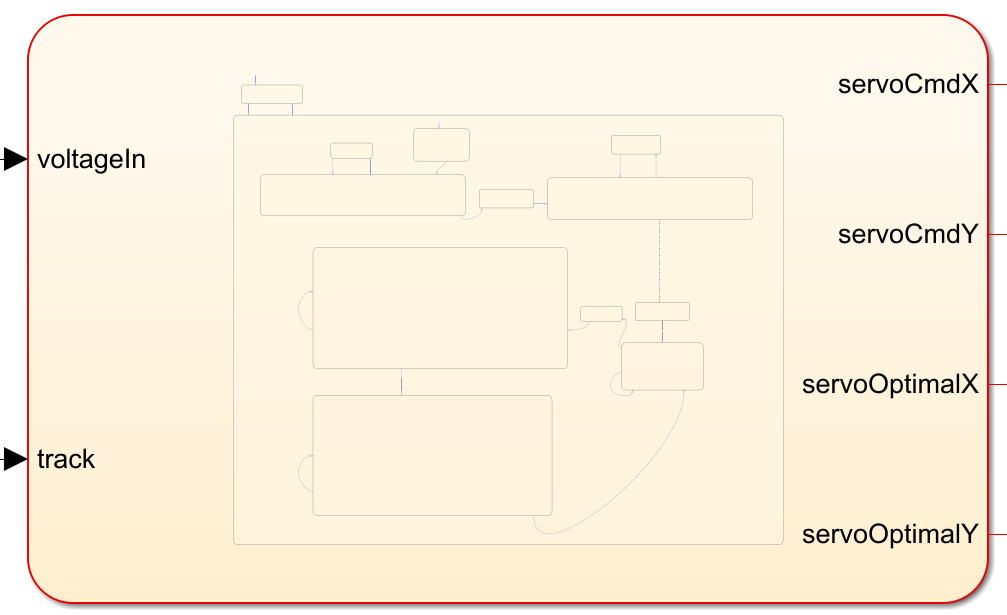
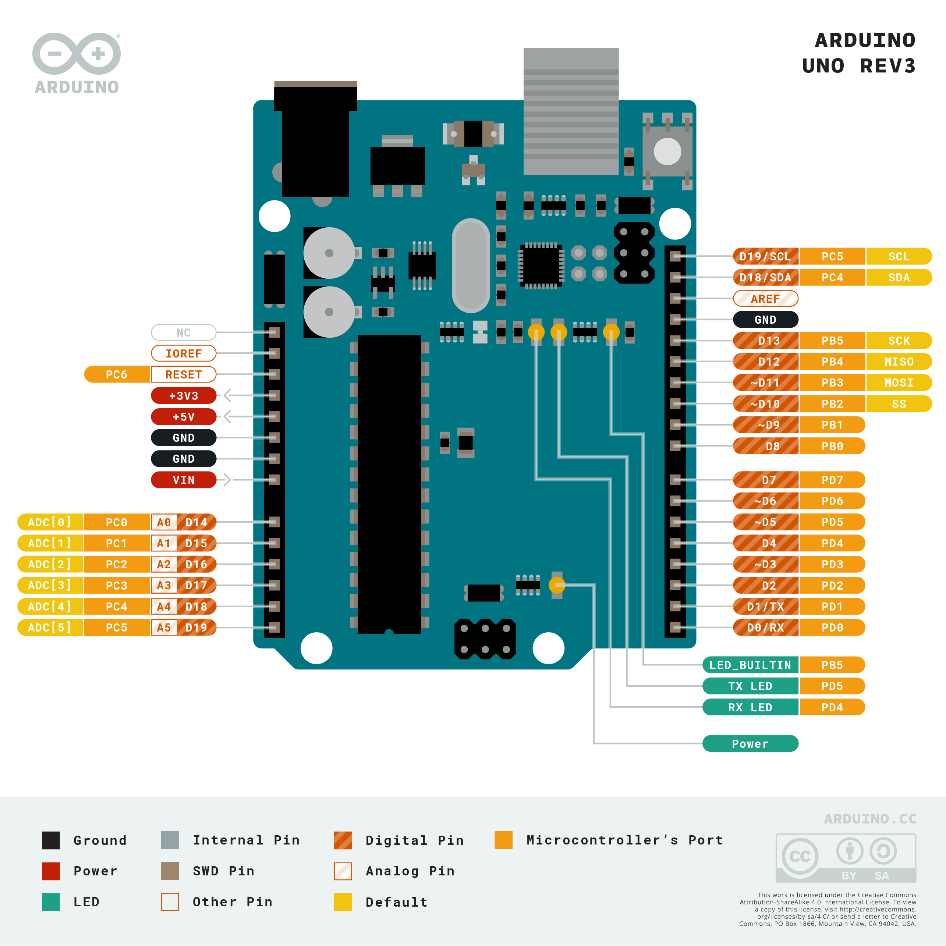
Алгоритм, що був розроблений в розділі 4, був реалізований в рамках одного блоку. Окрім локальних змінних, в ньому були використані дві вхідні змінні – напруга від сенсора та положення перемикача, та чотири вихідні змінні – виходи сервоприводів, два з яких використовуються для керування самим пристроєм, і ще два для можливості підключення іншого пристрою, що буде використовувати знайдені оптимальні значення кутів.

Рис. 5.1. Блок алгоритму роботи пристрою

Для підключення програмного блоку, що містить в собі алгоритм роботи приладу, необхідно використати компоненти з набору *Simulink Support Package for Arduino Hardware*, а саме:

* Один Analog Input;
* Один Digital Input;
* Чотири Servo Motor.

При підключенні Servo Motor Simulink автоматично призначить порти з підтримкою ШІМ.

Для підключення Analog Input та Digital Input необхідно вручну обрати порт. Щоб дізнатися, які з портів плати підтримують дискретний та аналоговий вхід можна скористатися офіційною документацією(рис.5.2) або заглянути в Pin Map Simulink’а(рис.5.3).

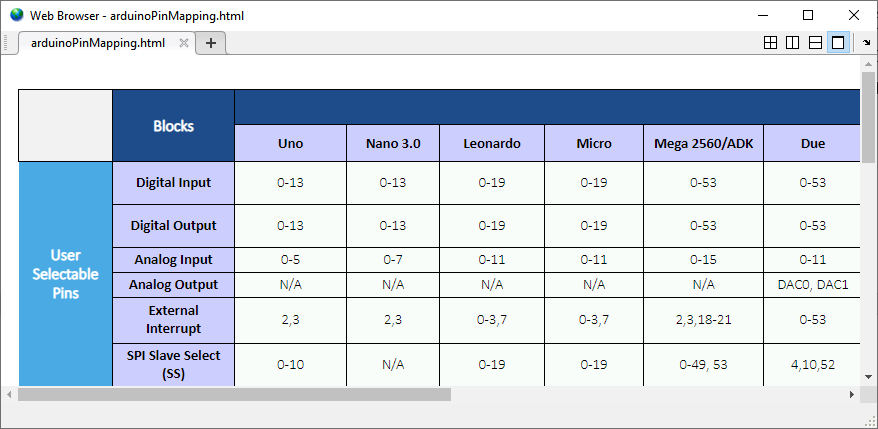
Рис. 5.2. Офіційна діаграма розпіновки Arduino Uno

Рис. 5.3. Допоміжна таблиця User Selectable Pins для кількох розповсюджених плат Arduino

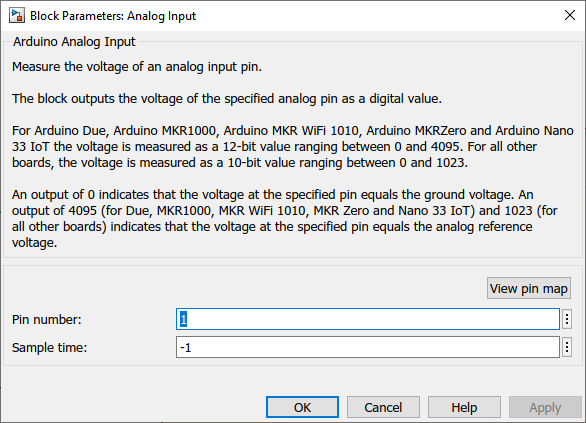
Бачимо, що для Digital Input на платі Arduino Uno можна використати 14 портів, а для Analog Input доступно лише шість.

Рис. 5.4. Вибір порта підключення аналогового входу

Для Analog Input оберемо порт 1, зі списку доступних портів для аналогового входу. Значення Sample time відповідає за частоту оклику порта платою. Значення «-1» для аналогового входу фактично означає неперервне опитування.

Для Digital Input оберемо порт 2, який може слугувати як аналоговим, так і дискретним входом, що залишить порти 3, 5, 6 і 9 вільними для підключення сервоприводів. Значення Sample time в умовах реалізованого алгоритму буде означати максимальну затримку до вимкнення роботи, так як підключатися до цього порта буде перемикач. Ця затримка не є критичною, тому можна залишити значення за замовчуванням – «0.1».

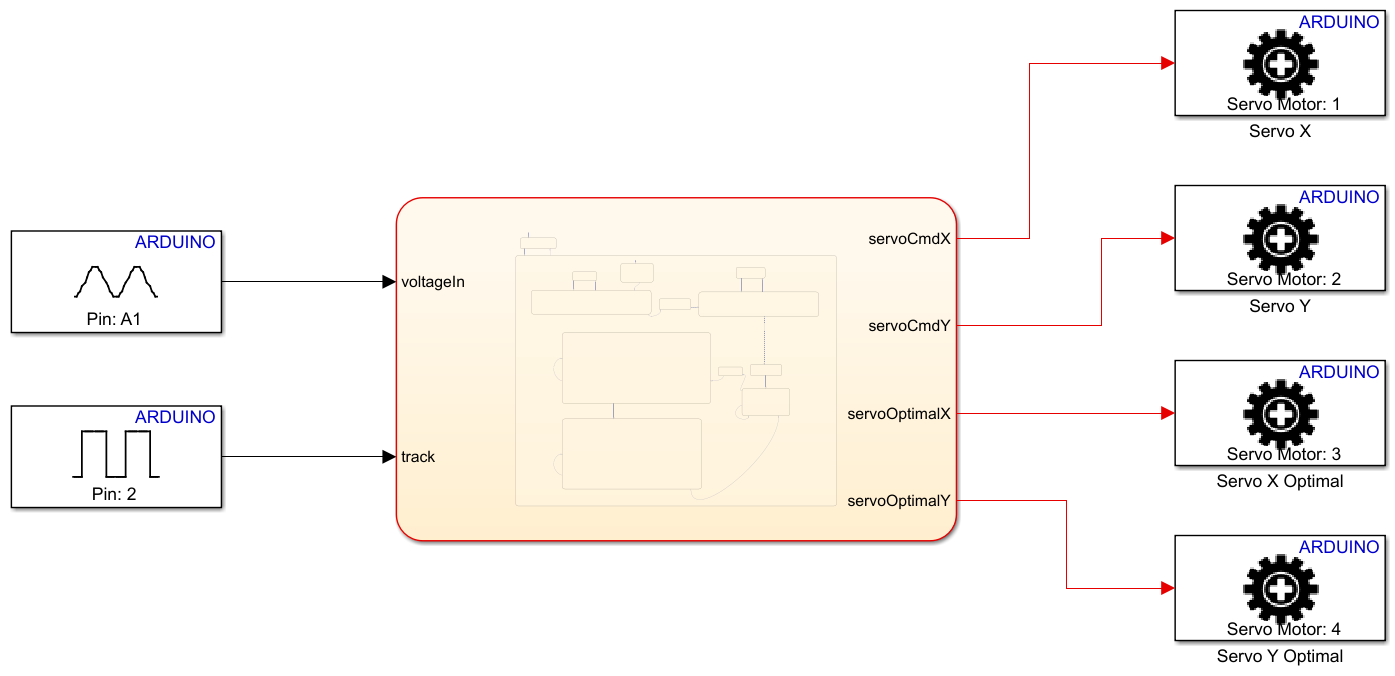
В результаті схема програмного підключення буде мати наступний вигляд:

Рис. 5.5. Схема програмного підключення змінних алгоритму до портів плати Arduino Uno

## 5.2. Апаратна схема підключення

Як було сказано в розділі 2, для розробки апаратної схеми підключення буде використовуватися програмне забезпечення Tinkercad.

Tinkercad маю досить обмежену кількість компонентів. В число відсутніх компонентів, наприклад, входять сонячна батарея та дискретний перемикач, тому для їх умовного позначення буде використаний фоторезистор та кнопку відповідно.

Відповідно до схеми програмного підключення маємо:

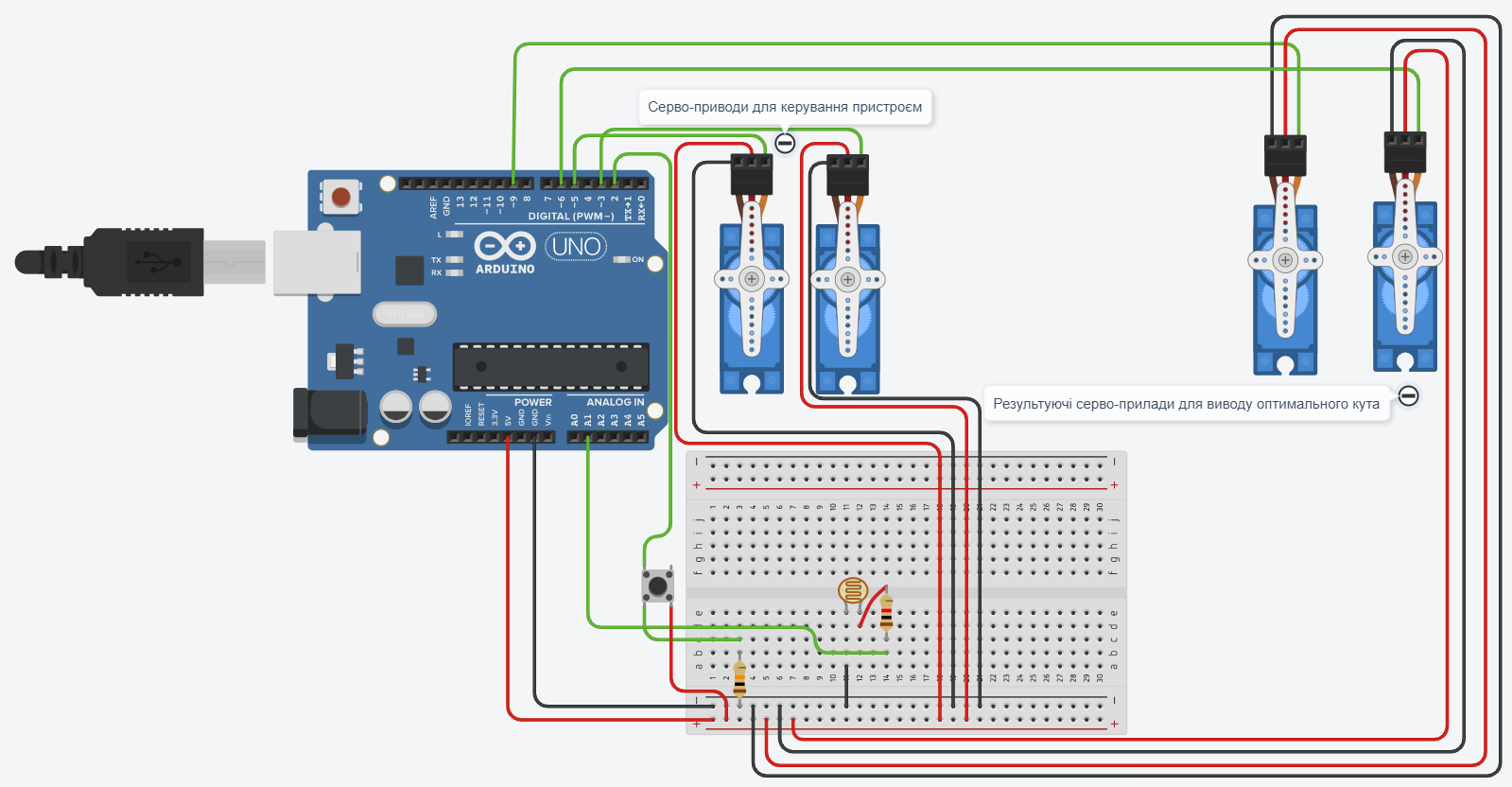
* Перемикач, підключений до дискретного входу 2;
* Сонячну батарею, підключену до аналогового входу 1 через резистор;
* Чотири сервоприводи, підключені до дискретних портів 3, 5, 6 та 9.

Рис. 5.6. Схема підключення компонентів кола

Резистор між аналоговим входом та сонячною батареєю потрібен для того, щоб обмежити максимальну напругу, яку може отримати сонячна батарея, адже повна напруга від сонячної панелі може вивести плату зі строю. Опір цього резистора необхідно буде обирати в залежності від моделі сонячної батареї.

# **Розділ 6. Вибір компонентів пристрою**

## 6.1. Вибір плати Arduino

Серцем пристрою, як вже неодноразово було вказано, буде виступати плата Arduino Uno, а саме, якщо бути більш специфічним, Arduino Uno Rev3 – плата Arduino Uno третьої ревізії, в якій були виправлені помилки попередніх ревізій. Якщо купувати цю плату на офіційному сайті виробника, вона обійдеться у 20 євро, не враховуючи доставку та податки.

Виробник описує плату Arduino Uno як «найкращу плату для початку роботи з електронікою та кодингом»[7], а для підтвердження цих слів була опублікована дуже детальна документація, в якій, між іншим, наявні повна схематика різних рівнів та приклади простих програм, розроблених за допомогою Arduino IDE.

Рис. 6.1. Плата Arduino Uno в офіційному пакуванні

## 6.2. Вибір сервопривода

При виборі сервопривода необхідно звернути увагу на наступні фактори:

* Тип обертання сервопривода;
* Робоча напруга.

Сервоприводи за типом обертання діляться на два види – сервоприводи з обмеженим обертанням та сервоприводи без обмеження на обертання. Сервоприводи без обмеження обертання можуть постійно обертатися в одному напрямку без зупинки, а сервоприводи з обмеженим обертанням, як можна зрозуміти з назви, мають можуть обертатися лише протягом обмеженої кількості кутів, після чого необхідно зупинитися.

Алгоритм, розроблений в розділі 4, розрахований на сервопривід з обмеженням обертання у межах 180 градусів.

Робоча напруга сервопривода повинна бути рівною 5V. Більшість сервоприводів не вимагають якоїсь точної напруги, вони здатні працювати при будь-якій напрузі в заданих межах. Нам необхідно підібрати сервопривід, межі робочої напруги якого включають 5V.

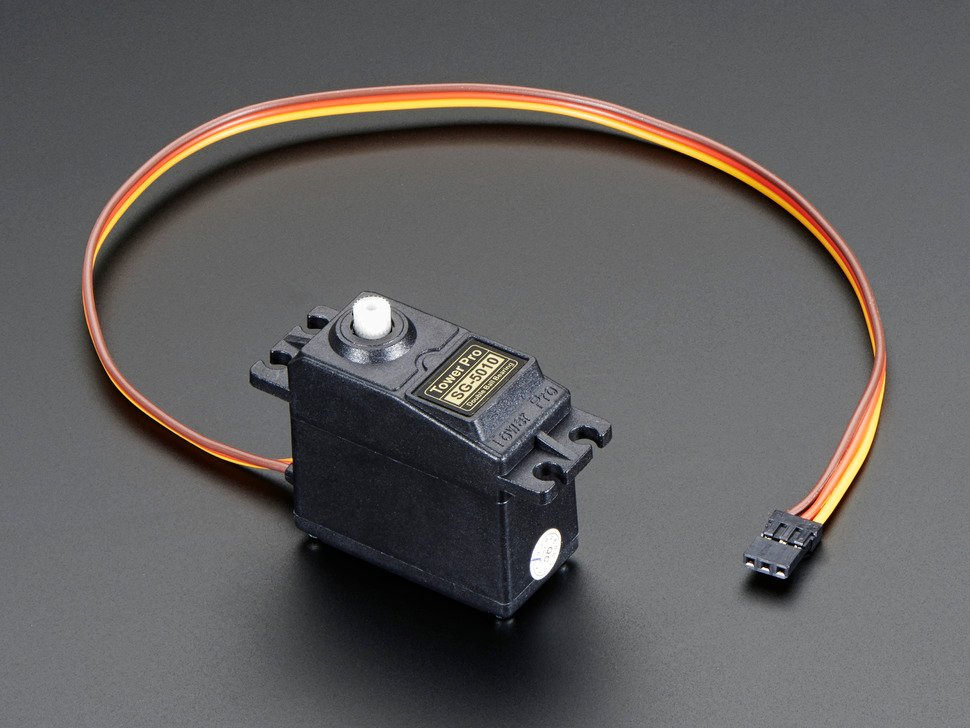
Підходящим пристроєм буде TowerPro SG-5010(рис 6.2). Він досить потужний для руху сонячної батареї, здатен працювати при стандартних дожинах пульсу ШІМ для сервоприводів(1-2мс), протестований при роботі на платах Arduino і чудово працює при напрузі 5V[8].

Рис. 6.2. TowerPro SG-5010

## 6.3. Вибір сонячної панелі

Сонячна панель(батарея) – найдорожчий елемент пристрою. На щастя, особливість її використання в даному проекті не передбачає її прямого використання як джерела електроенергії, використовуватися буде лише різниця даваної на аналоговий вхід напруги при різних положеннях сервоприводів. Слід також взяти до уваги те, що надто слабкі сонячні панелі(напруга яких зазвичай 5-7V) можуть виявити сонце лише коли воно знаходиться у найвищій точці, значно зменшуючи можливості пристрою.

Оптимальним варіантом буде сонячна панель СП KM5(6), основні характеристики якої:

* Номінальна потужність - 5Вт;
* Номінальна напруга під навантаженням - 17.56В;
* Напруга холостого ходу (при розімкнутому контурі) - 21.52В;
* Номінальний генерований струм - 0.286А;
* Струм короткого замикання - 0.31А;
* Габаритні розміри - 220х250х18 мм.

## 6.4. Вибір резистора

Аналоговий вхід Arduino Uno розрахований на максимальну напругу 5В. Напруга сонячної панелі під навантаженням дорівнює 17.56В при силі струму 0.286А. Розрахуємо необхідний опір резистора для обмеження напруги на аналоговому вході плати Arduino Uno.

При послідовному з’єднанні провідників повна напруга дорівнює сумі напруг провідників, тому для зменшення напруги від сонячної панелі до входу на платі необхідно щоб резистор працював при напрузі:

За законом Ома, резистор буде мати напругу при стурмі маючи опір:

Маємо, що для нашого пристрою підійде будь-який резистор з опором 54.41Ом і більше.

Залишився лише один елемент схеми – перемикач. Для нашого проекту підійде будь-який перемикач з робочою напругою 5V. Більшість таких мають вбудований у корпус резистор, що спростить підключення.

# **ВИСНОВОК**

Під час роботи над даним проектом мною був розроблений пристрій відслідковування сонця, в тому числі:

* Розроблений алгоритм роботи пристрою;
* Побудована схема підключення компонентів до плати Arduino;
* Проведений підбір підходящих компонентів з доступних на ринку на момент публікації роботи.

Розроблений пристрій здатен відслідковувати положення сонця на небі за допомогою показань однієї сонячної панелі(батареї) з мінімальним впливом інших джерел освітлення, чого вдалося досягти за допомогою відмови від стандартного підходу до вирішення проблеми – використання датчиків освітлення, замінивши тестування освітленості на тестування отриманої в різних положеннях електроенергії у вигляді напруги на аналоговому ході плати Arduino.

Ротація пристрою здійснюється за допомогою двох сервоприводів, що дозоляє повністю охопити всю доступну небесну площу. Запрограмовано було також два додаткові виходи для підключення двох сервоприводів допоміжного пристрою, що дозволить зчитувати покази положення сонця, які пристрій вважає оптимальними. Дані на вихід подаються за допомогою виходів з підтримкою ШІМ у вигляді імпульсів періодичністю 1-2мс, що відповідає куту положення сервопривода від 0 до 180 відповідно.

Вихідний код проекту можна знайти у репозиторії за посиланням: <https://github.com/randomanimedude/SolarTracker.git>.

# **СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ**

1. What Is Green Power? [Електронний ресурс] // United States Environmental Protection Agency – Режим доступу до ресурсу: https://www.epa.gov/greenpower/what-green-power.
2. Arduino Solar Tracker [Електронний ресурс]. – 2021. – Режим доступу до ресурсу: <https://duino-projects.com/arduino-solar-tracker/>.
3. Hammoumi A. Arduino Solar Tracker [Електронний ресурс] / A. Hammoumi, S. Motahhir. – 2020. – Режим доступу до ресурсу: https://create.arduino.cc/projecthub/336271/arduino-solar-tracker-41ef82.
4. Simulation and Model‑Based Design [Електронний ресурс] // The MathWorks, Inc.. – 2021. – Режим доступу до ресурсу: <https://www.mathworks.com/products/simulink.html>.
5. Широтно-імпульсна модуляція [Електронний ресурс]. – 2020. – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/Широтно-імпульсна\_модуляція.
6. ARDUINO UNO REV3 [Електронний ресурс] – Режим доступу до ресурсу: https://store.arduino.cc/arduino-uno-rev3.
7. Arduino Uno [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/Arduino_Uno>.
8. STANDARD SERVO - TOWERPRO SG-5010 [Електронний ресурс] // Adafruit – Режим доступу до ресурсу: https://www.adafruit.com/product/155.