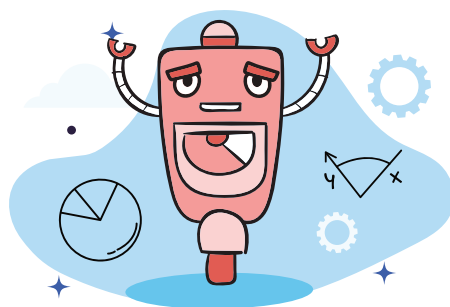




Sterowanie kątem za pomocą serwomechanizmu



Zagadnienia informatyczne

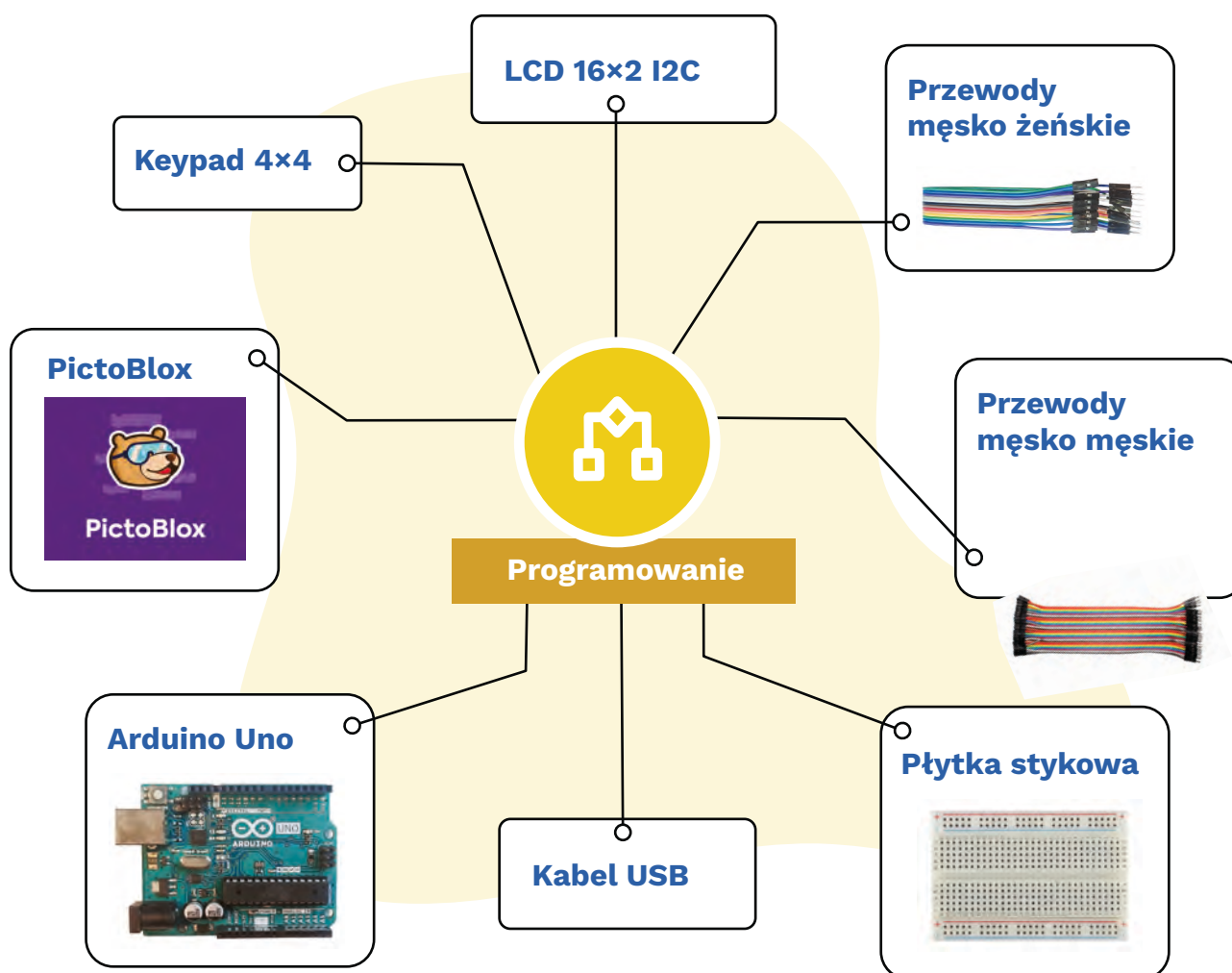
W tej części uczniowie poznają serwomechanizm jako element wykonawczy, który pozwala precyzyjnie sterować ruchem w określonym zakresie kątowym. Uczą się, w jaki sposób Arduino wysyła sygnał sterujący określający pozycję serwa oraz jak zmiana wartości w programie wpływa na rzeczywisty ruch urządzenia. Pracując w środowisku PictoBlox, uczniowie sterują serwomechanizmem, analizują zależność między zmienną w programie a położeniem osi oraz tworzą programy umożliwiające zarówno skokowe, jak i płynne przejścia między kolejnymi pozycjami. Dzięki temu poznają, jak mikrokontroler może kontrolować ruch elementów mechanicznych.



Zagadnienia matematyczne

Część matematyczna koncentruje się na pojęciu kąta oraz jego zastosowaniu w opisie ruchu obrotowego. Uczniowie wykorzystują serwomechanizm jako fizyczny model wskazówki zegara, aby zobaczyć, jak wartości wyrażone w stopniach odpowiadają rzeczywistemu położeniu. Na tej podstawie uczą się obliczać różnice kątów między wskazaniem zegara, interpretować miarę kąta jako wielkość opisującą obrót oraz wykorzystywać zależności matematyczne do przewidywania ruchu urządzenia sterowanego przez program.

Co będzie potrzebne do module 7?



Poznajemy świat arduino: Sterowanie serwomechanizmem i kontrola położenia

Co to jest serwomechanizm

Serwomechanizm (w skrócie: serwo) to niewielki silnik elektryczny, który potrafi **ustawić swoją oś w dokładnie określonej pozycji kątowej**.

W przeciwieństwie do zwykłego silnika, który obraca się cały czas, serwo może zatrzymać się w konkretnym miejscu, np.:

- 0° – pozycja początkowa
- 90° – pozycja środkowa
- 180° – pozycja końcowa

Dzięki temu serwo może pełnić rolę:

- sterowanej dźwigni
- wskazówki
- ramienia robota
- mechanizmu otwierającego lub zamykającego element

Serwomechanizmy są powszechnie stosowane w:

- robotyce
- modelarstwie
- automatyce
- urządzeniach elektronicznych



Jak działa serwomechanizm

Wewnątrz serwa znajdują się:

- silnik elektryczny
- przekładnia (zestaw zębatek zwiększających precyzję)
- układ elektroniczny sterujący pozycją
- czujnik położenia osi

Arduino wysyła sygnał sterujący, który określa, na jaki kąt serwo ma się ustawić. Serwo automatycznie obraca się do tej pozycji i zatrzymuje się.

Zakres pracy serwa

Typowe serwo edukacyjne pracuje w zakresie: **0° do 180°**

Oznacza to, że może obrócić się o pół pełnego obrotu.

Nie obraca się w nieskończoność – ma ograniczniki mechaniczne.

Budowa i przewody serwa

Serwo posiada trzy przewody:

- czerwony – zasilanie (5V)
- brązowy lub czarny – masa (GND)
- pomarańczowy, żółty lub biały – sygnał sterujący

Przewód sygnałowy podłącza się do pinu cyfrowego Arduino.

Serwa SG90, MG90S i S90 – różnice

W projekcie ArduMathLab używane są różne modele serw.

Najczęściej spotykane to SG90 i MG90S.





SG90

to najpopularniejsze serwo edukacyjne.

Cechy

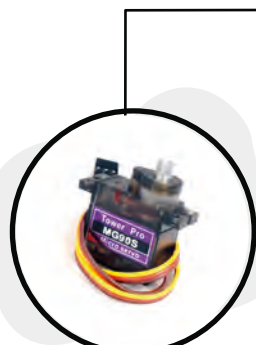
- lekkie
- wykonane z plastikowych elementów
- wystarczające do większości zastosowań szkolnych
- tanie i powszechnie dostępne

Zalety

- bardzo dobre do nauki
- niskie zużycie energii
- wystarczająca dokładność

Wady

- mniejsza wytrzymałość mechaniczna



MG90S

to mocniejsza wersja SG90.

Najważniejsza różnica: MG90S posiada metalowe zębatki, podczas gdy SG90 posiada plastikowe.

Cechy

- większa wytrzymałość
- większa siła
- większa trwałość
- większa masa

Zalety

- bardziej odporne na uszkodzenia
- lepsze do projektów mechanicznych

Wady

- większy pobór prądu
- nieco cięższe

Orczyki, co to jest i do czego służą

Orczyki to plastikowe elementy montowane na osi serwa.

Mogą mieć różne kształty, np.:

- pojedyncze ramię



- podwójne ramię
- krzyż
- tarcza

Orczyk jest mocowany do osi serwa za pomocą małej śruby.

Funkcja orczyka

Orczyk umożliwia przeniesienie ruchu serwa na inny element.

Można do niego przymocować:

- wskaźnik
- ramię robota
- dźwignię
- element konstrukcyjny

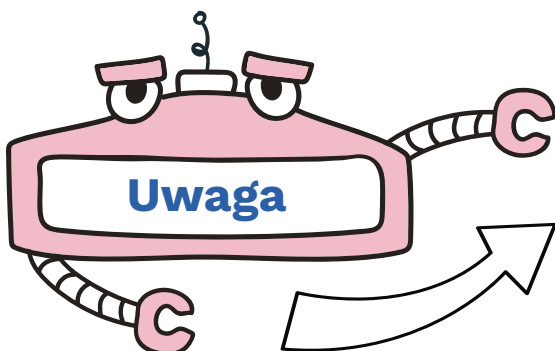
Gdy serwo obraca się, orczyk obraca się razem z nim. Dzięki temu można sterować ruchem innych elementów.

Sterowanie serwem w PictoBlox – blok „set servo”

Na początku modułu używane są bloki sterujące urządzeniami.

Znajdują się one w kategorii:

Actuators, w tłumaczeniu na język polski – elementy wykonawcze.

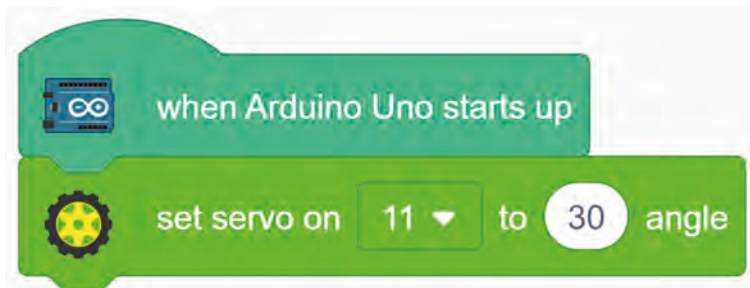


Serwomechanizm można obsługiwać w obu trybach pictoblox tzn zarówno stage jak i upload.

Blok:

Znaczenie:

ustawia serwo połączone do pinu 11 w pozycji 30°. Serwo obróci się do tej pozycji i zatrzyma.



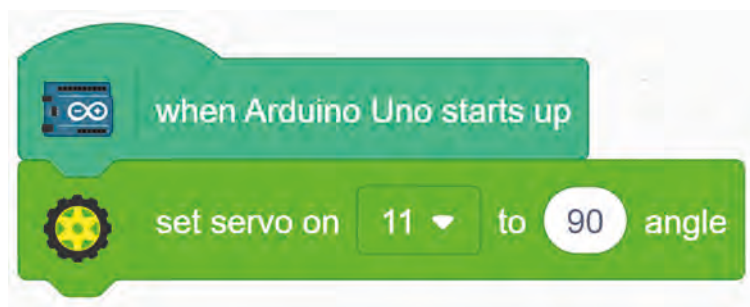
Rysunek 51

Jedyny blok Pictoblox odpowiedzialny za sterowanie ruchem serwomechanizmu

Kalibracja serwomechanizmu:

Pierwszą czynnością po podłączeniu serwomechanizmu do Arduino powinna być kalibracja, czyli ustawienie serwa w pozycję startową.

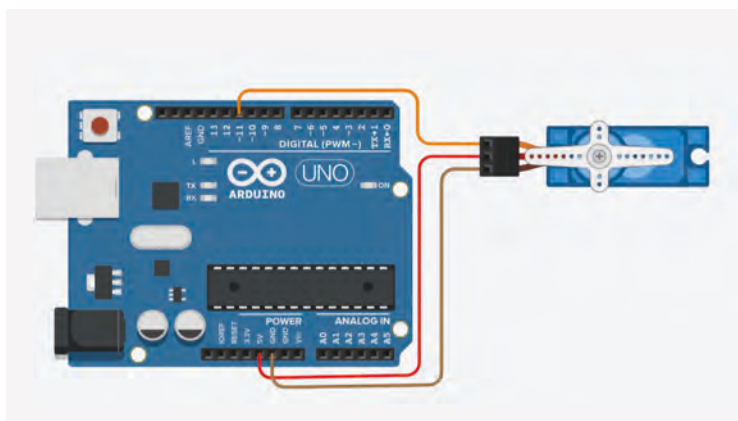
Najlepszym wyjściem będzie ustawienie początkowe na pozycję 90 stopni i dopiero wtedy zamontowanie orczyka (zaśrubowanie) tak, aby był ustawiony prostopadłe do obudowy. Uruchomiony program kalibracji będzie stale utrzymywał pozycję 90 stopni, zbyt duży opór może złamać zębatki.



Rysunek 52 Kalibracja serwomechanizmu przed zamontowaniem orczyka

Podłączenie Serwomechanizmu do Arduino

- czerwony → 5V
- brązowy → GND
- pomarańczowy → D11

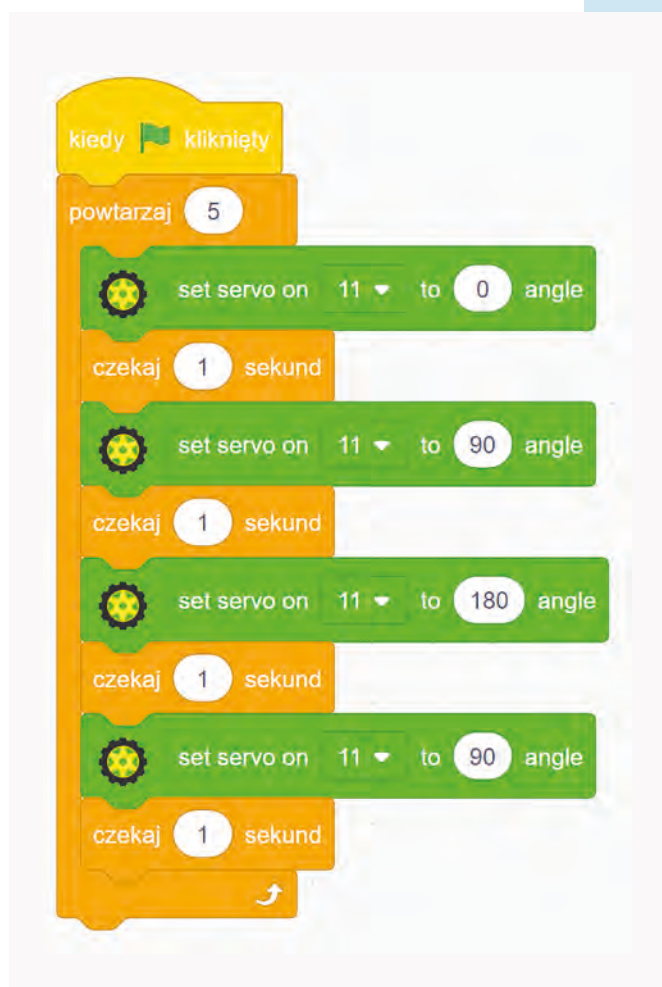


Rysunek XX



Zadanie 1

Sterowanie serwomechanizmem – dodawanie pozycji pośrednich



Rysunek 53
Szkic programu moduł7infzad1

Cel:

Poznać sposób sterowania serwomechanizmem w PictoBlox oraz zrozumieć, jak zmiana wartości kąta wpływa na ruch serwa. Nauczyć się modyfikować gotowy program tak, aby działał w pętli nieskończonej i zawierał pozycje pośrednie.

Krok 1. Uruchom program bazowy

1. Wgraj do Arduino program
2. Obserwuj działanie serwomechanizmu. Program wykonuje ruch:
 $0^{\circ} \rightarrow 90^{\circ} \rightarrow 180^{\circ} \rightarrow 90^{\circ}$
Powtarza cykl 5 razy, a następnie zatrzymuje się.

Krok 2. Zmodyfikuj program – pętla nieskończona

1. Zmień program tak, aby ruch serwomechanizmu powtarzał się bez końca (pętla nieskończona).

Krok 3. Dodaj pozycje pośrednie

1. Dodaj pozycje pośrednie znajdujące się dokładnie w połowie pomiędzy:
 - 0° i 90°
 - 90° i 180°



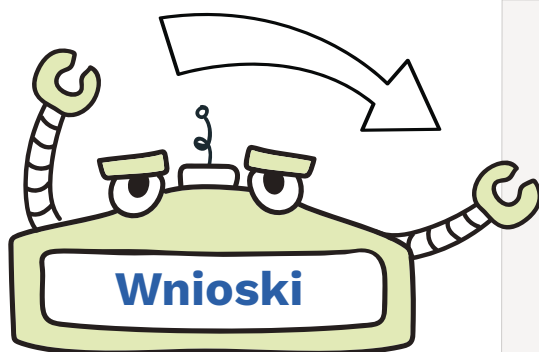
2. Po modyfikacji program ma wykonywać ruch w kolejności:
 $0^\circ \rightarrow \underline{\hspace{1cm}} \rightarrow 90^\circ \rightarrow \underline{\hspace{1cm}} \rightarrow 180^\circ$
 $\rightarrow \underline{\hspace{1cm}} \rightarrow 90^\circ \rightarrow \underline{\hspace{1cm}} \rightarrow 0^\circ$

Krok 4. Ustaw czas zatrzymania

1. W każdej pozycji serwo ma zatrzymać się na 1 sekundę.

Kryterium osiągnięcia celu

- Serwomechanizm wykonuje ruch w nieskończonej pętli (nie zatrzymuje się po 5 cyklach).
- W sekwencji ruchu widać dodatkowe pozycje pośrednie (pomiędzy $0-90$ oraz $90-180$).
- W każdej pozycji serwo zatrzymuje się na około 1 sekundę.



1. Co stanie się z ruchem serwa, jeśli usuniesz wszystkie opóźnienia między kolejnymi pozycjami?
2. Co się stanie, jeśli ustawisz pozycję wykraczającą poza zakres pracy serwomechanizmu?



Zadanie 2

Płynne przejścia serwomechanizmu



Rysunek 54
Szkielet programu moduł7infzad2

Cel:

- zrozumieć, jak uzyskać płynny ruch serwa przez stopniową zmianę kąta,
- przeanalizować rolę: zmiennej, pętli, kroku zmiany i opóźnienia,
- powiązać konkretny fragment kodu z obserwowanym ruchem serwomechanizmu.

Krok 1. Uruchom program i obserwuj

1. Wgraj do Arduino program (serwomechanizmem można sterować zarówno w trybie stage jak i upload).
2. Uruchom program.
3. Obserwuj, jak serwo porusza się od 0° do 180°, a potem wraca do 0°.

Po uruchomieniu programu obserwuj ruch serwomechanizmu. Zwróć uwagę na sposób poruszania się serwa.

Krok 2. Analiza kodu (na podstawie blochków)

Odpowiedz krótko, zapisz na karcie odpowiedzi

1. Od jakiej wartości rozpoczyna się ruch serwa?



2. Jak zmienia się zmienna stopnie w pierwszej pętli: rośnie czy maleje?
3. Jak zmienia się zmienna stopnie w drugiej pętli?
4. Dlaczego zastosowano powtarzaj 18? (powiąż to z zakresem 0–180 i krokiem zmiany)
5. O ile stopni zmienia się pozycja serwa w jednym kroku?
6. Co w tym programie decyduje o tym, że ruch wygląda “płynniej” niż w zadaniu 1?

Karta odpowiedzi

Krok 3. Dwie krótkie modyfikacje (test w praktyce)

1. Zmień krok ruchu z 10 na 30.
Uruchom program i porównaj ruch serwa.
2. Przywróć krok 10, a następnie usuń (albo ustaw na 0) opóźnienie
czekaj 0.2 sekundy. Uruchom program i porównaj zachowanie.

Kryterium osiągnięcia celu

1. Serwomechanizm wykonuje ruch krokowy: $0^{\circ} \rightarrow 180^{\circ} \rightarrow 0^{\circ}$ (bez “przeskoku” od razu do końca).
2. Uczeń potrafi wskazać w kodzie:
 - gdzie ustawiany jest kąt startowy,
 - gdzie realizowany jest krok zmiany kąta,
 - gdzie sterowana jest prędkość/ “płynność” ruchu (opóźnienie).
3. Po zmianie kroku na 30°
i po usunięciu opóźnienia uczeń potrafi opisać różnicę w zachowaniu serwa (ruch mniej płynny / zbyt szybki / “szarpany”).

Rysunek 54
Szkic programu moduł7infzad2





1. Co decyduje o „płynności” ruchu serwomechanizmu — wielkość kroku zmiany kąta, opóźnienie między krokami czy oba te elementy jednocześnie?
2. Dlaczego zastosowano dokładnie 18 powtórzeń w pętli? Jak ta liczba wynika z zakresu ruchu serwa ($0-180^\circ$) i kroku zmiany?

Ciekawostka

Co jeszcze można zrobić z serwomechanizmem

W tym module do serwomechanizmu został zamocowany orczyk, a następnie przyklejono do niego drewnianą wykałaczkę. Pełni ona rolę wskaźnika, który pokazuje kierunek i zakres ruchu serwa.

Dzięki temu łatwo można obserwować zmiany położenia i dokładnie zobaczyć, jak obraca się serwomechanizm. W praktyce do serwomechanizmu można przymocować wiele różnych elementów, w zależności od zastosowania.

Servomechanizm może sterować:

- ramieniem robota
- wskaźnikiem kierunku
- pokrywą pojemnika np. śmietnika
- drzwiami w modelu budynku
- zapadką lub blokadą
- elementem obracającym czujnik

Na przykład: Serwomechanizm może unosić i zamykać pokrywę kosza na śmieci, otwierać drzwi w modelu domu lub obracać czujnik odległości, który skanuje otoczenie. Taki układ działa podobnie do radaru.

Odkrywamy matematykę: Wyznaczanie kąta między wskazaniami zegara

Kąt i model zegara

Kąt jest miarą obrotu. Informuje nas, o ile jeden kierunek został obrócony względem innego. Jednostką kąta są **stopnie** ($^{\circ}$).

Pełny obrót wokół własnej osi wynosi: **360°**

Półowa obrotu wynosi: **180°**

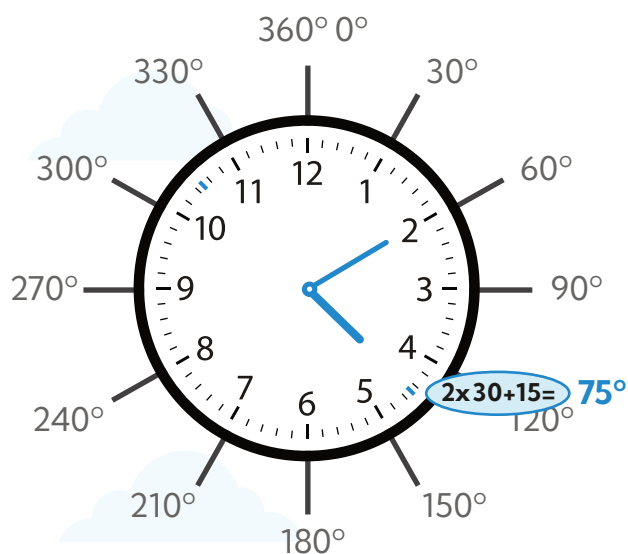
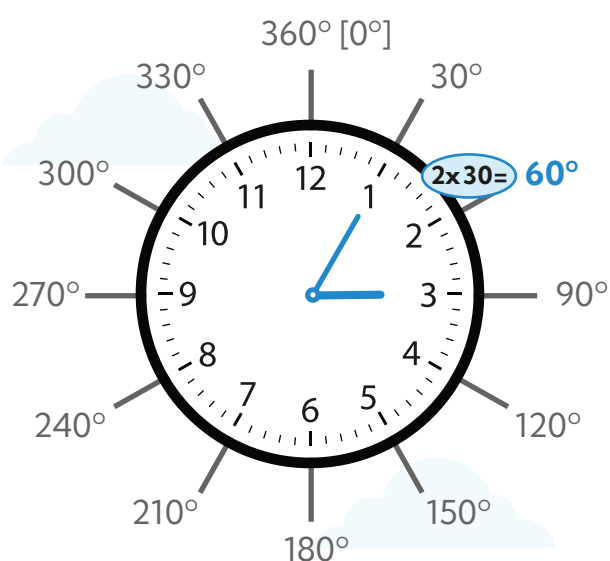
Ćwierć obrotu wynosi: **90°**

Zegar jako model matematyczny kąta

Tarcza zegara jest podzielona na **12 równych części (godzin)**. Ponieważ pełny obrót ma 360° , możemy obliczyć, jaki kąt odpowiada jednej godzinie:

$$360^{\circ} \div 12 = 30^{\circ} \quad 360^{\circ} \div 12 = 30^{\circ}$$

Oznacza to, że każda kolejna godzina na zegarze jest oddalona o 30° .



Godzina	Kąt względem godziny 12
12:00	0°
1:00	30°
2:00	60°
3:00	90°
4:00	120°
5:00	150°
6:00	180°

Serwomechanizm jako fizyczny model kąta

W tym ćwiczeniu serwomechanizm pełni rolę wskazówki zegara.

Gdy ustawiamy serwomechanizm na określoną wartość, np.: 0°, 30°, 90° lub 180° jego ramię ustawia się dokładnie pod tym kątem.

Dzięki temu możemy:

- zobaczyć kąt fizycznie,
- zmierzyć go,
- oraz sprawdzić poprawność naszych obliczeń.

Serwomechanizm staje się więc rzeczywistym modelem matematycznym kąta.

Różnica kątów między godzinami

Aby obliczyć, o ile stopni wskazówka musi się obrócić z jednej godziny na drugą, należy obliczyć różnicę odpowiadających im kątów.

Przykład:

przejście z godziny 2:00 do 5:00

2:00 → 60°

5:00 → 150°

różnica:

$$150^{\circ} - 60^{\circ} = 90^{\circ} \quad 150^{\circ} - 60^{\circ} = 90^{\circ} \quad 150^{\circ} - 60^{\circ} = 90^{\circ}$$

Wskazówka musi obrócić się o 90°.





Zadanie 1

Rysowanie tarczy zegara za pomocą serwomechanizmu

Cel:

Zobaczyć, jak kąty wyrażone w stopniach odpowiadają położeniom godzin na zegarze, oraz stworzyć fizyczny model tarczy zegara z wykorzystaniem serwomechanizmu jako wskazówki. W tym zadaniu serwomechanizm będzie pełnił rolę wskazówki, która zatrzymuje się kolejno na pozycjach odpowiadających godzinom od 12 do 6.

Krok 1. Przygotowanie wskazówki

1. Nałóż orczyk na serwomechanizm.
2. Do orczyka przymocuj wykałaczkę (np. taśmą dwustronną).
3. Wykałaczka będzie pełnić rolę wskazówki zegara.

Wskazówka powinna być dobrze widoczna i nie może ocierać o kartkę.

Krok 2. Przygotowanie kartki

1. Połóż serwomechanizm na środku kartki papieru.
2. Upewnij się, że wskazówka może się swobodnie obracać.
3. Jeśli serwo się przesuwą, możesz delikatnie przykleić je taśmą lub odrysować jego pozycję

Rysunek 53
Szkic programu moduł7infzad1



początkową, aby łatwo ustawić je ponownie.

4. Kartka będzie służyła do narysowania tarczy zegara.

Krok 3. Uruchomienie programu

1. Otwórz program w PictoBlox.
2. Program działa w trybie stage.
3. Podłącz Arduino do komputera.
4. Uruchom program, klikając zieloną flagę.

Serwomechanizm zacznie obracać wskazówkę co 30° i zatrzymywać się kolejno na różnych pozycjach. Każda pozycja odpowiada jednej godzinie na zegarze.

Krok 4. Rysowanie tarczy zegara

Za każdym razem, gdy wskazówka zatrzyma się w nowym miejscu:

1. Zaznacz na kartce punkt, na który wskazuje wykałaczka.
2. Obok punktu zapisz odpowiednią godzinę:
 - pierwsza pozycja — 12
 - druga pozycja — 1
 - trzecia pozycja — 2
 - czwarta pozycja — 3
 - piąta pozycja — 4

- szósta pozycja — 5
- siódma pozycja — 6

Program zatrzyma się na każdej pozycji na kilka sekund, aby umożliwić zaznaczenie punktu.

Kryterium osiągnięcia celu

- ☑ Na kartce powstał układ punktów odpowiadający kolejnym godzinom od 12 do 6.
- ☑ Punkty są rozmieszczone na okręgu wokół serwomechanizmu.
- ☑ Odległości między kolejnymi punktami są zbliżone, co odpowiada równym podziałom kąta pełnego.
- ☑ Powstał fizyczny model połowy tarczy zegara.



1. W jaki sposób kolejne zatrzymania serwomechanizmu co 30° odpowiadają położeniom godzin na zegarze?
2. Dlaczego wykonanie fizycznej tarczy zegara może pomóc w obliczaniu kątów między wskazaniemi w następnym zadaniu?
3. Dlaczego wszystkie punkty są w tej samej odległości od środka?



Zadanie 2

Obliczanie kąta między godzinami na zegarze

Cel:

Zrozumieć, że położenia wskazówki zegara można opisać za pomocą kątów oraz nauczyć się obliczać, o ile stopni należy obrócić wskazówkę, aby przejść między dwiema godzinami. W tym ćwiczeniu serwomechanizm ustawi wskazówkę na określonej godzinie początkowej, a następnie Twoim zadaniem będzie obliczyć, o ile stopni należy ją przesunąć, aby osiągnąć wskazaną godzinę docelową.

Krok 1. Przygotowanie stanowiska

1. Użyj kartki z narysowaną tarczą zegara z Zadania 1.
2. Serwomechanizm z wykałaczką pełni rolę wskazówki zegara.
3. Upewnij się, że:
 - o serwomechanizm jest podłączony do Arduino,
 - o program jest uruchomiony w PictoBlox, o stanowisko jest stabilne.

Krok 2. Obserwacja wskazania początkowego

1. Po uruchomieniu programu serwomechanizm ustawi

01

02

03

04

05

06

07

08

09

10

11

12



wskazówkę na godzinie początkowej.

2. Na ekranie pojawi się informacja o godzinie początkowej oraz docelowej.

Krok 3. Obliczenie różnicy kątów

1. Na podstawie położenia wskazówki i godziny docelowej oblicz, o ile stopni należy obrócić wskazówkę.
2. Możesz korzystać z narysowanej tarczy zegara, aby ułatwić obliczenia.

Krok 4. Wprowadzenie odpowiedzi

1. Wpisz obliczoną wartość kąta w oknie programu.
2. Zatwierdź odpowiedź.

Krok 5. Sprawdzenie wyniku

1. Program poinformuje, czy odpowiedź jest poprawna.
2. Następnie wskazówka ustawi się na godzinie docelowej.
3. Rozpocznie się kolejna runda z nowymi danymi.





1. W jaki sposób można obliczyć kąt obrotu bez znajomości stopni — tylko na podstawie liczby „przeskoczonych” godzin?
2. Czy różnica kątów między godzinami zależy od miejsca na tarczy, czy tylko od liczby godzin między nimi?

Kryterium osiągnięcia celu

- ☑ Uczeń potrafi obliczyć różnicę kątów między położeniami wskazówki zegara.
- ☑ Wprowadzona wartość odpowiada rzeczywistemu przesunięciu wskazówki.
- ☑ Uczeń korzysta z modelu tarczy zegara do weryfikacji obliczeń.
- ☑ Program poprawnie reaguje na wprowadzoną odpowiedź i przechodzi do kolejnych zadań.

Wskazówka matematyczna

Na zegarze odległość między kolejnymi godzinami wynosi 30°

Przykłady:

$$12 \rightarrow 1 = 30^\circ$$

$$12 \rightarrow 2 = 60^\circ$$

$$3 \rightarrow 6 = 90^\circ$$

Odległość między połówkami godzin wynosi 15° :

$$12 \rightarrow \text{w pół do } 1 = 15^\circ$$

$$1 \rightarrow \text{w pół do } 2 = 15^\circ$$

Możesz obliczyć wynik, licząc liczbę „kroków” między godzinami i mnożąc przez 30° a następnie dodając 15° w razie potrzeby.

Czego się nauczyłeś?



Matematyka

- czym jest kąt jako miara obrotu i jak wyraża się go w stopniach
- jak tarcza zegara może być traktowana jako model podziału pełnego kąta
- że odległość między kolejnymi godzinami na zegarze wynosi 30°
- jak obliczać różnicę kątów między dwoma położeniami wskazówki
- jak fizyczny model (serwomechanizm) może pomóc w zrozumieniu pojęć geometrycznych



Informatyka

- czym jest serwomechanizm i do czego służy w układach elektronicznych
- jak sterować pozycją serwa za pomocą programu i wartości kąta
- czym różni się ruch skokowy od płynnego oraz jak uzyskać płynny ruch poprzez stopniową zmianę wartości
- jak zmienne, pętle i opóźnienia wpływają na zachowanie fizycznego urządzenia
- jak program komputerowy steruje ruchem elementów mechanicznych

