

Budowa i programowanie robota sterowanego learn:bitem w oparciu o mikrokontroler micro:bit

Grupa docelowa: uczniowie klas 7–8 SP

Liczba zajęć: 8–10 x 45 min

Forma pracy: projektowa

Zakres podstawy programowej: informatyka, technika, fizyka

Cele ogólne:

- Rozwijanie kompetencji STEAM, czyli integracja nauki, technologii, inżynierii, sztuki i matematyki.
- Nauka programowania w edytorze MakeCode oraz edytorze Python (MicroPython) na mikrokontrolerach micro:bit.
- Poznanie działania i zastosowanie elementów elektroniki, takich jak serwomechanizmy, diody LED, czujniki światła i temperatury.
- Integracja sztucznej inteligencji (AI) z mikrokontrolerami poprzez wykorzystanie Teachable Machine.
- Rozwijanie umiejętności pracy zespołowej, kreatywnego myślenia i rozwiązywania problemów.

Cele szczegółowe:

- Praktyczne zastosowanie diod wyświetlacza mikrokontrolera.
- Programowanie wyświetlacza LED.
- Programowanie reakcji przycisków do wykonywania konkretnych poleceń.
- Umiejętność sterowania serwomechanizmami.
- Zrozumienie działania potencjometru oraz praktyczne zastosowania.
- Programowanie i sterowanie diodami LED.
- Wykorzystanie czujników mikrokontrolera do interakcji z otoczeniem.
- Tworzenie systemu komunikacji radiowej między micro:bitami.
- Wykorzystanie modelu AI do sterowania gestami poprzez aplikację Teachable Micro:Bit App.

Metody pracy:

- *Praca zespołowa:* uczniowie pracują w grupach 4-6 osobowych.
- *Eksperymenty i testy:* sprawdzanie poprawności działania kodu i mechaniki konstrukcji.
- *Dyskusja i prezentacja:* uczniowie dzielą się swoimi doświadczeniami i przedstawiają gotowe roboty.

Materiały i narzędzia:

- *Materiały edukacyjne:* microbit.org, kurs Learn:bit Arcade
- *Elektronika:* learn:bity, micro:bity, przewody, serwomechanizmy, diody
- *Narzędzia programistyczne:* edytor MakeCode lub Python
- *Materiały konstrukcyjne:* kartony, nożyczki, kleje, taśmy, patyczki, drut itp.
- *AI:* Teachable Machine i aplikacja Teachable Micro:Bit App.
- Komputer, telefon, tablet

Efekty końcowe:

- Każdy zespół zaprogramuje i skonstruuje działającego robota opartego na learn:bitcie i mikrokontrolerze micro:bit.
- Uczniowie zaprezentują swoją pracę zespołowo, omawiając proces budowy i kodowanie.
- Każdy uczeń zdobywa doświadczenie zarówno w programowaniu, jak i w projektowaniu konstrukcji, a praca zespołowa staje się kluczowym elementem sukcesu.

Uwagi:

- Elastyczność podejścia – prezentowane scenariusze zajęć są jedynie sugestią, która koncentruje się przede wszystkim na stronie programistycznej projektu. Można je dowolnie modyfikować, dostosowując do możliwości grupy oraz stopnia zaawansowania uczniów.
- Zrównoważony podział ról i praca zespołowa – celem zajęć jest zapoznanie każdego ucznia z podstawami programowania micro:bita, jednak trudniejsze zagadnienia są przeznaczone dla zespołu programistycznego. W tym czasie uczniowie zajmujący się konstrukcją mogą modyfikować rozwiązania techniczne robota.
- Interdyscyplinarność – zajęcia łączą elementy informatyki, techniki i fizyki, angażując uczniów w projektowanie, budowę i kodowanie. Dyskusja między zespołami jest kluczowa, by zapewnić skuteczne działanie robota.
- Eksperymentowanie i adaptacja – w trakcie realizacji projektu mogą pojawić się nowe pomysły na rozbudowę robota, ulepszenie jego funkcji, czy optymalizację kodu. Wskazane jest by uczniowie testowali różne rozwiązania i wspólnie analizowali osiągnięte efekty.

Lekcja 1: Wprowadzenie do projektu i planowanie pracy

Cele lekcji:

- Zapoznanie uczniów z celem projektu – wyjaśnienie, jakie kompetencje STEAM będą rozwijane w trakcie budowy robota.
- Analiza działania robota – omówienie kluczowych mechanizmów działania.
- Podział na zespoły – wstępne przypisanie ról i odpowiedzialności w grupach.
- Planowanie projektu – ustalenie harmonogramu i stworzenie wstępnego szkicu konstrukcji robota.

Materiały:

- komputer lub telefon
- tablica, kartki, markery
- przykładowe komponenty: learn:bit, micro:bit, serwa, diody LED
- instrukcje projektowe i harmonogram

Uwagi metodyczne:

- *Interdyscyplinarność:* lekcja łączy elementy informatyki, fizyki i techniki.
- *Aktywność zespołowa:* uczniowie pracują w grupach, co rozwija umiejętność współpracy.
- *Praktyka:* uczniowie nie tylko słuchają, ale od razu projektują i planują pracę.

Przebieg lekcji:**1. Wstęp (10 min) – zapoznanie z projektem**

Nauczyciel prezentuje temat projektu: „Budujemy robota w oparciu o learn:bita i micro:bita”.

Prezentacja przykładowych projektów (3–5 min).

Krótką rozmowa kierowana:

- Co będzie robił nasz robot?
- Czym dysponujemy?
- Jakie technologie wykorzystamy?
- Jak go zbudujemy?

2. Omówienie elementów robota (15 min) – analiza posiadanego sprzętu i technologii, jaka dysponujemy

Prezentacja i omówienie komponentów możliwych do wykorzystania w projekcie:

- learn:bit – płytki - „pad” wspierający działanie mikrokontrolera, wyposażony w diody LED, potencjometr, dodatkowe przyciski, wyświetlacz LCD
- micro:bit – mikrokontroler sterujący funkcjami robota, wyposażony w czujniki światła, temperatury, akcelerometr, głośnik
- serwomechanizmy – mechanizmy sterujące ruchem
- AI (Teachable Machine) – model sztucznej inteligencji, który można wykorzystać do sterowania

Dyskusja zespołowa:

- Jakie funkcje powinien mieć nasz robot?
- Jakie technologie mogą być dla nas wyzwaniem?

3. Tworzenie zespołów (5 min) – podział ról

Podział uczniów na zespoły 4–6 osobowe.

Każda grupa wybiera osoby odpowiedzialne za programowanie oraz osoby odpowiedzialne za konstrukcję

Wstępne omówienie obowiązków każdej osoby w grupie:

- sterowanie serwomechanizmami odpowiedzialnymi za ruch
- programowanie efektów świetlnych i muzycznych
- zadania dotyczące komunikacji radiowej i sterowania modelem AI
- konstrukcja robota

4. Planowanie projektu (15 min) – pierwsze szkice konstrukcyjne i harmonogram działań

Uczniowie na kartkach szkicują wstępny wygląd robota, uwzględniając:

- materiały i wszelkie pomysły konstrukcyjne w tym przede wszystkim mechanizmy ruchu
- miejsca na podzespoły (learn:bit, micro:bit, diody LED, serwa) i schematy możliwych połączeń

Tworzenie harmonogramu:

- Ustalanie kolejności prac
- Każdy zespół tworzy listę materiałów, które będą potrzebne do budowy robota (np. rodzaje kartonu, przewody, diody LED).

5. Podsumowanie lekcji (5min):

Każda grupa prezentuje swoje wstępne koncepcje i omawia wyzwania, które mogą się pojawić podczas realizacji projektu.

Lekcja 2: Budowa konstrukcji robota z kartonu

Cele lekcji:

- Budowa korpusu robota z materiałów recyklingowych, rozwijanie umiejętności technicznych.
- Uwzględnienie miejsc na podzespoły, przygotowanie konstrukcji do integracji elektroniki.
- Zastosowanie zasad projektowania mechanicznego – stabilna konstrukcja i możliwość ruchu wybranych elementów.
- Rozwijanie umiejętności manualnych i kreatywności poprzez tworzenie własnych modeli.

Materiały:

- Kartony różnej wielkości do budowy korpusu.
- Nożyczki, kleje, taśmy klejące (w tym taśma izolacyjna).
- Patyczki, gumki recepturki, druciki do wzmocnienia konstrukcji i mechanizmów ruchomych.
- Markery, linijki do rysowania i oznaczania miejsc montażu.

Uwagi metodyczne:

- *Praca zespołowa:* uczniowie współpracują nad projektowaniem i montażem robota.
- *Kreatywność i eksperymentowanie:* zachęcenie do testowania różnych mechanizmów ruchu.
- *Rozwój umiejętności technicznych:* lekcja łączy elementy inżynierii mechanicznej z podstawami elektroniki.
- *Bezpieczeństwo:* uczniowie uczą się odpowiedniego używania narzędzi i stabilnego konstruowania modeli.

Przebieg lekcji:**1. Omówienie zasad pracy z narzędziami (5 min) – bezpieczeństwo i technika**

Nauczyciel przypomina zasady BHP.

Krótką dyskusja na temat technik wykonania robotów:

- Jak wzmacniać konstrukcję, aby była stabilna?
- Jakie techniki zastosować do tworzenia ruchomych części?

2. Projektowanie konstrukcji robota (15 min) – szkicowanie i przygotowanie elementów

Uczniowie projektują własny model robota:

- szkicują na kartonie kształt robota i jego poszczególnych elementów
- określają mechanizmy ruchu

Dyskusja zespołowa:

- Jak zapewnić stabilność całej konstrukcji?
- Jakie dodatkowe elementy mogą poprawić wygląd lub funkcjonalność robota?

3. Wycinanie i montaż podstawowego szkieletu robota (20 min) – praktyczne działania

Uczniowie wycinają z kartonu elementy konstrukcyjne.

Łączenie elementów:

- montaż szkieletu przy użyciu kleju na gorąco, taśmy klejącej, gumek, czy drucików
- tworzenie prototypów mechanizmów ruchomych
- wyznaczanie miejsca na podzespoły elektroniczne (learn:bit, micro:bit, serwa, diody LED).

Sprawdzanie stabilności i poprawności budowy.

4. Podsumowanie lekcji (5 min) – wnioski i propozycje ulepszeń

Dyskusja:

- Czy konstrukcja jest stabilna? Czy wymaga dodatkowych wzmocnień?
- Jakie elementy trzeba poprawić przed montażem elektroniki?

Uczniowie przygotowują listę poprawek i modyfikacji.

Lekcja 3: Wprowadzenie do micro:bitów i sterowanie serwem

Cele lekcji:

- Zapoznanie uczniów ze środowiskiem programowania micro:bit (MakeCode lub edytor Python).
- Nauka sterowania serwomechanizmem.
- Praktyczne zastosowanie serwa w konstrukcji robota.
- Rozwijanie umiejętności programowania i analizy połączeń elektronicznych.

Materiały:

- micro:bity i komputery z dostępem do edytora MakeCode lub Python.
- serwomechanizmy
- przewody krokodylkowe (czerwony, czarny i sygnałowy)
- baterie i moduł zasilania do micro:bita
- kartony i elementy montażowe do przymocowania serwa

Uwagi metodyczne:

- *Praktyka:* uczniowie przekładają zdobytą wiedzę programistyczną na realne działanie urządzenia
- *Praca zespołowa:* każda grupa odpowiada za część zadania, sukces jest ich wspólnym dziełem
- *Eksperymentowanie:* uczniowie w praktyce testują działanie serwomechanizmów

Przebieg lekcji:**1. Wprowadzenie do micro:bita (10 min) – teoria**

Nauczyciel prezentuje micro:bit jako mikrokontroler do sterowania elektroniką.

Omówienie jego podstawowych elementów:

- wyświetlacz diodowy
- przyciski A i B
- piny wejścia/wyjścia (np. P0, P1, P2)

Ćwiczenie: Uczniowie piszą prosty program, w którym micro:bit wyświetli ikonę na wyświetlaczu diodowym po naciśnięciu przycisku A/B.

2. Podłączenie serwomechanizmu (10 min) – praktyka

Omówienie działania serwa:

- Serwomechanizm obraca się o określony kąt (np. 0°–180°).
- Sygnał sterujący wysyłany z micro:bita określa jego pozycję.

Podłączenie przewodów:

- Czerwony przewód - podłączony do VCC (zasilanie 3V).
- Czarny przewód - podłączony do GND (masa).
- Przewód sygnałowy - podłączony do pinu P0

Uwaga: Warto zwrócić uwagę na odpowiednie zasilanie mikrokontrolera z podłączonym serwomechanizmem!

3. Napisanie programu sterującego serwem (10 min) – kodowanie

Omówienie kodu sterowania:

- wciśnięcie A powoduje ruch serwa do góry.
- wciśnięcie B powoduje ruch serwa w dół.

Uczniowie tworzą kod programu w edytorze MakeCode lub Python.

4. Testowanie serwa (10 min) – wstępny montaż i testowanie działania serwa w praktyce

Uczniowie montują serwo do elementów testowych.

Sprawdzają, czy serwo poprawnie reaguje?

Uczniowie montują serwo do robota.

Sprawdzają, czy serwo poprawnie reaguje?

Jeśli występują problemy, wprowadzają zmiany w regulacji serwa lub testowanej konstrukcji.

5. Podsumowanie lekcji (5 min) – refleksja i pytania

Dyskusja grupowa:

- Czy sterowanie działa poprawnie?
- Co należy poprawić?

Lekcja 4: Obsługa potencjometru i odtwarzanie muzyki

Cele lekcji:

- Zapoznanie uczniów z funkcjami muzycznymi micro:bita – nauka generowania dźwięku
- Programowanie prostych motywów muzycznych
- Programowanie potencjometru i jego praktyczne zastosowanie – sterowanie intensywnością, częstotliwością dźwięku
- Dodanie funkcji aktywowanej potrząśnięciem pada – interaktywność sterowania.

Materiały:

- learn:bity, micro:bity i komputery z dostępem do edytora MakeCode lub Python
- baterie i moduł zasilania do micro:bita
- potencjometr (opcjonalnie)
- przewody krokodyłkowe
- głośniki/słuchawki

Uwagi metodyczne:

- *Interdyscyplinarność:* zastosowanie fizyki i zagadnień związanych z dźwiękiem przy programowaniu muzyki
- *Kreatywność:* uczniowie mogą eksperymentować z różnymi efektami dźwiękowymi
- *Praktyka:* połączenie elektroniki z programowaniem

Przebieg lekcji:

1. Wprowadzenie do funkcji muzycznych micro:bita (10 min) – teoria

Nauczyciel wyjaśnia, że micro:bit może generować dźwięk poprzez wbudowany głośniczek lub podłączone urządzenie zewnętrzne.

Omówienie podstawowych funkcji muzycznych:

- odtwarzanie pojedynczych nut
- programowanie sekwencji melodii
- regulacja intensywności i tempa utworu

Krótką demonstracją: nauczyciel pokazuje prosty motyw muzyczny na micro:bitcie

Omówienie sposobu podłączenia głośnika lub słuchawek:

Podłączenie przewodów:

- czarny przewód - podłączony do GND (masa) – górna część wtyczki słuchawkowej
- przewód sygnałowy - podłączony do pinu P0 – końcówka wtyczki słuchawkowej

2. Programowanie prostego motywu muzycznego (10 min) – kodowanie

Podłączenie głośnika/słuchawek

Omówienie kodu sterowania:

- wciśnięcie A odtwarza motyw DADADADUM
- wciśnięcie B odtwarza samodzielnie skomponowaną muzykę
- potrząśnięcie odtwarza dźwięk o zadanej częstotliwości

Uczniowie tworzą kod programu w edytorze MakeCode lub Python.

3. Sterowanie potencjometrem (5 min) – teoria

Nauczyciel krótko omawia zagadnienia związane z potencjometrem:

- zakres i jego sposoby skalowania
- możliwości podłączenia
- metody odczytu, zapisu wskazań

4. Sterowanie potencjometrem (15 min) – kodowanie

Omówienie kodu sterowania:

- skalowanie zakresu potencjometru na zakres częstotliwości lub głośność dźwięku
- zmiana położenia potencjometru zmienia częstotliwości lub głośność odtwarzanego dźwięku

Uczniowie tworzą kod programu w edytorze MakeCode lub Python.

5. Podsumowanie lekcji (5 min) – wnioski i pytania

- Jakie inne interaktywne elementy można dodać do systemu sterowania muzyką?
- Czy muzyka może być wykorzystana w naszym projekcie?
- Gdzie można użyć potencjometru?

Lekcja 5: Obsługa wyświetlacza diodowego micro:bita, sterowanie diodami LED i efekty świetlne

Cele lekcji:

- Tworzenie efektów świetlnych na wbudowanym wyświetlaczu micro:bita
- Sterowanie diodami micro:bita
- Zapoznanie uczniów z działaniem diod LED i ich programowaniem
- Poznanie sposobu mieszania barw RGB.
- Tworzenie efektów świetlnych w reakcji na interakcje użytkownika.
- Integracja diod LED z kodem micro:bita i zastosowanie w konstrukcji robota.

Materiały:

- learn:bity, micro:bity i komputery z dostępem do edytora MakeCode lub Python
- diody LED
- baterie i moduł zasilania do micro:bita
- kartony i elementy montażowe do mocowania diod

Uwagi metodyczne:

- *Eksperymentowanie*: uczniowie w praktyce testują efekty świetlne
- *Praktyka*: połączenie elektroniki z programowaniem
- *Interaktywność*: możliwość dynamicznej regulacji światła i animacji.

Przebieg lekcji:

1. Wprowadzenie wyświetlacz diodowy (5 min) – zasada działania matrycy micro:bita

Nauczyciel omawia budowę wyświetlacza i możliwości jego zaprogramowania:

- współrzędne diod
- funkcje sterowania diodami, ich jasnością i czasem wyświetlania

Krótką demonstracją: nauczyciel pokazuje przykładową animację na micro:bitcie

2. Programowanie sterowania wyświetlaczem micro:bita (10 min) – kodowanie

Omówienie kodu:

- uruchomienie rozszerzenia biblioteki neopixel
- potrząśnięcie zapala zaprogramowane diody
- przyciski A/B regulacja jasności 0-9

Uczniowie tworzą kod programu w edytorze MakeCode lub Python.

3. Zasady działania diod LED (10 min) – teoria

Nauczyciel omawia zagadnienia związane z diodami LED:

- sposoby mieszania kolorów RGB – czerwony, zielony, niebieski
- funkcje sterowania diodami LED – ich kolorami, jasnością

Krótką demonstracją: nauczyciel pokazuje przykładową animację świetlną

4. Programowanie sterowania diodami (10 min) – kodowanie

Omówienie kodu:

- naciśnięcie przycisku A – zapala diody LED
- naciśnięcie przycisku B – gasi diody LED
- potrząśnięcie – zmienia kolor diod
- (*) regulacja jasności diody poprzez potencjometr

Uczniowie tworzą kod programu w edytorze MakeCode lub Python.

5. Testowanie diod (5 min) – wstępny montaż i testowanie działania diod w praktyce

Uczniowie montują diody do robota.

- Sprawdzają, czy diody reagują poprawnie?

Jeśli występują problemy, wprowadzają zmiany efektów lub modyfikację testowanej konstrukcji.

6. Podsumowanie lekcji (5 min) – wnioski i propozycje ulepszeń

- Jakie animacje, efekty świetlne będą nam potrzebne?

Lekcja 6: Akcelerometr i czujniki

Cele lekcji:

- Obsługa czujników dźwięku, temperatury i światła – odczyt i interpretacja danych z otoczenia
- Wykorzystanie akcelerometru – robot reaguje na ruch i potrząśnięcie
- Efekty wizualne i dźwiękowe przy wykorzystaniu czujników

Materiały:

- micro:bity i komputery z dostępem do edytora MakeCode lub Python
- baterie i moduł zasilania do micro:bita

Uwagi metodyczne:

- *Interaktywność*: robot reaguje na ruch, dźwięk, światło, temperaturę
- *Interpretacja i praktyka*: uczniowie uczą się, jak interpretować i wykorzystać dane zebrane przez mikrokontroler

Przebieg lekcji:

1. Omówienie czujników wbudowanych w microbita (10 min) - teoria

Nauczyciel omawia możliwości interakcji mikrokontrolera z otoczeniem:

- akcelerometr – wykrywa ruch, przechylenie i potrząśnięcie micro:bita - może być używany do sterowania gestami.
- czujnik temperatury – mierzy temperaturę otoczenia
- czujnik światła – wykorzystuje diody na wyświetlaczu do pomiaru natężenia światła w otoczeniu
- mikrofon – mierzy natężenie dźwięku w otoczeniu
- kompas (magnetometr) – wykrywa pole magnetyczne i pozwala określić kierunek, w którym zwrócony jest micro:bit

2. Obsługa czujników temperatury i światła (15 min) – odczyt danych

Omówienie kodu:

- naciśnięcie przycisku A – wyświetla temperaturę
- naciśnięcie przycisku B – wyświetla natężenie światła
UWAGA! Warto zadbać o to, by w czasie pomiaru diody na wyświetlaczu były wyłączone.
- dotknięcie logo – wyświetla natężenie dźwięku otoczenia

Uczniowie tworzą kod programu w edytorze MakeCode lub Python.

3. Wykorzystanie akcelerometru (15 min) – reakcja na ruch

Nauczyciel przypomina działanie akcelerometru

- pozwala wykrywać ruch, potrząśnięcie i przechylenie
- omawia zakres danych przechyłu

Omówienie kodu:

- przechylenie mikrokontrolera sygnalizowane jest na wyświetlaczu strzałką zgodna z kierunkiem wychylenia

Uczniowie tworzą kod programu w edytorze MakeCode lub Python.

4. Podsumowanie lekcji (5 min) – wnioski i propozycje ulepszeń

- Jakie czujniki można wykorzystać w naszym robocie?
- Jak rozszerzyć interakcje? Czy czujniki można wykorzystać do sterowania robotem?

Lekcja 7: Obsługa wyświetlacza LED

Cele lekcji:

- Zapoznanie uczniów z zasadami działania wyświetlacza LCD
- Nauka sterowania wyświetlaczem LCD – wyświetlanie tekstu, liczb i dynamicznych komunikatów.
- Programowanie interakcji z użytkownikiem – zmiana wyświetlanych treści na podstawie przycisków lub czujników.
- Praktyczne zastosowanie wyświetlacza w projekcie edukacyjnym – wizualizowanie komunikatów i danych.

Materiały:

- learn:bit i micro:bity
- baterie i moduł zasilania do micro:bita
- komputer z dostępem do edytora MakeCode lub Python
- prototyp robota

Uwagi metodyczne:

- *Eksperymentowanie*: uczniowie w praktyce testują sterowanie wyświetlaczem LCD.
- *Interaktywność*: możliwość dynamicznej regulacji wyświetlanych komunikatów.
- *Logiczne myślenie*: uczniowie analizują i projektują interakcje między użytkownikiem, a mikrokontrolerem

Przebieg lekcji:

1. Wprowadzenie do wyświetlacza LCD (10 min) – teoria

Nauczyciel omawia budowę i funkcje wyświetlacza LCD - matryca znaków:

- 2 rzędy po 16 znaków
- każdy wyświetlany jest na siatce 5 x 8 pikseli
- można wyświetlać tekst, liczby, symbole oraz własne projekty

Demonstracja działania wyświetlacza: nauczyciel pokazuje przykładowe komunikaty i animacje na LCD.

2. Programowanie sterowania wyświetlaczem LCD (15 min) – kodowanie

Omówienie kodu:

- uruchomienie rozszerzenia biblioteki LCD
- wyświetlenie komunikatu tekstowego w pierwszym rzędzie
- wyświetlenie komunikatu „obrazkowego” stworzonego przez użytkownika

Można zmieniać pozycję tekstu, podając inne wartości x, y.

Uczniowie tworzą kod programu w edytorze MakeCode lub Python.

Omówienie kodu:

- Przycisk A – wyświetla komunikat
- Przycisk B – czyści ekran

Uczniowie modyfikują kod programu w edytorze MakeCode lub Python.

3. Tworzenie dynamicznych komunikatów na LCD (15 min) – programowanie animacji

Omówienie kodu:

- wykorzystanie akcelerometru lub potencjometru do sterowania obiektem na wyświetlaczu LCD

Uczniowie tworzą kod programu w edytorze MakeCode lub Python.

4. Podsumowanie lekcji (5 min) – wnioski i propozycje ulepszeń

- Gdzie można wykorzystać wyświetlacz LCD w naszym projekcie robota i jak go zamontować?
- Jakie komunikaty mogą wspomóc interakcję użytkownika z robotem?
- Jak można dynamicznie sterować wyświetlaczem?

Lekcja 8: Łączenie micro:bitów radiowo

Cele lekcji:

- Zapoznanie uczniów z komunikacją radiową w micro:bie – przesyłanie danych między urządzeniami.
- Konfiguracja kanałów radiowych – ustawienie parametrów transmisji.
- Programowanie przesyłania komend – sterowanie robotem za pomocą sygnałów radiowych.

Materiały:

- micro:bity i komputery z dostępem do edytora MakeCode lub Python
- baterie i moduł zasilania do micro:bita

Uwagi metodyczne:

- *Praca zespołowa*: współpraca w grupach 3–4 osobowych pozwala na rozdzielenie zadań.
- *Logiczne myślenie*: uczniowie analizują sposób działania systemu radiowego, ucząc się, jak przesyłać dane i sprawdzać ich poprawność.
- *Eksperymentowanie*: możliwość testowania różnych kanałów, filtrów wiadomości oraz rozszerzania kodu o dodatkowe funkcje sterujące robotem.

Przebieg lekcji:

1. Wprowadzenie do komunikacji radiowej (10 min) – teoria

Nauczyciel omawia zasadę działania radia w micro:bitach:

- komunikacja bezprzewodowa na określonych kanałach
- możliwość wysyłania wiadomości w formacie liczbowym lub tekstowym

Przykłady zastosowań:

- sterowanie innym micro:bitem
- wysyłanie danych między urządzeniami

2. Konfiguracja kanałów radiowych i przesyłanie/odbiór wiadomości (10 min) – praktyka

Omówienie kodu:

- konfiguracja kanału – wszystkie micro:bity muszą ustawiony ten sam kanał
- naciśnięciu przycisku A/B – wysłanie wiadomości
- odbiór wiadomości – sygnalizowany odpowiednim powiadomieniem na wyświetlaczu diodowym

Uczniowie tworzą kod programu w edytorze MakeCode lub Python.

Testowanie wysyłania wiadomości pomiędzy kilkoma mikrobitami.

3. Sterowanie micro:bitem na odległość (15 min) – praktyka

Omówienie kodu:

- konfiguracja kanału – wszystkie micro:bity muszą ustawiony ten sam kanał
- naciśnięciu przycisku A/B – wysłanie wiadomości, która u odbiorcy wywoła zadaną aktywność np. zapalenie diod, odtworzenie muzyki
- wykorzystanie acetometru do sterowania serwem podłączonym do odbiornika – przechyl do przodu obrót serwa o zadany kąt, przechyl do tyłu cofnięcie serwa na pozycję startową

Uczniowie tworzą kod programu w edytorze MakeCode lub Python.

Testowanie wysyłania wiadomości pomiędzy kilkoma mikrobitami.

4. Podsumowanie lekcji (10 min) – wnioski i propozycje ulepszeń

- Co zadziało, a co sprawia nam problem?
- Jakie inne funkcje można dodać do komunikacji radiowej?

Uczniowie tworzą schemat połączeń między micro:bitami.

Lekcja 9: Integracja AI – sterowanie gestami

Cele lekcji:

- Zapoznanie uczniów z działaniem Teachable Machine i nauka trenowania modelu AI
- Integracja micro:bita z modelem AI
- Połączenie komputera z micro:bitem komunikacja przez bluetooth i wykorzystanie aplikacji Teachable Micro:bit App

Materiały:

- komputer do trenowania modelu AI i obsługi Teachable Micro:bit App
- komputery z dostępem do edytora MakeCode (Python)
- micro:bity z modułami zasilania
- opcjonalnie elementy uzupełniające mikrokontroler np.: diody LED, serwomechanizmy itp.

Uwagi metodyczne:

- Interaktywność: wykorzystanie modelu AI do sterowania micro:bitem
- Logiczne myślenie: uczniowie analizują sposób działania modelu AI i możliwości zastosowania go w projekcie
- Eksperymentowanie: możliwość testowania różnych modeli opartych na obrazie lub dźwięku
- Rozwijanie umiejętności programistycznych: lekcja łączy elementy podstawowego programowania z bardziej zaawansowanymi mechanizmami AI i komunikacji bluetooth.

Przebieg lekcji:

1. Prezentacja działania modelu AI (10 min) – teoria

Nauczyciel prezentuje Teachable Machine - narzędzie online, które pozwala na łatwe trenowanie modeli sztucznej inteligencji do rozpoznawania obrazów, dźwięków, czy gestów.

Uczniowie uczą się:

- jak trenować model AI, który rozpozna kolory, kształty, gesty, czy dźwięki
- jak zapisać model i poprawić go do dalszego wykorzystania

Krótką demonstracją: nauczyciel pokazuje przykłady sterowania micro:bitem za pomocą modelu AI

2. Tworzenie modelu AI (15 min) – praktyka

Uczniowie uruchamiają Teachable Machine

- tworzą nowy projekt wykorzystujący gesty/kształty/kolory/dźwięki
- tworzą klasy i nagrywają przykładowe gesty/kształty/kolory/dźwięki
- trenują model
- testują poprawność działania modelu, rozpoznawania zadanych przykładów
- eksportują model do chmury

Dyskusja między grupami:

- Który model najlepiej sprawdzi się w naszym projekcie?
- Jakie klasy i przykłady będą najskuteczniej sterować robotem?

3. Połączenie modelu AI z micro:bitem (15 min) – kodowanie i obsługa Teachable Micro:bit App

Omówienie kodu:

- uruchomienie rozszerzenia biblioteki bluetooth
- zaprogramowanie reakcji na trenowane klasy modelu AI

Uczniowie tworzą kod programu w edytorze MakeCode (Python).

Uczniowie otwierają Teachable Micro:bit App:

- ładują model AI
- łączą aplikację z micro:bitem przez bluetooth
- testują sterowanie mikrokontrolera za pomocą wytrenowanego modelu AI

UWAGA Model AI czasami wymaga modyfikacji ustawień.

Urządzenie, na którym testujemy model powinno być wyposażone w tę samą kamerę, którą model był trenowany.

5. Podsumowanie lekcji (5 min) – wnioski i propozycje ulepszeń

- Jakimi elementami chcemy sterować modelem AI?
- Który model najlepiej sprawdzi się w naszym projekcie?
- Jakie klasy i przykłady będą najskuteczniej sterować robotem?

Uczniowie modyfikują schemat połączeń między micro:bitami.

Lekcja 10–11: Testowanie, prezentacja i ewaluacja

Cele lekcji:

- Integracja wszystkich funkcji robota – sprawdzenie poprawności działania układu sterowania.
- Prezentacja zespołowa – przygotowanie prezentacji o procesie tworzenia robota.
- Refleksja nad przebiegiem projektu – analiza trudności, sukcesów i propozycji ulepszeń.
- Ewaluacja pracy zespołowej – wypełnienie formularza oceniającego projekt i proces pracy.

Materiały:

- Gotowe, zmontowane roboty z wszystkimi funkcjami.
- Komputery do ewentualnych poprawek w kodzie.
- Formularze ewaluacyjne do analizy przebiegu projektu.
- Tablica i markery do zapisywania wniosków.

Uwagi metodyczne:

- *Podsumowanie całości projektu:* uczniowie integrują całą zdobytą wiedzę w praktycznym działaniu
- *Rozwój umiejętności prezentacyjnych:* uczniowie uczą się skutecznej prezentacji swoich pomysłów
- *Kreatywność i refleksja:* uczniowie analizują swój proces pracy, identyfikując mocne i słabe strony
- *Ocena i ewaluacja:* uczniowie zastanawiają się nad zdobytymi umiejętnościami i możliwościami dalszego rozwoju

Przebieg lekcji:**1. Testowanie działania wszystkich funkcji robota (50 min) – sprawdzanie poprawności integracji**

Każda grupa przeprowadza testy swojego robota:

- Czy robot poprawnie reaguje na komendy przycisków, akceleratora, czy potencjometru?
- Czy zamontowane serwa prawidłowo reagują na sterowanie?
- Czy diody LED zmieniają kolor zgodnie z zaprogramowanymi efektami?
- Czy dźwięki i animacje na wyświetlaczu działają poprawnie?
- Czy komunikacja radiowa między micro:bitami działa bez zakłóceń?
- Czy sterowanie modelem AI działa poprawnie?

Poprawki techniczne (jeśli wystąpią jakieś problemy):

- uczniowie dokonują niezbędnych poprawek w konstrukcji robota
- uczniowie dokonują zmian w kodzie

Nauczyciel wspiera uczniów w diagnozowaniu problemów i ich rozwiązywaniu.

2. Przygotowanie prezentacji zespołowej (10 min) – omówienie procesu tworzenia i funkcji robota

Każda grupa przygotowuje krótki scenariusz prezentacji swojego robota:

- Jakie funkcje posiada robot?
- Jakie wyzwania pojawiły się podczas budowy?
- Co było najtrudniejsze i jak udało się to rozwiązać?
- Jak można rozbudować projekt w przyszłości?

Każdy członek zespołu ma swoją rolę podczas prezentacji.

3. Prezentacja projektu (20 min) – pokaz robotów

Prezentacje zespołowe:

- Każda grupa przedstawia swojego robota, wyjaśniając jego działanie
- Pokaz sterowania robotem i prezentacja jego możliwości
- Odpowiedzi na pytania dotyczące funkcji robota i kodowania

4. Ewaluacja projektu i refleksja nad procesem pracy (10 min) – analiza doświadczeń

Dyskusja i porównanie projektów:

- Co wyróżnia poszczególne roboty?
- Jakie rozwiązania technologiczne były najbardziej innowacyjne?

Uczniowie wypełniają formularze ewaluacyjne, odpowiadając na pytania:

- Czego się nauczyliśmy podczas tego projektu?
- Co było największym wyzwaniem?
- Jak oceniasz pracę zespołu?
- Co można poprawić w przyszłości?

Wspólna analiza i podsumowanie najczęściej pojawiających się odpowiedzi.

Zapisanie propozycji ulepszeń do kolejnych projektów.