

PROGRAMOWANIE SONDY KOSMICZNEJ



135 minut (3 godziny lekcyjne)



szkoła podstawowa (VII–VIII klasa)



informatyka,technika



Tinkercad – Circuits | Arduino – programowanie | projektowanie układów elektronicznych | pomiar temperatury



www.esero.kopernik.org.pl



PROGRAMOWANIE SONDY KOSMICZNEJ

Autor: Marcin Duda

Opracowane dla ESERO-Polska

Poruszane wątki

- podstawy programowania systemu Arduino
- symulacja działania układów elektronicznych w programie Tinkercad – Circuits
- podstawowe parametry podzespołów elektronicznych
- pomiar temperatury w systemie Arduino

Rozwijane umiejętności

- posługiwanie się aplikacją Tinkercaddo symulacji układów elektronicznych
- programowanie mikrokontrolerów
- dostrzeganie zależności między zjawiskami
- obserwacja i wnioskowanie
- współpraca w grupie
- analiza danych

Metody pracy

- burza mózgów
- obserwacja
- praca w grupach
- metoda doświadczalna (symulacja)









CZAS

135 minut (3 godziny lekcyjne)



NIEZBĘDNE MATERIAŁY

- komputer lub tablet z dostępem do internetu
- arkusz do dyskusji załącznik 1
- arkusz ćwiczeniowy (dla zespołu)
 załącznik 2
- arkusz ćwiczeniowy (dla zespołu)
 załącznik 3
- arkusz ćwiczeniowy (dla zespołu)
 załącznik 4



MIEJSCE

sala lekcyjna / pracownia komputerowa

- arkusz Dioda LED (dla zespołu) załącznik 5
- arkusz ćwiczeniowy (dla zespołu)
 załącznik 6
- zdjęcia przykłady konstrukcji satelitów (dla zespołu)

Przygotowawnie zajęć

Do przeprowadzenia wszystkich ćwiczeń potrzebny ci będzie komputer lub tablet z dostępem do internetu. Program Tinkercad uruchamiany jest bezpośrednio w dowolnej przeglądarce internetowej. Przed pierwszym uruchomieniem programu należy założyć w nim konto. Procedura rejestracji jest standardowa i nie odbiega od stosowanych w innych aplikacjach. Adres aplikacji znajdziesz na końcu scenariusza w wykazie przydatnych linków.

Zacznij projektować Jak będziesz używać aplikacji Tinkercad?	
Jesteš w szkole? Nauczyciele mogą zacząć tutaj Jesteš uczniem? Dołącz do zajęć	
Samodzielnie Utwórz konto osobiste	
Masz już konto? Zaloguj się	



Wskazówka

Do efektywnego przeprowadzenia zajęć z projektowania układów elektronicznych w aplikacji Tinkercad – Circuits i obsługi systemu Arduino zaleca się wykorzystanie rzutnika, tablicy multimedialnej lub monitora multimedialnego. Ułatwi to demonstrację uczennicom/ uczniom prowadzonych symulacji, a także poprawę ewentualnych błędów pojawiających się w pisanym kodzie.





Co to jest satelita?

Zapowiedz uczennicom/uczniom, że na dzisiejszych zajęciach spróbują swoich sił jako konstruktorki/konstruktorzy – zaprojektują i przetestują komputerowy system pomiarowy dla satelity z wykorzystaniem aplikacji Tinkercad.



Zacznij zajęcia od burzy mózgów. Zadaj uczennicom/uczniom pytania:

- co im się kojarzy ze słowem "satelita"?
- co to jest "sztuczny satelita"?
- w jakim celu można stosować satelity?

Wypisz na tablicy ich skojarzenia.

Wyświetl lub przyklej do tablicy zdjęcie CanSata (możesz skorzystać z załącznika 1 lub ze zdjęć z internetu). Spytaj, czy coś im mówi ta nazwa. Wyjaśnij, że jest to minisatelita wielkości puszki napoju. Jego zadaniem jest przeprowadzenie różnego rodzaju badań i eksperymentów. Urządzenia te wykorzystywane są do nauki i zapoznawania z technologiami kosmicznymi poprzez rywalizację w konkursach organizowanych na całym świecie. W Polsce konkurs CanSat organizowany jest przez Biuro Edukacji Kosmicznej ESERO-Polska. Zadaniem drużyn konkursowych jest zbudowanie urządzenia, które uwzględnia wszystkie główne podsystemy obecne na prawdziwym satelicie, takie jak: moduł zasilania, komunikacji, komputer pokładowy. Minisatelita powinien być w stanie przeprowadzić badania i eksperymenty.

Więcej o CanSatach znajdziesz na: https://esero.kopernik.org.pl/konkurs-cansat/.

Arduino

Powiedz uczennicom/uczniom, że do pomiaru wybranych przez zespół wielkości fizycznych CanSat wykorzystuje system minikomputera Arduino, do którego możemy podłączyć różne czujniki, a następnie go zaprogramować, aby przeprowadzał pomiary w czasie lotu ministaelity. Arduino jest mikrokomputerem mającym postać płytki wyposażonej w złącze USB (tzn. uniwersalną magistralę szeregową, z ang. Universal Serial Bus) służące do komunikacji z komputerem, a także inne złącza wykorzystywane do podłączania zewnętrznych elementów elektronicznych, takich jak silniki, przekaźniki, fotodiody, diody laserowe, głośniki, mikrofony itp. Więcej informacji znajdziesz tu: https://www.arduino.cc/.





Wskazówka

Jeżeli ktoś z klasy będzie potrafił podać informacje lub powiedzieć parę słów o komputerze Arduino, poproś, by podzielił się swoją wiedzą z pozostałymi.



Ciekawostka

Możliwości zastosowania Arduino jest tak wiele, że trudno wymienić chociaż ułamek z nich – tak naprawdę każdym urządzeniem elektrycznym, które mamy w domu, jesteśmy w stanie sterować, korzystając z Arduino. Możemy również zbudować urządzenie z dostępem do internetu. Zakres możliwości płytki zaczyna się od sterowania prostymi elementami elektronicznymi, takimi jak diody, a kończy się na niezwykle precyzyjnych urządzeniach, takich jak frezarki czy drukarki 3D.



Podziel uczennice/uczniów na zespoły (2–4-osobowe). Zapewnij każdemu zespołowi komputer lub tablet z dostępem do internetu. Wyjaśnij, że ich zadaniem jest odszukanie w sieci informacji na temat minikomputera Arduino i uzupełnienie arkusza ćwiczeń. Rozdaj schemat Arduino (załącznik 2) i poproś o uzupełnienie. Zwróć uwagę, aby informacje umieszczane w arkuszu ćwiczeń były precyzyjne. Możesz skorzystać z linków zamieszczonych na końcu scenariusza. Krótkie filmy pokazują możliwości zastosowania systemu Arduino i współpracujących z nim czujników oraz podzespołów elektronicznych. Gdy uczennice/uczniowie wykonają zadanie, zespoły przedstawiają rezultaty swojej pracy. Poniżej znajduje się przykład wypełnionego załącznika 2.



[źródło: https://www.tinkercad.com/]



¹⁵_{min} Tinkercad – twój warsztat pracy

Przechodzimy do aplikacji Tinkercad, która pozwala na projektowanie konstrukcji 3D, tworzenie obwodów elektronicznych oraz oprogramowania. Wszystko to zostało przygotowane w sposób, który ma pomóc początkującym. Aby ułatwić wszystkim poznawanie możliwości Tinkercada, udostępniono darmowe kursy z zakresu zarówno części mechanicznej (3D), jak i części elektronicznej (Circuits). Dostępnych jest kilka lekcji, podczas których użytkownik krok po kroku zapoznaje się z możliwościami programu.

Aby uczennice/uczniowie rozpoczęły/rozpoczęli pracę, muszą założyć konto. Rozdaj zespołom załącznik 3 i poproś, aby postępowały zgodnie z instrukcją. Sprawdź, czy każdy zespół uruchomił aplikację Tinkercad. Po zakończeniu pracy z załącznikiem 3 spytaj, co było łatwe, a co sprawiło problemy podczas pracy z programem, i jakie dostrzegają zalety i wady wirtualnego projektowania i testowania układów elektronicznych. Zwróć uwagę młodych osób na bezpieczeństwo i brak kosztów przy błędnym zaprojektowaniu obwodu. W rzeczywistym środowisku uruchomieniowym błędnie podłączone podzespoły elektronicznie uległyby nieodwracalnemu uszkodzeniu. Tinkercad automatycznie zapisuje nasze projekty w chmurze, czyli możemy korzystać z nich wszędzie, na każdym komputerze (z dostępem internetu).

Zapowiedz, że na następnej lekcji uczennice/uczniowie będą programować i testować system mikrokomputera Arduino za pomocą symulatora.

Czas zaprogramować Arduino!

Wyjaśnij uczennicom/uczniom, że każdy program komputerowy jest zbiorem rozkazów (poleceń). Rozkazy są wykonywane w ciągu – jeden po drugim. Jeden rozkaz to jeden wiersz kodu. Wyświetl poniższą ilustrację:



Źródło: The open-source Arduino Software (IDE)

Wytłumacz, że w Arduino programowanie jest uproszczone. Otóż każdy zbiór rozkazów, czyli program, składa się tu z minimum dwóch funkcji: setup() i loop(). **Funkcja** to zestaw instrukcji, które mają zostać wykonane. Funkcja setup() wywoływana jest tylko raz podczas uruchamiania płytki. Natomiast loop() działa do momentu przerwania jej przez użytkownika.





Wskazówka

Symbolami "//" oznaczamy komentarze, czyli informacje mieszczące się w jednej linijce, które pomagają użytkownikowi w zrozumieniu programu. Podczas kompilacji zostają one pominięte. Jeśli chcemy umieścić dłuższy komentarz, to należy zawrzeć go /* **w takich symbolach** */. **Komentowanie jest bardzo ważne!** Pamiętaj, aby komentować kod, nawet gdy programy piszesz tylko na własny użytek.

Poproś uczennice/uczniów o uruchomienie aplikacji Tinkercad i otworzenie modułu Circuits, a następnie o dodanie płytki Arduino Uno do obszaru roboczego. Sprawdź, czy młode osoby poradziły sobie z zadaniem. Pojawi się nowa opcja w menu: Kod – obok przycisku – Uruchom symulację. Po wciśnięciu tego przycisku rozwinie się pole umożliwiające pisanie programu w następujący sposób:

- opcja BLOKI umożliwia programowanie Arduino w sposób graficzny (przez dodawanie kolejnych rozkazów programu umieszczonych w "klockach" w stylu Scratcha),
- opcja BLOKI + TEKST jest to połączenie programowania graficznego z tekstowym wprowadzaniem rozkazów programów. Opcja ta umożliwia podgląd kodu programu, który jest automatycznie tworzony, gdy uczennice/uczniowie "budują program z klocków",
- opcja TEKST (zalecana) uczennice/uczniowie programują Arduino, pisząc poszczególne komendy programu.

Teraz napiszemy program, który będzie włączał i wyłączał diodę LED. Diody znajdują zastosowanie w wielu projektach. Są one wyposażone w dwie elektrody (końcówki), do których musimy odpowiednio podłączyć zasilanie. Do anody (+) podłączamy biegun dodatni, natomiast do katody (-) biegun ujemny. Odwrotne podłączenie (polaryzacja) spowoduje, że dioda nie zaświeci. Dowodzi to, że diody przewodzą prąd tylko w jednym kierunku. Na płytce Arduino mamy do czynienia z napięciem 5 V, natomiast diody LED działają przy napięciu ok. 1,2 V. Jeżeli użyjesz rezystora, to zredukuje on napięcie do poziomu bezpiecznego dla diody. Rezystor ma prostą budowę – przypomina walec z dwoma końcówkami (elektrodami). Podstawowym parametrem rezystora jest opór (czyli rezystancja), a jego wartość wyraża się w omach (Ω). Do ograniczenia prądu diody LED najczęściej stosowany jest rezystor o wartości 220 Ω . Do wyjaśnienia budowy diody pomoże ci załącznik 3 umieszczony na końcu scenariusza. Poleć młodym osobom skonstruowanie obwodu elektrycznego w programie Tinkercad – Circuits zgodnie rysunkiem przedstawionym w załączniku 3. Tak wykonany obwód uczennice/uczniowie uruchomią i przetestują jego działanie, czego efektem będzie pulsowanie diody LED.

Sprawdź, czy wszyscy poradzili sobie z zadaniem.



Pokaż poniższą ilustrację:

```
1 void setup() {
2   pinMode(8, OUTPUT); //Konfiguracja pinu 8 jako wyjście
3 }
4
5 void loop() {
6   digitalWrite(8, HIGH); //Włączenie diody
7   delay(1000); //Odczekanie 1 sekundy
8   digitalWrite(8, LOW); //Wyłączenie diody
9   delay(1000); //Odczekanie jednej sekundy
10 }
```

Źródło: The open-source Arduino Software (IDE)

Wskaż, że program, który steruje diodą, jest umieszczony pod przyciskiem KOD. Wyjaśnij uczennicom/uczniom poszczególne polecenia programu:

<> E Arduino

- Funkcja pinMode(Pin, Tryb) umożliwia wybranie, czy dany pin jest wejściem, czy wyjściem. Pin może być liczbą całkowitą z zakresu od 0 do 13, zaś Tryb to INPUT lub OUTPUT. Jeżeli chcemy sterować wyjściem, to zawsze używamy trybu Output. Dzięki takiej konfiguracji możemy ustawić stan logiczny na wyjściu i dzięki temu włączyć diodę.
- 2. Funkcja digitalWrite(Pin, Stan). Stan jest stanem logicznym, który może być HIGH bądź LOW (wysoki 5 V bądź niski 0 V).
- 3. Funkcja **delay(Czas)** funkcja ta jako argument przyjmuje **liczbę milisekund**, na jaką ma zostać zatrzymane wykonywanie kolejnych linii kodu.



Powiedz uczennicom/uczniom, że mogą teraz poeksperymentować i sprawdzić:

Przy jakiej najmniejszej wartości opóźnień będą w stanie zauważyć miganie diody?

Co stanie się, gdy dioda będzie migała zbyt szybko?

Poproś młode osoby, aby omówiły spostrzeżenia z przeprowadzonych w programie zmian i wskazały, co zaobserwowały.





Optyczny system ostrzegawczy dla lądownika

Zapowiedz uczennicom/uczniom aby, wybrały/wybrali wolny pin i podłączyły/ podłączyli do niego drugą diodę. Poproś, aby napisały/napisali program, który będzie włączał obie diody LED, a następnie zmodyfikowały/zmodyfikowali go w taki sposób, aby obie diody migały na zmianę. Po wykonaniu zadania zespoły prezentują i omawiają swoje projekty.

Po zakończeniu pracy zapytaj:

- co było łatwe, a co sprawiło trudność w wykonaniu zadania?
- co wpływa na częstotliwość pulsowania diod?

Poproś uczennice/uczniów, aby zaprojektowały/zaprojektowali kolorowy system optyczny dla lądownika z wykorzystaniem w programie Tinkercad systemu Arduino i diody trójkolorowej. Program sterujący ma zaświecać poszczególne kolory (czerwony, niebieski, zielony) z zaprogramowaną przez uczennice/ uczniów częstotliwością.



Wskazówka

Dioda RGB (red, green, blue) zawiera w sobie trzy odrębne elementy (diody LED) wytwarzające światło. Każdy z nich emituje jedną podstawową barwę światła: czerwoną, zieloną lub niebieską. Z połączenia barw RGB (jednoczesna praca trzech źródeł barw) uzyskuje się światło białe.



Źródło: https://abc-rc.pl/product-pol-6849-Dioda-LED-RGB-5mm -wspolna-Anoda-matowa-ARDUINO-10-szt.html





Powiedz uczennicom/uczniom, że mogą teraz poeksperymentować i sprawdzić:

- co się stanie, jeżeli zaświecą jednocześnie dwa wybrane kolory?
- co się stanie, gdy zaświecą wszystkie trzy kolory?

Poproś młode osoby, aby omówiły spostrzeżenia z przeprowadzonych w programie zmian i wskazały, co zaobserwowały.



Czy leci z nami czujnik? Pomiar temperatury w systemie Arduino.

Wyjaśnij uczennicom/uczniom, że do tej pory przy konstruowaniu układów elektronicznych w systemie Arduino korzystały/korzystali tylko ze złączy cyfrowych, na których ustawiali stany: wysoki 5 V i niski 0 V. W kolejnym projekcie użyjecie złącz (pinów) analogowych. Wskaż, że zgodnie z informacjami wyszukanymi na poprzedniej lekcji za pomocą pinów analogowych możemy mierzyć napięcie przyłożone do nich w zakresie od 0 V do 5 V. Do minikomputera Arduino możemy podłączyć wiele czujników, które mierzą różne wielkości fizyczne, np.: temperaturę, ciśnienie, wilgotność, przyspieszenie, odległość itp. W tym projekcie uczennice/uczniowie wykorzystają analogowy czujnik temperatury oznaczony TMP 36. Wyjaśnij, że każdy czujnik i jego parametry są umieszczone w tzw. nocie aplikacyjnej (ang. application note) podzespołów elektronicznych, które są tworzone przez producentów podzespołów i dostępne w internecie. Łatwo wyszukać te informacje.

Uczennice/uczniowie w parach, korzystając z sieci internet, odszukują notę aplikacyjną dla czujnika TMP36. Po pobraniu dokumentu poproś młode osoby, aby odczytały następujące informacje:

- napięcie zasilania czujnika (od 2,7 V do 5,5 V),
- zakres pomiarowy mierzonej temperatury (od -40°C do +120°C),
- dokładność pomiarową (0,5°C),
- skalę pomiaru 10mV / 1°C.

Do wykonania tego zadania uczennice/uczniowie wykorzystają arkusz ćwiczeniowy umieszczony na końcu scenariusza (załącznik 4).



Po wykonaniu ćwiczenia omawiają dane, które odczytali z noty aplikacyjnej.



Następnie zespoły uruchamiają aplikację Tinkercad – Circuits. Z opcji programu – **ZESTAWY STARTOWE PODSTAWOWE** – wybierają obwód czujnika temperatury z zasilaniem i woltomierzem.





źródło: https://www.tinkercad.com/.

Wyjaśnij, że za pomocą tego obwodu można przetestować parametry czujnika związane z mierzonymi temperaturami i zmianą napięcia na czujniku. Uczennice/uczniowie, klikając podczas symulacji na rysunek czujnika, mogą zmieniać wartość wirtualnej temperatury, jednocześnie obserwując wskazania woltomierza. Porozmawiajcie, czy deklarowany parametr czujnika 10mV na 1°C jest prawidłowo przez program symulowany. Wyniki symulacji uczennice/ uczniowie notują, wykorzystując arkusz ćwiczeniowy znajdujący się na końcu scenariusza – załącznik 6.

Poproś młode osoby, aby omówiły swoje spostrzeżenia i porównały swoje wyniki.

Zwróć uwagę, że odczytane dane wskazują na liniowy (proporcjonalny) wzrost temperatury i napięcia czujnika.

Następnie poproś uczennice/uczniów, aby w programie Tinkercad – Circuits zbudowały/zbudowali obwód termometru elektronicznego dla systemu Arduino, wykorzystując rysunek z arkusza ćwiczeniowego – załącznik 4. Zwróć uwagę na prawidłowe podłączenie czujnika temperatury do systemu Arduino:

- pin czujnika GND (masy "–") do pinu Arduino GND,
- pin czujnika Vout-- do pinu analogowego A1,
- pin Vcc czujnika do pinu 5 V Arduino.

Gdy zespoły będą mały wykonany obwód termometru w aplikacji Tinkercad – Circuits, wspólnie z nimi uruchom edytor kodu naciskając przycisk: **KOD** i wybierz opcję **TEKST**. Wspólnie z młodymi osobami (w kilku krokach) napisz program sterujący termometrem, wyjaśniając użyte komendy języka programowania.



 Ustawiamy stałą odpowiadającą pinowi, do którego podłączono czujnik (A0), a w funkcji setup() definiujemy port szeregowy (Serial) łączący płytkę Arduino z komputerem.

[źródło: https://www.tinkercad.com/]

 W funkcji loop() wstawiamy kolejną funkcję, pozwalającą na odczyt napięcia proporcjonalnego do temperatury otoczenia. Temperaturze 0°C odpowiada 500mV (0,5 V). Dzięki temu możliwe jest zmierzenie temperatury ujemnej. Zmiana temperatury na wyższą o1°C oznacza dodanie 10mV, przy niższej – odjęcie tej wartości.

Temperatura się zmienia, musimy więc założyć, że wartość reprezentująca ją będzie przechowywana w zmiennej. Całość należy rozumieć następująco: odczytaj z czujnika temperatury wartość napięcia i przekaż ją do zmiennej odczyt.

```
Tekst
                               +
1 int czujnik = Al;
                          //pin analogowy Al połączony z sygnałem z czujnika
2
   void setup() {
     Serial.begin(9600); //inicjalizacja monitora szeregowego
3
4
5
   1
6
7
   void loop() {
8
       int odczyt = analogRead(czujnikPin); //odczytanie wartości z czujnika
9
10 }
```

[źródło: https://www.tinkercad.com/]

3. Wartość zwróconą przez funkcję analogRead() należy przekształcić na rzeczywiste napięcie, które uzyskujemy z czujnika:

odczytana wartość * 5 V,

gdzie 5 V oznacza zasilanie, którego dostarcza płytka Arduino.



Następnie uzyskany wynik dzielimy przez 1024. Tworzymy nową zmienną, tym razem jednak zastosujemy typ danych float. Poniżej zmiennej odczyt definiujemy zmienną napięcie.



[źródło: https://www.tinkercad.com/]



Wskazówka

Pamiętaj, aby w nazwach zmiennych nie stosować polskich znaków. Zmienne mają ściśle ustalone typy danych:

- bit przyjmuje wartość 0 lub 1,
- bool skrót od boolean, przyjmuje wartość true (prawda) lub false (fałsz),
- byte przechowuje wartość całkowitą, 8-bitową, z zakresu 0–255,
- char przechowuje pojedynczą literę,
- int przechowuje wartości z zakresu od –32768 do +32767,
- long przechowuje wartości ujemne i dodatnie znacznie dłuższe niż int.

Są też dodatkowe rodzaje zmiennych do liczb ułamkowych:

- float zapamiętuje ułamki z dokładnością do 7 cyfr po przecinku,
- **double** zapamiętuje ułamki z dokładnością do 15 cyfr po przecinku.
- 4. Przekształcamy wartości napięcia na temperaturę w skali Celsjusza.

Skorzystamy ze wzoru: obliczone napięcie (V – 0,5 V) × 100, gdzie: 0,5 V oznacza wartość początkową.



[źródło: https://www.tinkercad.com/]



5. Kiedy wszystko mamy już obliczone, pora ustalić, w jaki sposób będziemy wyświetlać wyniki na ekranie. Do wyświetlania wyników na ekranie monitora służy funkcja: Serial.print(wyświetlany tekst).



[źródło: https://www.tinkercad.com/]

- 6. Wskaż uczennicom/uczniom, aby uruchomili symulację i podgląd monitora portu, na którym pojawią się zmierzone dane. Aby uruchomić monitorowanie portów, należy wybrać opcję: konsola szeregowa.
- 7. Wytłumacz, że wprawdzie wyniki pomiaru temperatury się pojawiły, lecz są wypisywane w konsoli tak szybko, że jest problem z odczytem konkretnych wartość. Zapytaj, czy ktoś ma pomysł na rozwiązanie tego problemu. Jeżeli uczennice/uczniowie mają propozycję na rozwiązanie, niech zaprezentują ją reszcie klasy.

Należy wykorzystać znaną z poprzedniej lekcji funkcję delay(), dzięki której, prezentowane dane na ekranie monitora pozostaną przez określony czas.



Wskazówka

Tinkercad – Circuits ma również wbudowane narzędzie do automatycznego wykonywania wykresu danych szeregowych, pod warunkiem że strumień nie zawiera w sobie żadnego tekstu. Jest to przydatne narzędzie do wizualizacji zmian odczytów czujnika i innych danych wejściowych, a także do śledzenia zmiennych w programie.

Cały kod programu obsługującego analogowy czujnik temperatury TMP 36 w systemie Arduino jest podany poniżej wraz z komentarzem.



Tekst 🝷 🛓 🚔 👗	1 (Ardu	ino Uno R3	•
<pre>const int crujnikPin = A0; //pin analogowy A1 połączony z sygnałem z crujnika wold setup() { scrial.begin(9600); //inicjalizacja monitora szeregowego } voud loop() { int odczyt = analogRead(crujnikPin); //odczytanie wartości z crujnika float napiecie = odczyt + 5.0; //przeliczenie odczytanej wartości na napięcie w woltach (dla podłączenia pod 5V) napiecie = napiecie / 1024.0; float tempC = (napiecie - 0.6) + 1001 //konwersja z napięcia na temperaturę. Serial.println(tempC); //wyświetlenie wartości temp. monitorze delay(1000); //opóźnienie między kolejnymi odczytami } </pre>			
🔂 Konsola szeregowa			
27.15 24.71 24.71 24.71 24.71 22.75 23.69 23.69 23.69 23.69 23.69 23.69 23.69 23.69 23.69 23.69 23.69 25.69 25.59			_
	Wydiij	Wyczyść	₩



Zadanie dodatkowe

Zachęć młode osoby, by w domu samodzielnie próbowały eksperymentować z kodem programu termometru. Niech podłączą do płytki Arduino drugi czujnik temperatury i zmodyfikują program tak, aby odczytywał wartość z dwóch czujników na raz, a wyniki pomiarów wyświetlał w monitorze portu. Na kolejnych zajęciach porozmawiacie, jak uczennice/uczniowie rozwiązały/rozwiązali ten problem.



Ciekawostka

Podstawowa misja minisatelity CanSat jest taka sama dla wszystkich uczestniczących w konkursie zespołów i polega na pomiarze temperatury powietrza. Dane muszą być przekazywane przez CanSat przynajmniej raz na sekundę, tak aby zespół mógł je analizować i wyświetlić na wykresie. Drugorzędna misja CanSat może zostać wybrana przez każdy z uczestniczących zespołów. Na przykład mógłby to być pomiar zaawansowanych parametrów, takich jak przyspieszenie, lokalizacja GPS lub otaczających poziomów promieniowania.



źródło: https://www.esa.int/Education/CanSat/ Teams_for_the_2016_european_cansat_competition_selected

Podsumowanie

Zastanówcie się wspólnie, w jaki sposób można jeszcze wykorzystać program Tinkercad do realizacji rzeczywistej misji satelity CanSat. Jako podsumowanie zaprezentuj młodym osobom film ilustrujący pracę zespołów z całej Europy realizujący rzeczywiste misje ministaelitów CanSat. Skorzystaj z linków zamieszczonych na końcu scenariusza i złącznika prezentującego przykładowe fotografie wykonanych modeli satelitów.

Wykaz przydatnych linków:

• program Tinkercad: https://www.tinkercad.com/

• przykłady projektów Arduino:

https://www.youtube.com/watch?v=DRH-GdZxl-o https://www.youtube.com/watch?v=Ox-9eOc3bQU

• oficjalna strona projektu Arduino: https://www.arduino.cc/

• informacje o polskiej edycji konkursu CanSat: https://esero.kopernik.org.pl/konkurs-cansat/

- informacje o europejskich zawodach CanSat: https://www.esa.int/Education/CanSat
- rysunek płytki minikomputera Arduino Uno: https://pixabay.com/pl/illustrations/ arduino-arduino-uno-technologia-2168193/

• europejskie zawody CanSat – Norwegia 2012: https://www.youtube.com/watch?v=auCQMBL6dDc



Odniesienie do podstawy programowej

INFORMATYKA Cele kształcenia – wymagania ogólne

2) Programowanie i rozwiązywanie problemów z wykorzystaniem komputera oraz innych urządzeń cyfrowych: układanie i programowanie algorytmów, organizowanie, wyszukiwanie i udostępnianie informacji, posługiwanie się aplikacjami komputerowymi.

4) Rozwijanie kompetencji społecznych, takich jak komunikacja i współpraca w grupie, w tym w środowiskach wirtualnych, udział w projektach zespołowych oraz zarządzanie projektami.

Treści nauczania – wymagania szczegółowe KLASY VII I VIII

II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera i innych urządzeń cyfrowych. Uczeń:

- projektuje, tworzy i testuje programy w procesie rozwiązywania problemów. W programach stosuje: instrukcje wejścia/wyjścia, wyrażenia arytmetyczne i logiczne, instrukcje warunkowe, instrukcje iteracyjne, funkcje oraz zmienne i tablice;
- 2) projektuje, tworzy i testuje oprogramowanie sterujące robotem lub innym obiektem na ekranie lub w rzeczywistości;
- 3) korzystając z aplikacji komputerowych, przygotowuje dokumenty i prezentacje, także w chmurze, na pożytek rozwiązywanych problemów i własnych prac z różnych dziedzin (przedmiotów), dostosowuje format i wygląd opracowań do ich treści i przeznaczenia
- wyszukuje w sieci informacje potrzebne do realizacji wykonywanego zadania, stosując złożone postaci zapytań i korzysta z zaawansowanych możliwości wyszukiwarek.
- IV. Rozwijanie kompetencji społecznych. Uczeń:
- 1) bierze udział w różnych formach współpracy, jak: programowanie w parach lub w zespole, realizacja projektów, uczestnictwo w zorganizowanej grupie uczących się, projektuje, tworzy i prezentuje efekty wspólnej pracy;
- ocenia krytycznie informacje i ich źródła, w szczególności w sieci, pod względem rzetelności i wiarygodności w odniesieniu do rzeczywistych sytuacji, docenia znaczenie otwartych zasobów w sieci i korzysta z nich;
- 4) określa zakres kompetencji informatycznych, niezbędnych do wykonywania różnych zawodów, rozważa i dyskutuje wybór dalszego i pogłębionego kształcenia, również w zakresie informatyki.



TECHNIKA Cele kształcenia – wymagania ogólne

- I. Rozpoznawanie i opis działania elementów środowiska technicznego.
- 1) Postrzeganie elementów środowiska technicznego jako dobro materialne stworzone przez człowieka.
- 4) Rozróżnianie elementów budowy wybranych narzędzi, przyrządów i urządzeń technicznych.
- 9) Wyszukiwanie informacji na temat nowoczesnych dziedzin techniki, ciekawostek i wynalazków technicznych
- V. Rozwijanie kreatywności technicznej
- 1) Poznawanie siebie oraz swoich predyspozycji do wykonywania zadań technicznych.
- 2) Rozwijanie zainteresowań technicznych.
- 3) Przyjmowanie postawy twórczej, racjonalizatorskiej.

Treści nauczania – wymagania szczegółowe

- IV. Dokumentacja techniczna. Uczeń:
- 5) analizuje rysunki zawarte w instrukcjach obsługi i katalogach;
- 6) odczytuje i interpretuje informacje zamieszczone w instrukcjach obsługi urządzeń, na tabliczce znamionowej, opakowaniach żywności, metkach odzieżowych, elementach elektronicznych itp.;
- 7) projektuje i konstruuje modele urządzeń technicznych, w tym elektryczno- -elektronicznych.
- V. Mechatronika. Uczeń:

1) wyjaśnia na przykładach prostych urządzeń zasady współdziałania elementów mechanicznych, elektrycznych i elektronicznych;

3) konstruuje, m.in. z gotowych elementów, zabawki, roboty, modele mechaniczno-elektroniczne, w tym programowalne.

Źródło: Rozporządzenie Ministra Edukacji Narodowej z dnia 14 lutego 2017 r. w sprawie podstawy programowej wychowania przedszkolnego oraz podstawy programowej kształcenia ogólnego dla szkoły podstawowej, w tym dla uczniów z niepełnosprawnością intelektualną w stopniu umiarkowanym lub znacznym, kształcenia ogólnego dla branżowej szkoły I stopnia, kształcenia ogólnego dla szkoły specjalnej przysposabiającej do pracy oraz kształcenia ogólnego dla szkoły policealnej (Dz. U. z 2017 r., poz. 356).

















1. Konstrukcja płytki Arduino

Podstawą systemu Arduino jest **płyta główna**. To termin znany użytkownikom komputerów. Tutaj płytą główną nazywa się cały komputer Arduino. Odszukaj w internecie informację o zaznaczonych na poniższym rysunku komponentach (podzespołach) komputera Arduino. Nazwy poszczególnych elementów płytki Arduino wpisz w odpowiednie miejsca







Uruchomienie aplikacji Tinkercad

A. Uruchom dowolną przeglądarkę internetową, następnie wpisz adres strony: https://www. tinkercad.com/. Zobaczysz stronę powitalną aplikacji.



- B. Odszukaj i naciśnij przycisk: **DOŁACZ TERAZ**, a następnie przycisk: **UTWÓRZ KONTO OSOBISTE**.
- C. Masz teraz do wyboru:
- logowanie za pomocą dowolnego konta (jeśli je posiadasz) lub
- zarejestrowanie się z użyciem adresu e-mail (jeśli nie masz konta na wskazanych przez aplikację portalach).



- D. Kolejny etap to już tylko potwierdzenie utworzenia konta uruchamiamy swoją pocztę i potwierdzamy nadesłany e-mail z Grupy Autodesk. Możemy działać!
- E. Po prawidłowo wykonanej procedurze uzyskasz dostęp do aplikacji.



F. Następnie wybieramy moduł – Circuits (obwody elektryczne) z menu po lewej stronie:



G. Spróbujcie zaświecić diodę odpowiednią baterią

Z prawej strony ekranu do wyboru mamy różne podzespoły elektroniczne. Wybierając konkretny podzespół, umieszczamy go za pomocą wskaźnika myszki na środkowej części ekranu aplikacji, gdzie będziemy je łączyli w jeden obwód elektryczny. Aby połączyć podzespoły w jeden obwód elektryczny, ustawiamy wskaźnik myszki na wybranej elektrodzie (końcówce) podzespołu i naciskamy przycisk myszki. Następnie prowadzimy wirtualny przewód do następnej elektrody wybranego elementu, w ten sposób zamykając cały obwód elektryczny.

Po połączeniu wszystkich elementów możemy symulować ich pracę, włączając przycisk **uruchom/zatrzymaj symulację** i jednocześnie obserwując zachowanie poszczególnych elementów skonstruowanego obwodu.

Aby usunąć wybrany element elektroniczny z obwodu, należy jednokrotnie kliknąć na nim wskaźnikiem myszki, co spowoduje jego zaznaczenie, a następnie nacisnąć przycisk DELETE na klawiaturze komputera. Możecie poprosić nauczycielkę/nauczyciela o pomoc.

H. W tym samym menu znajdują się gotowe obwody. Jeśli wystarczy wam czasu, poeksperymentujcie z nimi. Możecie to też zrobić w domu po zajęciach.





Czujnik temperatury TMP 36 i sposób podłączania do Arduino.

Analogowy czujnik temperatury TMP 36			
parametr	wartość jednostki		
napięcie zasilania czujnika			
zakres mierzonej temperatury			
dokładność pomiarowa			
skala pomiaru			





Źródło: https://www.tinkercad.com/





Dioda LED – oznaczenie elektrod i sposób podłączenia do Arduino



Źródło: https://www.tinkercad.com/





Arkusz pomiarowy temperatury i napięcia na czujniku TMP 36

Nr pomiaru	Wartość ustawionej tempe- ratury na czujniku TMP 36	Wartość napięcia odczy- tana na woltomierzu
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		



Załącznik 7 Przykłady konstrukcji satelitów CanSat









Źródło: https://www.narom.no/undervisningsressurser/the-cansat-book/the-primary-mission/cansat-mechanics-design/, https://canduino.eu/index.php?id=startkit, https://kosmonauta.net/2017/04/final-polskiej-edycji-zawodow-cansat/#prettyPhoto, https://kosmonauta. net/2020/03/krakowscy-licealisci-konstruuja-wlasnego-cansata/#prettyPhoto

