



Czujniki PIR oraz DHT i analiza zmian w otoczeniu



Zagadnienia informatyczne

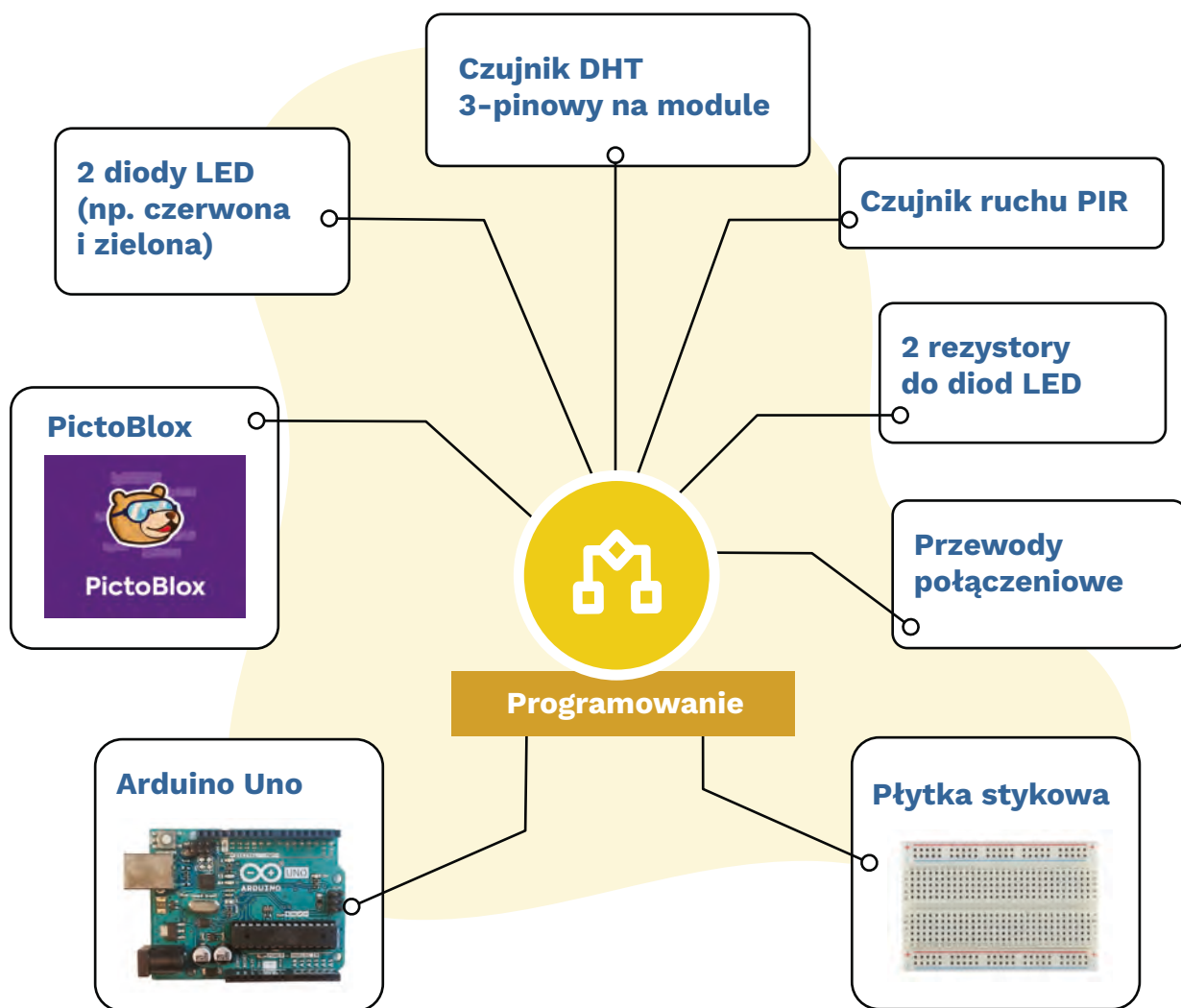
W tej części uczniowie poznają czujnik ruchu PIR jako element wejściowy, który pozwala wykrywać obecność ruchu w otoczeniu. Uczą się, w jaki sposób Arduino odczytuje sygnał z czujnika oraz jak program może reagować na pojawienie się ruchu poprzez wyświetlenie komunikatu, zmianę stanu diod LED lub zliczanie wykrytych zdarzeń. Pracując w środowisku PictoBlox, uczniowie tworzą programy reagujące na sygnał z czujnika i obserwują, że mikrokontroler może nie tylko odbierać informacje z otoczenia, ale także podejmować działania na ich podstawie. Dzięki temu rozumieją, jak działają proste systemy automatyki, alarmy oraz układy reagujące na zdarzenia w czasie rzeczywistym.



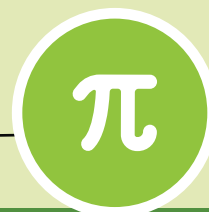
Zagadnienia matematyczne

Część matematyczna koncentruje się na analizie zmian temperatury i wilgotności przedstawionych na wykresie tworzonym na podstawie danych z czujnika DHT. Uczniowie uczą się odczytywać wartości początkowe, końcowe, maksymalne i minimalne, obliczać rozstęp oraz określać, o ile zmieniły się badane wielkości. Analizują także przebieg dwóch wykresów rysowanych jednocześnie, korzystając z dwóch różnych osi wartości, co pozwala im porównywać kierunek, szybkość i charakter zmian zachodzących w czasie. Dzięki temu uczniowie uczą się wykorzystywać matematykę do interpretacji danych pomiarowych, dostrzegać zależności widoczne na wykresie oraz wyciągać wnioski na podstawie rzeczywistych obserwacji.

Co będzie potrzebne do module 10?



W części matematycznej wykorzystywany będzie **czujnik DHT w wersji 3-pinowej, umieszczony na gotowym module**. Oznacza to, że na płytce czujnika znajdują się już dodatkowe elementy elektroniczne, takie jak **rezystor** i **kondensator**, dlatego nie trzeba dodawać ich osobno podczas podłączania układu.



Matematyka



Poznajemy świat arduino:

Czujnik ruchu PIR

Wygląd i budowa czujnika

Czujnik PIR posiada trzy podstawowe wyprowadzenia:

- **VCC** – zasilanie (5 V),
- **OUT** – sygnał wyjściowy,
- **GND** – masa.

Wyjście **OUT** jest podłączane do **pinu cyfrowego Arduino** (np. D2).

Czujnik działa w sposób bardzo prosty:

- gdy nie wykrywa ruchu → na wyjściu pojawia się **stan niski (LOW / 0)**
- gdy wykryje ruch → na wyjściu pojawia się **stan wysoki (HIGH / 1)**

Dzięki temu czujnik może być łatwo obsługiwany w programie jako **zwykły sygnał cyfrowy**.

Na płytce czujnika znajdują się dwa elementy regulacyjne:



Rysunek 69
Czujniki ruchu PIR

Potencjometr TIME

Potencjometr SENS (Sensitivity)

Pozwala regulować czułość czujnika, czyli zasięg wykrywania ruchu.

Określa **czas utrzymywania sygnału HIGH** po wykryciu ruchu. Przy minimalnym ustawieniu sygnał trwa zwykle około 2–3 sekund, a przy maksymalnym może utrzymywać się nawet przez kilka minut.

Na module znajduje się również zworka trybu pracy oznaczona jako H / L:

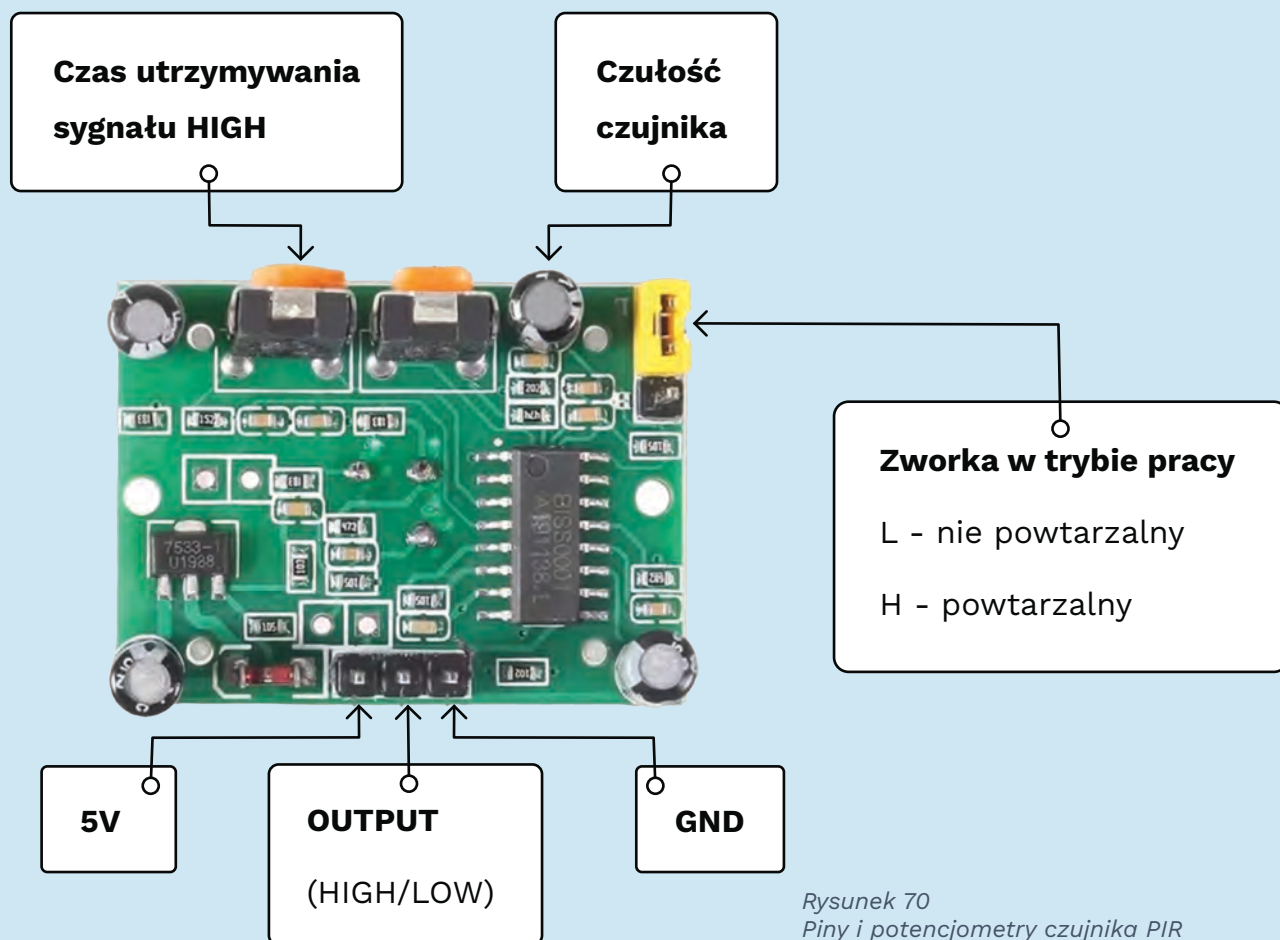
- **H – tryb retrigger (powtarzalny)**

Jeśli ruch nadal występuje, czujnik przedłuża sygnał HIGH.

- **L – tryb non-retrigger**

Czujnik generuje pojedynczy impuls o stałej długości, niezależnie od dalszego ruchu.

Potencjometry regulujące:



Przykłady zastosowań czujników PIR

Podczas pracy z Arduino czujnik PIR najczęściej podłącza się do pinu cyfrowego, ponieważ jego sygnał ma tylko dwa stany: 0 lub 1. Dzięki temu program może reagować na wykrycie ruchu, na przykład:

- zapalić diodę LED,
- uruchomić alarm,
- zwiększyć licznik wykrytych zdarzeń,
- włączyć inne urządzenie.

Po podłączeniu zasilania czujnik potrzebuje około **30–60 sekund na kalibrację**.

W tym czasie może generować niestabilne sygnały i nie powinien być jeszcze używany do pomiarów.

Czujniki PIR są powszechnie stosowane w wielu urządzeniach codziennego użytku, takich jak:

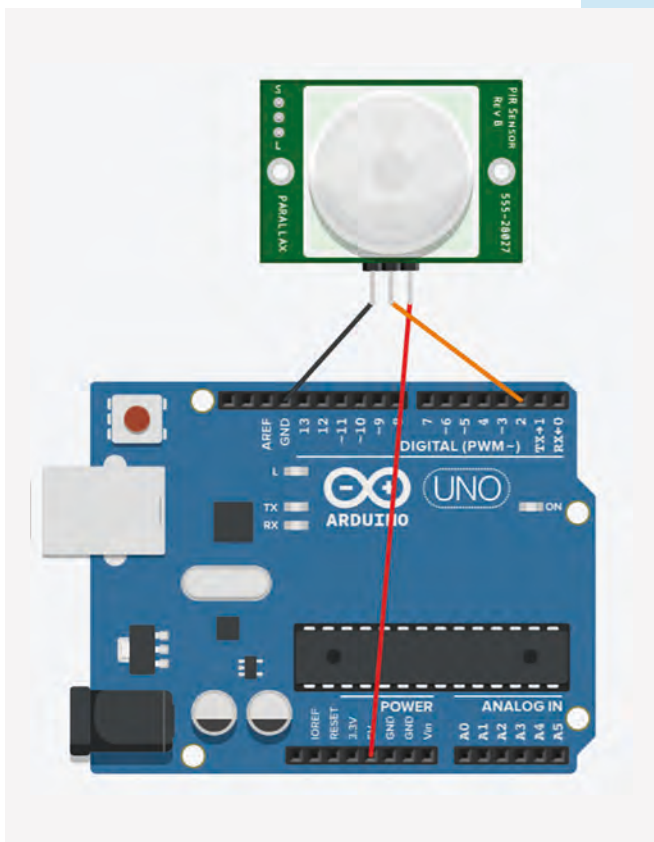
- automatyczne oświetlenie korytarzy,
- systemy alarmowe,
- czujniki obecności w budynkach,
- systemy oszczędzania energii w inteligentnych domach.



Zadanie 1 Odczyt stanu czujnika ruchu PIR

Cel:

Celem zadania jest poznanie sposobu działania czujnika ruchu PIR oraz nauczenie się odczytywania jego sygnału w programie PictoBlox.



Rysunek 71
Podłączenie czujnika ruchu pir

Jak działa program

Program w sposób ciągły sprawdza stan czujnika PIR podłączonego do pinu **D2**.

Jeśli czujnik wykryje ruch, na jego wyjściu pojawia się stan **1 (HIGH)**. Wtedy program wyświetla komunikat „**jest ruch**”.

Gdy czujnik nie wykrywa ruchu, jego wyjście ma stan 0 (LOW) i program wyświetla komunikat „brak ruchu”. Sprawdzanie czujnika odbywa się w pętli, dlatego komunikat jest aktualizowany co **1 sekundę**.

Krok 1 – Podłączenie czujnika

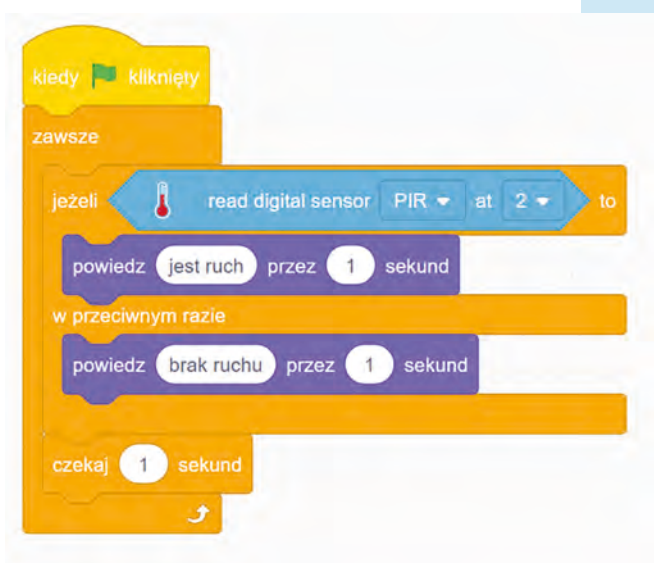
Podłącz czujnik PIR do Arduino:

- VCC → 5V
- GND → GND
- OUT → pin cyfrowy D2

Ustaw potencjometr **TIME** na możliwie najmniejszą wartość (kręć potencjometrem w lewo).

Krok 2 – Przygotowanie programu

W środowisku PictoBlox utwórz program przedstawiony na ilustracji. Program powinien:



Rysunek 72
Listing programu





1. Czy czujnik reaguje natychmiast na każdy ruch, czy pojawia się niewielkie opóźnienie?
2. Co dzieje się z komunikatem programu, gdy przez kilka sekund stoisz nieruchomo przed czujnikiem?

- sprawdzać stan czujnika PIR,
- wyświetlać komunikat „jest ruch” lub „brak ruchu”,
- wykonywać sprawdzenie co 1 sekundę.

Krok 3 – Uruchomienie programu

Kliknij zieloną flagę, aby uruchomić program.

Po uruchomieniu:

- przejdź przed czujnikiem, lub przyłóż rękę.
- następnie zatrzymaj się lub odejdź od czujnika.

Obserwuj komunikaty wyświetlane przez program.

Kryterium osiągnięcia celu

Program działa poprawnie, jeśli:

- ☒ po wykryciu ruchu pojawia się komunikat „jest ruch”,
- ☒ gdy w polu widzenia czujnika nic się nie porusza pojawia się komunikat „brak ruchu”.



Zadanie 2

Zliczanie wykrytych ruchów

Cel:

Celem zadania jest nauczenie się, jak program może reagować na sygnał z czujnika ruchu PIR, zliczać wykryte zdarzenia oraz prezentować wynik pomiaru po określonym czasie działania programu.

Jak działa program

Program odczytuje stan czujnika ruchu PIR podłączonego do pinu D2.

Gdy czujnik wykryje ruch:

- zapala się czerwona dioda LED,
- program zwiększa licznik wykrytych ruchów.

Gdy ruch nie jest wykrywany:

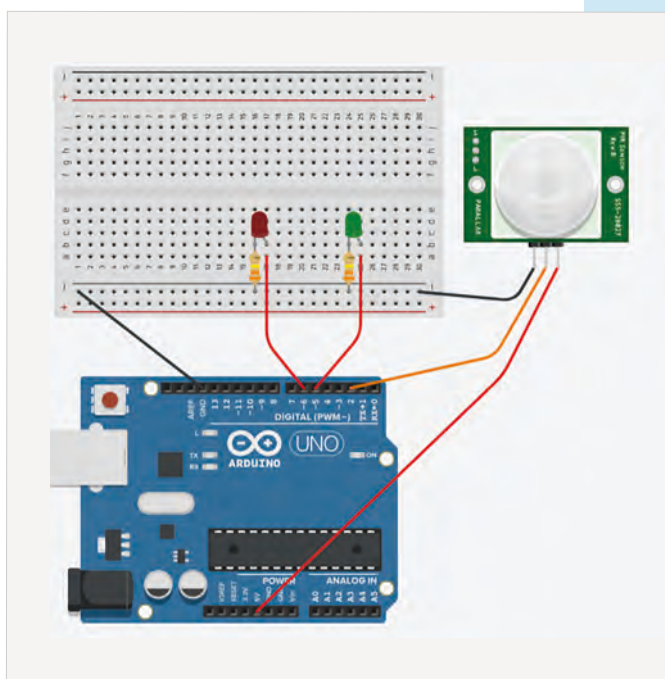
- zapala się zielona dioda LED.

Program zlicza wykryte ruchy przez określony czas. Po upływie tego czasu wyświetla na ekranie liczbę wykrytych zdarzeń.

Krok 1 – Podłączenie elementów

Podłącz elementy zgodnie ze schematem przedstawionym na ilustracji.





Rysunek 73
Schemat podłączenia do zadania2



Rysunek 74
Okno programu po upływie czasu.

Połączenia:

Czujnik PIR:

- **VCC → 5V**
- **GND → GND**
- **OUT → D2**

Diody LED:

zielona dioda:

- **anoda → D5**
- **katoda → rezystor → GND**

czerwona dioda:

- anoda → D6
- katoda → rezystor → GND

Krok 2 – Przygotowanie programu

W środowisku PictoBlox utwórz program, który:

- odczytuje stan czujnika PIR,
- zapala czerwoną diodę, gdy wykryty zostanie ruch,
- zapala zieloną diodę, gdy ruch nie jest wykrywany,
- zlicza liczbę wykrytych ruchów.

Krok 3 – Pomiar liczby ruchów

Uruchom program.

Przez 20 sekund wykonuj różne ruchy przed czujnikiem:

- przejdź przed czujnikiem,
- pomachaj ręką,

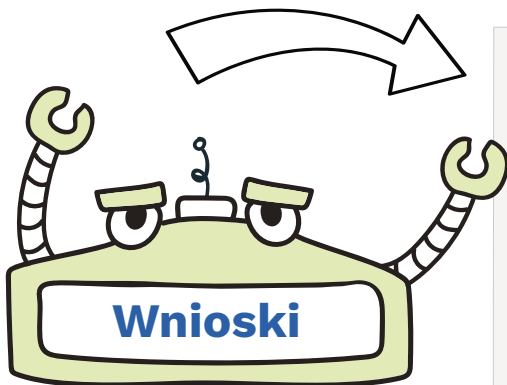
- odejść od czujnika.

Po upływie czasu program powinien wyświetlić liczbę wykrytych ruchów.

Kryterium osiągnięcia celu

Zadanie zostało wykonane poprawnie, jeśli:

- ☒ czerwona dioda zapala się w momencie wykrycia ruchu,
- ☒ zielona dioda świeci, gdy ruch nie jest wykrywany,
- ☒ program zlicza wykryte ruchy i wyświetla ich liczbę po zakończeniu pomiaru.



1. Czy czujnik zawsze zlicza dokładnie tyle ruchów, ile razy poruszyłeś się przed nim?
2. Jak zmienia się liczba wykrytych ruchów, gdy poruszasz się bardzo szybko lub bardzo powoli?

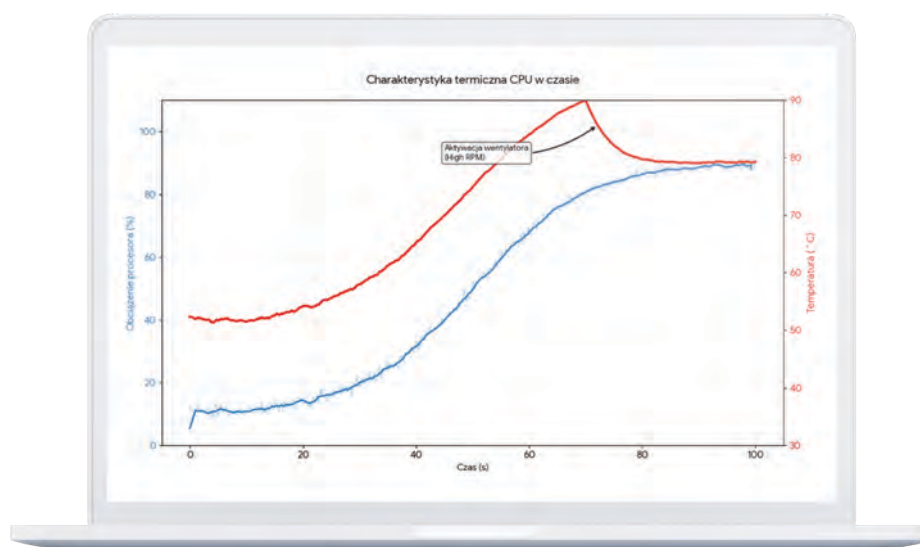


Odkrywamy matematykę: wykres zmian w czasie i dwie skale

W tym module dane pomiarowe są przedstawiane na wykresie, który pokazuje **zmiany wartości w czasie**. Oś pozioma (X) odpowiada kolejnym chwilom pomiaru, a oś pionowa (Y) przedstawia wartość danej wielkości. Dzięki temu można analizować nie tylko same wartości, ale też **trend** (czy rośnie/maleje) oraz **tempo zmian** (czy zmienia się szybko czy powoli).

Na siatce wykresu każda kratka odpowiada stałemu odcinkowi czasu. Aby poprawnie odczytywać czas z wykresu, trzeba znać skalę osi X, czyli ile sekund przypada na jedną kratkę.

Pozwala to np. oszacować, po ilu sekundach dana wartość osiągnęła określony poziom.



Rysunek XX
Opis

Na tym wykresie zastosowano **dwie osie Y**, ponieważ porównywane wielkości mają różne jednostki i zakresy: **obciążenie CPU (%)** oraz **temperatura (°C)**, a jednocześnie obserwacja zmian w czasie na jednym wykresie ułatwia analizę danych. Widać wyraźnie, że gdy obciążenie rośnie, temperatura również rośnie, a po przekroczeniu progu i uruchomieniu intensywniejszego chłodzenia temperatura zaczyna spadać mimo nadal wysokiego obciążenia. Taki zapis pozwala szybko zauważyć zależność oraz moment zadziałania mechanizmu regulacji temperatury.

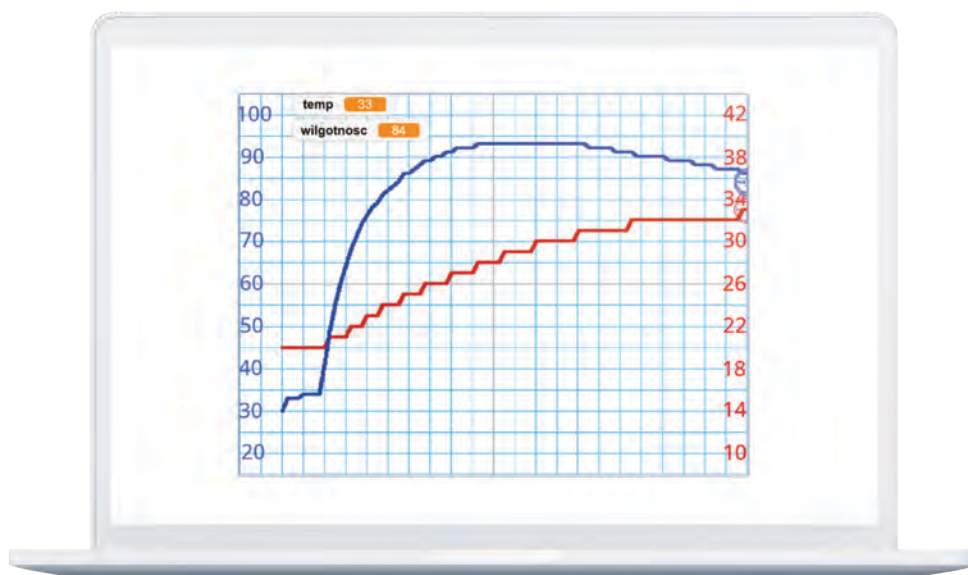
Na jednym wykresie mogą być przedstawione **dwie różne wielkości**, nawet jeśli mają inne jednostki i inne zakresy wartości. W takiej sytuacji stosuje się wykres z **dwoma skalami pionowymi (dwie osie Y)**:

- jedna oś Y opisuje pierwszą wielkość,
- druga oś Y opisuje drugą wielkość.

To rozwiązanie ułatwia porównywanie **momentu zmian** (kiedy coś zaczyna rosnąć lub się stabilizuje) oraz **kierunku zmian** (czy wartości rosną, maleją, czy jedna rośnie, a druga spada). Wymaga jednak uważnego odczytu, ponieważ każda wielkość ma swoją własną skalę.

W tym module dodatkowym ułatwieniem jest kolorystyczne oznaczenie osi:

- **kolor osi wilgotności** jest taki sam jak **kolor wykresu wilgotności - niebieski**,
- **kolor osi temperatury** jest taki sam jak **kolor wykresu temperatury - czerwony**.



Rysunek 75

Wykres wilgotności i temperatury gdy czujnik trzymany jest w rękach.

Podczas analizy wykresu należy pamiętać o dwóch zasadach:

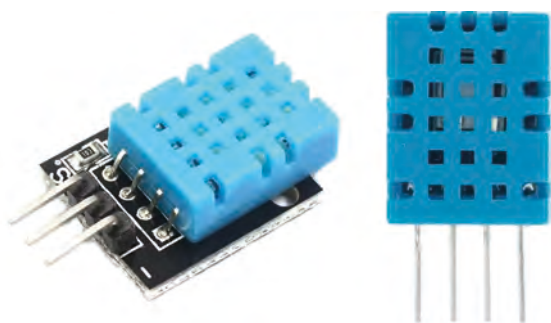
1. **Wartości zawsze odczytujemy z właściwej osi** (temperaturę z osi temperatury, wilgotność z osi wilgotności).
2. Przy dwóch różnych skalach nie porównujemy „kto jest wyżej”, tylko porównujemy **kierunek, moment i tempo zmian**.





Zadanie 1

Rejestracja zmian temperatury i wilgotności



Rysunek 76
Dwie wersje czujnika DHT11, pierwsza wygodniejsza w użyciu z uwagi na wbudowany rezystor, kondensator i mniejszą liczbę wymaganych połączeń.

Cel:

Celem zadania jest wykonanie pomiaru temperatury i wilgotności oraz obserwacja zmian tych wartości w czasie.

Polecenie

1. Podłącz czujnik **DHT** do płytki Arduino. Znajdź na module piny opisane jako: **VCC, GND, DATA** (czasem **OUT** lub **S**). Podłącz przewody:
 - **VCC → 5V** na Arduino
 - **GND → GND** na Arduino
 - **DATA/OUT/S → D3** na Arduino
2. Uruchom przygotowany program w środowisku **PictoBlox**.
3. Po uruchomieniu programu przez **10 sekund pozostaw czujnik nieruchomo**, aby ustabilizowały się pomiary.
4. Następnie **chwyć czujnik w palce i trzymaj go przez około 60 sekund**.
5. Obserwuj wykres pojawiający się na ekranie.

Na wykresie przedstawione są:

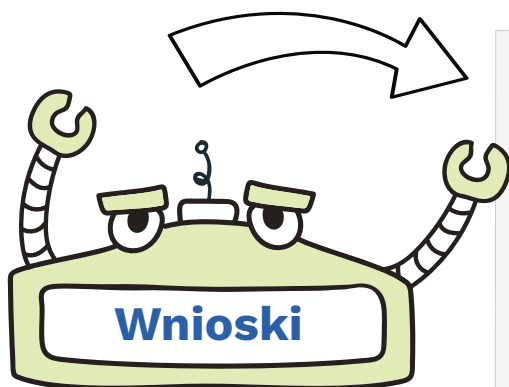
- **linia niebieska – wilgotność powietrza**
- **linia czerwona – temperatura**

Oś pozioma przedstawia **czas pomiaru**, a oś pionowa wartości temperatury i wilgotności.

Jedna kratka na osi czasu odpowiada **4 sekundom pomiaru**.

Kryterium osiągnięcia celu

- ☒ Program poprawnie rysuje wykres temperatury i wilgotności w czasie.
- ☒ Na wykresie powinien być widoczny **wzrost temperatury po ogrzaniu czujnika dłonią**.



1. Jak zmieniła się temperatura podczas trzymania czujnika w dłoni?
2. Jak zmieniła się wilgotność?
3. Czy zmiany następowały równomiernie przez cały czas pomiaru?



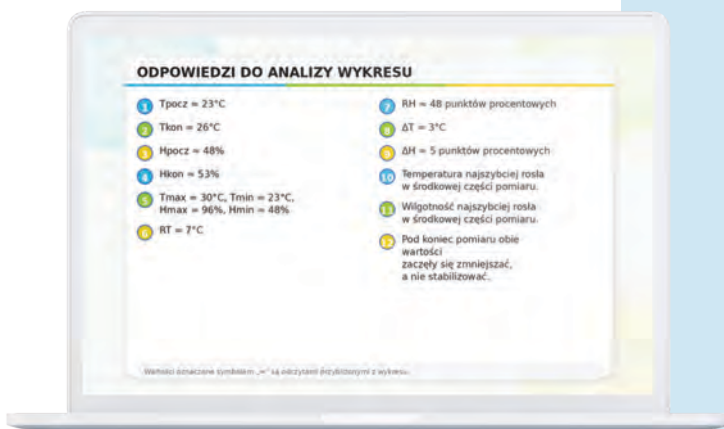


Zadanie 2

Analiza wykresu pomiarowego



Rysunek 77
Przykładowy wykres, przedstawiający dane z czujnika, na który dmuchano w bliskiej odległości.



Rysunek 78
Odczyt i analiza danych z przykładowego wykresu.

Cel:

Celem zadania jest analiza danych odczytanych z wykresu przedstawiającego zmiany temperatury i wilgotności w czasie.

Polecenie

Uruchom program tak jak w poprzednim zadaniu. Po kilku sekundach w wybrany sposób wpłynij na warunki wokół czujnika, aby spowodować zmianę odczytywanej temperatury i wilgotności. Możesz na przykład:

- ogrzewać czujnik w dłoniach (to scenariusz z poprzedniego ćwiczenia),
- zbliżyć go do kaloryfera,
- dmuchać w jego stronę,
- przybliżyć go do otwartego okna,
- ustawić go w pobliżu wiatraka komputerowego.

Obserwuj, jak zmienia się wykres, a następnie wykonaj jego analizę.

Przyjrzyj się jak wygląda przykładowy wykres i odpowiedzi oparte na danych z wykresu.

Na podstawie własnego wykresu otrzymanego podczas pomiaru odczytaj i zapisz

1. Temperaturę na początku pomiaru: T_{pocz} .
2. Temperaturę na końcu pomiaru: T_{kon} .
3. Wilgotność na początku pomiaru: H_{pocz} .
4. Wilgotność na końcu pomiaru: H_{kon} .
5. Najwyższą i najniższą temperaturę oraz wilgotność: T_{max} , T_{min} , H_{max} , H_{min} .

Następnie wykonaj obliczenia:

6. Rozstęp temperatury:
 $RT = T_{max} - T_{min}$.
7. Rozstęp wilgotności:
 $RH = H_{max} - H_{min}$.
8. Oblicz, o ile stopni zmieniła się temperatura:
 $\Delta T = T_{kon} - T_{pocz}$.
9. Oblicz, o ile punktów procentowych zmieniła się wilgotność:
 $\Delta H = H_{kon} - H_{pocz}$.

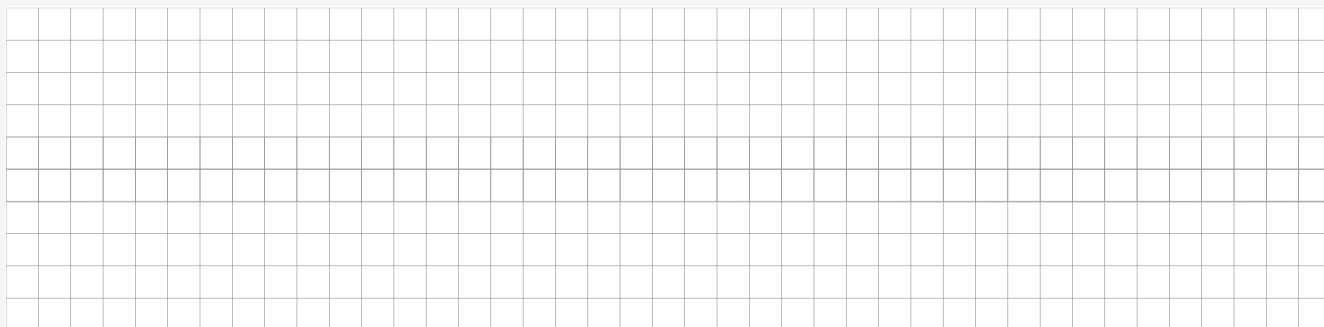
Na koniec opisz wykres:

10. W którym fragmencie pomiaru temperatura zmieniła się najszybciej?
11. W którym fragmencie pomiaru wilgotność zmieniła się najszybciej?



12. Czy pod koniec pomiaru wartości nadal się zmieniały, czy zaczęły się stabilizować?

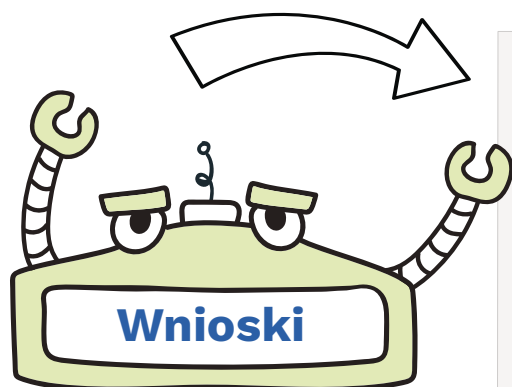
Pamiętaj: jedna kratka na osi czasu to **4 sekundy**.



Kryterium osiągnięcia celu

Uczeń poprawnie:

- ☒ odczytuje dane z wykresu
- ☒ dokonuje prawidłowych obliczeń w oparciu o odczytane dane z wykresu.



1. Dlaczego temperatura rośnie szybko na początku pomiaru, a później coraz wolniej?
2. Czy wilgotność zmienia się w taki sam sposób jak temperatura?

Czego się nauczyłeś?



Matematyka

- Nauczyłeś się odczytywać dane z wykresu przedstawiającego zmiany temperatury i wilgotności w czasie.
- Zrozumiałeś, jaką rolę na wykresie pełni oś czasu oraz dwie różne osie wartości.
- Nauczyłeś się, że na jednym wykresie można przedstawić dwie różne wielkości, jeśli mają one osobne skale.
- Potrafisz odczytać wartości początkowe, końcowe, maksymalne i minimalne.
- Nauczyłeś się obliczać rozstęp temperatury i wilgotności.
- Potrafisz obliczyć, o ile zmieniła się temperatura oraz wilgotność podczas pomiaru.
- Nauczyłeś się porównywać przebieg dwóch wykresów rysowanych jednocześnie.
- Potrafisz wskazać fragmenty, w których wartości zmieniają się szybciej, wolniej albo zaczynają się stabilizować.
- Zobaczyłeś, że wykres pozwala nie tylko odczytywać liczby, ale także analizować przebieg zjawiska i wyciągać wnioski.



Informatyka

- Dowiedziałeś się, jak działa czujnik ruchu PIR i do czego może być wykorzystywany.
- Nauczyłeś się, że czujnik PIR wykrywa zmiany promieniowania podczerwonego, a nie „widzi” ruchu jak kamera.
- Poznałeś podstawowe wyprowadzenia czujnika: VCC, GND i OUT.
- Nauczyłeś się podłączać czujnik PIR do Arduino i odczytywać jego stan w programie.
- Zobaczyłeś, że czujnik może zwracać dwa stany: wykrycie ruchu lub brak ruchu.
- Nauczyłeś się tworzyć program, który reaguje na wykrycie ruchu odpowiednim komunikatem.
- Potrafisz sterować diodami LED w zależności od sygnału z czujnika.
- Zobaczyłeś, że program może zliczać wykryte zdarzenia w określonym czasie.
- Dowiedziałeś się, że działanie czujnika zależy również od ustawień modułu, takich jak czas podtrzymania sygnału i tryb pracy.

