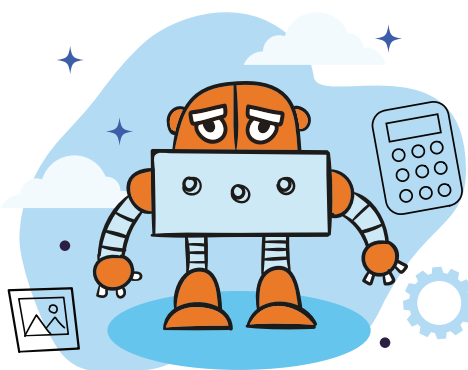




Reprezentacja obrazu i danych – piksele, siatka i pole powierzchni



Zagadnienia informatyczne

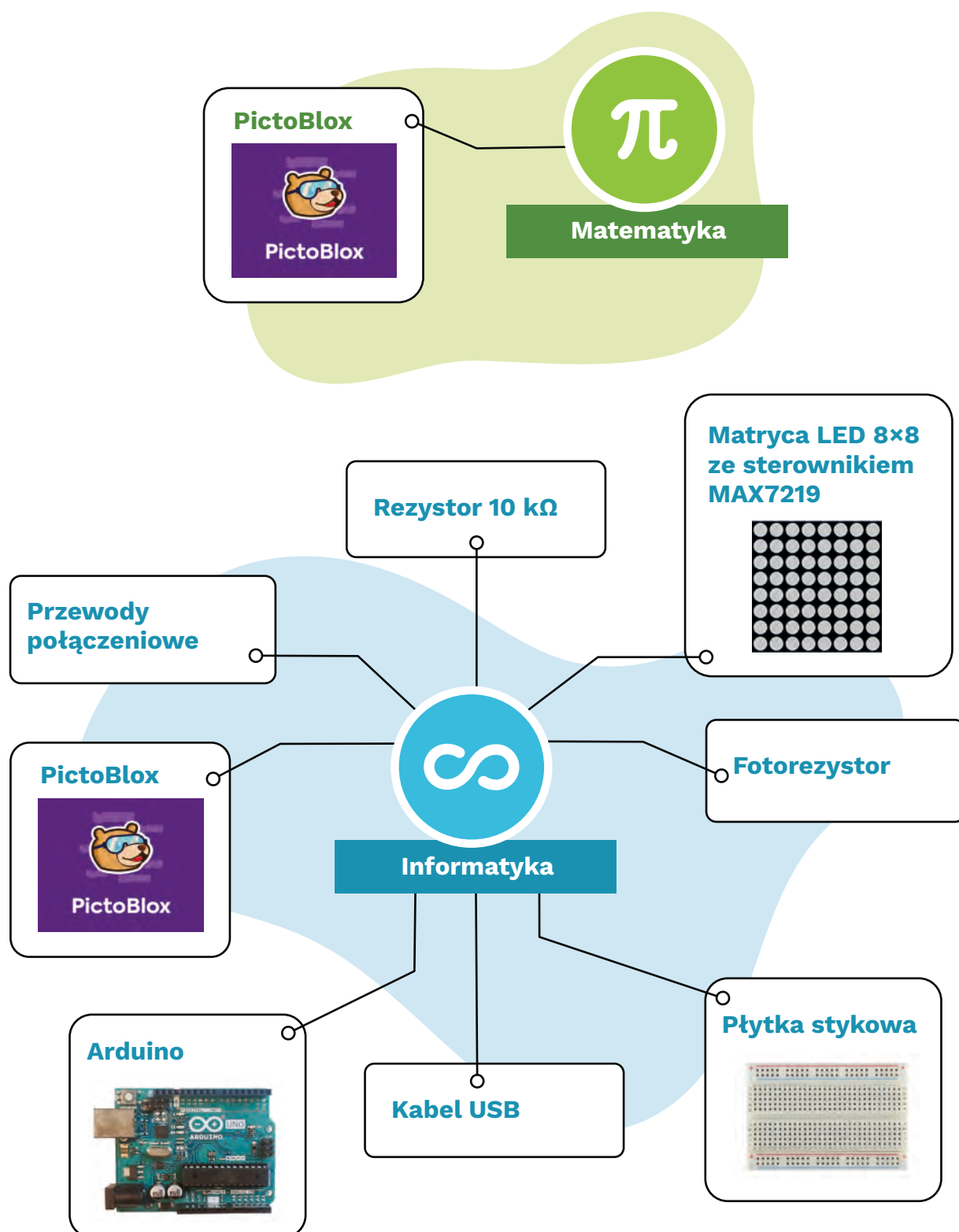
W tej części uczniowie poznają, jak komputer zapisuje i wyświetla obraz w postaci danych – na przykładzie pikseli, bitmapy i rozdzielczości. Pracując z matrycą LED 8×8 sterowaną układem MAX7219, uczą się traktować wyświetlany symbol jako siatkę 64 punktów oraz sterować pojedynczymi „pikselami” za pomocą współrzędnych wiersz–kolumna. Dodatkowo wykorzystują fotorezystor jako czujnik analogowy i tworzą prostą wizualizację natężenia światła na matrycy, dzięki czemu widzą bezpośrednie przejście od pomiaru (0–1023) do reprezentacji graficznej.



Zagadnienia matematyczne

Część matematyczna koncentruje się na polach figur przedstawionych na siatce 8×8. Uczniowie obliczają pole jako liczbę jednostkowych pól (zapalonych diod), a następnie weryfikują swoje wyniki w programie – poprzez Monitor portu szeregowego lub interfejs PictoBlox. Dzięki temu ćwiczą liczenie pól figur złożonych, sumowanie części oraz interpretację pola jako „zajętości” przestrzeni, co łączy geometrię z pojęciami znanymi z informatyki (obszar danych, rozdzielczość, zajętość pamięci).

Co będzie potrzebne do module 4?



Poznajemy świat Arduino: Piksel i bitmapa – jak komputer „widzi” obraz.

Piksel, bitmapa i rozdzielczość

Gdy patrzymy na obraz wyświetlany na ekranie komputera, telefonu czy telewizora, wydaje się on gładki i ciągły, podobnie jak obraz widziany ludzkim okiem.

W rzeczywistości komputer nie widzi obrazu jako jednej całości. Dla komputera obraz to zbiór bardzo małych elementów, z których każdy opisany jest za pomocą prostych informacji – najczęściej liczb.

Najmniejszym elementem obrazu cyfrowego jest **piksel**. Piksel to pojedynczy punkt obrazu, który może być włączony lub wyłączony, jasny lub ciemny, a w bardziej zaawansowanych przypadkach – mieć określony kolor. Pojedynczy piksel sam w sobie nie niesie informacji o kształcie czy znaczeniu obrazu. Dopiero duża liczba pikseli ułożonych obok siebie tworzy rozpoznawalny kształt, symbol lub literę.

Obraz cyfrowy składa się z wielu pikseli ułożonych w regularną siatkę. Taki sposób zapisu obrazu nazywamy **bitmapą**. Bitmapa to nic innego jak dwuwymiarowa tablica danych, w której każdy element odpowiada jednemu pikselowi. Można ją porównać do kartki w kratkę, gdzie każda kratka ma przypisaną informację, czy jest wypełniona, czy pusta.

Liczba pikseli w poziomie i w pionie określa **rozdzielczość obrazu**. Rozdzielczość mówi nam, z ilu elementów zbudowany jest obraz, ale nie mówi jeszcze nic o jego kolorach czy szczegółowości. Im większa rozdzielczość, tym więcej pikseli tworzy obraz, a tym samym więcej danych musi zostać zapisanych i przetworzonych przez komputer.

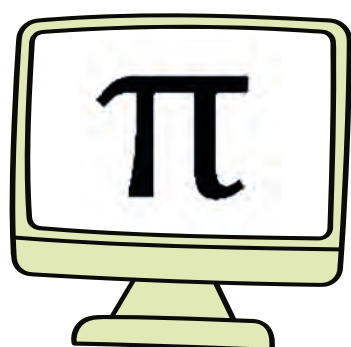
Dokładnie ten sam sposób myślenia możemy zastosować do fizycznych urządzeń elektronicznych. W matrycy LED 8×8 każdy świecący punkt jest odpowiednikiem jednego piksela. Cała matryca tworzy obraz o rozdzielczości 8 × 8, czyli złożony z 64 pikseli.



Każda dioda może być włączona lub wyłączona, a ich odpowiednie ułożenie pozwala wyświetlać proste symbole, litery i znaki.

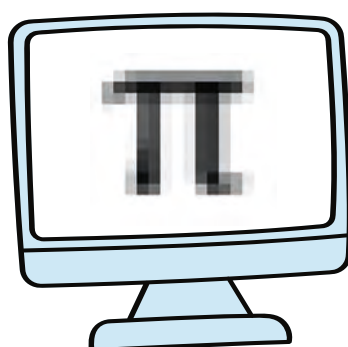
W tym ujęciu obraz nie jest rysunkiem, lecz **zestawem danych**, które sterują konkretnymi elementami sprzętowymi. Program nie „rysuje” obrazu w tradycyjnym sensie, lecz decyduje, które piksele – czyli diody LED – mają się zapalić, a które pozostać zgaszone. To dokładnie ten sam mechanizm, który działa w ekranach komputerów i telefonów, tylko w znacznie prostszej i łatwiejszej do zrozumienia formie.

W kolejnej części modułu zobaczymy, jak za pomocą Pictoblox i Arduino można sterować pojedynczymi punktami matrycy LED, tworząc obrazy zapisane jako proste tablice danych. Dzięki temu pojęcia takie jak piksel, bitmapa i rozdzielczość przestają być abstrakcyjne, a stają się czymś, co można zobaczyć, zaprogramować i zmienić własnymi rękami.



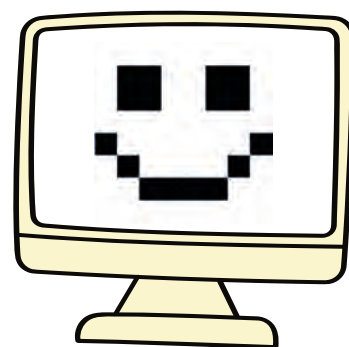
**Rozdzielczość
128 na 128
pikseli**

Rysunek 25
Symbol pi rozdzielczość
128 na 128



**Rozdzielczość
16 na 16
pikseli**

Rysunek 24
Symbol pi rozdzielczość 16 na 16



**Rozdzielczość
8 na 8
pikseli**

Rysunek 23
Ikona uśmiechniętej buźki
rozdzielczość 8 na 8 pikseli.

Matryca LED 8×8 – czym jest i jak działa

Matryca LED 8×8 to niewielki wyświetlacz złożony z **64 diod świecących**, ułożonych w **8 wierszy i 8 kolumn**. Każda dioda odpowiada jednemu „pikselowi” obrazu – dokładnie tak, jak piksel w grafice komputerowej. W przeciwieństwie do ekranu telefonu czy monitora, taka matryca jest zazwyczaj **monochromatyczna**, co oznacza, że świeci jednym kolorem i każda dioda może być tylko włączona lub wyłączona.



Moduł sterujący matrycą – co znajduje się na płytce

W zestawach Arduino bardzo często spotyka się matryce LED 8×8 zamontowane na **niewielkiej płytce drukowanej** (zwykle o wymiarach ok. 1 × 3 cm). Taka płytka nie zawiera jedynie samych diod, ale również **układ sterujący**, który przejmuje większość pracy związanej z ich obsługą. Tutaj przedstawiony zostanie specjalny sterownik LED MAX7219.

Dzięki temu Arduino **nie musi sterować każdą diodą osobno**, a jedynie wysła dane do sterownika, który zajmuje się:

- zapamiętaniem obrazu,
- odświeżaniem diod,
- sterowaniem wierszami i kolumnami.

Moduł sterujący matrycą – co znajduje się na płytce

Na płytce matrycy znajdują się zazwyczaj **5 pinów**:

- **VCC** – zasilanie (najczęściej 5 V),
- **GND** – masa,
- **DIN** (Data In) – linia danych, którą przesyłany jest obraz, **(D3)**
- **CS** (Chip Select) – sygnał wyboru układu, **(D5)**
- **CLK** (Clock) – sygnał zegarowy, który synchronizuje przesyłanie danych. **(D6)**



Zadanie 1

Uruchomienie matrycy led 8x8

Cel:

sprawdzić poprawne podłączenie i inicjalizację matrycy.

Polecenie:

1. Podłącz matrycę LED 8×8 do Arduino.
2. W Pictoblox zbuduj program, który po uruchomieniu Arduino:

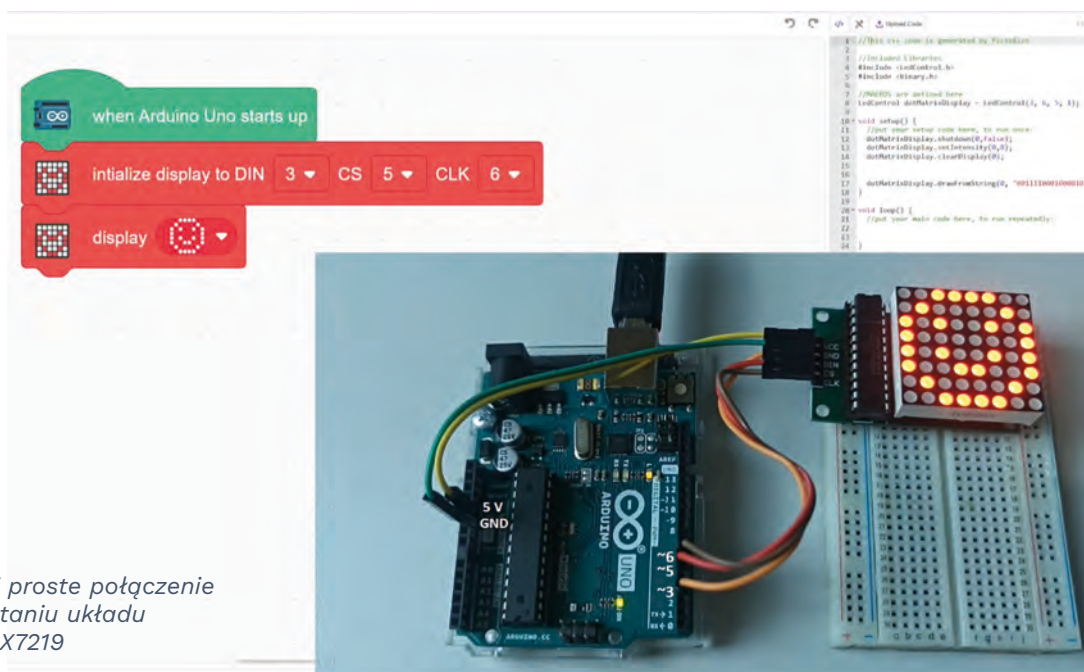
- inicjalizuje matrycę
(DIN = 3, CS = 5, CLK = 6),
- czyści ekran,
- wyświetla wybrany symbol
(np. buźkę).

Kryterium osiągnięcia celu:

Po starcie Arduino na matrycy pojawia się symbol i świeci stabilnie
(bez migania i losowych punktów).

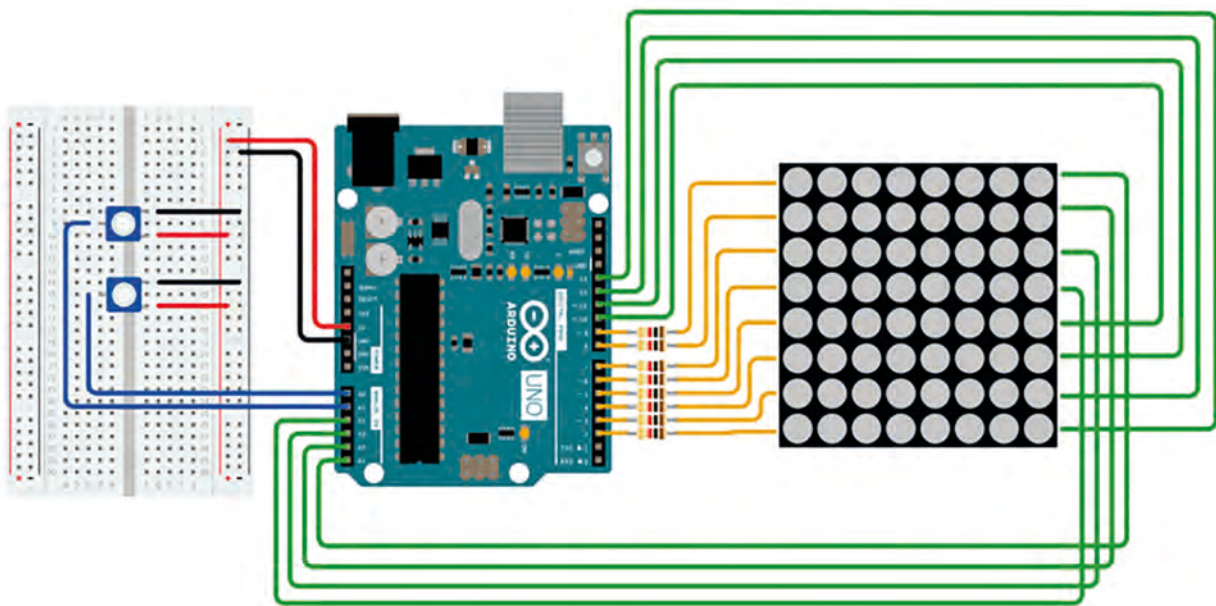


1. Czy matryca LED musi być podłączona dokładnie do tych samych pinów Arduino, czy można użyć innych pinów? Co należałoby wtedy zmienić?
2. Jaką zaletę daje zastosowanie sterownika MAX7219?



Rysunek 26
Kod programu i proste połączenie
dzięki wykorzystaniu układu
sterującego MAX7219





Rysunek 27

Skomplikowane podłączenie LED8x8 jeśli nie używamy układu sterującego MAX7219,

źródło: dokumentacja Arduino: <https://docs.arduino.cc/built-in-examples/display/RowColumnScanning/>



Zadanie 2

„Piksel” na matrycy
– współrzędne
wiersz/kolumna

Cel:

zrozumieć, że matryca to siatka 8×8,
a każdy punkt ma współrzędne.

Polecenie:

1. Po uruchomieniu programu wyczyść matrycę.
2. Zapal **jeden** punkt (LED) w podanym miejscu, np.: wiersz 1, kolumna 1.

3. Zmień współrzędne tak, aby punkt „przeskakiwał” kolejno w 4 narożniki:

- $(1,1) \rightarrow (1,8) \rightarrow (8,8) \rightarrow (8,1)$

Po każdej zmianie dodaj krótką pauzę i czyść ekran przed zapaleniem kolejnego punktu.

- Program ma zapalać diodę w jednym narożniku, gasić ją i zapalać drugi narożnik, gasić i tak dalej w nieskończonej pętli.

Kryterium osiągnięcia celu:

Widać jeden świecący punkt, który kolejno pojawia się w narożnikach.



1. Co stanie się na matrycy, jeśli w programie nie wyłączysz poprzedniego punktu przed włączeniem kolejnego?
2. Która wartość w programie odpowiada za ruch punktu w poziomie, a która za ruch w pionie?

Fotorezystor – czujnik światła

Fotorezystor (LDR – Light Dependent Resistor) to element elektroniczny, którego opór zmienia się w zależności od ilości światła.





Fotorezystor **nie jest gotowym czujnikiem** – sam z siebie nie wysyła żadnych danych. Aby Arduino mogło „zobaczyć” zmiany światła, musimy zamienić zmianę oporu na zmianę napięcia. Do tego służy **dzielnik napięcia**.

Arduino nie mierzy światła bezpośrednio – mierzy **napięcie na wejściu analogowym**, które następnie zamienia na liczbę z zakresu **0–1023**:

- mała wartość → mało światła,
- duża wartość → dużo światła.

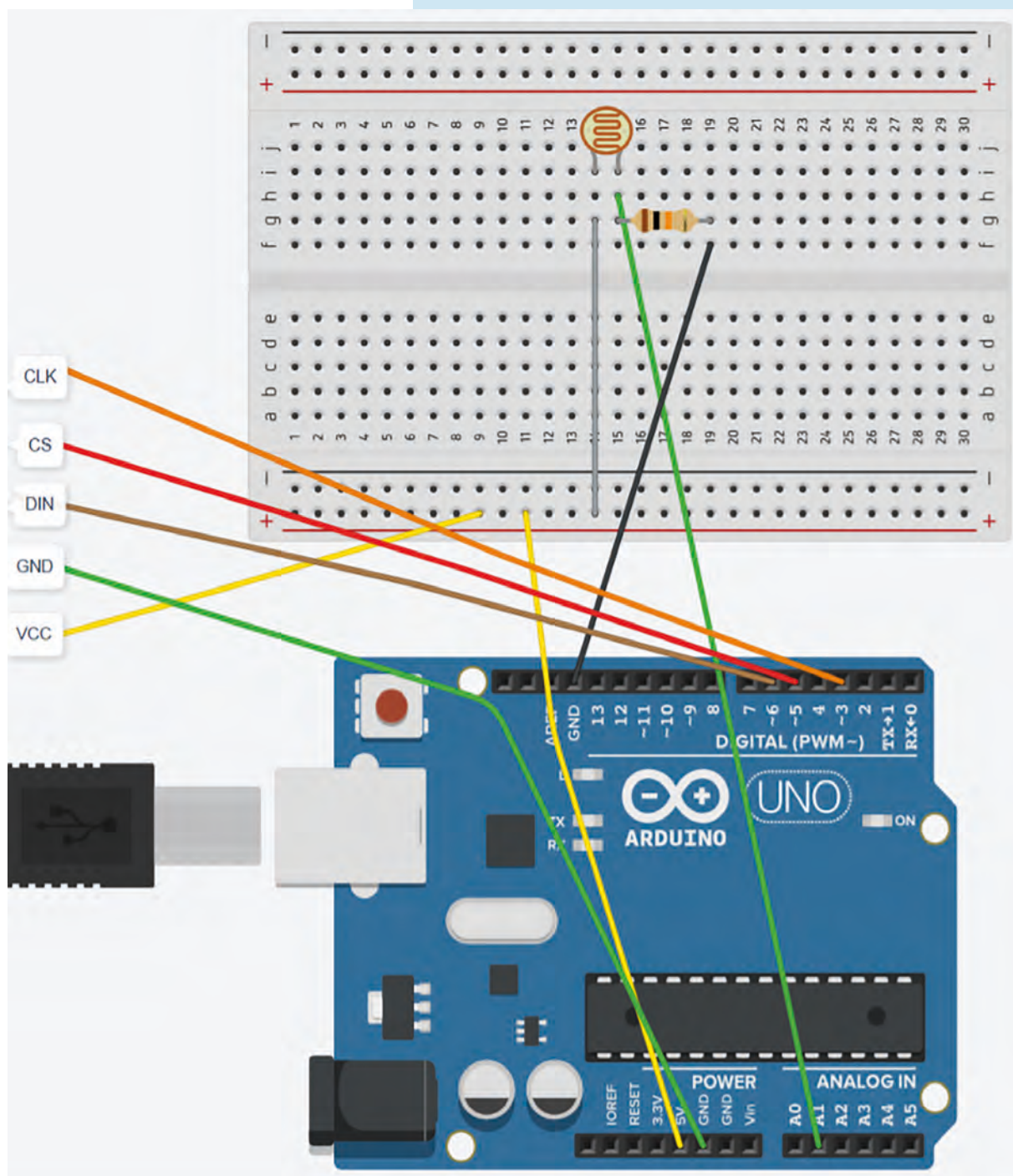
Dzielnik napięcia – po co jest potrzebny?

Dzielnik napięcia składa się z **dwóch elementów połączonych szeregowo**:

- fotorezystora,
- rezystora (np. 10 kΩ).

Połączenie tych dwóch elementów powoduje, że w zależności od oświetlenia zmienia się napięcie w punkcie pomiędzy nimi.

To właśnie ten punkt podłączamy do pinu analogowego Arduino. Dzięki temu Arduino może „odczytać” światło jako liczbę.



Rysunek 28
Podłączenie Arduino, LED8x8 oraz dzielnika napięcia z fotorezystorem



Zadanie 3

Wizualizacja natężenia światła na matrycy LED 8×8

Cel:

- odczyt natężenia światła za pomocą fotorezystora,
- podział odczytu na **cztery zakresy wartości**,
- wizualizacja poziomu światła na **matrycy LED 8×8** w formie prostego „wskaźnika”.

Krok 1. Podłączenie elementów

1. Podłącz **fotorezystor** do Arduino zgodnie z wcześniej omówionym schematem dzielnika napięcia:
 - 5V → fotorezystor → A1
→ rezystor 10 kΩ → GND.
2. Podłącz **matrycę LED 8×8** do Arduino (VCC, GND, DIN, CS, CLK) zgodnie z instrukcją modułu.

Krok 2. Założenia programu

Program powinien:

- odczytywać wartość z wejścia analogowego **A0**,
- aktualizować odczyt **co 1 sekundę**,
- rozróżniać **cztery poziomy natężenia światła** na podstawie wartości z zakresu 0–1023.





Rysunek 29
Przykład wyświetlania matrycy LED
dla trzech pierwszych poziomów
światła

Przyjmij następujące progi:

- **poziom 1:** wartość mniejsza niż 150
- **poziom 2:** wartość mniejsza niż 350
- **poziom 3:** wartość mniejsza niż 600
- **poziom 4:** wartość równa lub większa niż 600

Krok 3. Wizualizacja na matrycy LED

Na podstawie wykrytego poziomu światła zapal diody na matrycy LED w następujący sposób:

- **Poziom 1 (<150)**
Zapal **kolumny 1 i 2** do wysokości **2 diod**.
- **Poziom 2 (<350)**
Zapal **kolumny 3 i 4** do wysokości **4 diod**.
- **Poziom 3 (<800)**
Zapal **kolumny 5 i 6** do wysokości **6 diod**.



- **Poziom 4 (≥ 800)**

Zapal **kolumny 7 i 8** do pełnej wysokości **8 diod**.

Przy każdej zmianie poziomu:

- wyczyść wcześniej zapalone diody,
- wyświetl nowy układ odpowiadający aktualnemu natężeniu światła.

Kryterium osiągnięcia celu:

Matryca LED działa jak prosty, wizualny wskaźnik światła:

- im jaśniej w otoczeniu, tym więcej kolumn i diod jest zapalonych,
- zmiany są aktualizowane co 1 sekundę,
- uczniowie widzą bezpośrednie powiązanie między natężeniem światła a reprezentacją graficzną.



1. Jaką zaletę daje wykorzystanie matrycy LED jako „wskaźnika”, zamiast wyświetlania samej liczby?
2. W jakich sytuacjach wskazania na matrycy LED mogą szybko się zmieniać i z czego to wynika?

Odkrywamy matematykę: Pola figur

Pole powierzchni jako sposób opisu zajętości przestrzeni

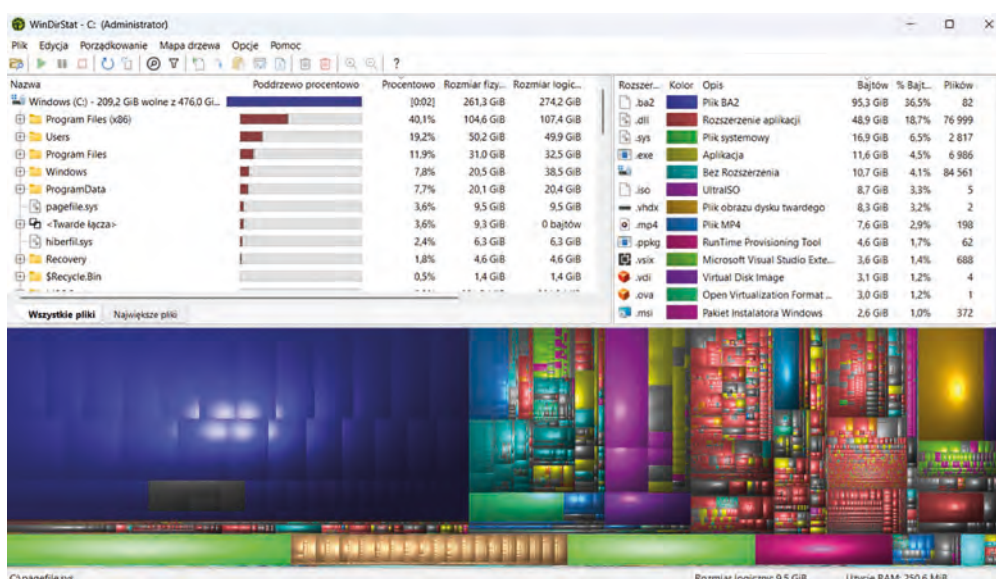
Pole powierzchni to pojęcie matematyczne, które opisuje, ile miejsca zajmuje dany obszar.

Najczęściej kojarzymy je z figurami geometrycznymi – prostokątami, kwadratami czy bardziej złożonymi kształtami. W praktyce pole oznacza liczbę jednostkowych pól, które wypełniają dany obszar. Podobną ideę obszaru, zajętości dysku, wielkości pliku znajdujemy w informatyce.

Dysk twardy, pendrive czy karta pamięci to również uporządkowana przestrzeń, podzielona na mniejsze komórki, w których zapisywane są dane. Każda z tych komórek ma swoje miejsce i adres, a pliki zajmują określoną liczbę takich komórek. Im więcej danych zawiera plik, tym większy obszar dysku musi zostać zajęty.

Graficzne narzędzia do analizy zajętości dysku (np. WinDirStat) pokazują tę zależność w bardzo czytelny sposób – jako mapę prostokątnych obszarów, gdzie:

- każdy prostokąt reprezentuje plik lub folder,
- powierzchnia prostokąta odpowiada ilości zajętego miejsca na dysku,
- większe pliki zajmują większą „powierzchnię” przestrzeni dyskowej.



Rysunek 30

Widok programu Windirstat, który graficznie prezentuje zajętość przestrzeni dyskowej



W takim ujęciu:

- pole figury w matematyce
- i zajętość pliku w informatyce

opisują to samo zjawisko: ile jednostek przestrzeni zostało wypełnionych.

Rozdzielczość obrazu cyfrowego

Rozdzielczość obrazu cyfrowego również można rozpatrywać w kategoriach pola.

Obraz składa się z pikseli ułożonych w siatkę – im większa rozdzielczość, tym więcej elementarnych pól (pikseli) tworzy obraz, a co za tym idzie – więcej miejsca zajmuje on na dysku.

To jak szybko wypełnia się przestrzeń dysku 30GB zależnie od rozdzielczości (czyli niejako powierzchni) pliku widać wyraźnie na wykresie. „Jak szybko znika pamięć telefonu przy różnych formatach zdjęć” w poprzednim rozdziale.



Zadanie 1

Pole figury na matrycy 8×8

Cel:

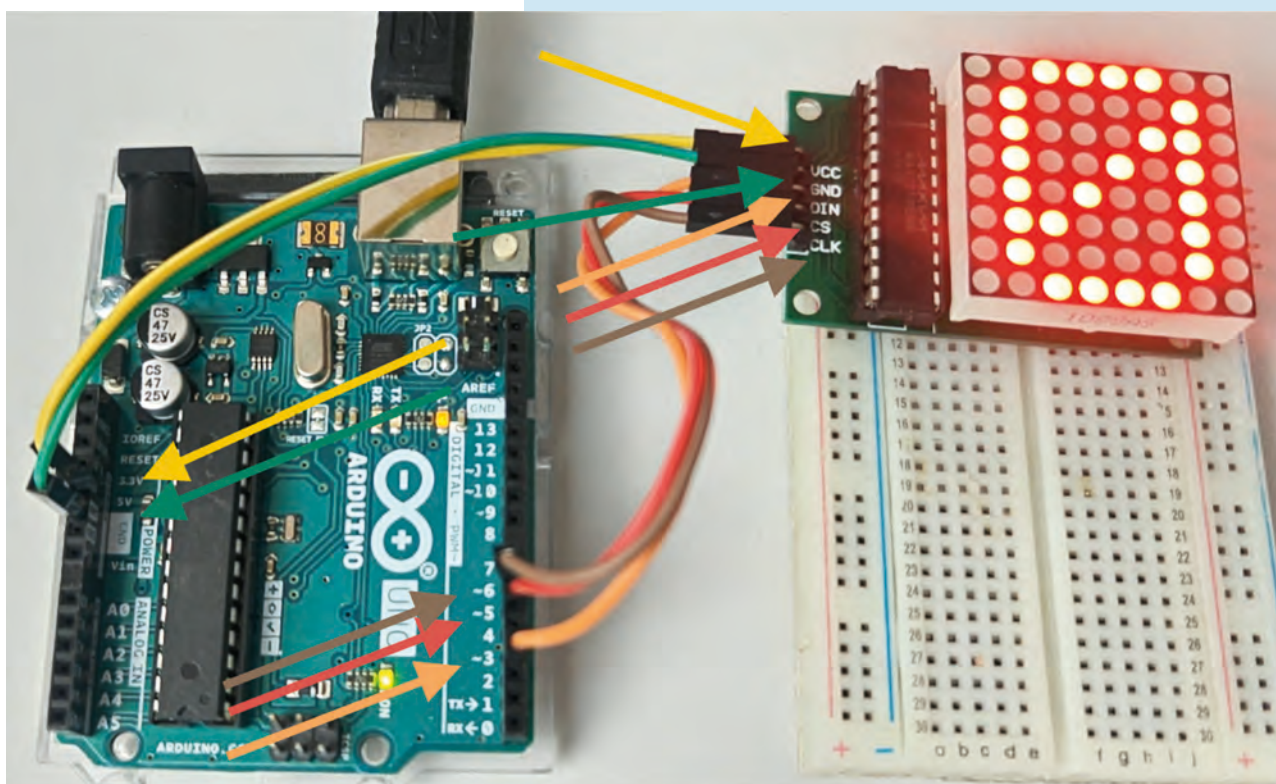
W tym ćwiczeniu Arduino wyświetla na matrycy LED 8×8 różne układy zapalonych diod (figury). Twoim zadaniem jest **policzyć**, ile pól (diod) świeci się na matrycy, a następnie **wpisać wynik** w Monitorze portu szeregowego. Program od razu sprawdzi, czy odpowiedź jest poprawna.

Uwaga: Połączenie matrycy LED 8×8 z Arduino wykonaj tak samo jak w części informatycznej modułu (sekcja „Podłączenie matrycy 8×8”). W tym zadaniu nie zmieniamy okablowania.



Krok 1. Podłącz Arduino

1. Podłącz Arduino do komputera kablem USB.
2. Upewnij się, że matryca LED 8×8 jest podłączona jak w części informatycznej modułu.



Rysunek 31
Schemat podłączenia matrycy led
z Arduino

Krok 2. Otwórz projekt / wgraj kod do Arduino (PictoBlox)

1. Uruchom PictoBlox, połącz się z płytką Arduino i włącz tryb upload.
2. Otwórz kod programu, który znajduje się w pliku obliczaniepolled8na8.ino. Możesz otworzyć w notatniku albo innym programie.



3. Skopiuj kod programu i wklej go do pictoblox w okienku znajdującym się w prawym górnym rogu, nadpisując cały znajdujący się tam kod.
4. Wgraj program do płytki (Upload Code).

Po wgraniu program zacznie działać automatycznie na Arduino.

Krok 3. Otwórz Monitor portu szeregowego

1. W PictoBlox znajdź opcję Serial Monitor / Monitor portu szeregowego (zwykle w menu narzędzi lub w zakładce Arduino).
2. Ustaw prędkość transmisji (baud rate) zgodną z programem:
 - jeśli w kodzie jest Serial.
begin(9600); → ustaw 9600
 - jeśli masz w kodzie inną wartość → ustaw taką samą
3. Otwórz monitor – zobaczysz komunikaty programu

Kryterium osiągnięcia celu:

1. Program wyświetla kolejne figury na matrycy LED 8×8,

2. Umożliwia wprowadzenie odpowiedzi w Monitorze portu szeregowego
3. Poprawnie informuje o wyniku w postaci smutnej lub uśmiechniętej buźki
4. Po zakończeniu serii zadań wyświetla podsumowanie liczby poprawnych odpowiedzi.



1. W jakich przypadkach możesz szybciej obliczyć liczbę świecących diod, używając mnożenia zamiast liczenia każdej diody osobno?
2. Dlaczego dzielenie figury na prostokątne fragmenty może ułatwić obliczenie całkowitej liczby świecących diod?



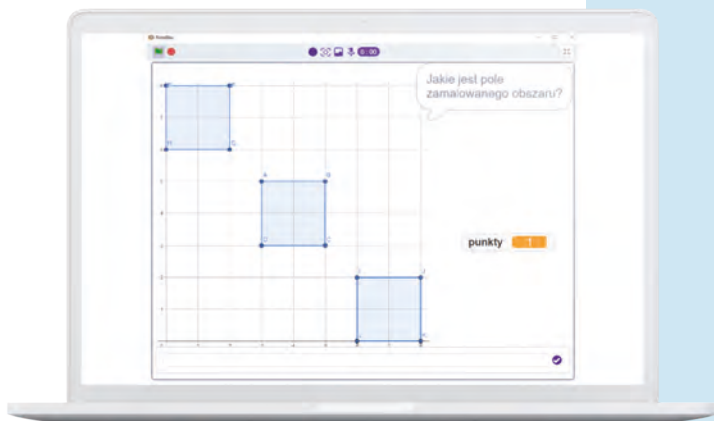
Zadanie 2

Oblicz pole figury
na siatce 8×8
(PictoBlox)

Cel:

ćwiczyć szybkie obliczanie pola na siatce, przechodząc od liczenia pojedynczych pól do analizy bardziej złożonych figur, ich grupowania oraz dostrzegania wzorców ułatwiających obliczenia.





Rysunek 32
Widok okna programu, do zadania 2

Polecenie

1. Po uruchomieniu programu, włącz pełen ekran i spójrz na wyświetlony obrazek (siatka 8×8 z zamalowanymi polami).
2. Policz, ile pól jest zamalowanych:
 - każde zamalowane pole traktuj jako 1 jednostkę pola,
 - jeśli figura składa się z kilku części, zsumuj pola wszystkich części.
3. Wpisz obliczoną wartość pola w oknie programu i zatwierdź odpowiedź.
4. Odczytaj komunikat programu:
 - jeśli odpowiedź jest poprawna → przechodzisz do kolejnego zadania,
 - jeśli odpowiedź jest błędna → program poinformuje, że wynik jest niepoprawny (i przejdziesz dalej)
5. Powtórz kroki 1–4 dla kolejnych wylosowanych obrazków.
6. Po rozwiązaniu wszystkich zadań odczytaj wynik końcowy (liczbę poprawnych odpowiedzi).



Kryterium osiągnięcia celu:

- Uczeń poprawnie oblicza pole jako liczbę zamalowanych pól.
- Program po każdej odpowiedzi wyświetla informację dobrze/źle.
- Na końcu program pokazuje łączny wynik (ile odpowiedzi było poprawnych).

1. W jaki sposób podział złożonej figury (np. równoległoboku lub deltoidu) na prostsze fragmenty pomaga szybciej obliczyć jej pole?
2. Dlaczego w przypadku figur takich jak trójkąt nie można policzyć pola przez proste mnożenie liczby wierszy przez liczbę kolumn?

Czego się nauczyłeś?



Matematyka

- czym jest matryca LED 8×8 i jak sterować pojedynczymi punktami,
- jak komputer reprezentuje obraz jako siatkę pikseli,
- jak dane z czujnika mogą być przedstawione w formie graficznej.



Informatyka

- jak obliczać pole figury jako liczbę pól jednostkowych,
- jak przyspieszyć obliczenia przez grupowanie i dostrzeganie wzorców,
- jak analizować złożone figury poprzez podział na prostsze części.

