

# Zielone światło dla micro:bita

dr Katarzyna OŁĘDZKA

Na stronie micro:bita – małej, programowalnej płytki – czytamy: *Get creative, get connected, get coding* – Bądź kreatywny, Bądź podłączony, Programuj. Spróbuję przekonać Czytelników, by dać się „wkręcić” w zajęcia z micro:bitem, by zapalić zielone światło dla nauki programowania z elementami robotyki. Warto zauważyć, że łączyąc programowanie i obsługę różnego rodzaju czujników, serwomechanizmów, silników czy innych urządzeń elektronicznych, mamy okazję rozbudzić u młodych ludzi pasję inżynierską.

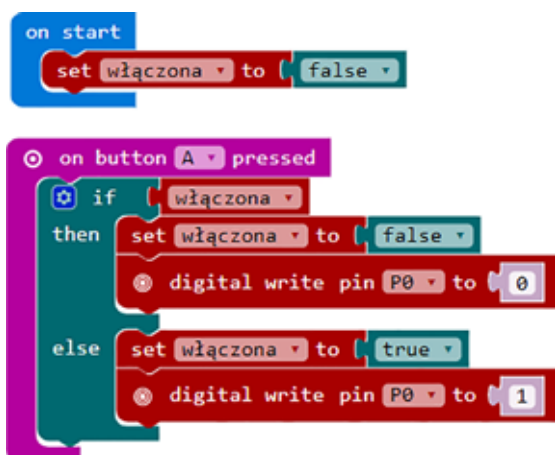
Płytką micro:bit została stworzona przez BBC z myślą o angielskich uczniach. Jest dedykowana uczniom szkoły podstawowej, aby zachęcić ich do eksperymentowania z elektroniką i programowaniem. Świetnie nadaje się do tworzenia własnych projektów, które pobudzą fantazję i ciekawość, sprzyjają rozwojowi kreatywności. W dobie rozwoju Internetu Rzeczy młódzież może poznać i zbudować własne układy oraz doskonalić umiejętność programowania w ujęciu praktycznym. Ponadto warto zwrócić uwagę, że zajęcia wykorzystujące elementy robotyki są propozycją rozwoju zainteresowań nie tylko dla uczniów ogólnie wyróżniających się, ale takich, którym przypisano łatkę „mniej zdolnych” lub „sprawiających trudności”. Zapropowanie uczniom udziału w atrakcyjnych zajęciach jest dobrym sposobem na zwiększenie ich motywacji do nauki oraz rozwijania zainteresowań.

Micro:bit to płytka o wymiarach 52×43 mm z 32-bitowym procesorem ARM Cortex M0. Na płytce znajduje się wyświetlacz LED 5×5, czyli 25 diod oraz dwa przyciski oznaczone jako A i B. Ponadto mamy do dyspozycji czujniki temperatury, światła oraz akcelerometr i magnetometr, a nawet Bluetooth. Micro:bit może być zasilany przez USB lub za pomocą baterii przez dedykowane złącze. Jest wyposażony w 5 złączy bananowych oraz w kilkanaście mniejszych, łącznie ma na swym pokładzie 23 złącza. Programy dla micro:bita możemy przygotowywać w przeglądarce WWW, korzystając z języka JavaScript lub środowiska wizualnego. Innym sposobem jest programowanie w Pythonie, zarówno online jak i offline, z wykorzystaniem biblioteki microPython. Po podłączeniu micro:bita przez interfejs USB zapisujemy skrypt w formacie HEX na urządzeniu, podobnie jak na pendrive.

Dalej przedstawiamy kilka projektów związanych z zapalaniem i gaszeniem światła, które można wykorzystać na zajęciach z uczniami.



Rysunek 1. Układ – lampka podłączona do mikro:bita.

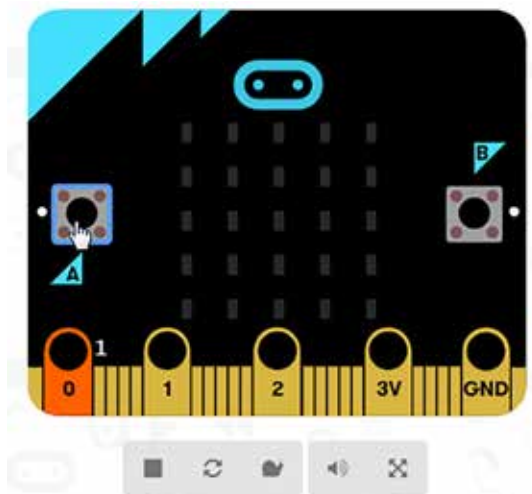


Rysunek 2. Skrypt – włączanie i wyłączenie lampki przyciskiem A.

## Włącz i wyłącz lampkę

Nasz eksperyment rozpoczniemy od podłączenia lampki. Będzie ona gaszona/zapalana, gdy użytkownik naciśnie przycisk. Co prawda standardowo lampka jest zasilana napięciem 5 V, podczas gdy micro:bit oferuje tylko 3 V, ale nie będziemy się tym zbytnio przejmować. Lampka będzie trochę słabiej świecić – budujemy przecież prototyp.

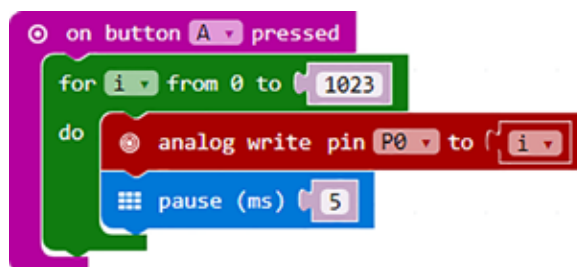
W skrypcie potrzebujemy zmiennej, na której będziemy pamiętać, czy lampka jest włączona, czy nie. Na początek lampka jest wyłączona. Gdy użytkownik wciśnie przycisk A, sprawdzamy, jaki jest stan zmiennej. Gdy lampka jest włączona, wyłączamy ją, lub odwrotnie. To oznacza, że musimy zmienić wartość zmiennej **włączona** z true (prawda) na false (fałsz) lub odwrotnie oraz przekazać sygnał na wyjście P0. Po napisaniu skryptu można go przetestować w emulatorze, a następnie w rzeczywistości na zbudowanym układzie.



Rysunek 3. Testowanie w emulatorze.

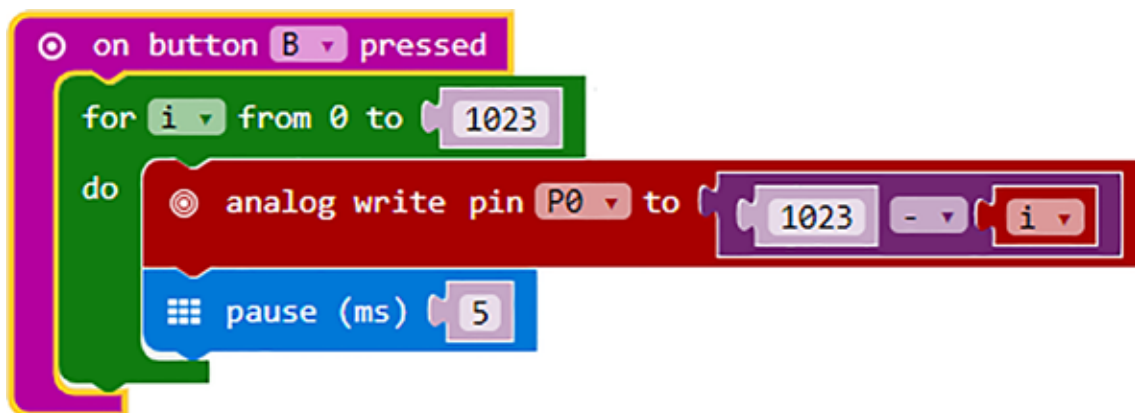
## Zabawa ze światłem

Spróbujmy przygotować skrypty, które będą realizowały stopniowe rozjaśnianie i przygaszanie światła (lampki), czyli łagodne przejście od stanu, gdy lampka jest wyłączona, do pełnego świecenia. W tym celu zamiast zapisu cyfrowego (*digital write*), gdzie mamy tylko wartości 0 lub 1, musimy wykorzystać zapis analogowy (*analog write*) z wartościami od 0 do 1023. Jeśli użytkownik naciśnie klawisz A, to program powinien na wyjściu P0 podawać coraz to większy sygnał od 0 do 1023. Przy czym długość pauzy wpływa na szybkość zmian.



Rysunek 4. Skrypt – stopniowe rozjaśnianie.

Podobnie postępujemy, gdy chcemy poprzez naciśnięcie przycisku B stopniowo wygasić światło. Korzystamy z pętli *for*, dla której wartość zmiennej sterującej *i* zmienia się od 0 do 1023. Możemy jednak skorzystać z odejmowania – od 1023 odejmujemy wartość *i*. W ten sposób osiągniemy stopniowe zmniejszanie się wartości. Najpierw mamy  $1023 - 0 = 1023$ , potem  $1023 - 1 = 1022$ , ... i tak dalej aż do 0.

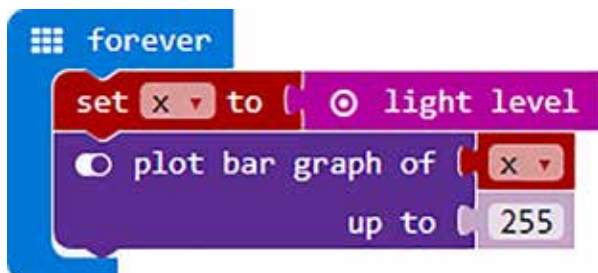


Rysunek 5. Skrypt – stopniowe przyciemnianie.

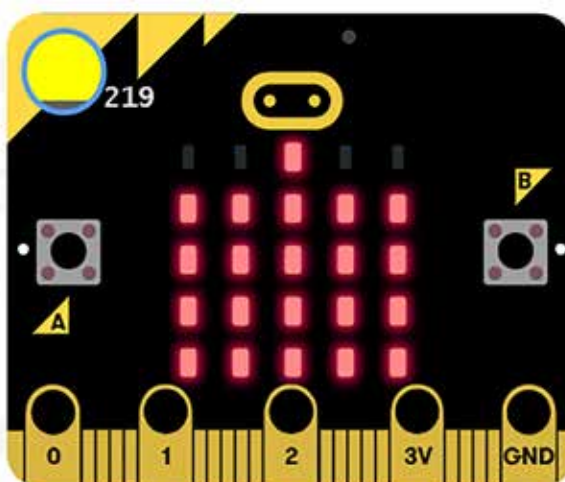
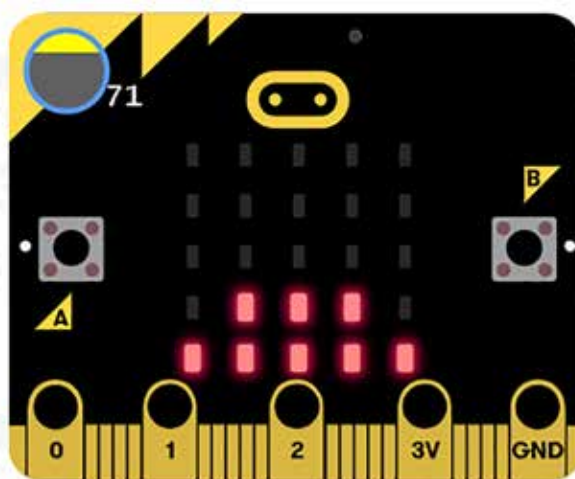
Zauważmy, że w tym projekcie wykorzystujemy pętlę *for* zarówno wtedy, gdy chcemy stopniowo zwiększać wartość, jak i wtedy, gdy ją zmniejszamy. Jest to praktyczny przykład wykorzystania instrukcji iteracji. Mamy też elementy elektroniki, możemy wyjaśnić uczniom różnicę między wyjściem cyfrowym, które przyjmuje tylko dwie wartości 0 i 1, a analogowym, które pozwala na uzyskanie wartości pośrednich.

### Zapal lampkę, gdy jest ciemno

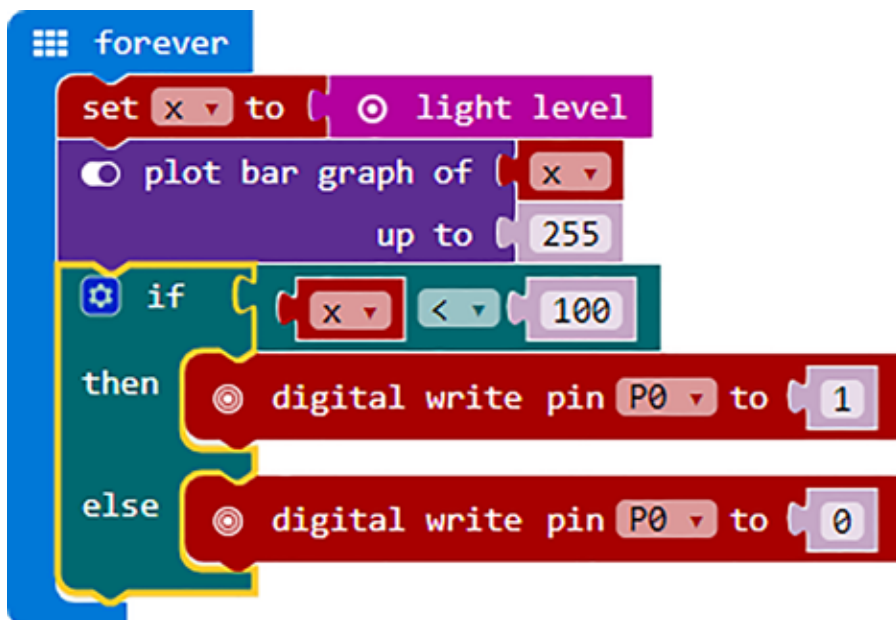
Trzecie zadanie będzie polegało na samoczynnym włączaniu się lampki wtedy, gdy jest ciemno. Będziemy korzystać z czujnika światła wbudowanego w płytkę. Odczytujemy odpowiednią wartość i zapamiętujemy ją na zmiennej *x*. Do celów testowych możemy wartości zmiennej wyświetlić w formie tzw. grafu za pomocą polecenia *plot bar graph*, które wywołujemy z dwoma parametrami: zmieniającą się wartością światła (zmienna *x*) oraz zakresem górnego – możliwie największej wartości, jaką może przyjmować zmienna *x*, w tym przypadku 255. Wartość światła mierzona przez micro:bit jest zakresu 0-255.



Rysunek 6. Skrypt – pomiar światła i graficzne przedstawienie wyniku.



Rysunek 7. Emulator – różne wskazania światła: u góry 71, u dołu 219.



**Rysunek 8.** Skrypt – zapalenie/gaszenie lampki w zależności od wyników pomiaru światła.

Następnie skrypt rozbudowujemy o instrukcję warunkową. Będzie ona definiować, kiedy lampka powinna być zapalana, a kiedy gaszona. Gdy zmierzona wartość światła jest mniejsza od wartości progowej, na przykład 100, czyli kiedy jest ciemno, to lampka powinna być zapalona, w przeciwnym przypadku – zgaszona.

## Na zakończenie

Cykl zajęć z robotyki prowadzonych dla uczniów to doskonała okazja do rozwinięcia kompetencji zarówno informatycznych, jak i inżynierskich. Młodzi ludzie mają okazję pobawić się w konstruktorów, by lepiej zrozumieć cyfrowy świat, nauczyć się twórczego myślenia. Ciekawe projekty działają motywująco, wspomagając uczniów w ich rozwoju. Pełnią także funkcje poznawcze, zaznajamiając użytkowników z rzeczywistym działaniem robotów dzięki ćwiczeniom praktycznym. Mają niewątpliwie funkcje kształtujące – wptywają bezpośrednio na proces poznawczy i rozwój umiejętności uczenia się. Współczesne trendy edukacyjne, które promują naukę programowania i robotykę już w szkole podstawowej, są godne polecenia i wdrażania. Przedstawione pomysły na zajęcia z uczniami z micro:bitem i światłem to tylko drobna część tego, co można zrobić na zajęciach. Podłączaliśmy światło białe, a przecież można też skorzystać z diod kolorowych: czerwonych, niebieskich czy

zielonych. Zachęcam, by rozpocząć swoją przygodę z micro:bitem już dziś, z projektowaniem i budowaniem układów, również tych z zielonym światłem.

**dr Katarzyna OLĘDZKA** jest nauczycielem konsultantem w Ośrodku Edukacji Informatycznej i Zastosowań Komputerów w Warszawie.