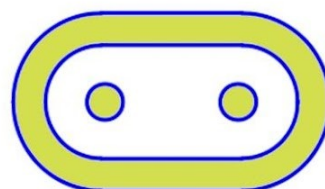
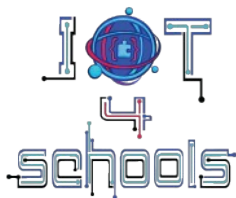


Technical guide for

BBC micro:bit





IoT4Schools
**“Bringing the Internet of Things in school education
as a tool to address 21st century challenges”**

Techniczny przewodnik o BBC Micro:bit

Autorzy: C. Papasarantou, R. Alimisi [EDUMOTIVA]

Współtwórcy: A. Tefelska, D. Tefelski [WUT]

License: CC BY-NC 4.0 LEGAL CODE, Attribution-NonCommercial 4.0 International



**Co-funded by
the European Union**

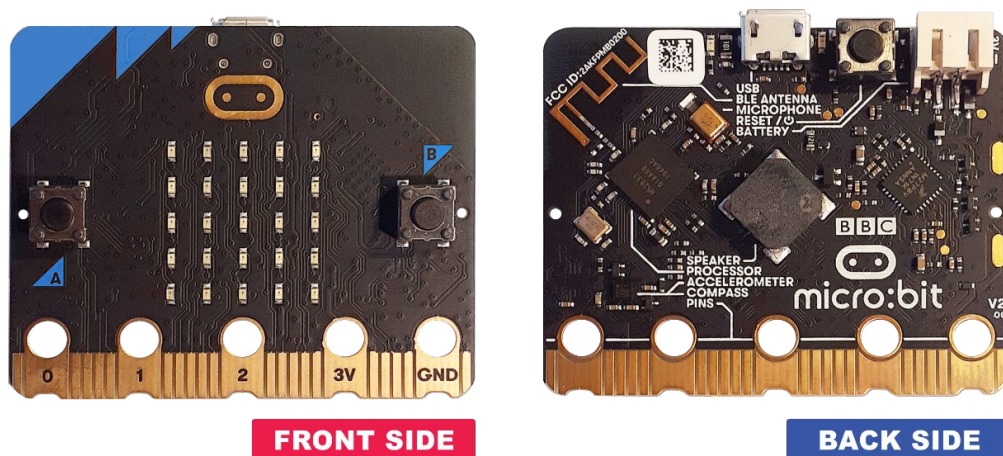
The European Commission's support to produce this publication does not constitute an endorsement of the contents, which reflect the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Spis treści

1. Introduction to BBC micro:bit	4
2. The BBC micro:bit board	5
3. Programming BBC micro:bit – The Microsoft Makecode Environment	7
3.1 Microsoft Makecode environment: becoming familiar with the interface	7
3.2 Microsoft Makecode environment: creating a new project	8
3.3 Microsoft Makecode environment: command blocks and groups	9
3.4 Microsoft Makecode environment: assembling a code/script	12
3.5 Microsoft Makecode environment: downloading and testing a project or a code/script	15
4. How to use micro:bit for introducing IoT	17
4.1 Bluetooth Radio communication	17
4.1.a. Radio command group	17
4.1.b. Bluetooth command group	19
4.2 WiFi communication via ESP8266 WiFi module	22
5. Indicative activities to become familiar with micro:bit and radio communication	24
5.1 Sending a smiley face	24
5.2 Sending a smiley and a sad face	27
5.3 Sending temperature data to another micro:bit	29

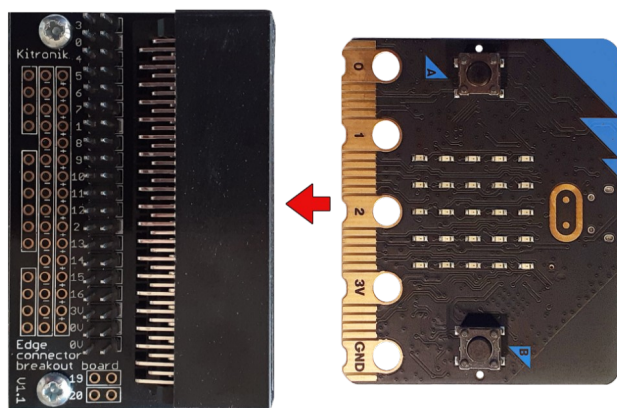
1. Wprowadzenie do BBC micro:bit

Płytką BBC micro:bit (rysunek 1) to kieszonkowy mikrokontroler, zaprojektowany głównie do celów edukacyjnych, aby wprowadzić młodych uczniów w dziedzinę ICT w prosty i intuicyjny sposób. Posiadając kilka wbudowanych urządzeń wejściowych i wyjściowych, micro:bit oferuje poręczne rozwiązanie do tworzenia wielu interesujących projektów związanych z STEM, bez potrzeby dodatkowego sprzętu. Ponadto micro:bit jest kompatybilny z wieloma czujnikami i komponentami elektronicznymi powszechnie używanymi w innych płytkach (takich jak Raspberry Pi Pico i Arduino), co jeszcze bardziej rozszerza jego możliwości. Obecnie istnieją dwie różne wersje płytki micro:bit: V1 i V2. W projekcie IoT4Schools używana jest wersja V2, ponieważ ma ona pewne dodatkowe komponenty.



Rysunek 1: Dwie strony płytki BBC micro:bit

W niektórych przypadkach zaleca się stosowanie adaptera breakout, ponieważ umożliwia on stabilne połączenie większej liczby podzespołów elektronicznych, ułatwiając proces projektowania obwodów. Podłączenie płytki micro:bit do adaptera breakout jest zazwyczaj prostym procesem, ponieważ obejmuje jedynie przypięcie micro:bit do złącza (patrz przykład na rysunku 2).

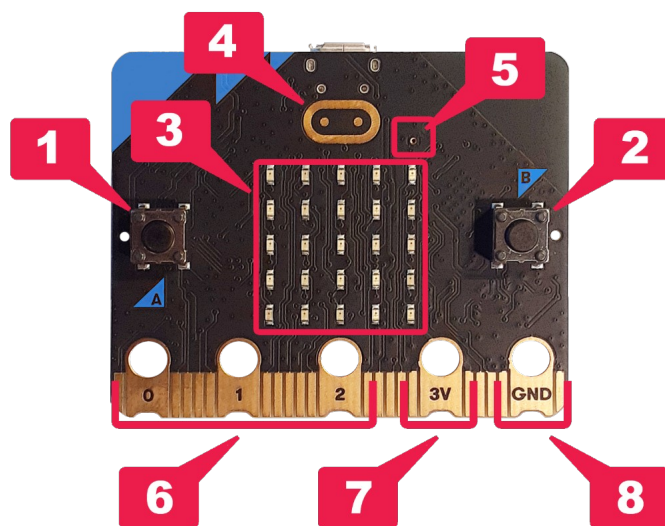


Rysunek 2: Podłączenie płytki BBC Micro:bit do adaptera breakout.

2. Płytki BBC micro:bit

Jak wspomniano, płyta BBC micro:bit ma wiele wbudowanych urządzeń wejściowych i wyjściowych. Niektóre z nich znajdują się z przodu płyty micro:bit, a niektóre z tyłu płyty micro:bit. Poniższe diagramy pokazują, gdzie znajduje się każde urządzenie.

Strona przednia (Rysunek 3):



Rysunek 3: Wbudowane urządzenia wejściowe i wyjściowe na przedniej ścianie micro:bit

- Przycisk A (1) i przycisk B (2); dwa urządzenia wejściowe, które można zaprogramować tak, aby wyzwały akcję po naciśnięciu
- Wyświetlacz LED 5x5 (3); działający jako urządzenie wejściowe (czujnik światła) i wyjściowe (ekran wyświetlacza)
- Czujnik dotykowy (4) (dostępny w wersji 2); urządzenie wejściowe, które umożliwia micro:bit reagowanie na dotyk
- Mikrofon (5) (dostępny w wersji 2); urządzenie wejściowe, które umożliwia micro:bit reagowanie na zewnętrzne zdarzenia audio
- Kilka pinów GPIO (6), w tym zasilanie 3 V (7) i uziemienie (8) do podłączania większej liczby elementów elektronicznych za pomocą zacisków krokodylkowych lub płytki breakout Edge Connector.

Tylna strona (Rysunek 4):

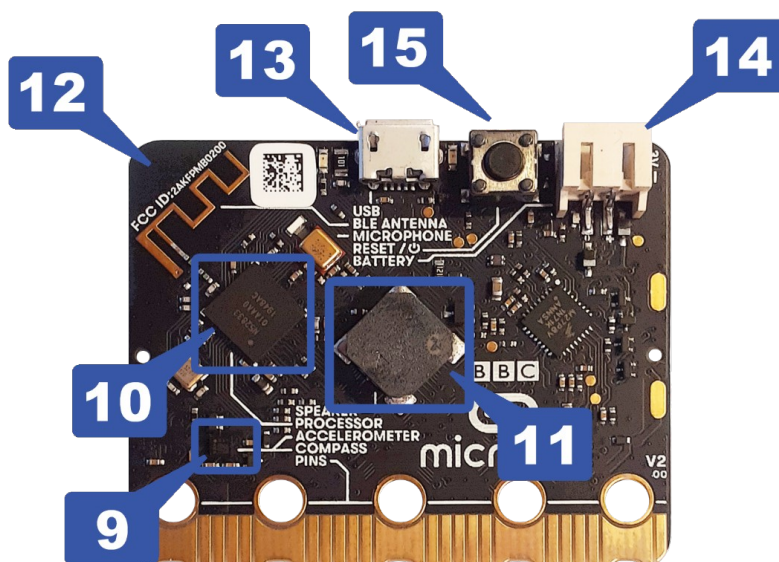


Figure 4: In-built input and output devices on the back side of the micro:bit

- Akcelerometr (9); czujnik, który wykrywa, czy micro:bit jest wstrząśnięty, upuszczony lub obrócony do góry nogami
- Kompas (9); czujnik, który wykrywa pole magnetyczne Ziemi, umożliwiając micro:bit wykrywanie kierunku, w którym jest zwrócony
- Procesor (10); „mózg” micro:bit-a, który przetwarza wszystkie informacje, odbierając dane wejściowe, uruchamiając programy i dając dane wyjściowe. Wewnątrz procesora znajduje się czujnik temperatury, umożliwiający micro:bit pomiar temperatury w stopniach Celsjusza
- Głośnik (11) (dostępny w wersji V2); urządzenie wyjściowe, które umożliwia odtwarzanie dźwięków
- Antena do połączenia radiowego bądź Bluetooth(12); umożliwiająca micro:bit bezprzewodową komunikację z innymi micro:bit lub innymi urządzeniami. To wbudowane urządzenie jest niezwykle cenne przy realizacji projektów IoT (patrz sekcja 4).
- Gniazdo USB (13); przez które microbit można podłączyć do komputera lub do zewnętrznego źródła zasilania
- Złącze baterii JST (14); gdzie można podłączyć źródło zasilania ze złączem JST
- Wyłącznik zasilania (15) (dostępny w wersji V2); przycisk, który może włączać/wyłączać lub resetować micro:bit po naciśnięciu.

3. Programowanie BBC micro:bit -środowisko Microsoft Makecode

Oficjalnym środowiskiem do programowania BBC micro:bit jest Microsoft Makecode (<https://makecode.microbit.org/>); oprogramowanie internetowe oferujące rozwiązania oparte zarówno na blokach, jak i na językach tekstowych (Javascript lub Python), co czyni je elastycznym i odpowiednim dla młodych uczniów narzędziem programistycznym. Aby korzystać z oprogramowania Makecode, wymagane jest połączenie z Internetem. Dodaje to pewne ograniczenia dotyczące dostępności. Makecode można również używać za pośrednictwem inteligentnych urządzeń mobilnych (telefonów lub tabletów) oraz za pośrednictwem aplikacji iOS i Android, które są dostępne do pobrania.

Płytę micro:bit można również programować w różnych środowiskach opartych na blokach, takich jak Scratch (za pośrednictwem aplikacji Scratch Link), pictoblox, microblocks.fun, Open Roberta, Art.bit i inne. W przeciwieństwie do wszystkich innych, Scratch jest oficjalnie wspierany przez micro:bit Educational Foundation.

3.1 Środowisko Microsoft Makecode: zapoznanie się z interfejsem

Poniższy diagram (rysunek 5) przedstawia przegląd środowiska Microsoft Makecode.



Rysunek 5: Interfejs oprogramowania Microsoft Makecode .

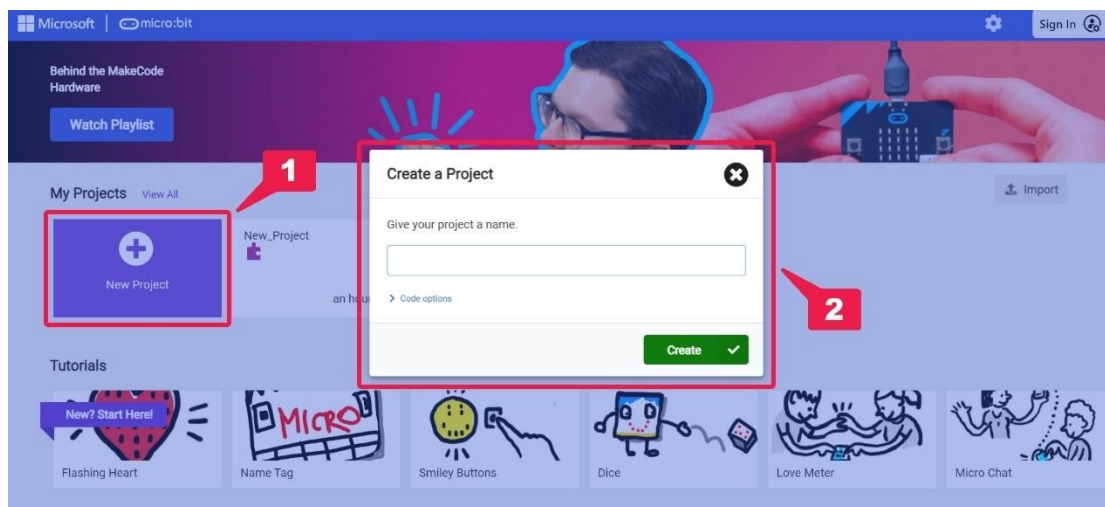
W szczególności, znajdziesz:

- obszar (1), w którym możesz złożyć swój kod/skrypt, przeciągając, upuszczając i przyciągając bloki poleceń
- kilka bloków poleceń (2) pogrupowanych w grupy oznaczone kolorami, aby zaprogramować swój projekt
- symulator (3), w którym możesz przetestować działanie swojego kodu/skryptu przed pobraniem go na płytkę micro:bit
- przycisk pobierania (4), aby bezpośrednio pobrać kod/skrypt na micro:bit
- przycisk zapisywania (5), aby zapisać kod/skrypt lokalnie na swoim komputerze jako plik .hex
- przełącznik (6), aby przełączyć się na środowisko kodowania oparte na tekście (JavaScript lub Python) i odwrotnie
- przycisk/menu „more...” (7), aby uzyskać dostęp do różnych funkcji, takich jak “project settings”, “extensions”, “language” i inne.

- przycisk logowania (8), aby zalogować się na swoje konto osobiste. Zdecydowanie zaleca się utworzenie konta, aby uniknąć utraty swoich projektów.

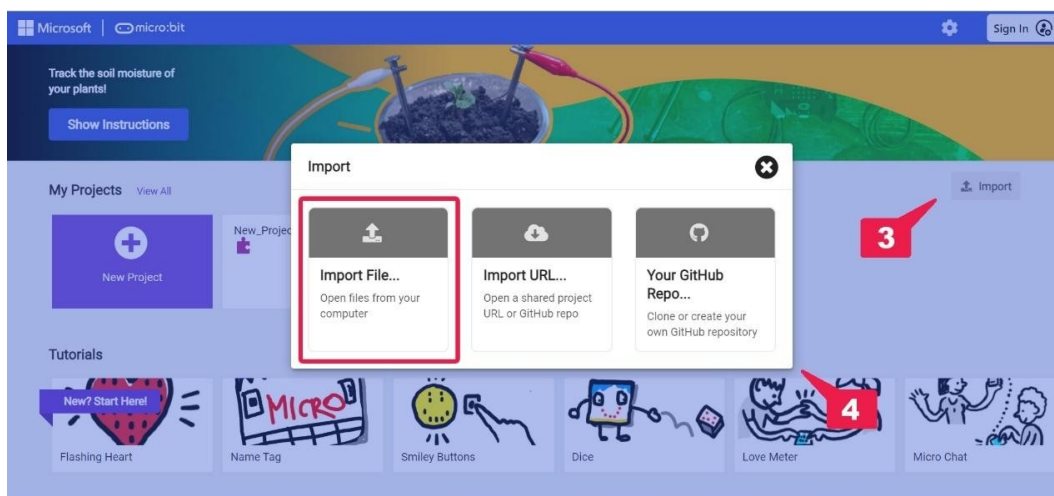
3.2 Środowisko Microsoft Makecode: tworzenie nowego projektu

Aby utworzyć i rozpocząć pracę nad nowym projektem, wejdź na stronę internetową Makecode (<https://makecode.microbit.org/>) w przeglądarce i na stronie głównej kliknij kartę New Project (1). W menu podręcznym „Create a Project” (2) wpisz nazwę swojego projektu, a następnie kliknij przycisk „Create” (Rysunek 6).



Rysunek 6: Tworzenie nowego projektu

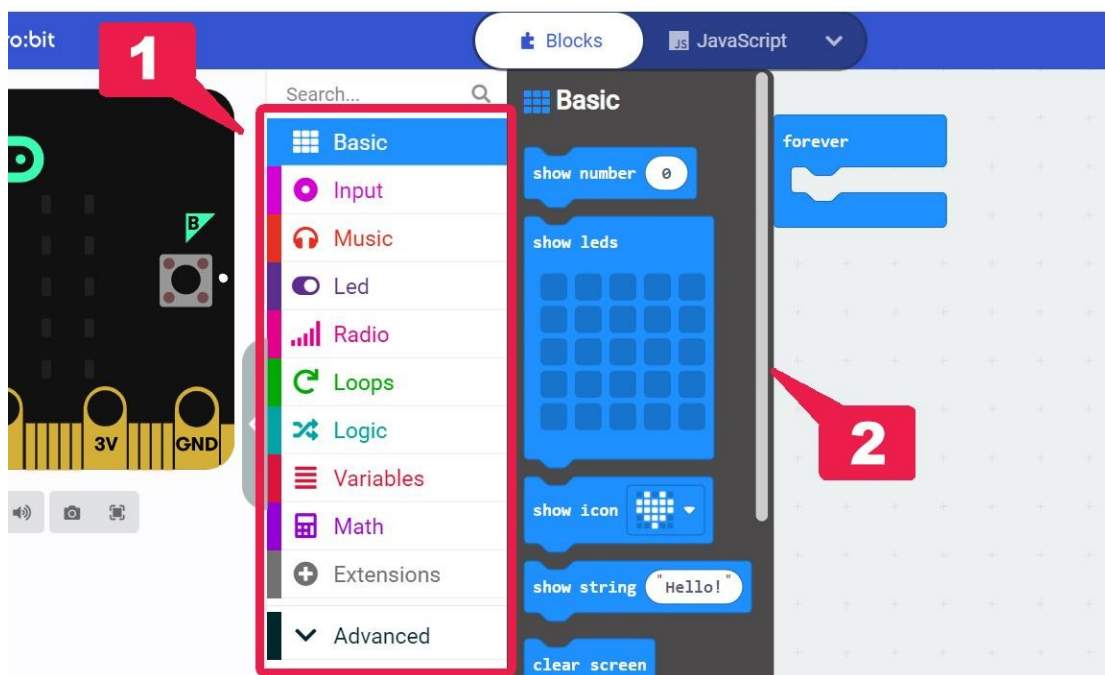
Możesz również zaimportować istniejący projekt, klikając przycisk „import” (3) i wybierając miejsce, z którego chcesz importować (np. ze swojego komputera, z chmury lub z GitHub) w pojawiającym się okienku zatytułowanym „Import” (4) (Rysunek 7).



Rysunek 7: Wybór miejsca, z którego importujemy projekt.

3.3 Środowisko Microsoft Makecode: bloki poleceń i grupy

Istnieje wiele różnych bloków poleceń, których możesz użyć w swoim projekcie (Rysunek 8). Bloki te są zorganizowane w różne grupy poleceń oznaczone kolorami (1). Poświęć trochę czasu na zbadanie, co wszystkie te grupy mogą zrobić i jakie rodzaje bloków się w nich znajdują. Na przykład grupa poleceń „Basic” zawiera bloki, takie jak bloki obsługi zdarzeń „On start” i „Forever” lub bloki „show number/string ...” (2), które pomagają zaprogramować podstawowe działania i funkcje. Grupa poleceń „Input” zawiera bloki poleceń, które pomagają zaprogramować kilka urządzeń wejściowych, takich jak przyciski, czujnik światła i temperatura, podczas gdy grupa poleceń „Logic” zawiera operatory i stałe, takie jak polecenia warunkowe „if...else..” i polecenia logiki boolowskiej.



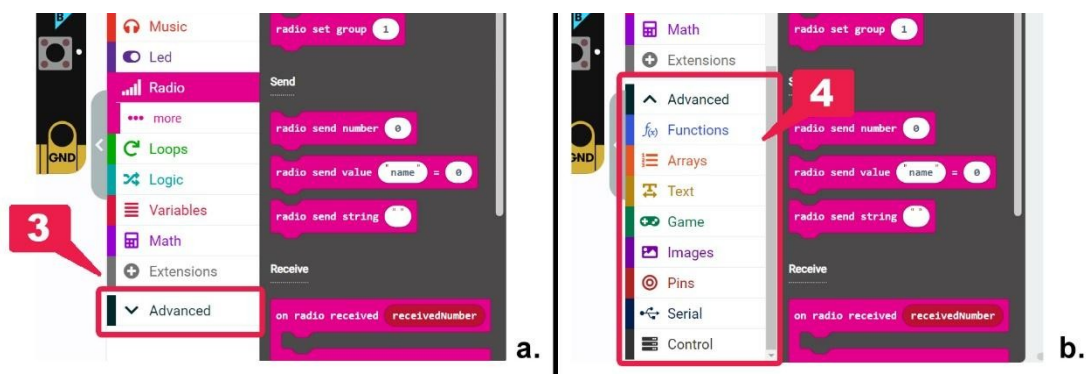
Rysunek 8: Grupy i bloki poleceń.

Po kliknięciu niektóre grupy poleceń (takie jak „Input” lub „Radio”) ujawnią podmenu „...more” (Rysunek 9). Grupy te zawierają drugą zakładkę z blokami poleceń.



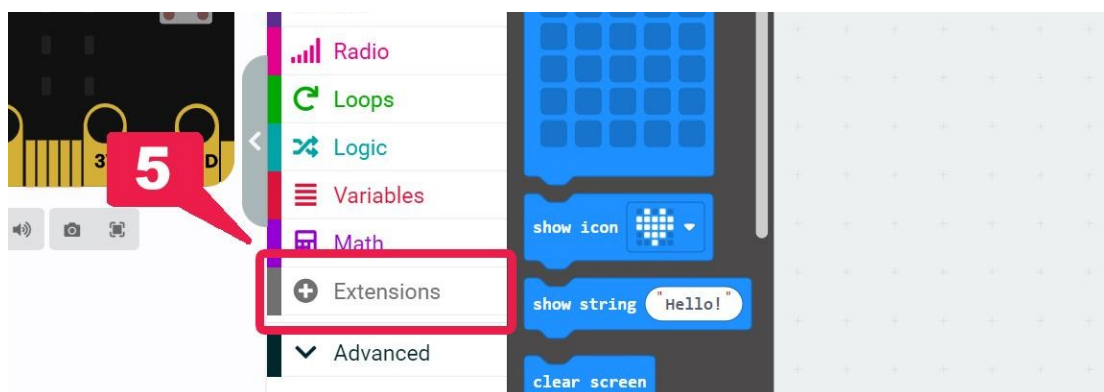
Rysunek 9: Podmenu „...more”

Istnieje czarna grupa poleceń o nazwie „Advanced” (rysunek 10a.). Kliknięcie jej przeniesie Cię do innych grup poleceń (rysunek 10b.), które mogą być używane w zaawansowanych projektach. Przykładowe zaawansowane grupy to „Functions” i „Arrays”.



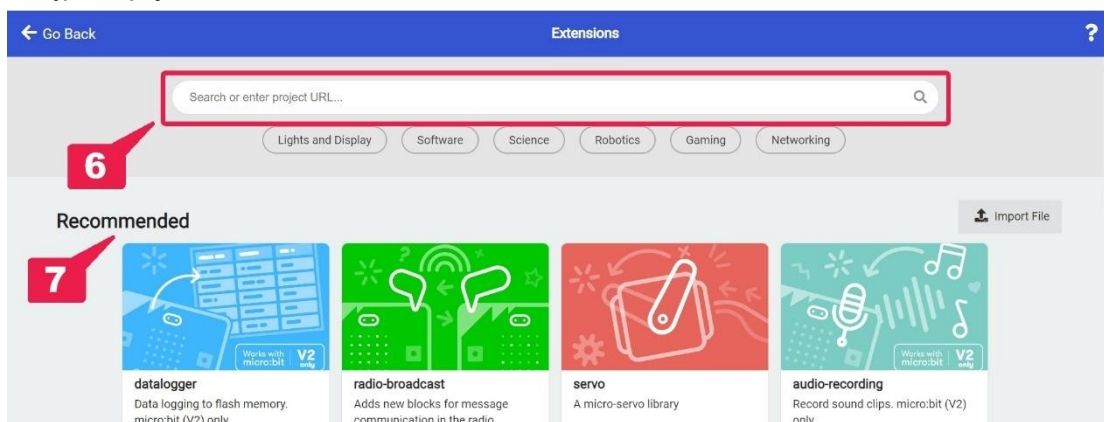
Rysunek 10: a. Wybór menu zaawansowanego; b. Grupy poleceń zawarte w menu zaawansowanym

Oprócz standardowych grup poleceń możesz importować inne bloki zaprojektowane i dostarczone przez innych programistów ze społeczności micro:bit. Bloki te znajdują się w grupie „Extensions” (5) (Rysunek 11) i mogą pomóc w programowaniu bardziej złożonej elektroniki i czujników, rozszerzając w ten sposób możliwości micro:bit.



Rysunek 11: Grupa „Extensions”

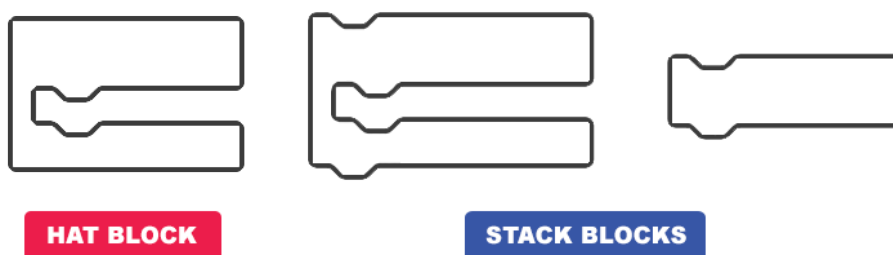
Kliknięcie grupy „Extensions” przeniesie Cię do menu wyszukiwania rozszerzeń (Rysunek 12). Tutaj możesz albo wyszukiwać wśród zalecanych rozszerzeń (7), albo wpisać nazwę (6) komponentu, który chcesz zaprogramować (np. ESP8266, Bluetooth itd.) i zobaczyć dostępne opcje.



Rysunek 12: Menu wyszukiwania rozszerzeń.

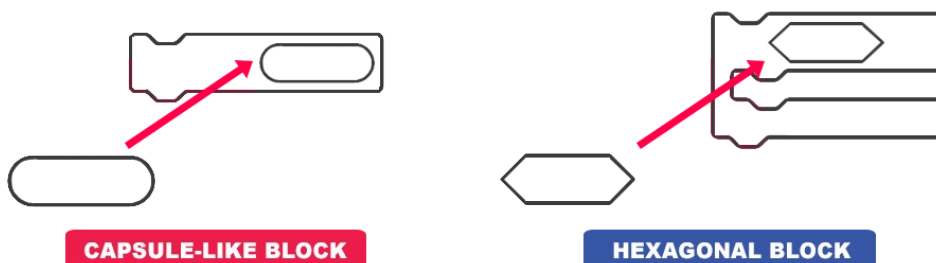
3.4 Środowisko Microsoft Makecode: składanie kodu/skryptu

Aby utworzyć/złożyć skrypt w wersji Makecode opartej na blokach, musisz połączyć różne bloki poleceń w sposób, który ma sens z perspektywy kodowania/programowania. Skrypt można złożyć, używając bloków hat i stack (Rysunek 13). Bloki hat są obowiązkowe, ponieważ to one rozpoczynają skrypt. Bloki stack, które są zwykle głównymi poleceniami kodu, to bloki przypominające łamigłówkę, które mieszczą się w bloku hat. Skrypt musi zawierać co najmniej jeden blok hat i tyle bloków stack, ile jest potrzebnych do realizacji projektu.



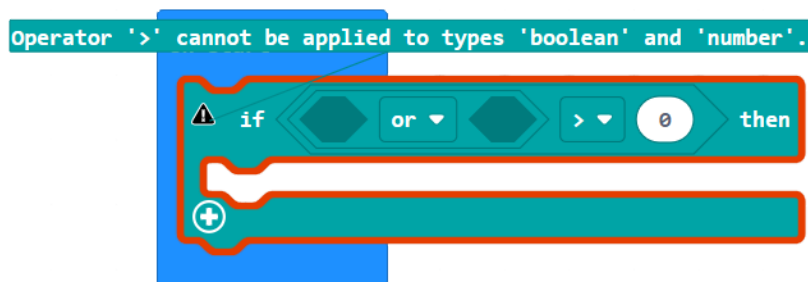
Rysunek 13: Przykłady, jak mogą wyglądać bloki Hat i Stack

Ponadto istnieją również pewne bloki capsule-like i hexagonal (Rysunek 14), których nie można używać niezależnie, ale należy je zintegrować w określonych blokach stosu. Te bloki zazwyczaj zawierają dane, takie jak ciągi znaków, dane liczbowe, zmienne, operatory Boole'a i porównania.



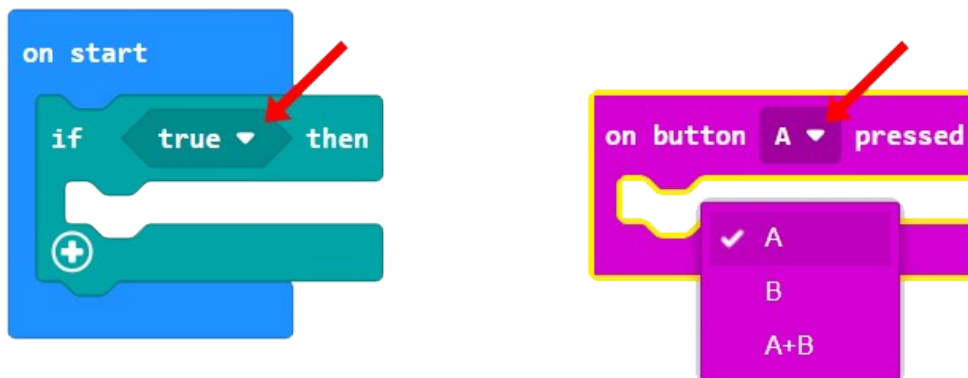
Rysunek 14: Bloki capsule-like i hexagonal .

Uwaga: Jeśli nie masz pewności, czy blok zmieści się w innym bloku, po prostu spróbuj. Jeśli poleceń nie da się połączyć, nie będziesz mieć możliwości zintegrowania bloku lub otrzymasz ostrzeżenie (Rysunek 15).



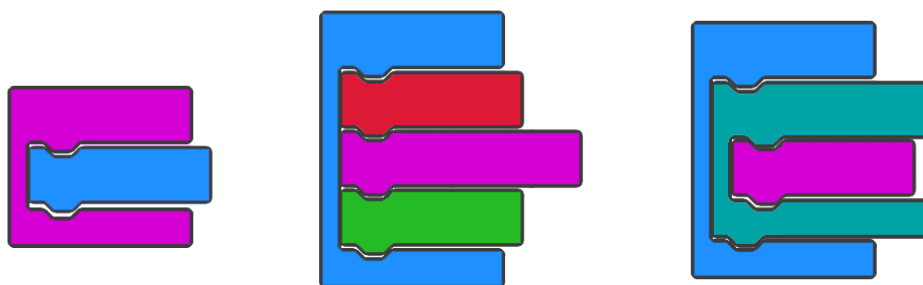
Rysunek 15: Błąd w kodzie. Operator heksagonalny nie może być zintegrowany w tej części skryptu

Niektóre polecenia mogą mieć menu rozwijane (Rysunek 16). Kliknij strzałkę i przeszukaj rozwijane menu, aby odkryć dodatkowe opcje programowania.



Rysunek 16: Polecenia w rozwijanym menu

Rysunek 17 przedstawia kilka koncepcyjnych przykładów struktur skryptów. Takie skrypty można złożyć, przeciągając i upuszczając bloki poleceń do obszaru skryptu i łącząc je ze sobą. Długość przedstawionych skryptów jest orientacyjna.



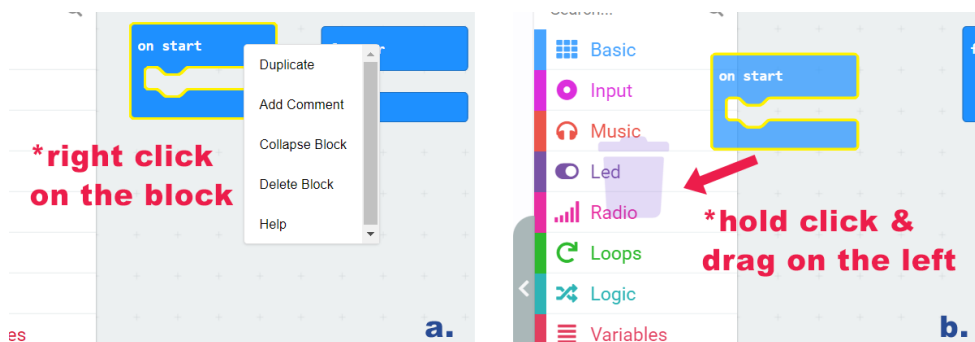
Rysunek 17: Przykłady złożonych skryptów.

Jeśli blok (lub stos bloków) jest szary, oznacza to, że skrypt jest nieaktywny i nie zostanie wykonany (rysunek 18).



Rysunek 18: Przykłady aktywnych i nieaktywnych skryptów/kodów. Polecenie clear screen nie zostanie wykonane

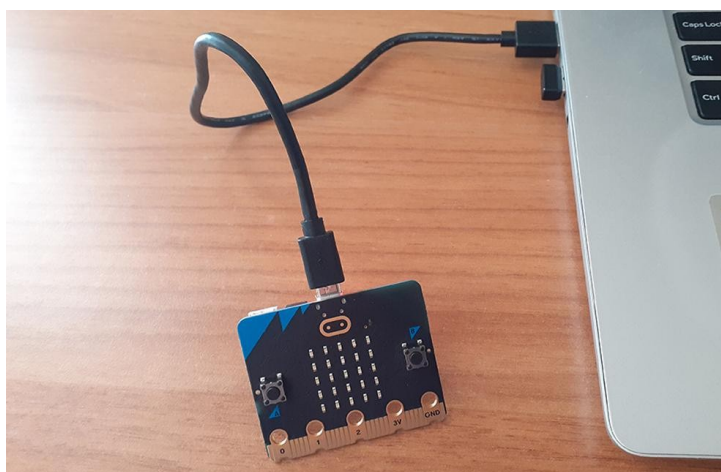
Aby usunąć blok lub cały skrypt, możesz kliknąć prawym przyciskiem myszy na bloku i wybrać opcję „delete block(s)” z menu podręcznego (Rysunek 19 a.) lub klikając na bloku (lub skrypcie) i przeciągnąć go w lewo, aż pojawi się fioletowy kosz na śmieci (Rysunek 19b.)



Rysunek 19: Usunięcie bloku poleceń.

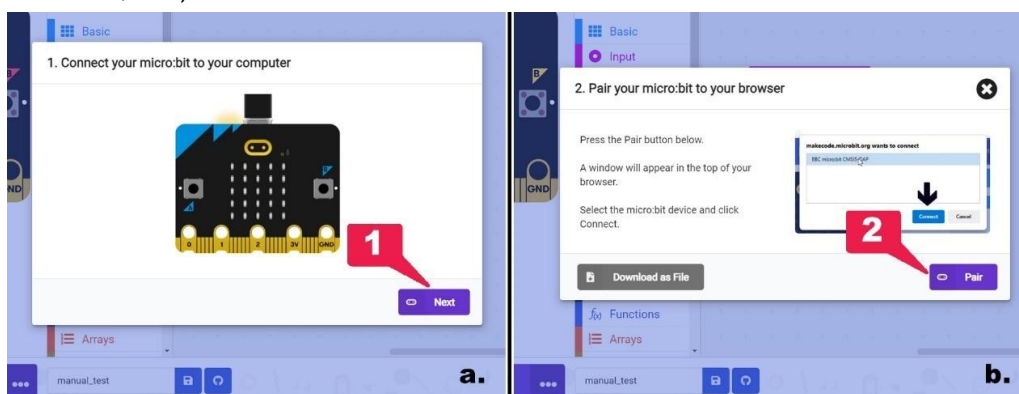
3.5 Środowisko Microsoft Makecode: pobieranie i testowanie projektu lub kodu/skryptu

Aby pobrać projekt lub kod/skrypt na micro:bit, należy podłączyć płytkę micro:bit do komputera za pomocą kabla USB (rysunek 20), a następnie kliknąć przycisk „download” [rysunek 5, (4)] w oprogramowaniu Makecode.



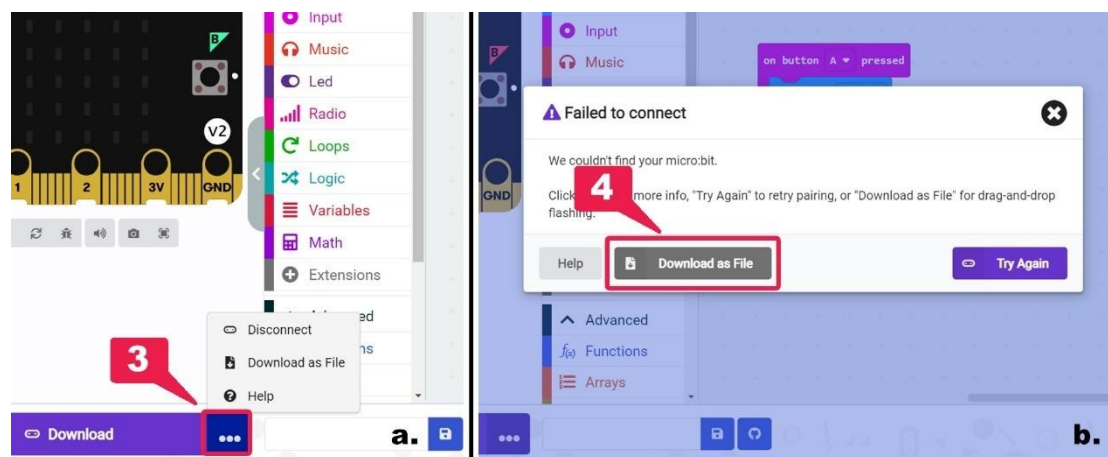
Rysunek 20: Podłączanie płytki micro:bit do komputera za pomocą kabla USB

Jeśli zapomnisz podłączyć micro:bit lub jeśli skrypt nie zostanie pobrany od razu, Makecode udostępnia instrukcje krok po kroku, jak sparować micro:bit ze środowiskiem Makecode (rysunek 21a, 21b).



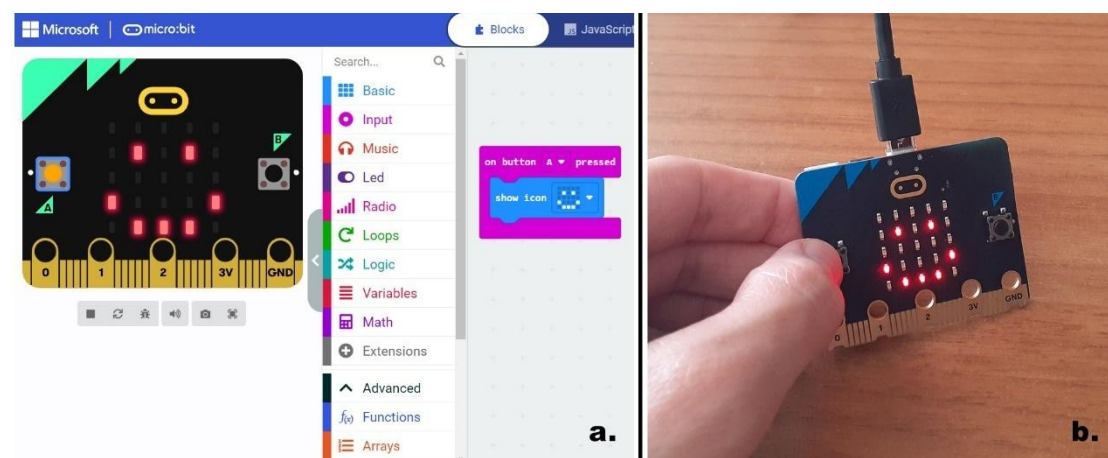
Rysunek 21: a. Krok, aby podłączyć micro:bit do komputera; b. Instrukcje, jak sparować micro:bit z przeglądarką.

Alternatywnie, jeśli połączenie micro:bit nie powiedzie się, możesz pobrać skrypt lokalnie na swój komputer jako plik .hex, klikając przycisk „...” (3) lub klikając przycisk „Download as File” (4) (Rysunek 22).



Rysunek 22: Pobieranie projektu jako plik o rozszerzeniu .hex .

Aby przetestować swój projekt lub kod/skrypt, możesz skorzystać z symulatora micro:bit (rysunek 23a) lub pobrać go na płytkę micro:bit (rysunek 23b).



Rysunek 23: a. Testowanie projektu na symulatorze; b. Testowanie projektu na micro:bit

4. Jak używać micro:bit do wprowadzania IoT

4.1 Komunikacja poprzez Bluetooth i Radio

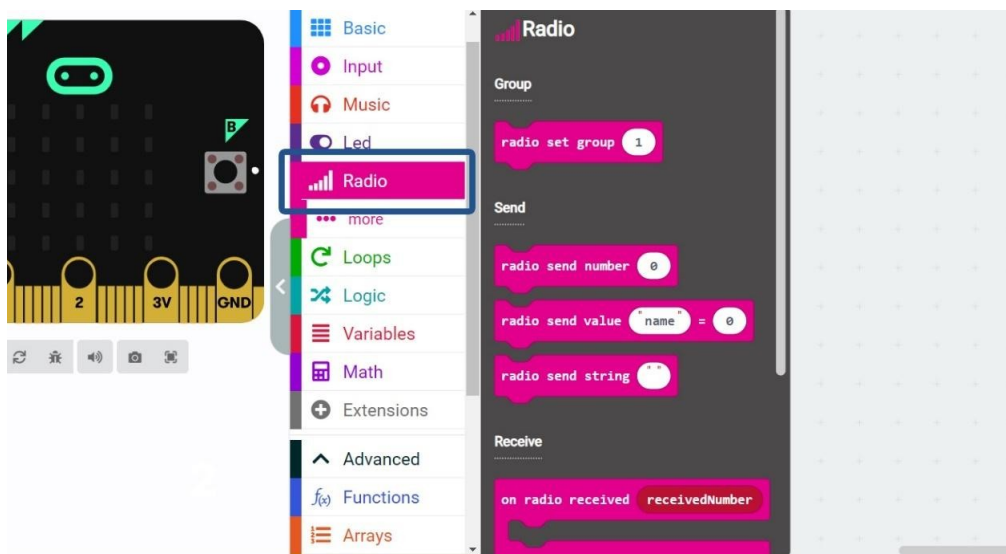
Jak wspomniano, płytkę BBC micro:bit ma bardzo niskonapięciową antenę Bluetooth Radio [Rysunek 4, (12)], która umożliwia micro:bit bezprzewodową komunikację z innymi micro:bitami lub innymi urządzeniami. Ten Bluetooth to to samo radio, którego smartfon używa do komunikacji z innymi urządzeniami bezprzewodowymi (takimi jak zestawy głośnomówiące lub smartwatche). Ta funkcja jest cenna nie tylko do wprowadzania IoT do uczniów, ale także do tworzenia szeregu projektów IoT.

Obecnie micro:bit można zaprogramować tylko do wysyłania lub odbierania wiadomości tekstowych i/lub wartości liczbowych do innych urządzeń. Można to zrobić za pomocą dwóch głównych grup poleceń: a. grupy poleceń radiowych i b. grupy poleceń Bluetooth.

Oto niektóre z kluczowych bloków poleceń z obu grup:

4.1.a. Grupa komend Radio

Korzystając z bloków zawartych w grupie poleceń Radio, możesz wprowadzić swoich uczniów w IoT i ideę komunikacji danych między dwoma lub większą liczbą urządzeń. Kliknij grupę poleceń Radio, aby zapoznać się z dostępnymi blokami poleceń (Rysunek 24).



Rysunek 24: Kliknięcie na grupę komend Radio

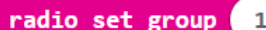
Na głównej karcie grupy Radio dostępnych jest 8 bloków poleceń, natomiast na karcie „... more” znajdują się 4 kolejne bloki poleceń.

Bloki hat, które są najczęściej używane w projektach to OnReceivedNumber i OnReceivedString (Rysunek 25). Te bloki hat wykonają część skryptu/kodu, gdy micro:bit otrzyma liczbę lub ciąg przez radio.



Rysunek 25: Bloki typu hat - OnReceivedNumber i OnReceivedString.

Blokiem, który powinien być używany we wszystkich projektach, jest blok poleceń „radio set group...” (Rysunek 26). Ten blok jest niezbędny, ponieważ ustawia kanał (lub ID) radia. Micro:bity na tym samym kanale mogą „rozmawiać” ze sobą i wymieniać informacje. Jeśli nie ustawisz ID, micro:bit losowo wybierze kanał, co prawdopodobnie spowoduje niepowodzenie projektu. Numeracja kanałów/ID mieści się w zakresie od 1 do 255.



Rysunek 26: Komenda radio set group.

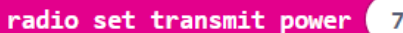
Inne bloki zablokowane często wykorzystywane w projektach to bloki poleceń „radio send number ...” i „radio send string ...” (Rysunek 27), które są używane do nadawania liczb i ciągów znaków odpowiednio do innych micro:bits podłączonych do tego samego identyfikatora/kanału.

Uwaga: pamiętaj, że ciąg znaków może mieć długość do 19 znaków




Rysunek 27: Bloki poleceń: radio send number i radio send string.

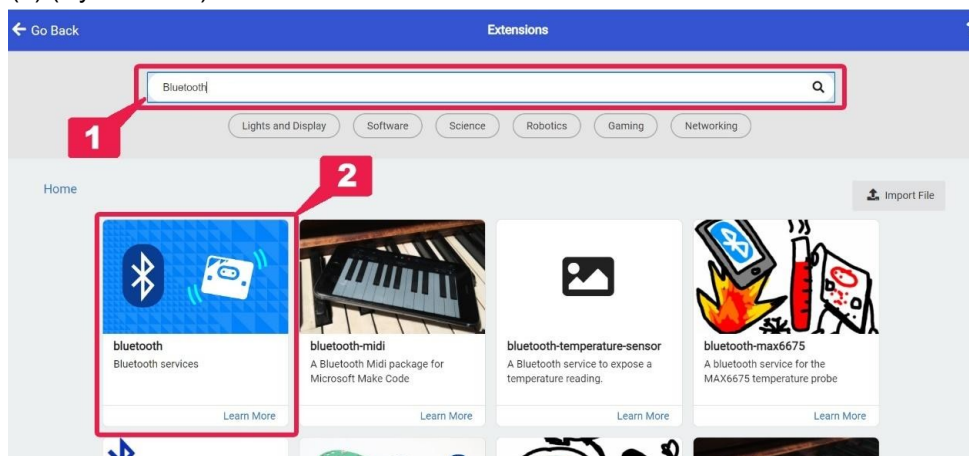
Innym ciekawym blokiem (znajdującym się na karcie „...more”) jest blok polecenia „radio set transmit power...” (Rysunek 28). Ten blok określa, jak słaby lub jak silny jest sygnał radiowy micro:bit, w skali od 0 do 7. Możesz użyć tego bloku, aby rozpocząć dyskusję w klasie na temat siły sygnału i tego, jak odnosi się ona do odległości, na jaką dwa (lub więcej) urządzenia mogą się komunikować.



Rysunek 28: Blok polecenia radio set transmit power

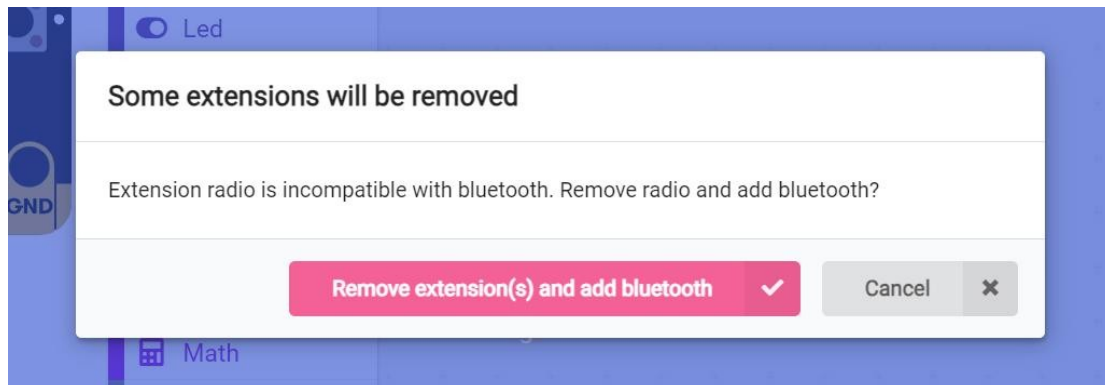
4.1.b. Blok poleceń Bluetooth

Aby tworzyć projekty, w których micro:bit wymienia dane z innymi urządzeniami Bluetooth, należy użyć bloków zawartych w grupie poleceń Bluetooth, oferujących dodatkowe możliwości. Ta grupa nie jest zawarta w głównych grupach poleceń i powinna zostać zaimportowana za pośrednictwem grupy „extensions”. Dlatego kliknij „Extensions” i wpisz „Bluetooth” w pasku wyszukiwania (1) (Rysunek 29). Następnie wybierz pierwszy zwrócony wynik (2) (Rysunek 29).



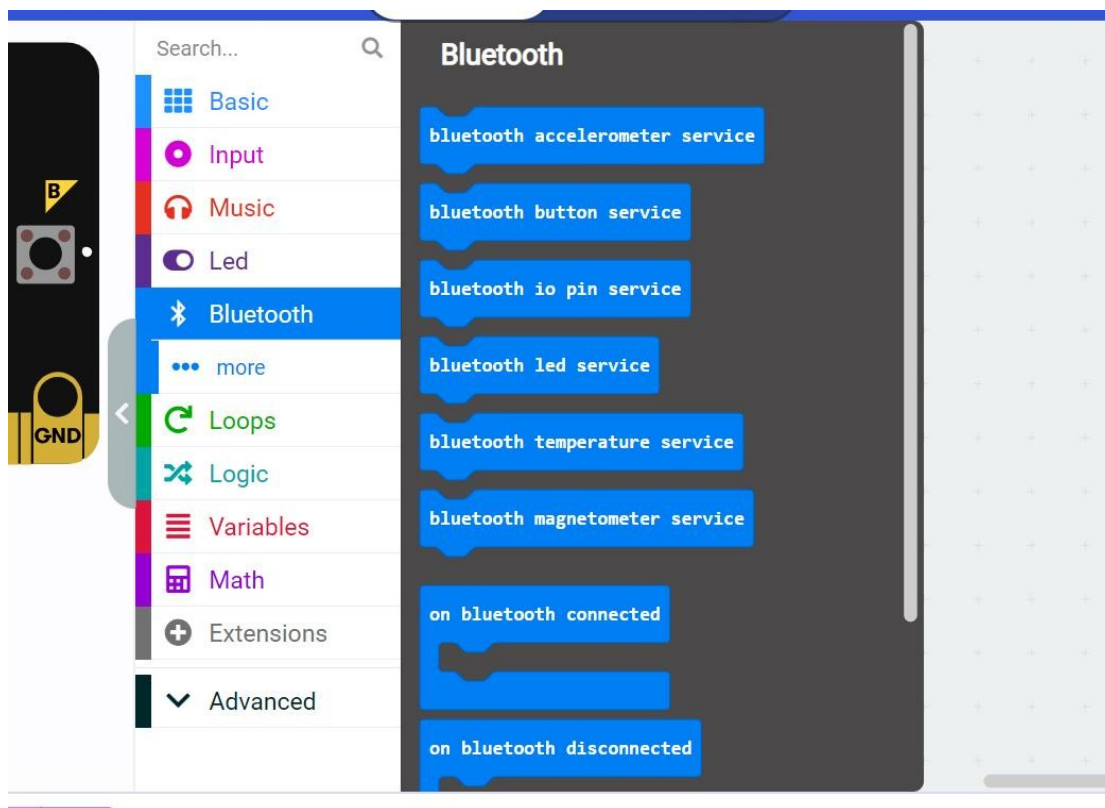
Rysunek 29: Szukanie rozszerzenia Bluetooth

Otrzymaś wiadomość informującą, że radio jest niezgodne z Bluetooth i powinno zostać usunięte. Kliknij „Remove extension(s) and add Bluetooth” (Rysunek 30), aby załadować grupę poleceń Bluetooth do swojego projektu.



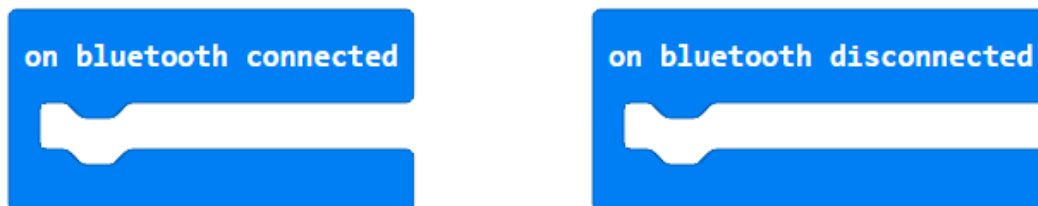
Rysunek 30: Ostrzeżenie i usuwanie rozszerzenia Radio.

Kliknij grupę poleceń Bluetooth, aby wyświetlić dostępne bloki poleceń (rysunek 31).



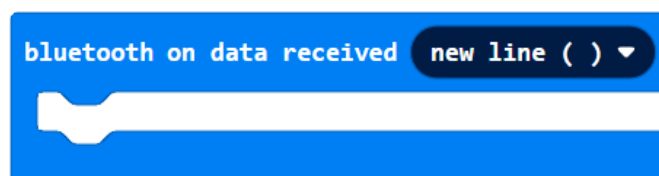
Rysunek 31: Eksplorowanie dostępnych bloków poleceń Bluetooth

Dwa z najbardziej przydatnych bloków typu hat to OnBluetoothConnected i OnBluetoothDisconnected (rysunek 32). Pierwszy z nich wykona część skryptu/kodu, gdy micro:bit zostanie podłączony do innego urządzenia przez Bluetooth. Drugi wykona obsługę zdarzeń, gdy micro:bit podłączony do innego urządzenia przez Bluetooth zostanie rozłączony.



Rysunek 32: Bloki typu hat „on Bluetooth connected” i „on Bluetooth disconnected”

Innym ważnym blokiem typu hat jest „Bluetooth on data received...” (Rysunek 33). Gdy znak ograniczający zostanie dopasowany do odebranych danych, ten blok wykonuje część kodu. Ograniczniki to znaki w odebranym ciągu danych, które dzielą ciąg na mniejsze ciągi, aby utworzyć oddzielne elementy danych. Niektóre ograniczniki to przecinki (,), hashtagi (#), dwukropki (:) i średniki (;).



Rysunek 33: Blok Bluetooth on data received ...

Blokiem, który powinien być używany we wszystkich projektach, jest blok poleceń „Bluetooth UART service” (rysunek 34). Ten blok jest niezbędny, ponieważ pozwala innemu urządzeniu, takiemu jak smartfon, na wymianę danych z micro:bit. UART to skrót od Universal Asynchronous Receiver Transmitter i jest jednym ze sposobów nawiązywania komunikacji danych szeregowych.



Rysunek 34: Blok Bluetooth uart service

Bloki poleceń „Bluetooth uart write number” i „Bluetooth uart write string” (rysunek 35) umożliwiają innemu urządzeniu Bluetooth wymianę danych liczbowych lub ciągów znaków z micro:bit, tylko jeśli usługa uart jest uruchomiona.



Rysunek 35: Bloki poleceń „Bluetooth uart write number” i „Bluetooth uart write string”.

Dostępne są również inne usługi, takie jak pomiar temperatury i magnetometr (rysunek 36), dzięki którym micro:bit może udostępniać dane dotyczące tych konkretnych parametrów innym urządzeniom Bluetooth.

bluetooth temperature service

bluetooth magnetometer service

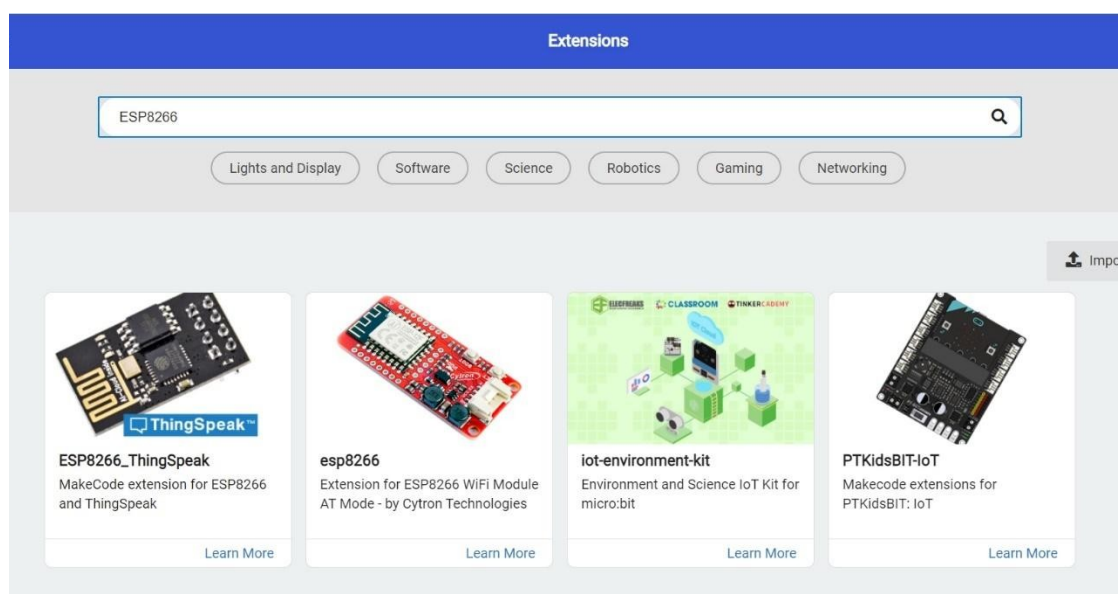
Rysunek 36: Bloki poleceń "Bluetooth temperature service" i "Bluetooth magnetometer service" .

Uwaga: Aby dowiedzieć się więcej o wszystkich dostępnych blokach poleceń Bluetooth, odwiedź ten link (<https://makecode.microbit.org/reference/bluetooth>).

4.2 Komunikacja WiFi poprzez moduł WiFi ESP8266

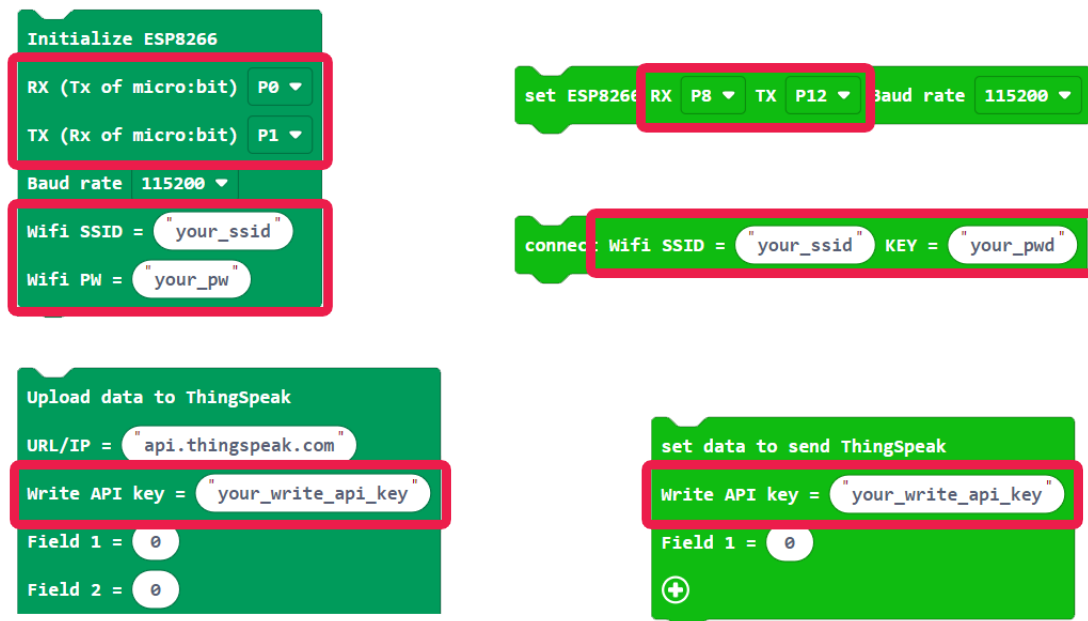
Oprócz radia Bluetooth, micro:bit może wymieniać dane z innymi urządzeniami przez WiFi, używając modułu WiFi ESP8266. Poprzez ESP8266, micro:bit może komunikować się z Internetem, wysyłać i odbierać dane oraz wchodzić w interakcje z usługami internetowymi i interfejsami API.

ESP8266 można znaleźć jako oddzielny moduł lub zintegrować z wyprowadzonymi złączami krawędziowymi, takimi jak IoT Bit. Aby zaprogramować ESP8266, należy wyszukać odpowiednie rozszerzenie w menu Rozszerzenia (Rysunek 37).



Rysunek 37: Różne rozszerzenia do programowania ESP8266

W zależności od wybranego rozszerzenia dostępne bloki poleceń mogą się różnić. Jednak we wszystkich przypadkach istnieje kilka wspólnych pól, które należy wypełnić. Są to „Wifi SSID”, czyli nazwa sieci Wi-Fi, „Wifi PW”, czyli hasło do sieci Wi-Fi, oraz „Write API key”, który można znaleźć w kanale utworzonym na platformie ThingSpeak1 (rysunek 38). Konieczne będzie również ustawienie pinów RX i TX, aby umożliwić komunikację szeregową między micro:bit a modulem WiFi (rysunek 38).



Rysunek 38: Bloki zlokalizowane w różnych rozszerzeniach i wymagane pola wspólne

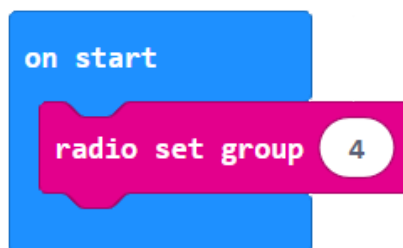
5. Ćwiczenia orientacyjne mające na celu zapoznanie się z micro:bit i komunikacją radiową

W tej sekcji zawarto szereg ćwiczeń mających na celu zaznajomienie się z płytką micro:bit i komunikacją radiową.

5.1 Przesyłanie uśmiechniętej buźki

Celem tej aktywności jest wysłanie uśmiechniętej buźki do innego microbita, gdy zostanie naciśnięty przycisk A.

Pierwszym krokiem jest ustawienie kanału (lub ID), na którym będzie nadawać Twój micro:bit. Z grupy poleceń „Basic” przeciągnij blok „on start” do obszaru kodu. Następnie z grupy poleceń „Radio” przeciągnij polecenie „radio set group...” i umieść je wewnątrz bloku „on start”. Ustaw kanał na dowolną liczbę od 1 do 255 (4 w poniższym przykładzie).

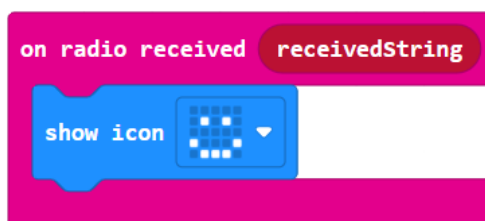


Następnym krokiem jest zaprogramowanie przycisku A tak, aby wysyłał wiadomość po naciśnięciu. Z grupy poleceń „Input” przeciągnij blok „on button A pressing” do obszaru kodu. Następnie z grupy poleceń „Radio” przeciągnij polecenie „radio send string...” i umieść je wewnątrz bloku „on button A pressing”. Następnie napisz coś znaczącego, na przykład „smile”.

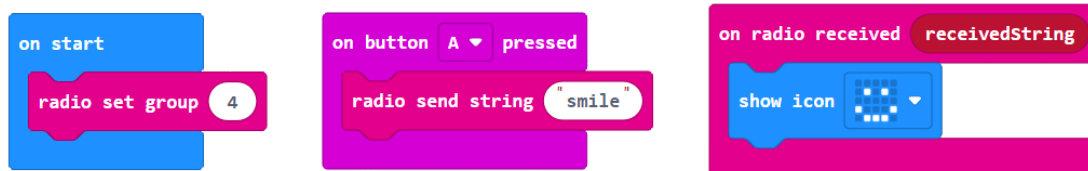
Ta część kodu umożliwi Ci wysłanie wiadomości „uśmiech” po naciśnięciu przycisku A.



Ostatnim krokiem jest zaprogramowanie tego, co się stanie, gdy inny micro:bit otrzyma wiadomość „smile”. Z grupy poleceń „Radio” przeciągnij blok kapelusza „on radio received receivedString” do obszaru kodu. Następnie z grupy poleceń „Basic” przeciągnij polecenie „show icon...” i umieść je wewnątrz bloku kapelusza „on radio received...”. Następnie z menu rozwijanego wybierz ikonę uśmiechniętej buźki.

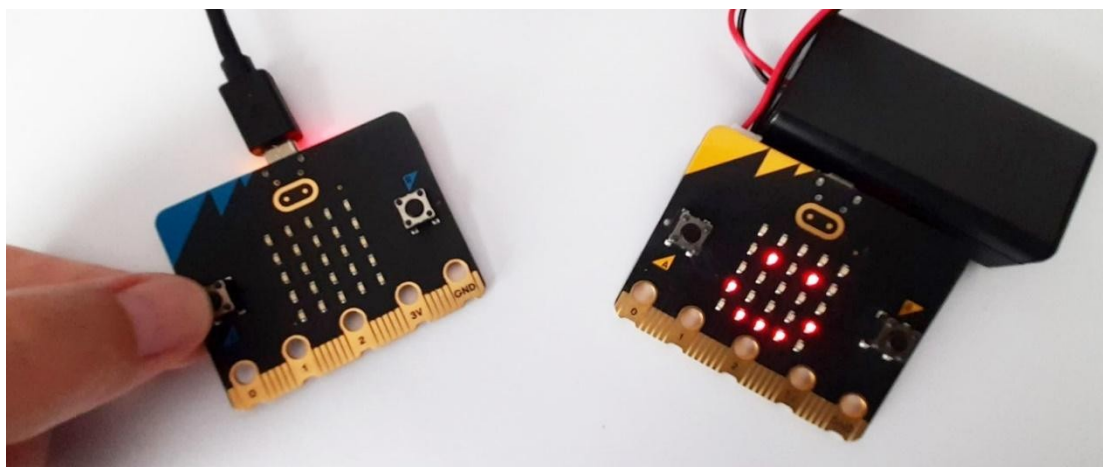


Pobierz cały kod (rysunek 39) do dwóch micro:bits i obserwuj wynik.

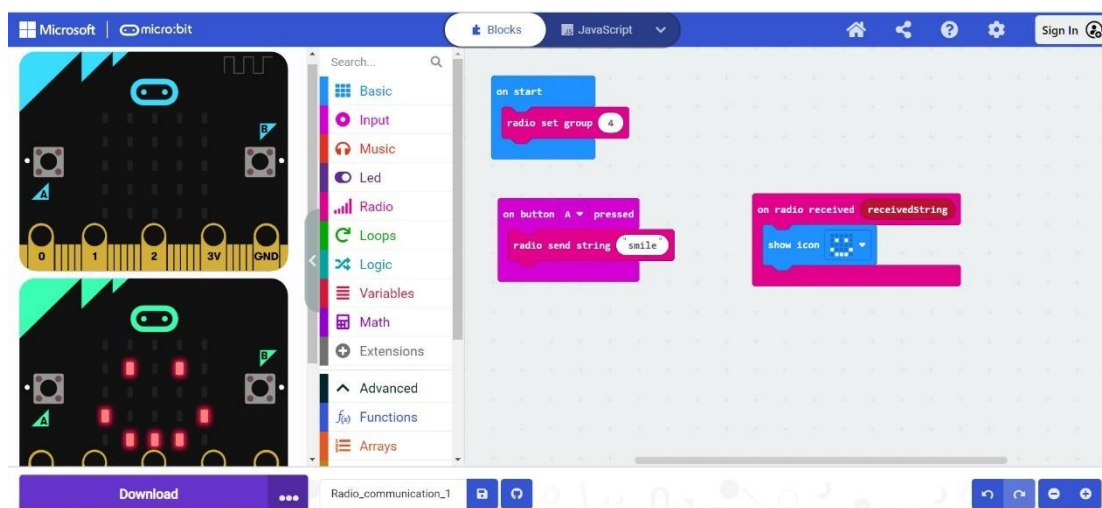


Rysunek 39: Cały kod do ćwiczenia nr 1.

Aby przetestować kod można skorzystać z dwóch micro:bits (rysunek 40) lub symulatora Makecode (rysunek 41).

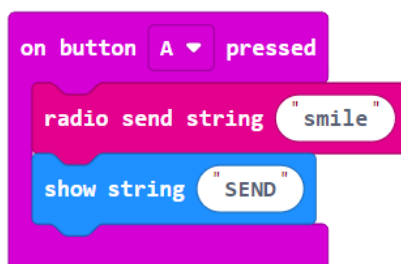


Rysunek 40: Wysłanie uśmiechniętej buźki do żółtego mikrobíta poprzez naciśnięcie przycisku A na niebieskim mikrobíta



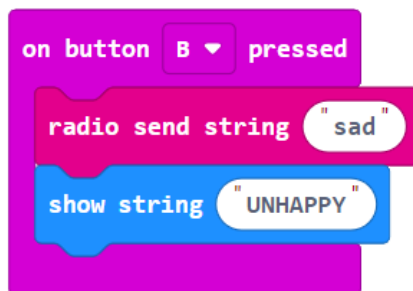
Rysunek 41: Testowanie kodu na symulatorze Makecode

Wskazówka: Możesz również dodać polecenie „show string...” wewnątrz bloku „on button A pressing”, aby uczynić komunikację bardziej żywą/aktywną. W takim przypadku micro:bit wysyłający wiadomość wyświetli wstawiony ciąg (SEND w przykładzie), dzięki czemu interakcja stanie się bardziej intuicyjna.



5.2 Wysyłanie uśmiechniętej i smutnej buźki

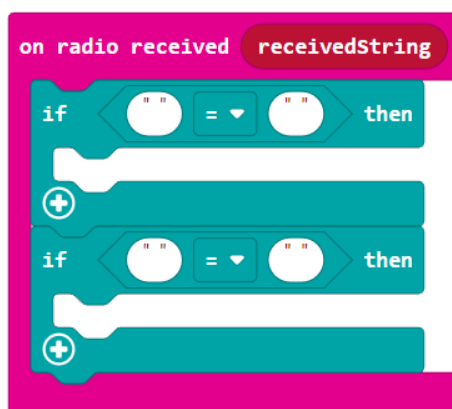
Te ćwiczenie jest rozszerzoną wersją pierwszego ćwiczenia. Celem tego drugiego ćwiczenia jest rozbudowa poprzedniego kodu i zaprogramowanie przycisku B tak, aby po naciśnięciu wysyłał smutną buźkę. Dodaj kolejny blok typu hat „on button A pressing” i wybierz „B” z jego rozwijanego menu. Wewnątrz tego nowego bloku typu hat dodaj polecenie „radio send string...” i napisz coś znaczącego, na przykład „sad”. Dodaj również polecenie „show string...” i wpisz „unhappy”.



Następnym krokiem jest zaprogramowanie micro:bit, który odbiera wiadomości, aby wyświetlał uśmiechniętą buźkę, gdy otrzyma wiadomość „smile” i smutną buźkę, gdy otrzyma wiadomość „sad”. Aby to zrobić, musimy użyć dwóch poleceń „if..then” z grupy poleceń „Logic” i dwóch operatorów porównania ciągów (Rysunek 42) i umieścić je wewnątrz bloku typu hat „on radio received receivedString” (Rysunek 43).

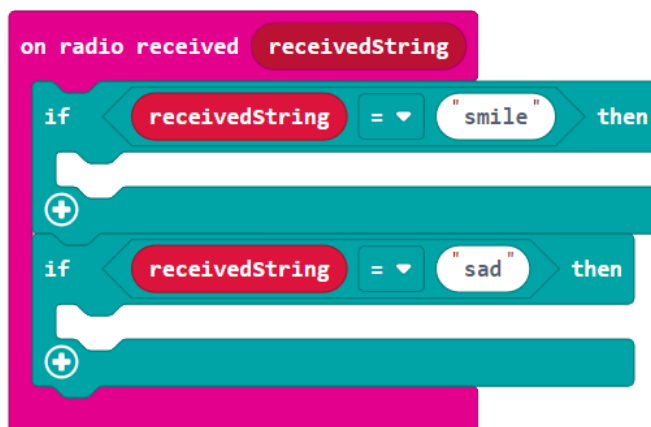


Rysunek 42: Polecenie logiczne „if...then” i operator porównania ciągów znaków



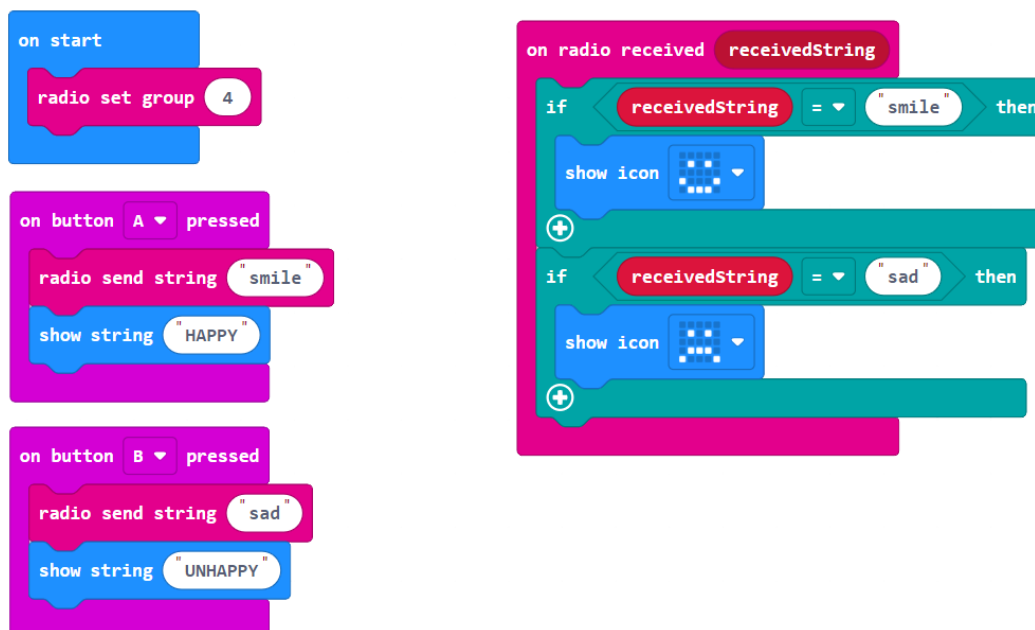
Rysunek 43: Blok typu hat „on radio received...” po dodaniu polecenia „if...then” .

Na podstawie otrzymanej wiadomości (uśmiech lub smutek) należy zaprogramować każde z poleceń „if...then”, aby wyświetlić odpowiadającą im ikonę (uśmiechniętą lub smutną buźkę). Wewnątrz każdego operatora porównania ciągów znaków należy dodać polecenie „receivedString” do lewej części równania i wpisać ciąg „uśmiech” lub „smutny” do prawej części równania.



Na koniec, wewnątrz każdego polecenia logicznego „if...then” należy umieścić odpowiadające mu polecenie „show icon”.

Po tym ostatnim kroku cały kod powinien wyglądać jak ten pokazany na Rysunku 44.



Rysunek 44: Cały kod do ćwiczenia nr 2.

Pobierz kod na dwa micro:bity lub skorzystaj z symulatora Makecode i obserwuj rezultaty.

5.3 Wysyłanie danych o temperaturze do innego micro:bit

Celem tego ćwiczenia jest pokazanie, jak można wysłać dane dotyczące zmierzonej temperatury do innego micro:bit, gdy naciśnięty zostanie przycisk A.

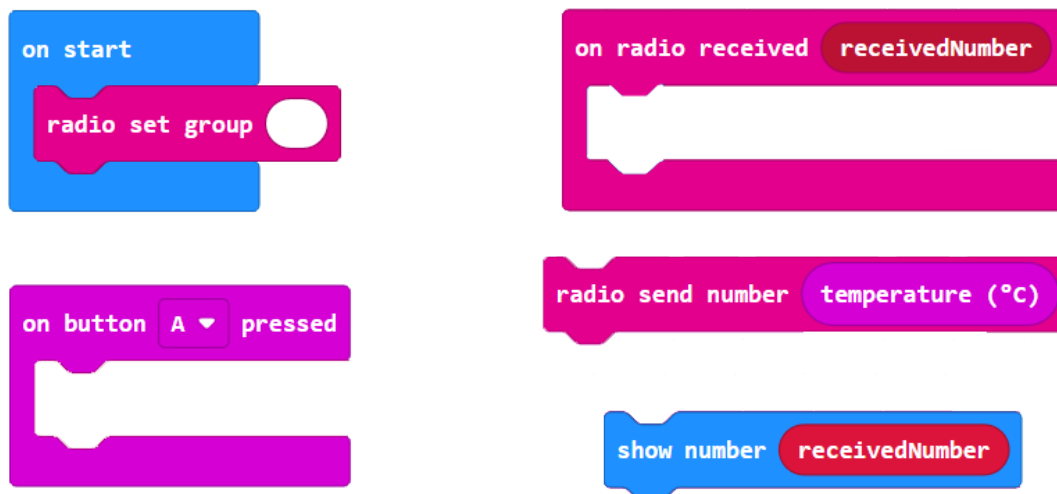
Najpierw należy ustawić kanał (lub ID), na którym będzie nadawał Twój micro:bit.

Następnie należy zaprogramować przycisk A, aby wysyłał wartości temperatury za pomocą radia.

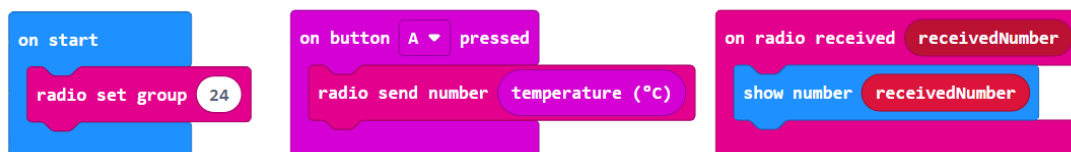
Na koniec należy zaprogramować inny micro:bit, aby wyświetlał temperaturę, gdy otrzyma odpowiadającą jej wartość.

Mając te informacje na uwadze, spróbuj złożyć następujący półustrukturyzowany kod.

Wskazówka: blok poleceń „temperature” znajduje się w grupie poleceń „Input”.



Przetestuj kod (który powinien wyglądać jak ten pokazany na rysunku 45, pobierając go do dwóch micro:bits lub korzystając z symulatora Makecode.

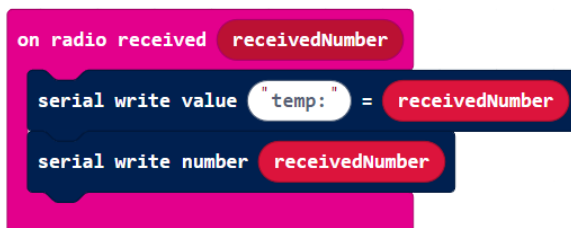


Rysunek 45: Całkowity kod do ćwiczenia nr 3.

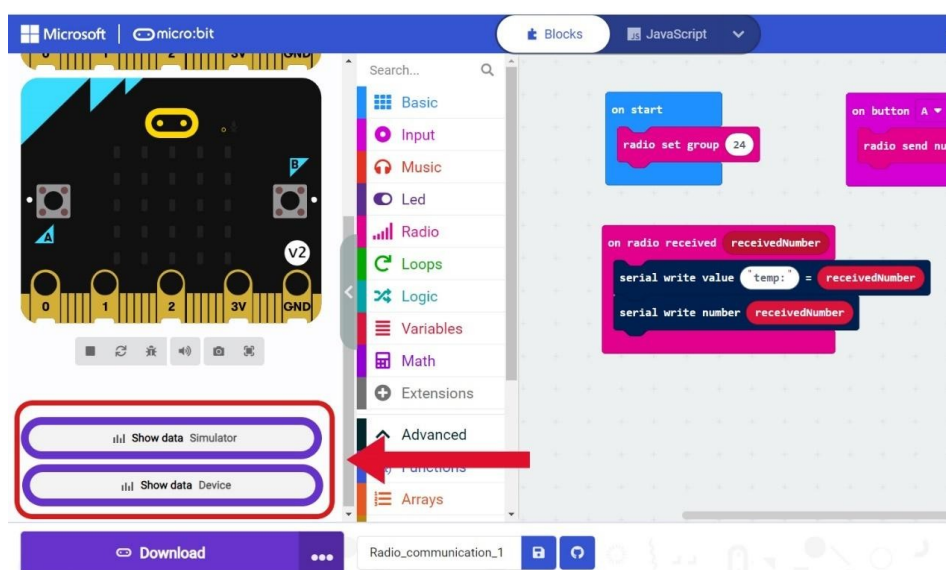
Wyświetlanie danych na porcie szeregowym Makecode

Ponadto możesz monitorować zarejestrowane dane na porcie szeregowym Makecode, używając niektórych poleceń blokowych z grupy poleceń „Serial”.

Wewnątrz bloku „on radio received receivedNumber” umieść bloki poleceń „serial value x = ...” i „serial write numbers ...”. Następnie wpisz „temp” w polu x i umieść „receivedNumber” w dwóch pozostałych polach.

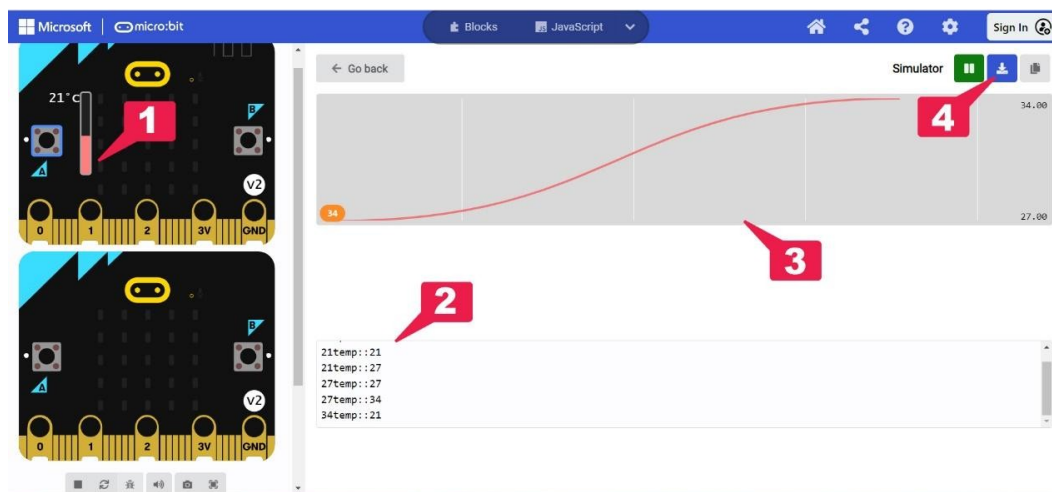


Przetestuj nowy kod, pobierając go do dwóch micro:bits lub używając symulatora. Możesz monitorować otrzymane dane, wybierając „Show data device” lub „Show data simulator” (Rysunek 46).



Rysunek 46: Wybieranie źródła danych, które chcesz obserwować.

Wskazówka: Wybierz „Show data simulator” (Pokaż symulator danych) i zmień temperaturę na termometrze symulatora (1), aby zaobserwować, jak różne dane mogą być monitorowane (2) i wyświetlane (3) w czasie na monitorze szeregowym (Rysunek 47). Możesz również kliknąć przycisk pobierania (4), aby pobrać zarejestrowane dane jako arkusz kalkulacyjny.



Rysunek 47: Port szeregowy w Makecode