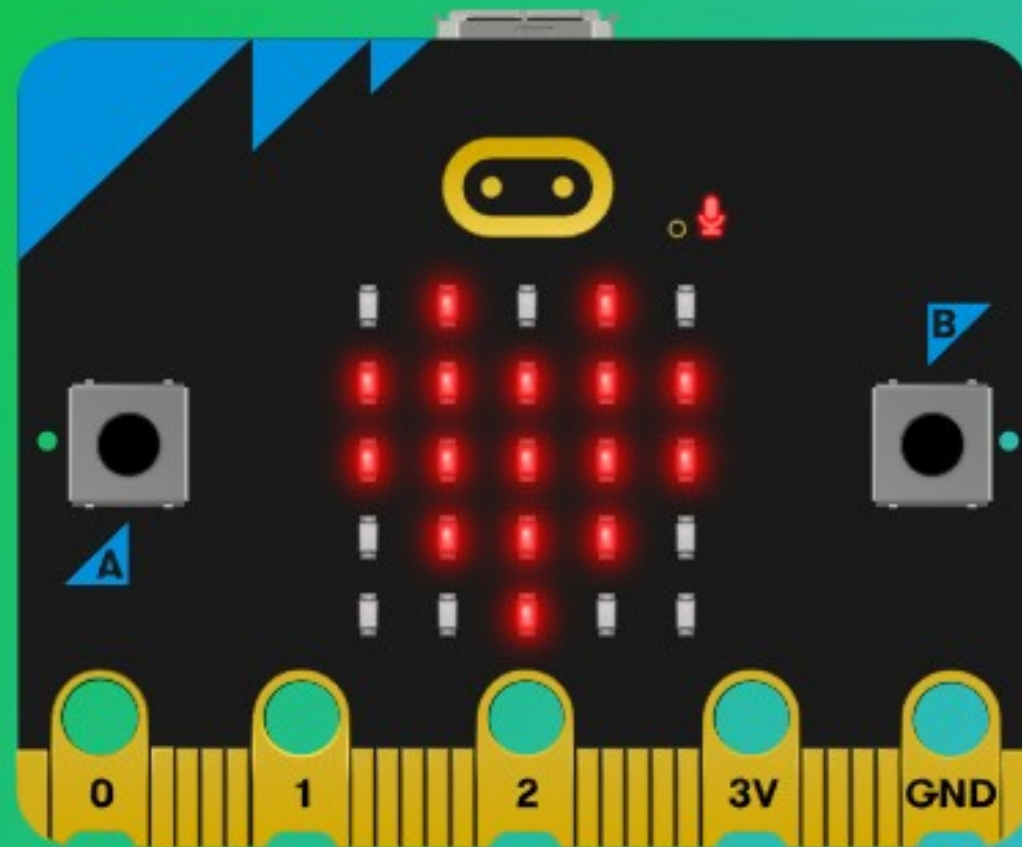


# BBC micro:bit v2



**Create**

**Learn**

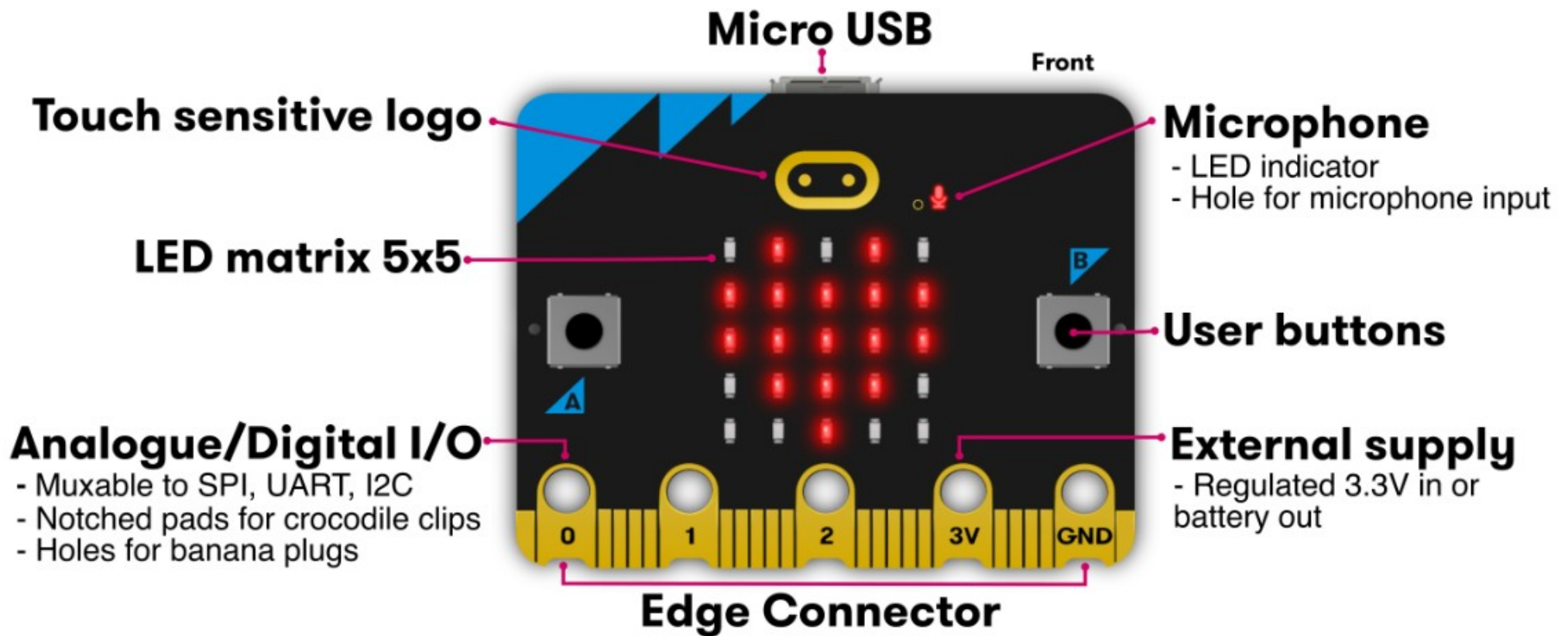
**Code**

# Felhasznált és ajánlott irodalom

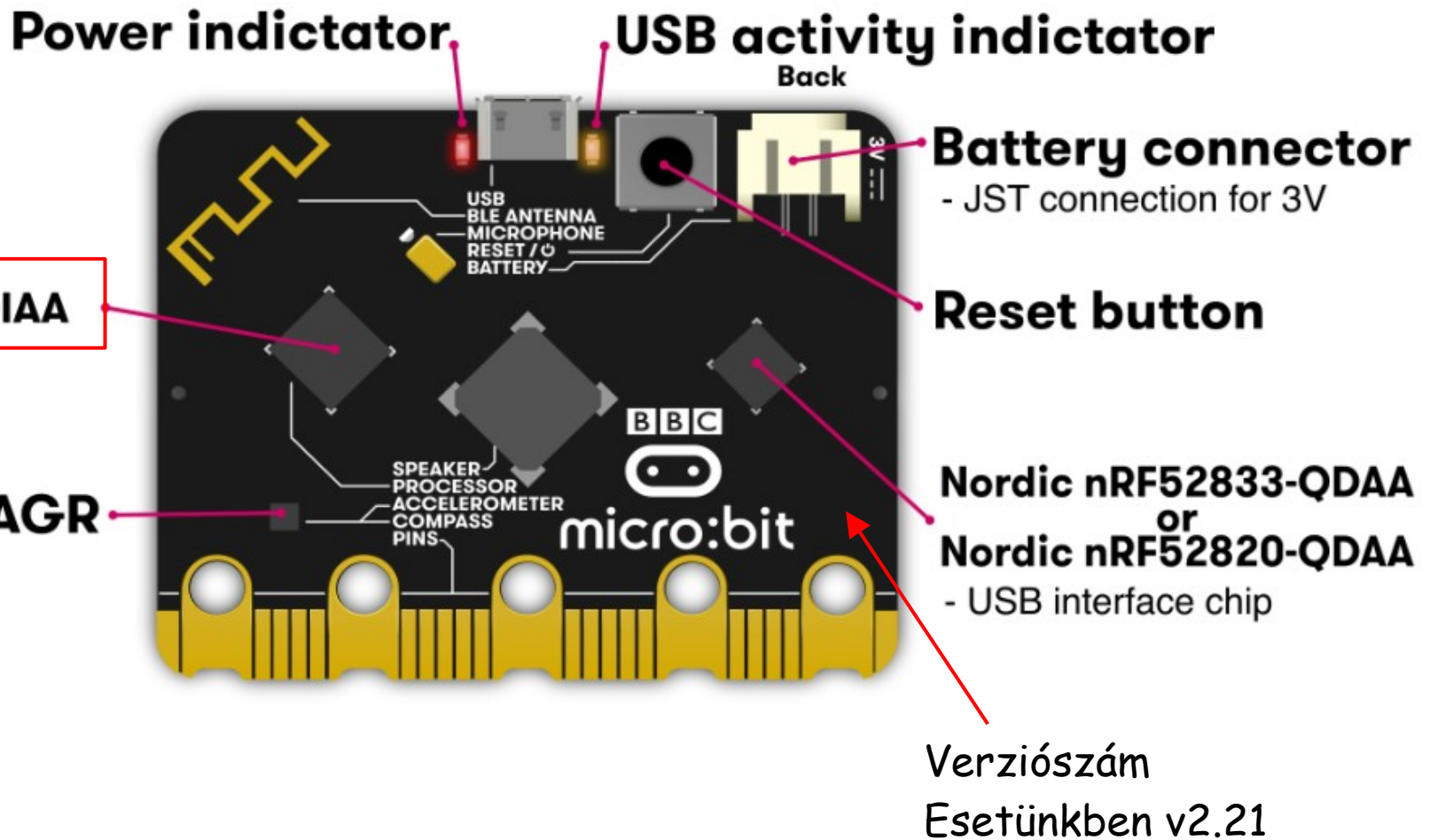
---

- Micro:bit botorkálás: [Segédanyagok](#)
- Dr. Abonyi-Tóth Andor: [Programozzuk micro:biteket!](#)
- Cytron Technologies: [EDU:BIT könyv](#)
- BBC Micro:bit: [Microbit.org](#)
- Micro:bit developer community: [Circuit Schematics](#)
- Microsoft MakeCode: [Documentation](#)
- BBC Micro:bit [Python Reference](#)
- BBC Micro:bit [MicroPython documentation](#)
- Codewith.mu: [Mu and micro:bit](#)

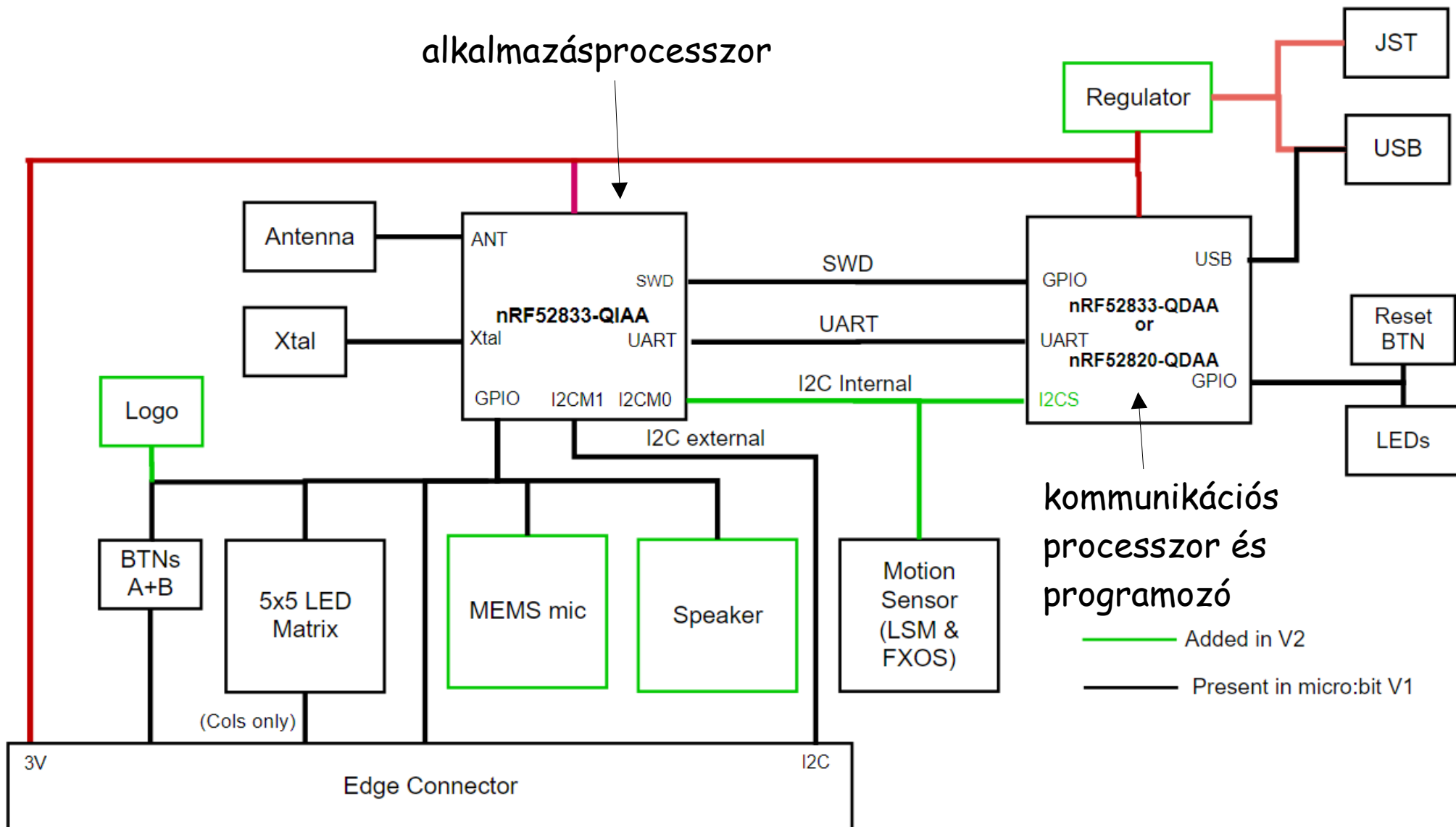
# A micro:bit v2 kártya



# A kártya hátoldala



# A hardver blokkvázlata



# Az nRF52 alkalmazásprocesszor

- A **Nordic nRF52 833** alkalmazásprocesszorban futnak a felhasználói programok. A felhasználói és a futtató kód, valamint a Bluetooth-csomag, a chipen lévő flash memóriába töltődik. Az összes, a felhasználó által elérhető GPIO kivezetést ez a processzor biztosítja. Van egy beépített 2,4 GHz-es rádió-periféria, amely Bluetooth és egyedi rádióképességek biztosítására szolgál a chipen kívüli nyomtatott áramköri antennán keresztül

item	details
Model	<a href="#">Nordic nRF52833</a>
Core variant	<a href="#">Arm Cortex-M4 32 bit processor with FPU</a>
Flash ROM	512KB
RAM	128KB
Speed	64MHz
Debug	<a href="#">SWD, J-Link/OB</a>
More Info	<a href="#">Software, nRF52 datasheet</a>

# A kijelző

- A kijelző egy 5x5-ös LED mátrix. A futtató szoftver nagy sebességgel frissíti ezt a mátrixot úgy, hogy a felhasználó nem észlel villogást. Ezt a LED-mátrixot a környezeti fény érzékelésére is használják, a LED-eket vezérlő kivezetések egy részét ismételten bemenetre kapcsolva, és mintavételezve a feszültségcsökkenési időt, ami nagyjából arányos a környezeti fény szintjével

item	details
Type	miniature surface mount red LED
Physical structure	5x5 matrix
Electrical structure	5x5
Intensity control	Software controlled up to 255 steps
Sensing	ambient light estimation via software algorithm
Sensing Range	TBC, 10 levels from off to full on
Colour sensitivity	red centric, red is 700nm

# USB kapcsolat

- Az USB kapcsolatot a kommunikációs processzor biztosítja, ami lehetővé teszi, hogy a fájlokat húzzunk át a **MICROBIT** meghajtóra, vagy a soros adatok streamelését az alkalmazásprocesszorba, illetve és támogatja a **CMSIS-DAP** specifikációt az alkalmazásprogramok hibakereséséhez

item	details
Connector	USB micro, MCR-B-S-RA-SMT-CS5-TR
USB version	2.0 Full Speed device
Speed	12Mbit/sec
USB classes supported	Mass Storage Class (MSC) Communications Device Class (CDC) CMSIS-DAP HID & WinUSB WebUSB CMSIS-DAP HID



# Mozgásérzékelő

- A mozgásérzékelő 3-tengelyű mágneses érzékelőt és egy 3-tengelyű gyorsulásmérőt tartalmaz (ez 2/4/6/16g érzékenységi fokozatban)
- A *szabadesés* gesztust a chip érzékeli, a többi gesztust (tilt, shake, stb.) szoftver algoritmusok állítják elő a szenzor adatokból

item	details
Model	<a href="#">LSM303AGR</a>
Features	3 magnetic field and 3 acceleration axes , 2/4/8/16g ranges
Resolution	8/10/12 bits
On board gestures	'freefall'
Other gestures	Other gestures are implemented by software algorithms in the runtime.

# MakeCode – szinte gyerekjáték!

- A **MakeCode** egy ingyenes online programozási és szimulációs platform, ahol bárki könnyen készíthet programot a **micro:bit** kártyához

Microsoft | micro:bit

Blocks JavaScript

Search...

- Basic
- Input
- Music
- Led
- Radio
- Loops
- Logic
- Variables
- Math
- Extensions
- Advanced

on start

serial redirect to USB

forever

set tc to 0

set comp to 0

for index from 0 to 99

do

change comp by compass heading (°)

change tc by temperature (°C)

serial write value "Temp" = tc / 100

serial write value "Compass heading" = comp / 100

pause (ms) 2000

Két módban használhatjuk

Show data Simulator

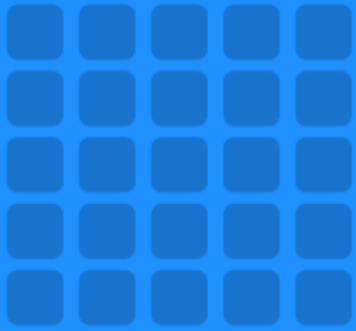
# Basic blokkok

## Basic

show number

0

show leds



show icon



show string

"Hello!"

pause (ms)

100

clear screen

show arrow

North

# LED blokkok

**Led**

- plot x 0 y 0
- toggle x 0 y 0
- unplot x 0 y 0
- point x 0 y 0
- plot bar graph of 0 up to 0

+

**more**

- plot x 0 y 0 brightness 255
- point x 0 y 0 brightness
- brightness
- set brightness 255
- led enable false
- stop animation
- set display mode black and white

# Input blokkok

A screenshot of the Scratch 'Input' category. It displays several event and sensor blocks:

- on button **A** ▼ pressed
- on **shake** ▼
- on pin **P0** ▼ pressed
- button **A** ▼ is pressed
- acceleration (mg) **x** ▼
- pin **P0** ▼ is pressed
- light level
- compass heading (°)
- temperature (°C)
- is **shake** ▼ gesture

A screenshot of the Scratch 'micro:bit (V2)' category. It displays several event and sensor blocks:

- on **loud** ▼ sound
- on logo **pressed** ▼
- logo is pressed
- sound level

A screenshot of the Scratch 'more' category. It displays several sensor and control blocks:

- calibrate compass
- magnetic force ( $\mu\text{T}$ ) **x** ▼
- rotation (°) **pitch** ▼
- running time (ms)
- running time (micros)
- on pin **P0** ▼ released
- set accelerometer range **1g** ▼
- micro:bit (V2)
- set **loud** ▼ sound threshold to **128**

# Kő, papír, olló

## microbit-Rock-Paper-Scissors-V2.hex

- Az *on shake* esemény bekövetkeztekor
  - ❖ Generálunk egy véletlen számot (1, 2, vagy 3) és eltároljuk a *hand* nevű változóban
  - ❖ Megjelenítjük az adott számhoz tartozó ikont és lejátszunk egy hangeffektet
- A kártya megrázásakor véletlenszerűen megjelenik valamelyik ikon

The image displays three segments of Scratch code blocks for a micro:bit program. The first segment is an 'on shake' event block followed by a 'set hand to pick random 1 to 3' block. The second segment is an 'if hand = 1 then' block containing a 'show leds' block with a 4x4 grid of LEDs and a 'play sound' block with a square wave sound effect. The third segment is an 'else if hand = 2 then' block containing a 'show leds' block with a 4x4 grid of LEDs and a 'play sound' block with a sine wave sound effect. A fourth 'else' block is partially visible, containing a 'show leds' block with a 4x4 grid of LEDs and a 'play sound' block with a sawtooth wave sound effect.

# Python Editor

- A micro:bit Python Editor szintén egy online, fejlett és könnyen használható kódfejlesztő környezet és szimulátor

The screenshot displays the micro:bit Python Editor web interface. The browser address bar shows the URL <https://python.microbit.org/v/3>. The interface is divided into three main sections:

























- Left Sidebar (Project Ideas):** Contains several project categories with icons and brief descriptions:
  - Data logging V2:** Log sensor data on your...
  - Pins:** Expand your micro:bit
  - NeoPixels:** Colourful LED strips
  - Data types:** Ways of storing different data...
  - String manipulation:** Working with text in Python
  - Text input and output:** Use your computer's keyboard...
  - Advanced maths:** Add more maths functions to...
- Center (Code Editor):** Titled "Untitled project", it contains the following Python code:

```
1 # Imports go at the top
2 from microbit import *
3
4
5 # Code in a 'while True:' loop repeats forever
6 while True:
7     display.show(Image.HEART)
8     sleep(1000)
9     display.scroll('Hello')
10
```
- Right (Simulation):** Shows a virtual micro:bit board with a yellow heart on the LED matrix and the text "Hello" scrolling across the display. Below the board are control buttons for running, pausing, and refreshing. At the bottom right, there is a "Show serial" panel with a dropdown menu set to "shake" and several sliders for sensor settings.

At the bottom of the editor, there are three buttons: "Send to micro:bit", "Save", and "Open..."

# Python támogatói könyvtárak

- A beépített könyvtárak széleskörű támogatást nyújtanak az I/O eszközök és a szenzorok kezeléséhez: „*van itt minden, mint a búcsúban*”

 <b>Variables</b> → Keep track of data that changes	 <b>Comments</b> → Explain your Python code	 <b>Temperature</b> → Measure your micro:bit's...	 <b>Pins</b> → Expand your micro:bit
 <b>Display</b> → The micro:bit's LED display...	 <b>Maths</b> → Basic maths in Python	 <b>Compass</b> → Measure direction and streng...	 <b>NeoPixels</b> → Colourful LED strips
 <b>Buttons</b> → Use button inputs in your code	 <b>Lists (arrays)</b> → Organise data in lists	 <b>Sound</b> → Play music, speech and other...	 <b>Data types</b> → Ways of storing different data...
 <b>Loops</b> → Count and repeat sets of...	 <b>Functions</b> → Make your code more efficient	 <b>Microphone V2</b> → Measure and react to sound...	 <b>String manipulation</b> → Working with text in Python
 <b>Logic</b> → Making decisions in code	 <b>Radio</b> → Send messages between...	 <b>Touch logo V2</b> → Use the gold logo as an extra...	 <b>Text input and output</b> → Use your computer's keyboard...
 <b>Accelerometer</b> → Detect gestures and movement	 <b>Light level</b> → Measure light falling on your...	 <b>Data logging V2</b> → Log sensor data on your...	 <b>Advanced maths</b> → Add more maths functions to...



# brightness\_setting.hex

- Amint látjuk, az előlapi LED-ek fényereje szabályozható
- A **MakeCode**-hoz hasonlóan a **.hex** fájlokból a forráskód visszatölthető

The screenshot displays the micro:bit Python Editor interface. The central code editor shows a Python script named 'brightness\_setting' with the following code:

```
1 # Imports go at the top
2 from microbit import *
3
4
5 # Code in a 'while True:' loop repeats forever
6 while True:
7     sleep(500)
8     display.show(Image('00300:'
9                         '03630:'
10                        '36963:'
11                        '03630:'
12                        '00300'))
13
14     sleep(500)
15     display.show(Image('96069:'
16                        '63036:'
17                        '33333:'
18                        '63036:'
19                        '96369'))
```

Below the code editor, the text "Letöltés a kártyára" (Download to the card) is displayed with a downward arrow pointing to the "Send to micro:bit" button. Other buttons visible include "Save" and "Open...".

The left sidebar shows the "Reference" section for the "Radio" module, with a code snippet: `import radio` and `radio.config(group=23)`. Below it, the "On and off" section shows a code snippet: `import radio` and `radio.on()`.

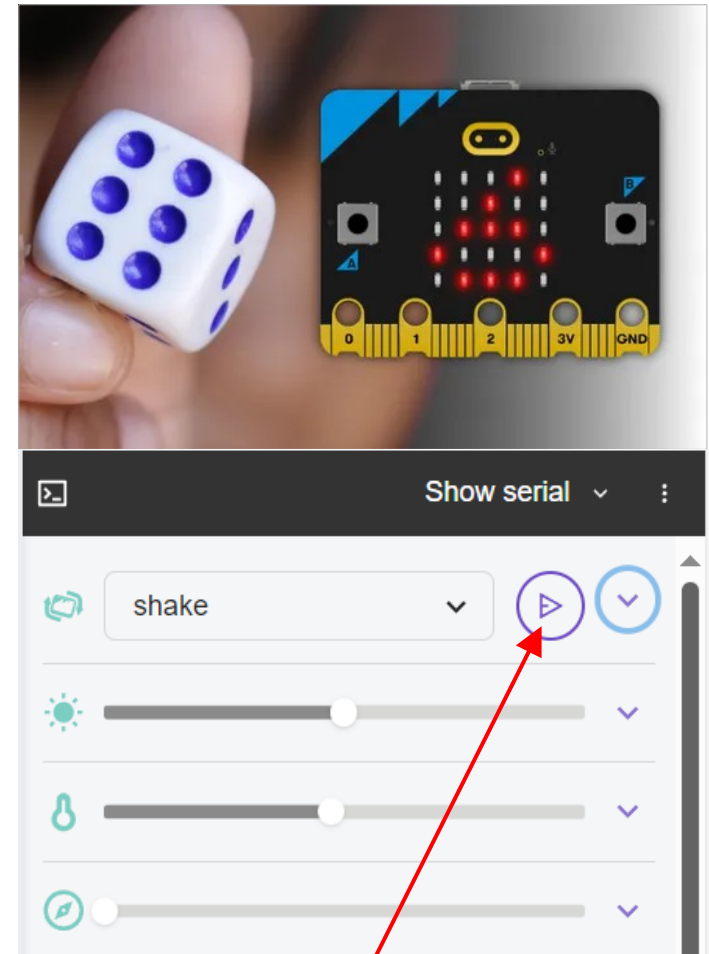
The right sidebar shows a virtual micro:bit board with a grid of LEDs. Below the board, there are control buttons for "shake", "show serial", and various sliders and buttons for controlling the board's functions.

# Dobókocka: dice.hex

```
from microbit import *
import random

while True:
    if accelerometer.was_gesture('shake'):
        display.show(random.randint(1, 6))
```

- A gyorsulásmérő jelzi, ha „megrázzuk” a kártyát (a szimulátorban ezt a nyíllal jelzett gombra kattintva érhetjük el)
- Minden „shake” esemény után generálunk egy véletlen számot és kiíratjuk a kijelzőn

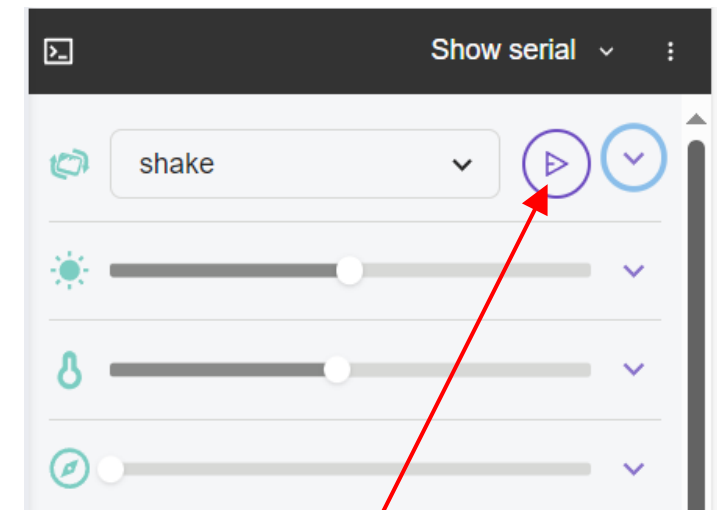
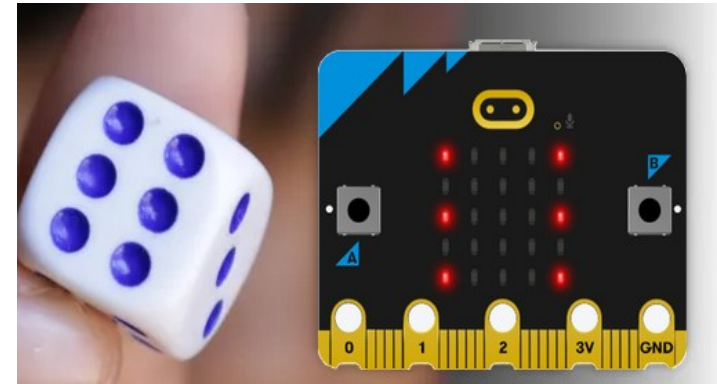


A szimulátorban ide kattintva válthatjuk ki az eseményt

# Dobókocka másképp: dice2.hex

```
from microbit import *
import random
myimage = [Image('00000: '
                 '00000: '
                 '00900: '
                 '00000: '
                 '00000'),
           . . .
           Image('90009: '
                 '00000: '
                 '90009: '
                 '00000: '
                 '90009')]

while True:
    if accelerometer.was_gesture('shake'):
        n = random.randint(1, 6)
        display.show(myimage[n-1])
```



A szimulátorban ide kattintva válthatjuk ki az eseményt

# Zene lejátszás: play\_music.hex

- Zene lejátszásánál a hangok ABC-s jelén kívül meg kell adni az oktávot (a zongorabillentyűzet bal széle az 1. oktáv) és legalább a kezdőhangnál az időtartamot (1/16 egységekben értve)

```
# Playing the Frère Jacques melody
from microbit import *
import music

for x in range(2):
    music.play(['C5:4', 'D5', 'E5', 'C5'])

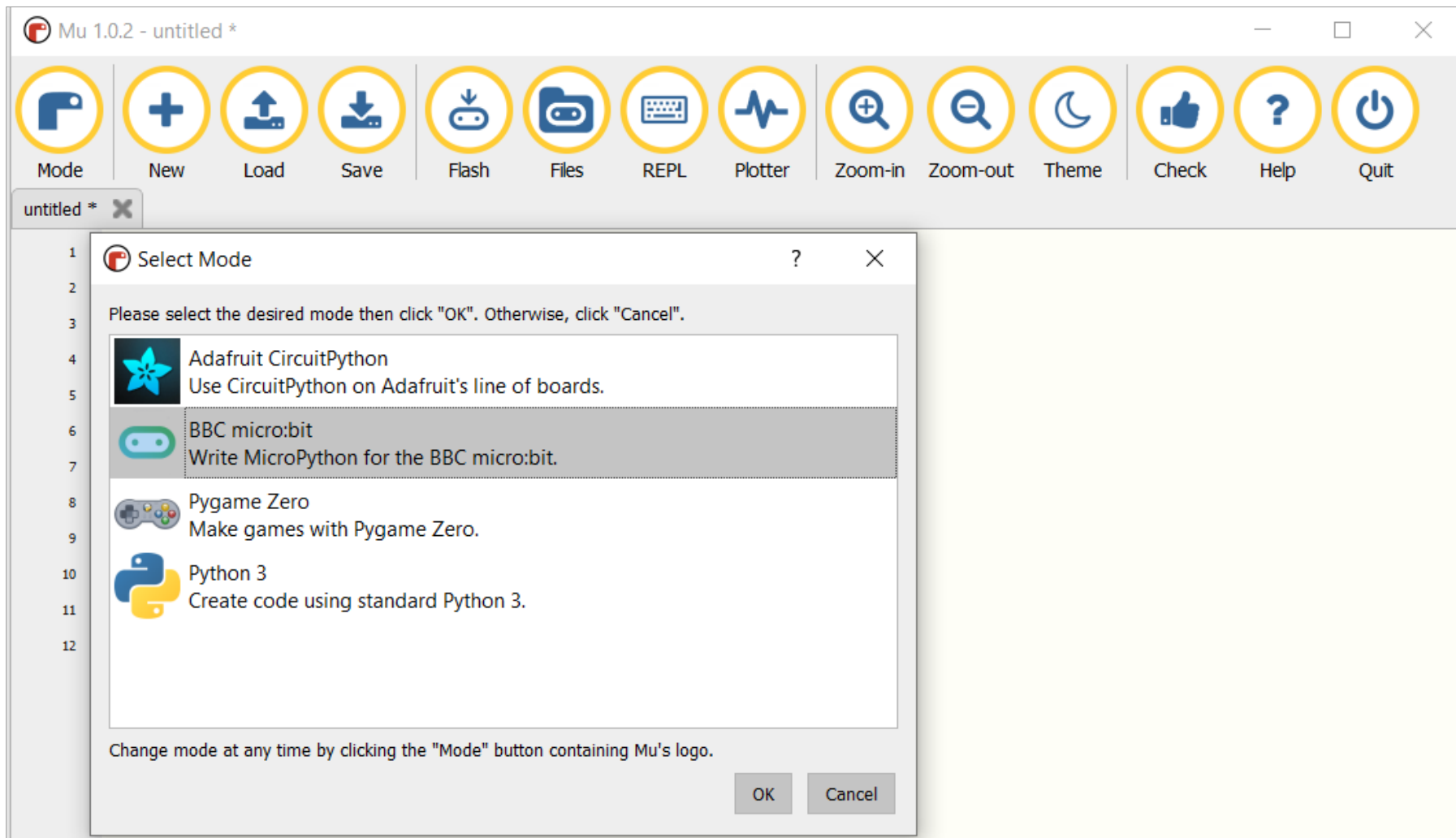
for x in range(2):
    music.play(['E5:4', 'F5', 'G5:8'])

for x in range(2):
    music.play(['G5:2', 'A5', 'G5', 'F5', 'E5:4', 'C5:4'])

for x in range(2):
    music.play(['D5:4', 'G4', 'C5:8'])
```

# Mu és a micro:bit kártya

- A Mu editor **Mode** gombjával **BBC micro:bit** üzemmódot is választhatunk. Ezután a menüben megjelenő **Flash** gomb a programletöltés, a **REPL** az interaktív **MicroPython** futtatás, a **Plotter** pedig a grafikonrajzoló.



# accel\_in\_mu.py

- Kiolvassuk és kiíratjuk a gyorsulásmérő adatait
- Az **A** nyomógommbal válthatunk, hogy csak az **x** komponens, vagy mindhárom komponens (**x**, **y**, **z**) legyen kiíratva
- Ahogy a **CircuitPython** előadásokban láttuk, a **Mu editor** plot funkciója **tuple** típusú adatokat vár, ezért egyetlen adatcsatorna esetén így kell megadni egy adatpontot: (**adat**, )

```
from microbit import *
flag = True # jelző

while True: # Végtelen ciklus
    sleep(20)
    if button_a.was_pressed(): # Ha A gombot lenyomtuk
        flag = not flag
    if flag: # Ha a jelző True
        print((accelerometer.get_x(), )) # Csak az x komponens kell
    else:
        print(accelerometer.get_values()) # Mindhárom komponens kell
```

# accel\_in\_mu.py futási eredmény

