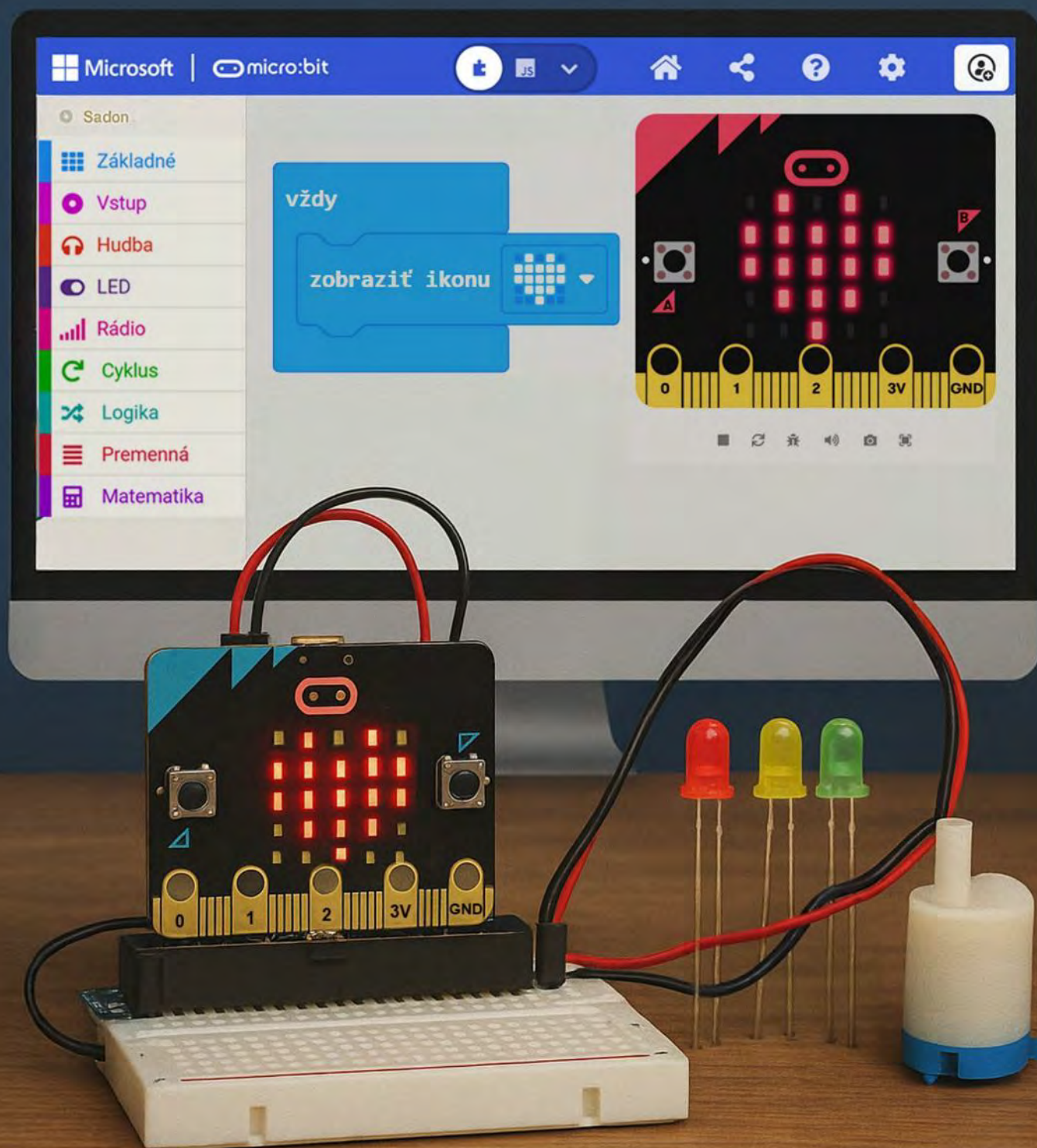


MIKROKONTROLÉRY V INTERDISCIPLINÁRNEJ EDUKÁCIÍ 1



Miroslav Šebo - Erik Krajínčák

UNIVERZITA KONŠTANTÍNA FILOZOFA V NITRE

**Mikrokontroléry
v interdisciplinárnej edukácii
1**

Nitra 2025

Názov: Mikrokontroléry v interdisciplinárnej edukácii 1

Autor: Mgr. Miroslav ŠEBO, PhD.
PaedDr. Erik Krajínčák, PhD.

Recenzent: prof. PaedDr. Jarmila Honzíková, Ph.D.
prof. PhDr.PaedDr. Jiří DOSTÁL, Ph.D.

Všetky práva vyhradené. Toto dielo ani žiadnu jeho časť nemožno reprodukovat' bez súhlasu majiteľov práv.

ISBN 978-80-558-2331-7

Obsah

Úvod.....	8
1 Makecode	11
2 Základné.....	16
3 Vstup	25
4 Hudba	45
5 LED	57
6 Rádio	69
7 Cyklus.....	81
8 Logika.....	88
9 Premenné.....	94
10 Matematika.....	99
11 Funkcie	114
12 Polia.....	120
13 Text.....	142
14 Kolíky.....	149
15 Sériové rozhranie v blokovom programovaní micro:bit	156
Záver.....	164
Literatúra	166

Zoznam obrázkov

Obrázok 1 Rozhranie MakeCode	13
Obrázok 2 Hlavné prostredie MakeCode	14
Obrázok 3 Príkaz „Vždy“	16
Obrázok 4 Príkaz „Zobraziť číslo“	17
Obrázok 5 Príkaz "zobraziť LED"	18
Obrázok 6 Príkaz "zobraziť Ikonu"	19
Obrázok 7 Príkaz "zobraziť reťazec"	20
Obrázok 8 Príkaz "vymazať obrazovku"	21
Obrázok 9 Príkaz "počas spustenia"	22
Obrázok 10 Príkaz "pozastaviť (ms)"	23
Obrázok 11 Príkaz "zobraziť šípku"	24
Obrázok 12 Príkaz "keď sa tlačidlo A stlačí"	25
Obrázok 13 Príkaz "keď potrasie"	26
Obrázok 14 Príkaz "on pin P0 pressed"	27
Obrázok 15 Príkaz "tlačidlo A je stlačené"	28
Obrázok 16 Príkaz "akcelerácia (mg) x"	29
Obrázok 17 Príkaz "kolík P0 je stlačený"	30
Obrázok 18 Príkaz "úroveň osvetlenia"	31
Obrázok 19 Príkaz "smerovanie kompasu"	32
Obrázok 20 Príkaz "teplota"	33
Obrázok 21 Príkaz "je gesto potrasenie"	34
Obrázok 22 Príkaz "keď hlasný zvuk"	35
Obrázok 23 Príkaz "keď je logo stlačené"	36
Obrázok 24 Príkaz "logo je stlačené"	37
Obrázok 25 Príkaz "úroveň zvuku"	38
Obrázok 26 Príkaz "kalibrovať kompas"	39
Obrázok 27 Príkaz "magnetická sila (μT) x"	40
Obrázok 28 Príkaz "otáčania ($^{\circ}$) náklon"	41
Obrázok 29 Príkaz "keď sa kolík P0 uvoľní"	42
Obrázok 30 Príkaz "nastaviť akcelerometer rozsah 1g"	43
Obrázok 31 Príkaz "nastaviť pre hlasný zvuk prahovú hodnotu na 128"	44
Obrázok 32 Príkaz "play melody at tempo (bpm) until done"	45
Obrázok 33 Príkaz "play tone stredné C for 1 úder until done"	46
Obrázok 34 Príkaz "vyzváňací tón (Hz) stredné C"	47
Obrázok 35 Príkaz "rest for 1 úder"	48
Obrázok 36 Príkaz "1 úder"	49
Obrázok 37 Príkaz "tempo (bpm)"	50
Obrázok 38 Príkaz "play melody dadadum in background"	51
Obrázok 39 Príkaz "zastaviť melódiu všetky"	52
Obrázok 40 Príkaz "hudba na nota melódie sa prehrala"	53
Obrázok 41 Príkaz "play chichot until done"	54
Obrázok 42 Príkaz "play zvuková vlna"	55
Obrázok 43 Príkaz "sound is playing"	56
Obrázok 44 Príkaz "vykresliť X Y"	57
Obrázok 45 Príkaz "prepnúť X Y"	58
Obrázok 46 Príkaz "zrušiť vykreslenie X Y"	59
Obrázok 47 Príkaz "bod X Y"	60
Obrázok 48 Príkaz "plot bar graph of 0 up to 0"	61
Obrázok 49 Príkaz "vykresliť X Y jas 255"	62

Obrázok 50 Príkaz "jas bodu X Y"	63
Obrázok 51 Príkaz "jas"	64
Obrázok 52 Príkaz "nastaviť jas 255"	65
Obrázok 53 Príkaz "zapnúť LED nepravda"	66
Obrázok 54 Príkaz "zastaviť animáciu"	67
Obrázok 55 Príkaz "set display mode čierna a biela"	68
Obrázok 56 Príkaz "rádio nastaviť skupinu 1"	69
Obrázok 57 Príkaz "rádio odoslať číslo 0"	70
Obrázok 58 Príkaz "rádio odoslať hodnotu name = 0"	71
Obrázok 59 Príkaz "rádio odoslať reťazec"	72
Obrázok 60 Príkaz "rádio pri prijatí čísla"	73
Obrázok 61 Príkaz "rádio pri prijatí name value"	74
Obrázok 62 Príkaz "rádio pri prijatí reťazca receiveString"	75
Obrázok 63 Príkaz "prijatý paket sila signálu"	76
Obrázok 64 Príkaz "rádio nastaviť silu prenosu"	77
Obrázok 65 Príkaz "rádio nastaviť sériové číslo prenosu pravda"	78
Obrázok 66 Príkaz "rádio nastaviť frekvenčné pásmo 0"	79
Obrázok 67 Príkaz "rádio volať udalosť"	80
Obrázok 68 Príkaz "opakovať 4 krát"	81
Obrázok 69 Príkaz "zatiaľ čo pravda"	82
Obrázok 70 Príkaz "pre index od 0 do 9"	83
Obrázok 71 Príkaz "pre prvok hodnota z list"	84
Obrázok 72 Príkaz "každých 500 ms"	85
Obrázok 73 Príkaz "prerušiť"	86
Obrázok 74 Príkaz "pokračuj"	87
Obrázok 75 Príkaz "ak pravda potom"	88
Obrázok 76 Príkaz "ak pravda potom inak"	89
Obrázok 77 Príkaz "ak s porovnávacou logikou"	90
Obrázok 78 Príkaz "ak s logickými premennými"	91
Obrázok 79 Príkaz "ak s logickými premennými a/alebo"	92
Obrázok 80 Použite príkaz "ak s logickou premennou nie"	93
Obrázok 81 Číselné premenné	94
Obrázok 82 Textové premenné	95
Obrázok 83 Premenné typu boolean	95
Obrázok 84 príklad premennej typu boolean	96
Obrázok 85 Pole (Array)	96
Obrázok 86 Príklad prístupu k jednotlivým hodnotám v poli	97
Obrázok 87 Premenné typu enum	97
Obrázok 88 Príklad premennej typu enum	98
Obrázok 89 Funkcia sčítania	99
Obrázok 90 Funkcia odčítania	100
Obrázok 91 Funkcia násobenia	101
Obrázok 92 Funkcia delenia	102
Obrázok 93 Funkcia zvyšku po delení (remainder of)	103
Obrázok 94 Funkcia minimálnej hodnoty (min)	104
Obrázok 95 Funkcia maximálnej hodnoty (max)	105
Obrázok 96 Funkcia absolútnej hodnoty	106
Obrázok 97 Funkcia druhej odmocniny	107
Obrázok 98 Funkcia zaokrúhľovania	108
Obrázok 99 Funkcia výberu náhodného čísla	109

Obrázok 100 Funkcia obmedzenia hodnoty na interval	110
Obrázok 101 Funkcia mapovania z intervalu od-do do iného intervalu	111
Obrázok 102 Funkcia náhodného výberu medzi pravdou a nepravdou	113
Obrázok 103 Funkcia $f(x)$ s parametrom text	114
Obrázok 104 Funkcia $f(x)$ s parametrom logické premenné.....	115
Obrázok 105 Funkcia $f(x)$ s parametrom číslo	116
Obrázok 106 Funkcia $f(x)$ s parametrom pole	117
Obrázok 107 Funkcia $f(x)$ s parametrom LED sprite.....	118
Obrázok 108 Funkcia $f(x)$ s parametrom image.....	119
Obrázok 109 Funkcia "nastaviť zoznam na pole"	120
Obrázok 110 Funkcia "nastaviť textový zoznam na pole"	121
Obrázok 111 Funkcia "prázdne pole"	122
Obrázok 112 Funkcia "dĺžka poľa (zoznam)"	123
Obrázok 113 Funkcia "získať hodnotu na"	124
Obrázok 114 Príkaz "odobrať hodnotu na indexe"	125
Obrázok 115 Funkcia "načítať a odobrať poslednú hodnotu zo zoznamu"	126
Obrázok 116 Príkaz "načítať a odobrať prvú hodnotu zo zoznamu"	128
Obrázok 117 Príkaz "získať náhodnú hodnotu z"	130
Obrázok 118 Príkaz "nastaviť hodnotu v 0 na"	132
Obrázok 119 Príkaz "pridať hodnotu na koniec"	134
Obrázok 120 Príkaz "odstrániť poslednú hodnotu z"	135
Obrázok 121 Príkaz "odstrániť prvú hodnotu z"	136
Obrázok 122 Príkaz "vložiť hodnotu na začiatok"	137
Obrázok 123 Príkaz "vložiť na index hodnotu"	138
Obrázok 124 Príkaz "odobrať hodnotu na indexe"	139
Obrázok 125 Príkaz "najdi index hodnoty"	140
Obrázok 126 Príkaz "otočiť zoznam"	141
Obrázok 127 Príkaz "Vytvoriť textový reťazec"	142
Obrázok 128 Príkaz ako spojiť texty	142
Obrázok 129 Príkaz na zistenie dĺžky textového reťazca	143
Obrázok 130 Príkaz na získanie znaku z určitého miesta v texte	143
Obrázok 131 Príkaz na získanie časti textového reťazca.....	144
Obrázok 132 Príkaz na porovnanie dvoch textových reťazcov	144
Obrázok 133 Príkaz na odstránenie nepotrebných medzier z textu	145
Obrázok 134 Príkaz na zistenie prítomnosti určitého textu v reťazci.....	145
Obrázok 135 Príkaz na zmenu textu na veľké písmená.....	146
Obrázok 136 Príkaz na zmenu textu na malé písmená	146
Obrázok 137 Príkaz na nahradenie určitého textu iným textom v reťazci.....	147
Obrázok 138 Príkaz na rozdelenie textu na jednotlivé časti	147
Obrázok 139 Príkaz na pripojenie textu pred alebo po inom texte	148
Obrázok 140 Príkaz rozdelenie textu na jednotlivé znaky.....	148
Obrázok 141 Príkaz „digitálne načítať kolík P0“	151
Obrázok 142 Príkaz „digitálne zapísať kolík“	152
Obrázok 143 Príkaz „analogovo čítať kolík“	153
Obrázok 144 Príkaz „analogovo zapísať kolík“	154
Obrázok 145 Príkaz "seriové zapísať riadok"	157
Obrázok 146 Príkaz "seriové zapísať číslo"	158
Obrázok 147 Príkaz "seriové zapísať hodnotu"	159
Obrázok 148 Príkaz: "seriové zapísať reťazec"	160
Obrázok 149 Príkaz: "seriové po prijatí údajov"	161

Úvod

V posledných rokoch sa čoraz častejšie stretávame s pojmami ako programovanie, mikrokontroléry či digitálna gramotnosť. Možno sa ti tieto slová zdajú zložité, no ak si niekedy zapol lampu cez pohybový senzor, pozrel si sa na teplomer alebo prehrál pesničku na reproduktore, bol si v kontakte s technológiami, ktoré riadia práve mikropočítače – teda jednoduché „mozgy“ zariadení, ktoré reagujú na prostredie a vykonávajú rôzne úlohy. Jedným z najdostupnejších a najobľúbenejších mikrokontrolérov pre školy a žiakov je **BBC micro:bit** – a práve jemu je venovaná táto učebnica.

Prečo vznikla táto učebnica?

Hlavným cieľom tejto učebnice je pomôcť ti pochopiť, ako fungujú digitálne zariadenia, ktoré nás dennodenne obklopujú, a zároveň ťa naučiť základom programovania v jednoduchom, zrozumiteľnom a vizuálne atraktívnom prostredí s názvom **MakeCode**. Nejde o teoretickú knihu plnú pojmov, ale o praktický sprievodca, ktorý ťa krok za krokom povedie svetom blokového programovania a mikropočítačovej logiky.

Na rozdiel od bežných učebníc informatiky, ktoré sa často zameriavajú len na prácu s počítačom ako textovým alebo výpočtovým nástrojom, táto učebnica ti ukáže, ako môžeš digitálnu technológiu *tvoriť*. Staneš sa konštruktérom, programátorom aj výskumníkom. Využiješ svoju predstavivosť aj logické myslenie. A možno po prvýkrát pocítiš, že informatika nie je len o písaní na klávesnici, ale aj o **tvorbe niečoho, čo funguje v reálnom svete**.

Čo je BBC micro:bit?

BBC micro:bit je malá doska – o niečo väčšia ako kreditná karta –, ktorá v sebe ukrýva množstvo súčiastok:

- LED displej z 25 svetielok,
- dve tlačidlá,
- akcelerometer na meranie pohybu,
- magnetometer (kompas),
- rádiový a Bluetooth modul,
- kolíky (tzv. „piny“) na pripojenie ďalších zariadení.

Je navrhnutá tak, aby bola jednoduchá na pochopenie a zároveň dostatočne výkonná na realizáciu desiatok nápadov – od animácií cez hry až po ovládanie svetiel alebo meranie teploty. Mikropočítač vie „vnímať“ svoje okolie, „rozhodovať sa“ a následne „konať“ – presne ako živý organizmus.

Programovanie cez MakeCode

MakeCode je online platforma od spoločnosti Microsoft, ktorá umožňuje programovanie micro:bitu pomocou tzv. blokov. Pracuje sa v prehliadači a nie je potrebné nič inštalovať. Kód

sa vytvára tak, že presúvaš farebné bloky s príkazmi ako stavebnicu – podobne ako keď skladáš lego.

Toto blokové programovanie je mimoriadne vhodné pre deti a začiatočníkov. Nepotrebuješ poznať zložité príkazy ani syntax. Všetko robíš vizuálne – presúvaš, vkladáš a spájaš. Ak ťa programovanie začne baviť, môžeš neskôr prejsť aj do textového režimu, napríklad v jazyku JavaScript alebo Python.

Čo ťa v učebnici čaká?

Každá kapitola sa venuje inej skupine blokov v prostredí MakeCode a ich praktickému využitiu. Postupne sa naučíš:

- používať **základné bloky** na spustenie programu a zobrazovanie obrázkov na LED displeji,
- ovládať **vstupy** – ako sú tlačidlá, pohyb, naklonenie alebo gesto,
- prehrávať **zvuky a melódie**,
- riadiť svietiace **LED diódy** a vytvárať animácie,
- posielat' a prijímať **správy cez rádio** – napríklad medzi dvoma micro:bitmi,
- vytvárať **opakujúce sa činnosti** pomocou cyklov,
- zapisovať **podmienky** pomocou logiky typu „ak – potom – inak“,
- pracovať s **premennými**, číslami a výpočtami,
- vytvárať **vlastné funkcie** a znovupoužiteľný kód,
- ukladať a čítať údaje z **polí**,
- manipulovať s **textom** a zobrazit' ho na displeji,
- používať **kolíky (piny)** na pripojenie externých súčiastok,
- prenášať údaje cez **sériové rozhranie** – napríklad do počítača alebo do cloudu.

Každý nový pojem je vysvetlený jednoducho a ilustrovaný na konkrétnom príklade. Zároveň sú v učebnici pripravené úlohy, v ktorých si všetko vyskúšaš v praxi.

Ako s učebnicou pracovať?

Túto učebnicu nemusíš čítať naraz. Môžeš si ju otvoriť pri konkrétnom probléme alebo projekte. Odporúčame ti však začať od začiatku a postupne si prechádzať jednotlivé bloky a ich použitie.

Každá kapitola obsahuje:

- teoretický výklad (čo daný blok robí),
- názorný príklad (krátky program),
- tipy a triky pre pokročilejších,

- samostatné úlohy,
- prehľad naučeného.

Môžeš si kód vyskúšať najprv v simulátore a potom ho nahrať do skutočného micro:bitu. V každej úlohe sa snaž pochopiť, **prečo niečo funguje**, nie len „ako“. Tvoje porozumenie je dôležitejšie než zapamätanie postupu.

Čo všetko sa naučíš?

- Učebnica ti pomôže rozvíjať:
- **logické myslenie** – budeš vytvárať podmienky, cykly a riešenia,
- **kreativitu** – môžeš navrhnúť vlastné animácie, hry či experimenty,
- **praktické zručnosti** – napríklad prácu s mikropočítačom a zapájanie obvodov,
- **matematické a technické myslenie** – pri práci s číslami, premennými a dátami,
- **digitálne kompetencie** – ako pracovať s dátami a technológiou zodpovedne.

Pre koho je táto učebnica?

Táto učebnica je určená žiakom 2. stupňa základných škôl. Nevyžaduje žiadne predchádzajúce znalosti programovania. Je vhodná aj na domáce samoštúdium, prácu v školských krúžkoch, na hodinách informatiky alebo techniky. Môžeš ju využívať s pomocou učiteľa, ale aj úplne samostatne.

Začiatok programovania môže vyzerat' ako výstup na horu – zo začiatku sa to zdá ťažké, ale keď sa naučíš používať správne „výstroj“ (bloky a logiku), budeš postupovať stále vyššie a z každého vrcholu uvidíš nové možnosti.

Veríme, že ťa táto cesta bude baviť, že získaš nové zručnosti a možno objavíš svoj záujem o techniku, ktorý ti raz pomôže vybrať si tvoje budúce povolanie. Prajeme ti veľa radosti, trpezlivosti a odvahy na tvojej mikropočítačovej ceste!

1 Makecode

V súčasnej dobe, keď je technológia neoddeliteľnou súčasťou každodenného života, rastie dôležitosť digitálnej gramotnosti u žiakov základných škôl. Schopnosť rozumieť princípom fungovania počítačov, ako aj schopnosť tvoriť programy, otvára mladým ľuďom nové možnosti a rozvíja ich kreativitu, kritické myslenie a schopnosť riešiť problémy. Jedným z nástrojov, ktorý umožňuje deťom na základných školách jednoducho začať s programovaním, je BBC micro:bit. Tento malý počítač, v spojení s programovacím prostredím, predstavuje efektívny a zábavný spôsob, ako vstúpiť do sveta programovania a elektroniky.

BBC micro:bit je malý mikropočítač, ktorý bol vyvinutý s cieľom podporiť výučbu programovania a výpočtovej techniky medzi deťmi a mladými ľuďmi. Tento projekt bol iniciovaný britskou spoločnosťou BBC v rámci snahy o rozvoj technickej gramotnosti. Od svojho uvedenia v roku 2016 sa BBC micro:bit stal populárnym nielen v Spojenom kráľovstve, ale aj na školách po celom svete. BBC micro:bit je malá doska o veľkosti kreditnej karty, vybavená rôznymi senzormi, LED displejom, tlačidlami a bezdrôtovými komunikačnými technológiami, ktoré umožňujú vytvárať rôzne interaktívne projekty. Vďaka jednoduchému rozhraniu a dostupným učebným materiálom je micro:bit vhodný pre deti už od ôsmich rokov (Nominet Trust, 2017).

Čo je MakeCode?

MakeCode je blokové programovacie prostredie vyvinuté spoločnosťou Microsoft, ktoré umožňuje programovať micro:bit intuitívnym spôsobom. MakeCode ponúka dva základné režimy: blokové programovanie, ktoré je vhodné pre začiatočníkov, a JavaScript, ktorý môžu používať pokročilejší žiaci, ktorí chcú preniknúť hlbšie do sveta programovania. Blokové programovanie je veľmi podobné stavebnici, kde žiaci skladajú jednotlivé príkazy vo forme farebných blokov, ktoré následne vytvárajú program. Tento vizuálny prístup pomáha deťom ľahko pochopiť logiku programovania bez nutnosti učiť sa zložité syntaxe programovacích jazykov (Smith, 2018).

Prečo používať BBC micro:bit na školách?

Používanie BBC micro:bit a MakeCode v školskom prostredí má niekoľko významných výhod:

Podpora kreativity a riešenia problémov: micro:bit a MakeCode umožňujú žiakom vytvárať vlastné projekty, čo podporuje ich tvorivé myslenie. Či už ide o jednoduché animácie, hry alebo komplexnejšie projekty využívajúce senzory a tlačidlá, žiaci sa učia riešiť problémy a hľadať nové spôsoby, ako dosiahnuť cieľ (Brown, 2019).

Jednoduchosť používania: Vizuálny charakter blokového programovania v MakeCode robí toto prostredie veľmi prístupným pre začiatočníkov. Žiaci si môžu rýchlo osvojiť základné koncepty programovania, ako sú cykly, podmienky alebo premenné, a ihneď ich aplikovať v praxi (Smith, 2018).

Rôznorodé možnosti aplikácie: micro:bit ponúka širokú škálu možností vďaka svojim integrovaným senzorom, LED displeju a možnostiam pripojenia k ďalším zariadeniam. Žiaci môžu napríklad vytvoriť jednoduchý teplomer, krokmer alebo dokonca malé robotické

vozidlo. Týmto spôsobom sa učia nielen programovať, ale aj chápať princípy fungovania elektroniky a fyziky (Thompson, 2020).

Podpora tímovej práce a spolupráce: Projekty, ktoré vyžadujú programovanie micro:bit, často motivujú žiakov spolupracovať, deliť sa o nápady a navzájom si pomáhať pri riešení úloh. Takáto spolupráca rozvíja komunikačné a sociálne zručnosti, ktoré sú kľúčové pre úspech v modernej spoločnosti (Barrett, 2017).

Začíname s MakeCode

Začiatok práce s MakeCode a BBC micro:bit je veľmi jednoduchý. Stačí navštíviť webovú stránku <https://makecode.microbit.org>, kde si môžete spustiť programovacie prostredie priamo vo webovom prehliadači. Nie je potrebné nič inštalovať, čo robí toto prostredie prístupné pre školy bez ohľadu na technické vybavenie.

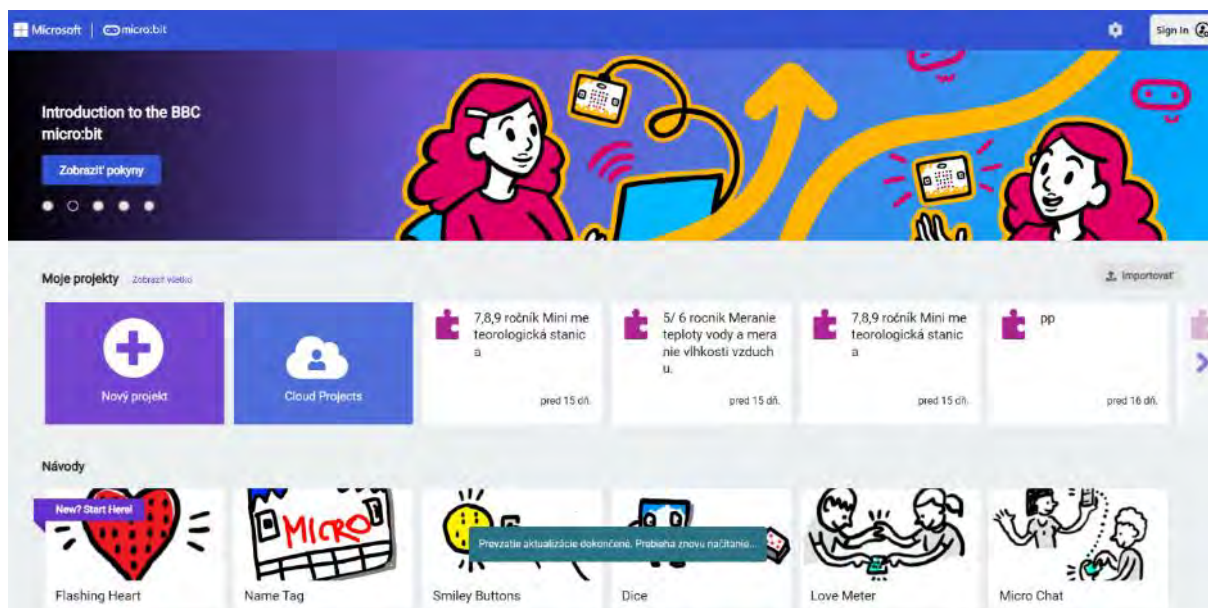
V úvodných krokoch môžu žiaci experimentovať s jednoduchými projektmi, ako je vytvorenie animácií na LED displeji alebo naprogramovanie reakcie na stlačenie tlačidla. Následne môžu postupne prejsť k zložitejším úlohám, ktoré zahŕňajú využitie senzorov alebo bezdrôtovú komunikáciu medzi viacerými BBC micro:bit. Vďaka interaktívnym tutoriálom a množstvu predpripravených lekcií je pre učiteľov ľahké začleniť MakeCode a BBC micro:bit do bežných vyučovacích hodín.

BBC micro:bit v kombinácii s programovacím prostredím MakeCode predstavuje silný nástroj na rozvoj digitálnej a technickej gramotnosti u detí. Využívanie tejto technológie v školách nielen podporuje učenie sa programovania, ale tiež rozvíja kreatívne a analytické schopnosti žiakov. Vďaka svojmu jednoduchému používaniu a širokým možnostiam aplikácie má micro:bit potenciál stať sa kľúčovým nástrojom vo výučbe programovania a technológií v školách.

Práca v prostredí MakeCode pre BBC micro:bit je možná dvoma spôsobmi – s prihlásením a aj bez prihlásenia.

Používatelia môžu pracovať v prostredí aj bez vytvorenia účtu, čo umožňuje rýchly a jednoduchý prístup k základným funkciám editoru, ako je programovanie pomocou blokov, simulácia kódu a stiahnutie programu do zariadenia BBC micro:bit. Avšak, prihlásenie prostredníctvom účtu Microsoft (napr. školského Office 365 účtu) alebo google účtu, prináša ďalšie výhody, ako je možnosť ukladať projekty do cloudu, zdieľať ich s ostatnými, synchronizovať prácu medzi viacerými zariadeniami a využívať pokročilé nástroje pre učiteľov a žiakov.

Táto flexibilita umožňuje využívať MakeCode v rôznych podmienkach – v škole, doma alebo na workshopoch – bez nutnosti zložitej administrácie.



Obrázok 1 Rozhranie MakeCode

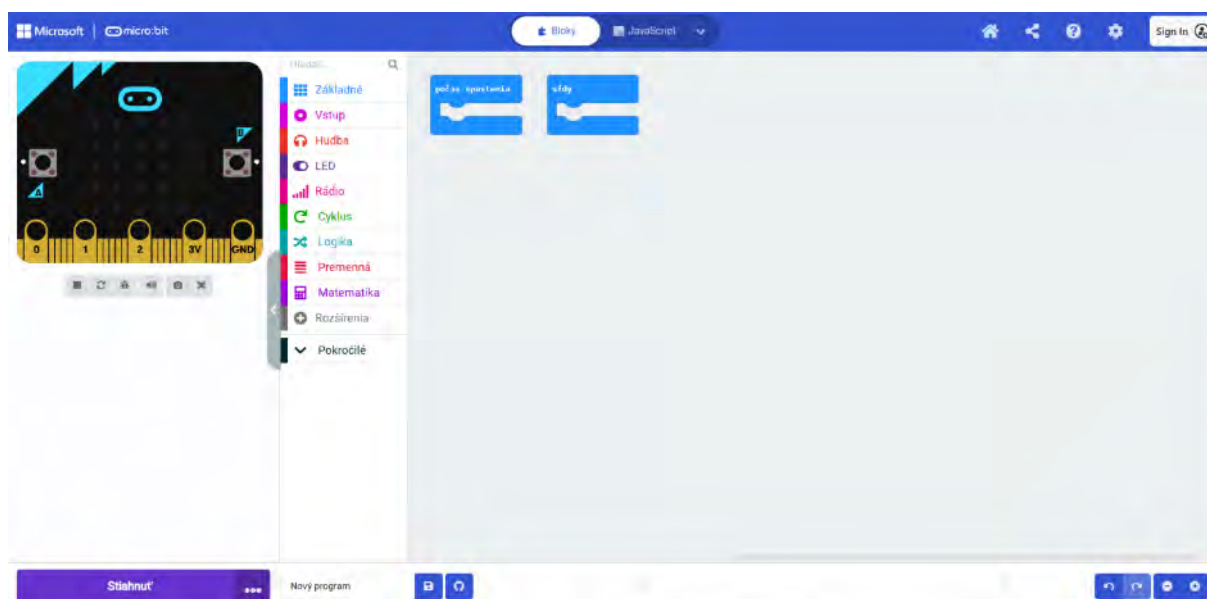
Popis rozhrania MakeCode

- **Horná časť obrazovky** obsahuje logo Microsoft a micro:bit, ako aj možnosť prihlásenia (Sign In).
- V strede vidíme farebný banner s názvom **"Introduction to the BBC micro:bit"** a tlačidlom **„Zobrazit pokyny“**, ktoré slúži na spustenie úvodného tutoriálu.
- **Sekcia „Moje projekty“** zobrazuje už vytvorené projekty používateľa:
 - o Napríklad projekty:
 - „7,8,9 ročník Mini meteorologická stanica“,
 - „5/6 ročník Meranie teploty vody a meranie vlhkosti vzduchu“
 - o Tlačidlo **„Nový projekt“** slúži na vytvorenie nového programu.
 - o **„Cloud Projects“** umožňuje ukladať projekty online (v cloude).
- V dolnej časti sa nachádza **galéria návodov** ako napr.:
 - o „Flashing Heart“ – blikajúce srdce
 - o „Name Tag“ – menovka
 - o „Smiley Buttons“ – úsmevné tlačidlá
 - o „Dice“ – kocka
 - o „Love Meter“ – merač lásky
 - o „Micro Chat“ – jednoduchý chat pomocou micro:bitov

 Toto prostredie je ideálne na výučbu základov programovania. Učitelia a žiaci môžu:

- tvoriť vlastné projekty pomocou blokov alebo JavaScriptu,
- skúšať pripravené návody na jednoduché elektronické obvody,
- učiť sa interaktívne cez praktické aktivity.

Základné časti prostredia MakeCode pre BBC micro:bit



Obrázok 2 Hlavné prostredie MakeCode

Po otvorení stránky <https://makecode.microbit.org> sa zobrazí hlavné pracovné prostredie. Tu je popis jeho hlavných častí:

1. Simulátor (vľavo hore)

Simulátor zobrazuje virtuálny micro:bit. Umožňuje vidieť, ako sa bude program správať v reálnom zariadení – bez potreby fyzického micro:bitu. Po zmene kódu sa simulátor automaticky aktualizuje. Môžeš naň aj kliknúť (napr. na tlačidlá A a B) a vyskúšať si, ako program reaguje.

2. Panel s blokmi (stred – hore)

Toto je hlavný priestor na programovanie. Bloky sa skladajú ako puzzle. Každý typ bloku má svoju kategóriu, na **Príklad:**

- Základné (Basic) – zobrazenie textu, čakanie, zapnutie LED.
- Vstupy (Input) – reakcie na stlačenie tlačidla, pohyb, otrasy.
- Logika (Logic) – podmienky (ak...tak), porovnávanie hodnôt.
- Slučky (Loops) – opakovanie činností.
- Premenné (Variables) – vytváranie a používanie vlastných hodnôt.
- Hudba (Music) – prehrávanie tónov, melódií.
- A iné – podľa toho, aké rozšírenia si zapneš.

3. Pracovná plocha (stred – dole)

Sem sa skladajú jednotlivé bloky. Na začiatku uvidíš základné štruktúry ako napr.:

- pri štarte (on start) – vykoná sa raz pri zapnutí micro:bitu.
- navždy (forever) – opakuje sa stále dokola.

Bloky môžeš presúvať myšou a vkladať do seba. Nezabudni, že farebne musia ladiť – inak blok nezapadne.

4. Tlačidlá pre stiahnutie a správu projektu (dole)

- Stiahnuť – vygeneruje súbor .hex, ktorý sa nahráva do micro:bitu (ako keby si kopíroval súbor na USB kľúč).
- Uložiť (Save) – umožňuje uložiť projekt do počítača (vo formáte .mkcd).
- Zdieľať (Share) – vygeneruje odkaz na tvoj projekt.
- Prepnúť jazyk – v pravom dolnom rohu možno nastaviť jazyk rozhrania (slovenčina je dostupná).

5. Menu a prihlásenie (pravý horný roh)

- Domov (Home) – návrat na hlavnú stránku s projektmi.
- Prihlásenie – ak máš školský alebo Microsoft účet, môžeš sa prihlásiť a ukladať projekty do cloudu.
- Rozšírenia – umožňujú pridať ďalšie funkcie pre rôzne senzory, súčiastky alebo moduly.

2 Základné

Príkaz „Vždy“

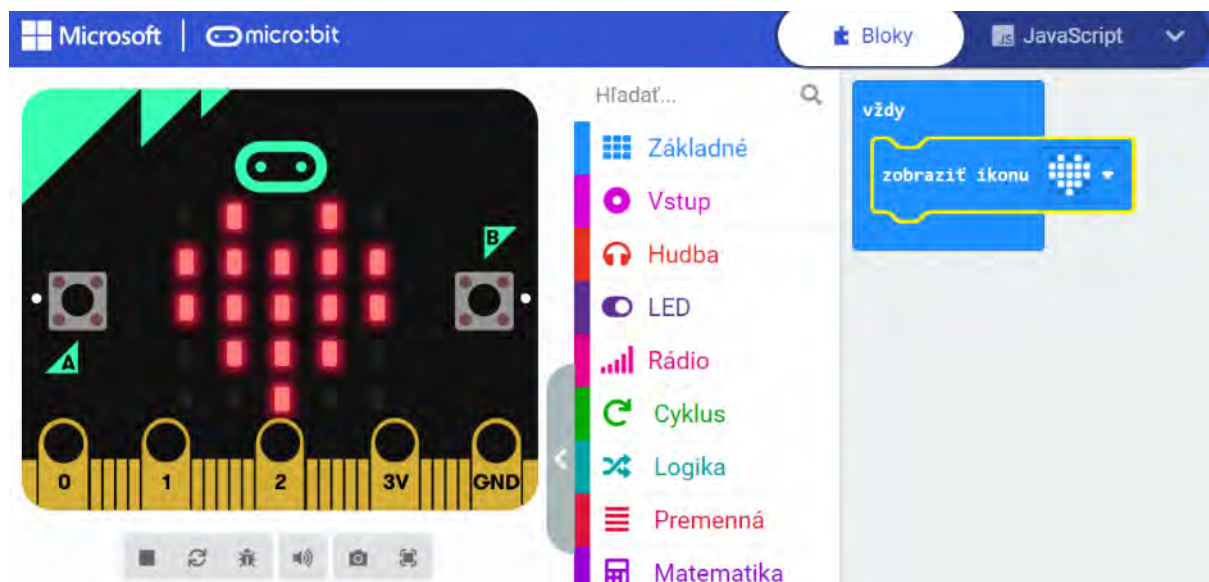
Príkaz "Vždy" v programovaní BBC micro:bit je veľmi dôležitý a užitočný. Slúži na vykonávanie určitých úloh neustále, bez prerušenia.

Ako to funguje?

1. Nepretržité opakovanie: Kód, ktorý umiestnime do príkazu "Vždy", sa bude opakovat' neustále, znova a znova. Hovoríme mu aj "nekonečný cyklus".
2. Realizácia úloh: Tento príkaz je ideálny na úlohy, ktoré potrebujú neustále kontrolovanie alebo vykonávanie, ako napríklad zobrazenie animácie alebo meranie teploty.

Príklad:

Ak chceme, aby BBC micro:bit neustále zobrazoval srdiečko, použijeme príkaz "Vždy" nasledovne:



Obrázok 3 Príkaz „Vždy“

Po spustení tohto kódu sa bude srdiečko na LED matici BBC micro:bitu zobrazovať neustále, bez zastavenia.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Vyberte príkaz "Vždy" a vložte ho do svojho programu.
3. Do príkazu "Vždy" vložte blok "zobrazíť ikonu" a vyberte ikonu srdiečka.
4. Spustite svoj program a sledujte, ako BBC micro:bit neustále zobrazuje srdiečko!

Príkaz "Vždy" je skvelý na vytváranie programov, ktoré potrebujú nepretržité sledovanie alebo opakovanie určitých úloh.

Príkaz „Zobraziť číslo“

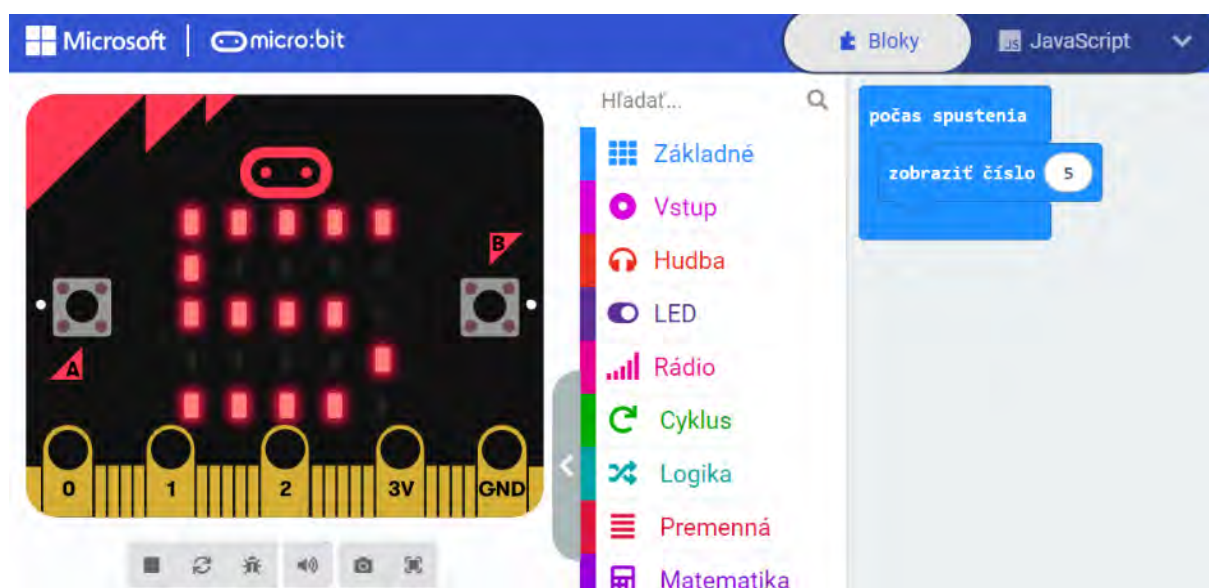
Príkaz „**Zobraziť číslo**“ slúži na zobrazovanie čísel na LED matici BBC micro:bitu. Predstavte si LED maticu ako malú obrazovku, na ktorej sa zobrazia čísla.

Ako to funguje?

1. Výber čísla: Vyberieme číslo, ktoré chceme zobraziť. Môže to byť akékoľvek číslo, napríklad 5.
5. Zobrazenie čísla: Použitím príkazu "zobraziť číslo" sa vybrané číslo zobrazí na LED matici.

Príklad:

Ak chceme na BBC micro:bitu zobraziť číslo 5, použijeme nasledujúci blok kódu:



Obrázok 4 Príkaz „Zobraziť číslo“

Po spustení tohto kódu sa na LED matici BBC micro:bitu objaví číslo 5. Tento príkaz je veľmi užitočná pri vytváraní rôznych projektov, napríklad počítačiel alebo jednoduchých hier.

Vyskúšajte si to!

6. Otvorte MakeCode editor na stránke micro:bit.
7. Vyberte blok "zobraziť číslo".
8. Vložte ho do svojho programu a zadajte číslo, ktoré chcete zobraziť.
9. Spustite svoj program a sledujte, ako sa číslo zobrazí na LED matici vášho micro:bitu!

Príkaz "zobrazit' LED"

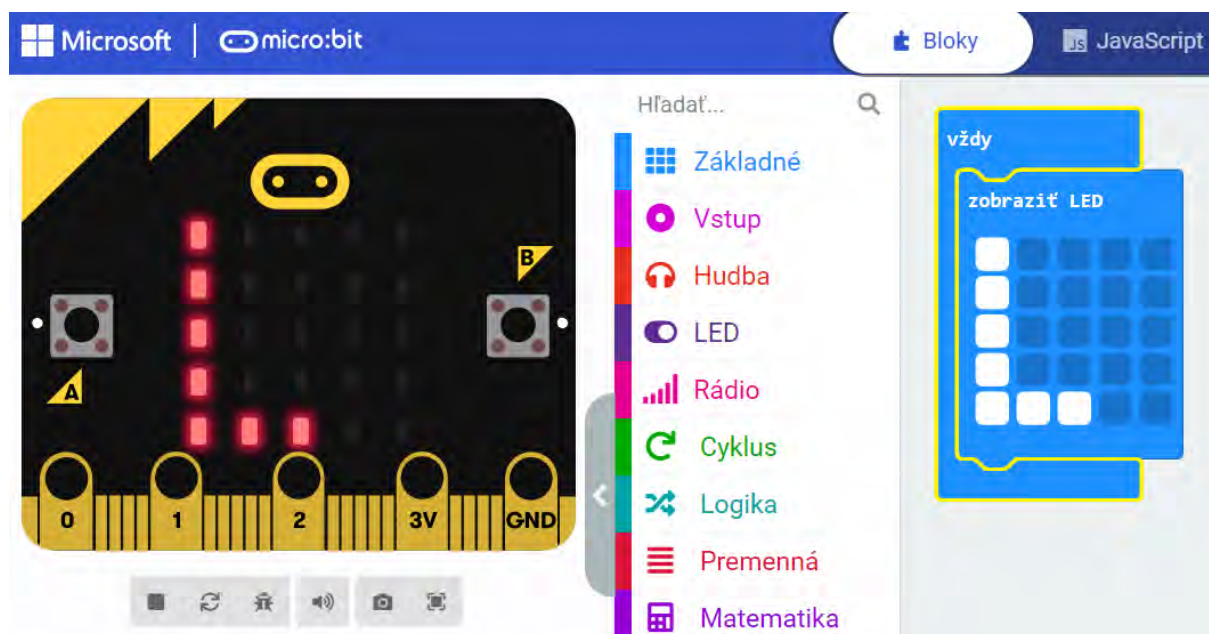
Príkaz "zobrazit' LED" slúži na rozsvietenie konkrétnych LED diód na LED matici micro:bitu. LED matica pozostáva z 25 diód usporiadaných do mriežky 5x5.

Ako to funguje?

1. Výber diód: Každá dióda má svoju pozíciu, ktorá je určená dvoma číslami: riadkom (0-4) a stĺpcom (0-4).
2. Rozsvietenie: Pomocou príkazu "zobrazit' LED" môžeme rozsvietiť konkrétnu diódu alebo viaceré diódy naraz.

Príklad:

Ak chceme rozsvietiť LED diódy v tvare písmena "L", použijeme nasledujúci blok kódu:



Obrázok 5 Príkaz "zobrazit' LED"

Po spustení tohto kódu sa na LED matici micro:bitu zobrazí písmeno "L".

Vyskúšajte si to!

1. Otvorte Makecode editor na stránke micro:bit.
2. Vyberte príkaz "zobrazit' LED" a vložte ho do svojho programu.
3. Vložte do príkazu rozmiestnenie diód, ktoré chcete rozsvietiť.
4. Spustíte svoj program a sledujte, ako sa diódy na LED matici rozsvietia podľa vášho návrhu!

Pomocou príkazu "zobrazit' LED" môžete vytvárať rôzne tvary, písmená a vzory.

Príkaz "zobraziť Ikonu"

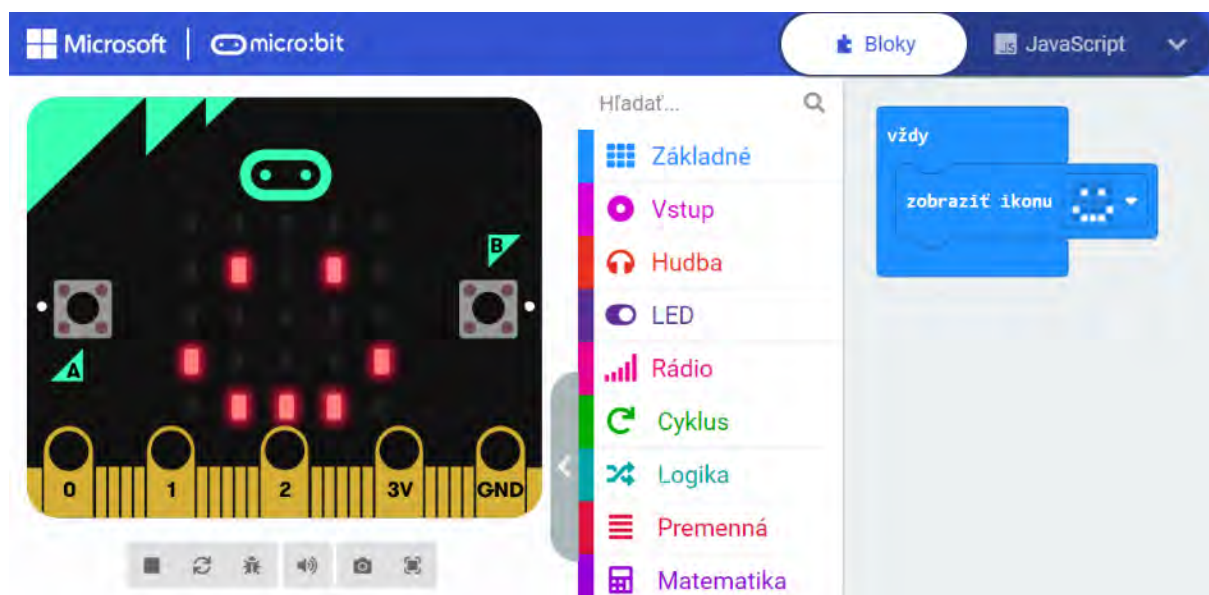
Príkaz "zobraziť Ikonu" slúži na zobrazenie preddefinovaných ikon na LED matici BBC micro:bitu. Tieto ikony predstavujú jednoduché obrázky, ako sú srdce, smajlík, šípka a mnoho ďalších a môžu byť použité na rôzne účely, napríklad na signalizáciu alebo zábavu.

Ako to funguje?

Príklad:

1. Výber ikony: V editori si vyberiete z rôznych preddefinovaných ikon, ktoré chcete zobraziť.
2. Zobrazenie ikony: Pomocou príkazu "zobraziť Ikonu" sa vybraná ikona objaví na LED matici BBC micro:bitu.

Ak chceme zobraziť ikonu smajlíka, použijeme nasledujúci blok kódu:



Obrázok 6 Príkaz "zobraziť Ikonu"

Po spustení tohto kódu sa na LED matici micro:bitu objaví ikona smajlíka.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Vyberte príkaz "zobraziť Ikonu" a vložte ho do svojho programu.
3. Vyberte ikonu, ktorú chcete zobraziť, napríklad smajlík.
4. Spustíte svoj program a sledujte, ako sa ikona zobrazuje na LED matici vášho BBC micro:bitu!
5. Tento príkaz je veľmi jednoduchý na použitie a umožňuje rýchlo vytvárať vizuálne efekty na vašom micro:bite, čo ho robí zábavnejším a interaktívnejším.

Tento príkaz je veľmi jednoduchý na použitie a umožňuje rýchlo vytvárať vizuálne efekty na vašom BBC micro:bite.

Príkaz "zobrazit' reťazec"

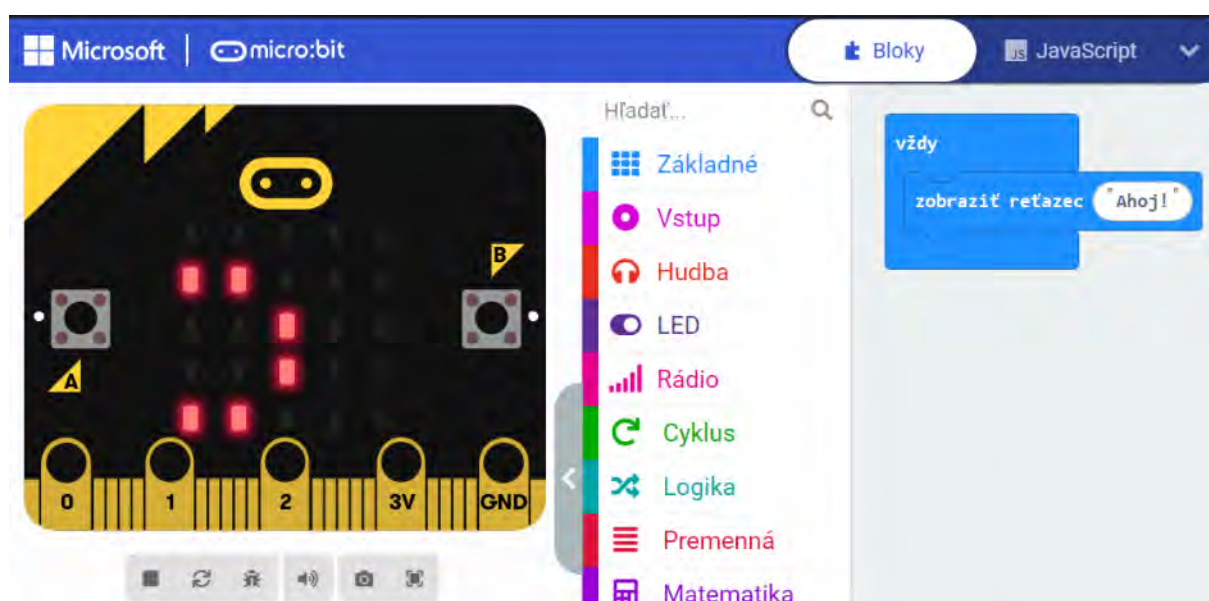
Príkaz "zobrazit' reťazec" umožňuje zobrazit' textové reťazce (slová alebo vety) na LED matici BBC micro:bitu. Text sa zobrazuje postupne, písmeno po písmene, pohybom zľava doprava.

Ako to funguje?

3. Výber textu: Vyberieme text, ktorý chceme zobrazit', napríklad "Ahoj".
4. Zobrazenie textu: Pomocou príkazu "zobrazit' reťazec" sa text objaví na LED matici a bude sa posúvať zľava doprava.

Príklad:

Ak chceme zobrazit' text "Ahoj!", použijeme nasledujúci blok kódu:



Obrázok 7 Príkaz "zobrazit' reťazec"

Po spustení tohto kódu sa text "Ahoj!" bude postupne zobrazovať na LED matici micro:bitu.

Vyskúšajte si to!

1. Otvorte Makecode editor na stránke micro:bit.
2. Vyberte príkaz "zobrazit' reťazec" a vložte ho do svojho programu.
3. Do príkazu zadajte text, ktorý chcete zobrazit', napríklad "Ahoj!".
4. Spustite svoj program a sledujte, ako sa text posúva na LED matici vášho micro:bitu!

Tento príkaz je skvelý na zobrazovanie rôznych správ a oznamov na vašom micro:bite.

Príkaz "vymazať obrazovku"

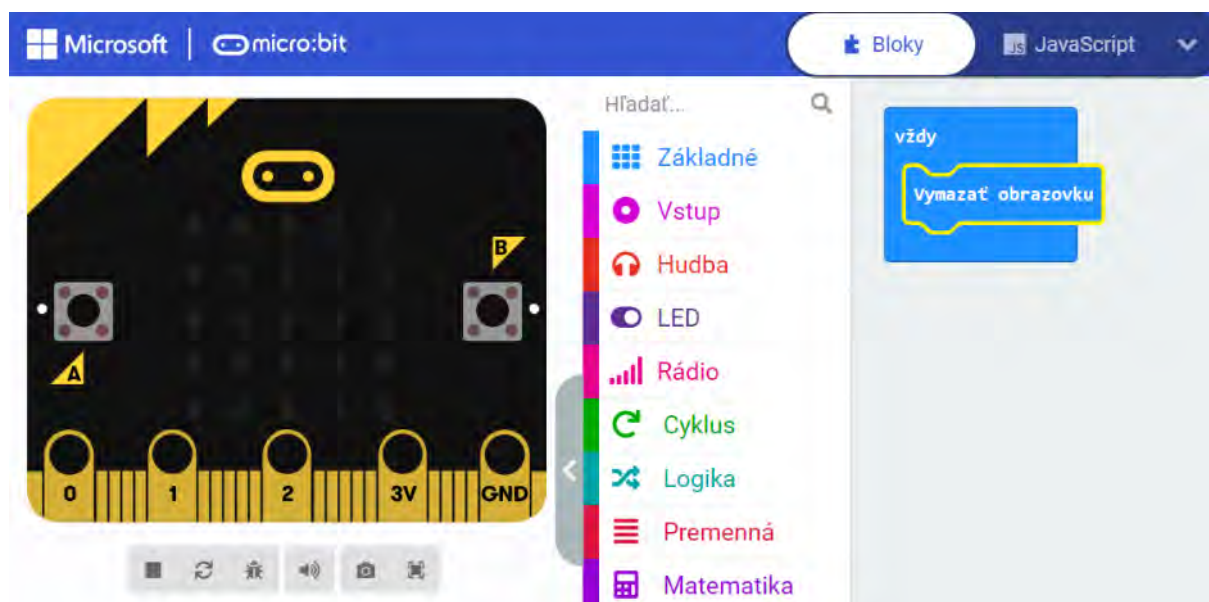
Príkaz "vymazať obrazovku" slúži na vypnutie všetkých LED diód na LED matici micro:bitu. Tento príkaz je užitočný, keď chceme zobrazit' čistú obrazovku pred začiatkom novej úlohy alebo projektu.

Ako to funguje?

1. Vymazanie diód: Po vykonaní tohto príkazu sa všetky diódy na LED matici vypnú, čo znamená, že žiadne svetlá nebudú svietiť.
2. Čistá obrazovka: Tento príkaz zabezpečí, že LED matica bude pripravená na zobrazenie nových informácií alebo ikon.

Príklad:

Ak chceme vymazať obrazovku pred začiatkom nového zobrazenia, použijeme nasledujúci blok kódu:



Obrázok 8 Príkaz "vymazať obrazovku"

Po spustení tohto kódu sa všetky diódy na LED matici micro:bitu vypnú, a obrazovka bude čistá.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Vyberte príkaz "vymazať obrazovku" a vložte ho do svojho programu.
3. Spustite svoj program a sledujte, ako sa obrazovka LED matice micro:bitu vymaže!

Tento príkaz je jednoduchý, ale veľmi užitočný pri tvorbe interaktívnych programov, kde potrebujeme pravidelne čistiť obrazovku pred zobrazením nových informácií alebo vzorov.

Príkaz "počas spustenia"

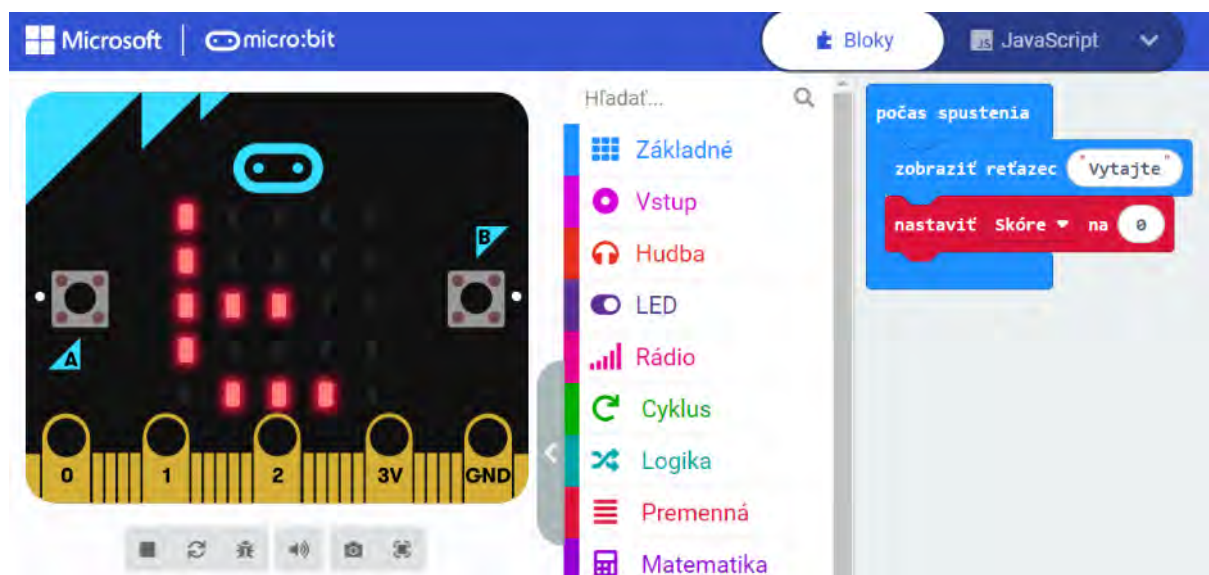
Príkaz "počas spustenia" je jedným z najdôležitejších príkazov, pretože určuje, čo sa má vykonať hneď po spustení programu na micro:bit.

Ako to funguje?

1. Inicializácia: Akonáhle micro:bit zapneme alebo reštartujeme, všetko, čo je v bloku "počas spustenia", sa vykoná iba raz na začiatku programu.
2. Príprava: Tento príkaz sa často používa na nastavenie počiatočných podmienok, ako sú premenné, zobrazenie úvodného textu alebo ikony, či iné dôležité nastavenia pred spustením hlavného programu.

Príklad:

Ak chceme po spustení programu zobraziť nápis "Vitajte!" a potom nastaviť počiatočné hodnoty pre premenné, použijeme nasledujúci blok kódu:



Obrázok 9 Príkaz "počas spustenia"

Po spustení tohto kódu sa najprv zobrazí text "Vitajte!" a potom sa nastaví počiatočná hodnota premennej "Skóre" na 0.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Vyberte príkaz "počas spustenia" a vložte ho do svojho programu.
3. Vložte do príkazu akcie, ktoré chcete vykonať na začiatku programu, napríklad zobrazenie textu a nastavenie premenných (vyberte príkaz "Premenná" kliknite na "vytvoriť premennú" zvolte názov premennej "Skóre" vyberte príkaz "nastaviť").
4. Spustite svoj program a sledujte, ako sa tieto akcie vykonajú hneď po spustení micro:bitu!

Tento príkaz je kľúčový pre inicializáciu a prípravu vášho programu, aby všetko fungovalo presne tak, ako ste si predstavovali.

Príkaz "pozastaviť (ms)"

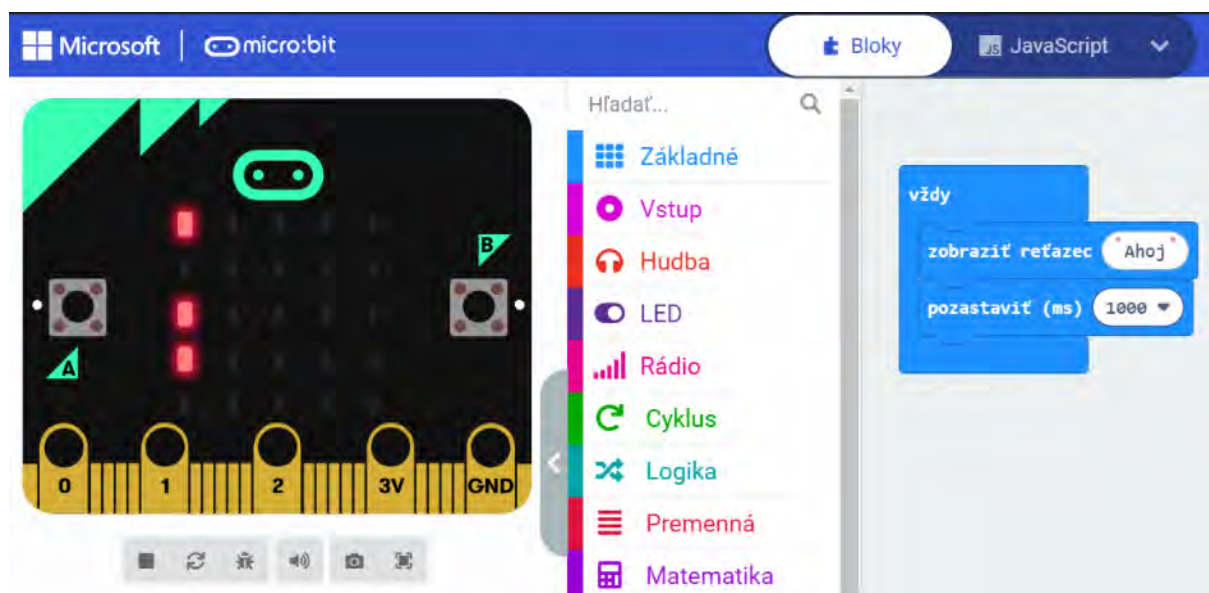
Príkaz "pozastaviť (ms)" slúži na dočasné zastavenie vykonávania programu na určitý čas, ktorý je uvedený v milisekundách (ms). Tento príkaz je veľmi užitočný, ak potrebujete vytvoriť časové oneskorenie medzi akciami.

Ako to funguje?

1. Nastavenie času: Vyberiete si, na ako dlho chcete pozastaviť vykonávanie programu. Čas sa uvádza v milisekundách (1 sekunda = 1000 milisekúnd).
2. Pozastavenie programu: Program sa zastaví na zadaný čas a potom pokračuje ďalej.

Príklad:

Ak chceme pozastaviť program na 1 sekundu, použijeme nasledujúci blok kódu:



Obrázok 10 Príkaz "pozastaviť (ms)"

Po spustení tohto kódu sa program zastaví na 1 sekundu a potom bude pokračovať ďalej.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Vyberte príkaz "pozastaviť (ms)" a vložte ho do svojho programu.
3. Nastavte čas pozastavenia, napríklad 1000 ms (1 sekunda).
4. Spustite svoj program a sledujte, ako sa program na zadaný čas pozastaví a potom pokračuje ďalej!

Príkaz "pozastaviť (ms)" je veľmi užitočný na vytvorenie časových oneskorení medzi akciami vo vašom programe, čo môže byť užitočné pri tvorbe animácií, hier a iných interaktívnych projektov.

Príkaz "zobraziť šípku"

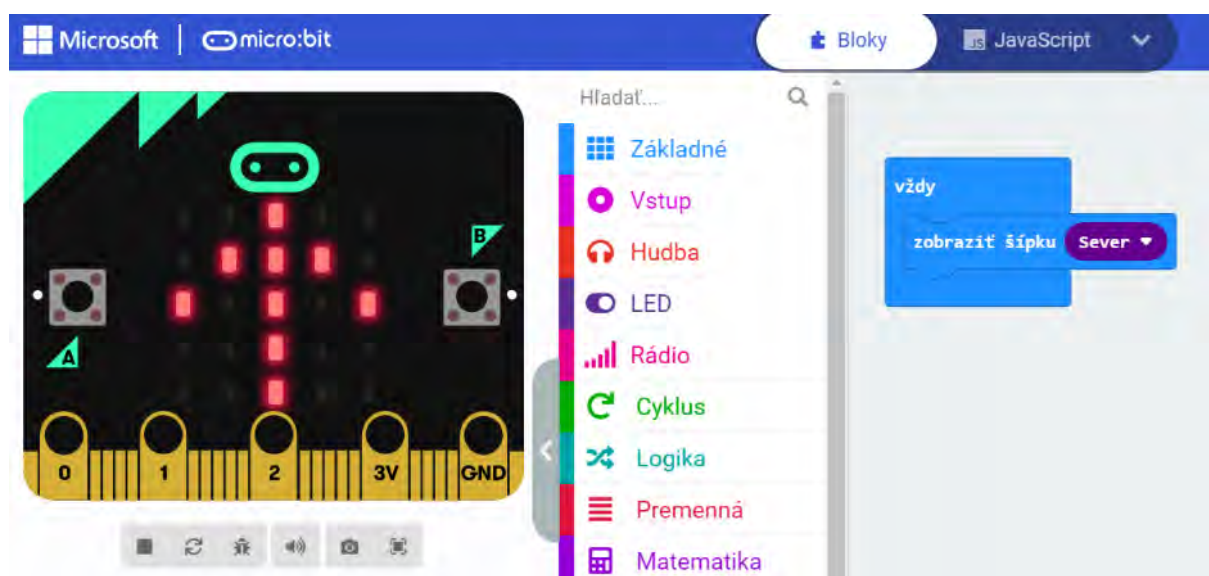
Príkaz "zobraziť šípku" slúži na zobrazenie preddefinovanej šípky na LED matici micro:bitu. Šípka môže ukazovať rôznymi smermi, čo je užitočné pri tvorbe projektov, ktoré potrebujú vizuálne smerovanie.

Ako to funguje?

1. Výber smeru: Vyberiete si smer, ktorým má šípka ukazovať (hore, dole, doľava, doprava alebo diagonálne).
2. Zobrazenie šípky: Pomocou príkazu "zobraziť šípku" sa vybraná šípka objaví na LED matici micro:bitu.

Príklad:

Ak chceme zobraziť šípku, ktorá ukazuje hore, použijeme nasledujúci blok kódu:



Obrázok 11 Príkaz "zobraziť šípku"

Po spustení tohto kódu sa na LED matici BBC micro:bitu objaví šípka, ktorá ukazuje na sever (hore).

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Vyberte príkaz "zobraziť šípku" a vložte ho do svojho programu.
3. Vyberte smer šípky, ktorú chcete zobraziť, napríklad hore.
4. Spustíte svoj program a sledujte, ako sa šípka zobrazuje na LED matici vášho BBC micro:bitu!

Tento príkaz je veľmi užitočný na vytváranie interaktívnych projektov, kde potrebujete smerovanie alebo navigáciu. Teraz ste pripravení experimentovať s rôznymi smermi šípok vo vašich programoch!

3 Vstup

Príkaz "keď sa tlačidlo A stlačí"

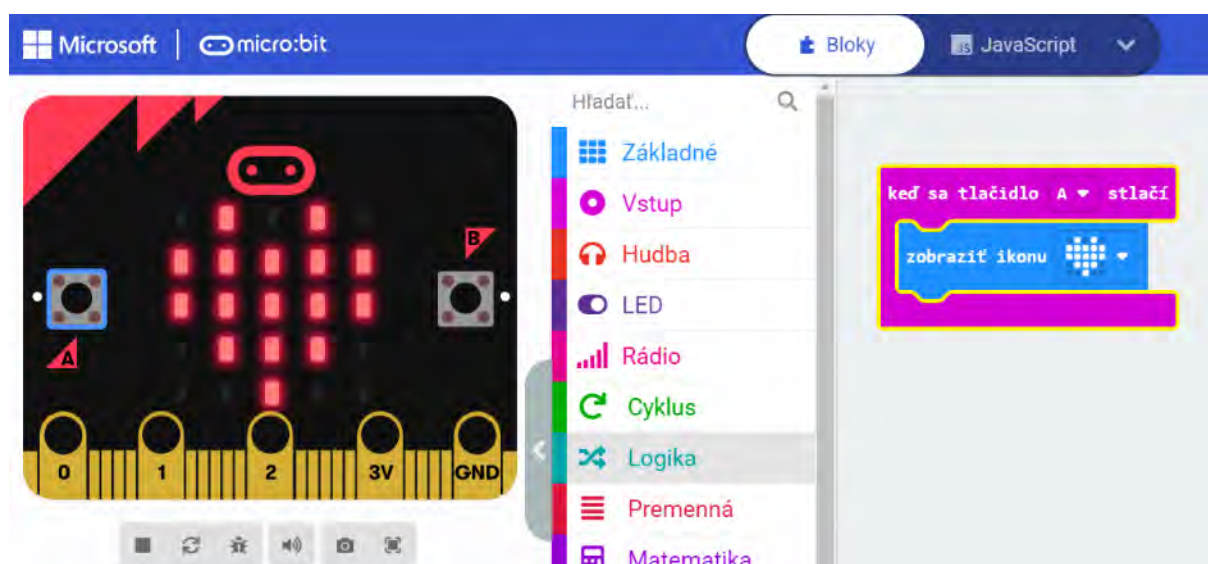
Príkaz "keď sa tlačidlo A stlačí" slúži na vykonanie určitej akcie, keď používateľ stlačí tlačidlo A na BBC micro:bitu. Toto tlačidlo sa nachádza na prednej strane BBC micro:bitu a je označené písmenom A.

Ako to funguje?

1. Detekcia stlačenia: Akonáhle sa tlačidlo A stlačí, BBC micro:bit detekuje túto udalosť.
2. Vykonanie kódu: Po detekcii stlačenia tlačidla A sa vykoná kód, ktorý je vložený do tohto príkazu.

Príklad:

Ak chceme zobraziť srdiečko, keď sa stlačí tlačidlo A, použijeme nasledujúci blok kódu:



Obrázok 12 Príkaz "keď sa tlačidlo A stlačí"

Po spustení tohto kódu sa na LED matici BBC micro:bitu zobrazí srdiečko vždy, keď stlačíte tlačidlo A.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Vyberte príkaz "keď sa tlačidlo A stlačí" a vložte ho do svojho programu.
3. Do príkazu vložte akciu, ktorú chcete vykonať, napríklad zobrazenie ikony srdiečka.
4. Spustíte svoj program a stlačíte tlačidlo A na micro:bitu, aby ste videli, ako sa akcia vykoná!

Tento príkaz je veľmi užitočný na vytváranie interaktívnych projektov, kde používateľ môže ovládať program pomocou tlačidiel na micro:bitu. Teraz ste pripravení experimentovať a tvoriť vlastné interaktívne programy!

Príkaz "keď potrasie"

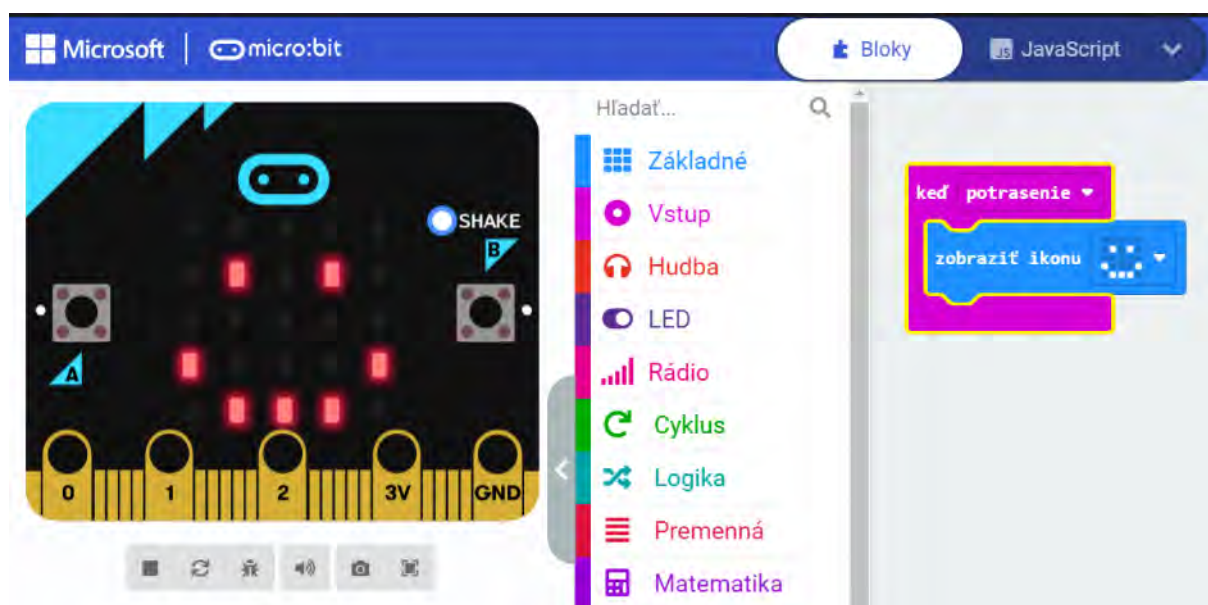
Príkaz "keď potrasie" slúži na vykonanie určitej akcie, keď používateľ potrasie micro:bitom. Micro:bit má zabudovaný akcelerometer, ktorý deteguje pohyby, vrátane trasenia.

Ako to funguje?

1. Detekcia trasenia: Akonáhle micro:bit deteguje trasenie, spustí sa kód vložený do tohto príkazu.
2. Vykonanie kódu: Po detekcii trasenia sa vykoná akcia, ktorú sme do príkazu vložili.

Príklad:

Ak chceme zobraziť smajlík, keď potrasíme micro:bitom, použijeme nasledujúci blok kódu:



Obrázok 13 Príkaz "keď potrasie"

Po spustení tohto kódu sa na LED matici micro:bitu zobrazí smajlík vždy, keď potrasiete micro:bitom.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Vyberte príkaz "keď potrasie" a vložte ho do svojho programu.
3. Do príkazu vložte akciu, ktorú chcete vykonať, napríklad zobrazenie ikony smajlíka.
4. Spustíte svoj program a potrasíte micro:bitom, aby ste videli, ako sa akcia vykoná!

Tento príkaz je veľmi užitočný na vytváranie interaktívnych projektov, kde môžete reagovať na pohyb BBC micro:bitu.

Príkaz "on pin P0 pressed"

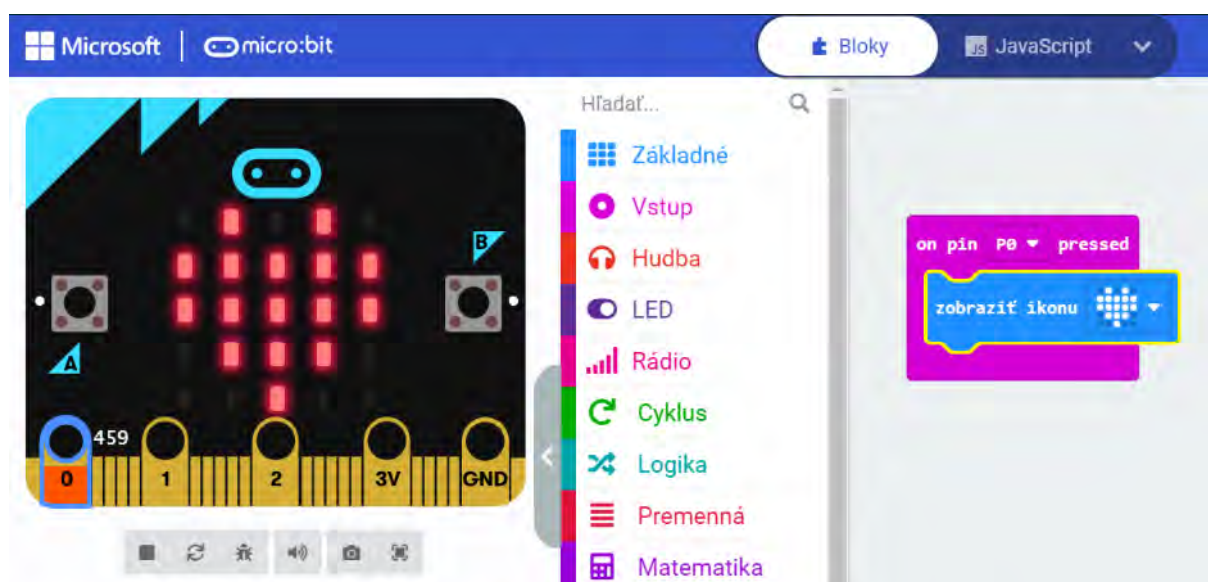
Príkaz "on pin P0 pressed" slúži na vykonanie určitej akcie, keď používateľ stlačí alebo pripojí niečo k pinu P0 na micro:bit. Piny P0, P1 a P2 sú malé kovové kontakty na spodnej strane BBC micro:bitu, ktoré sa dajú použiť na pripojenie rôznych externých komponentov, ako sú tlačidlá, svetelné senzory alebo motory.

Ako to funguje?

1. Detekcia stlačenia/pripojenia: Akonáhle micro:bit deteguje, že bol pin P0 stlačený alebo sa k nemu niečo pripojilo, spustí sa kód vložený do tohto príkazu.
2. Vykonanie kódu: Po detekcii tejto udalosti sa vykoná akcia, ktorú sme do príkazu vložili.

Príklad:

Ak chceme zobrazit' srdiečko, keď sa stlačí alebo pripojí k pinu P0, použijeme nasledujúci blok kódu:



Obrázok 14 Príkaz "on pin P0 pressed"

Po spustení tohto kódu sa na LED matici BBC micro:bitu zobrazí srdiečko vždy, keď stlačíte alebo pripojíte niečo k pinu P0.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Vyberte príkaz "on pin P0 pressed" a vložte ho do svojho programu.
3. Do príkazu vložte akciu, ktorú chcete vykonať, napríklad zobrazenie ikony srdiečka.
4. Spustite svoj program a stlačte pin P0 na micro:bitu alebo k nemu pripojte nejaký komponent, aby ste videli, ako sa akcia vykoná!

Tento príkaz je veľmi užitočný na vytváranie interaktívnych projektov, kde micro:bit reaguje na rôzne externé vstupy. Teraz ste pripravení experimentovať a tvoriť vlastné interaktívne programy!

Príkaz "tlačidlo A je stlačené"

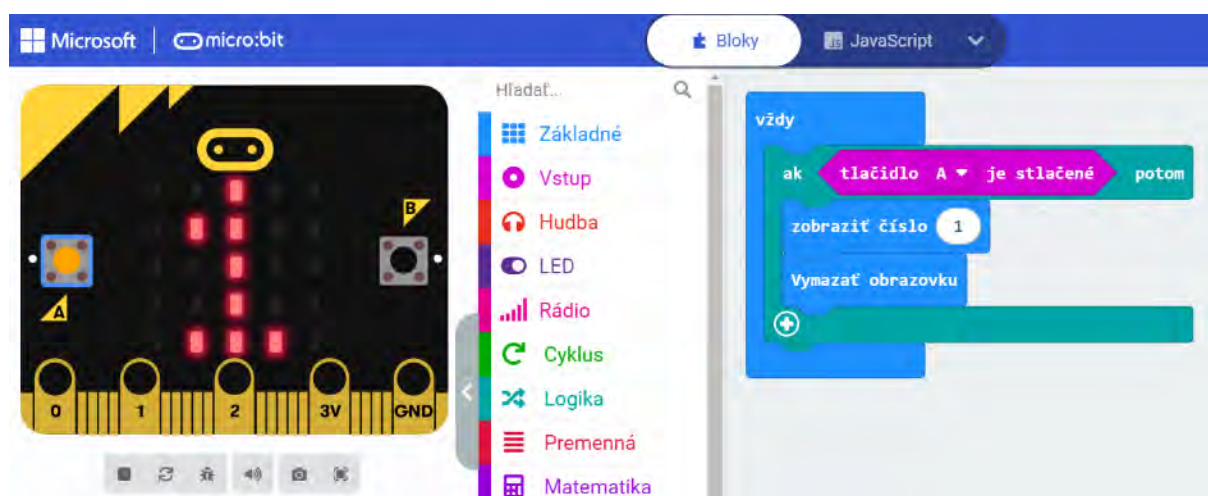
Príkaz "tlačidlo A je stlačené" slúži na zistenie, či je tlačidlo A na micro:bitu stlačené v danom okamihu. Tento príkaz sa používa na vykonanie určitej akcie, len keď je tlačidlo A stlačené.

Ako to funguje?

1. Kontrola stlačenia: Program neustále kontroluje, či je tlačidlo A stlačené.
2. Vykonanie kódu: Ak je tlačidlo A stlačené, vykoná sa kód vložený do tohto príkazu.

Príklad:

Ak chceme zobraziť číslo 1, keď je tlačidlo A stlačené, použijeme nasledujúci blok kódu:



Obrázok 15 Príkaz "tlačidlo A je stlačené"

Po spustení tohto kódu sa na LED matici micro:bitu zobrazí číslo 1, keď je tlačidlo A stlačené.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Vyberte príkaz "tlačidlo A je stlačené" a vložte ho do svojho programu.
3. Do príkazu vložte akciu, ktorú chcete vykonať, napríklad zobrazenie čísla.
4. Spustíte svoj program a stlačíte tlačidlo A na micro:bite, aby ste videli, ako sa akcia vykoná!

Tento príkaz je veľmi užitočný na vytváranie interaktívnych programov, kde používateľ môže ovládať program pomocou tlačidiel na BBC micro:bite.

Príkaz "akcelerácia (mg) x"

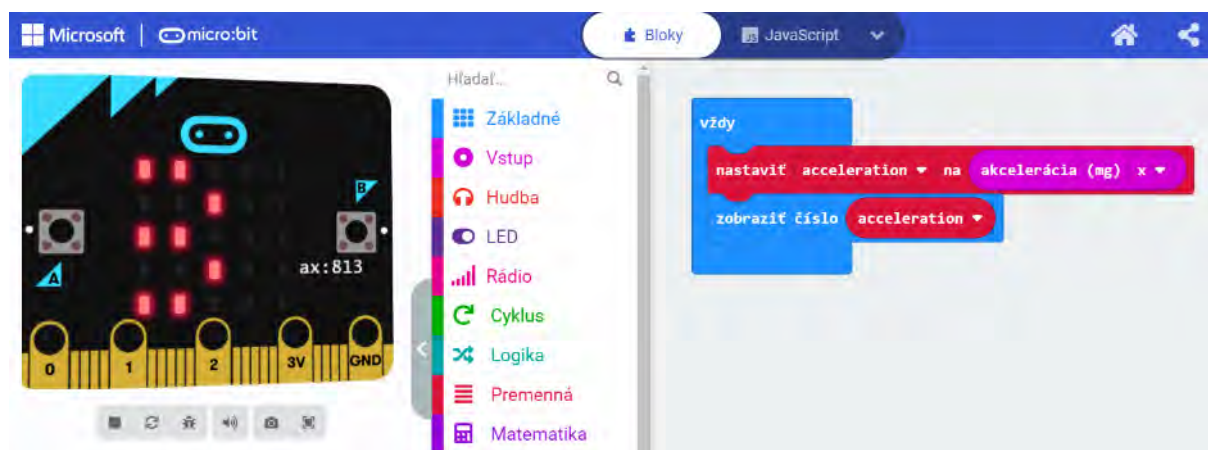
Príkaz "akcelerácia (mg) x" slúži na meranie zrýchlenia pozdĺž osi x na micro:bitu. Micro:bit má zabudovaný akcelerometer, ktorý môže merať zrýchlenie v miligravitoch (mg) pozdĺž osi x, y a z.

Ako to funguje?

1. Detekcia zrýchlenia: Akcelerometer na BBC micro:bitu meria zrýchlenie pozdĺž osi x.
2. Návrat hodnôt: Príkaz vráti hodnotu zrýchlenia v miligravitoch (mg), čo je tisícina g (gravitačnej sily).

Príklad:

Ak chceme zobrazit' zrýchlenie pozdĺž osi x na micro:bitu, použijeme nasledujúci blok kódu:



Obrázok 16 Príkaz "akcelerácia (mg) x"

Po spustení tohto kódu sa na LED matici BBC micro:bitu zobrazí hodnota zrýchlenia pozdĺž osi x v miligravitoch.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Vyberte príkaz "akcelerácia (mg) x" a vložte ho do svojho programu.
3. Uložte hodnotu zrýchlenia do premennej a zobrazte ju na LED matici.
4. Spustite svoj program a sledujte, ako sa hodnota zrýchlenia pozdĺž osi x zobrazuje na vašom micro:bit!

Tento príkaz je veľmi užitočný na meranie pohybu a tvorbu interaktívnych projektov, kde potrebujete vedieť, ako sa micro:bit pohybuje.

Príkaz "kolík P0 je stlačený"

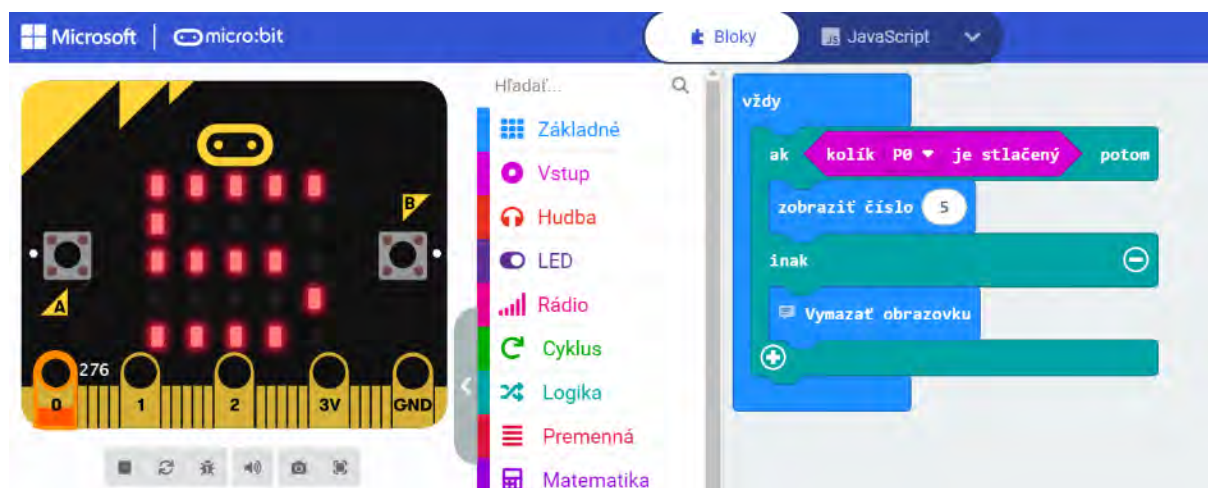
Príkaz "kolík P0 je stlačený" slúži na zistenie, či je pin (kolík) P0 na micro:bitu stlačený alebo pripojený v danom okamihu. Tento príkaz sa používa na vykonanie určitej akcie, len keď je kolík P0 stlačený alebo pripojený.

Ako to funguje?

1. Detekcia stlačenia/pripojenia: Program neustále kontroluje, či je kolík P0 stlačený alebo pripojený.
2. Vykonanie kódu: Ak je kolík P0 stlačený, vykoná sa kód vložený do tohto príkazu.

Príklad:

Ak chceme zobrazit' číslo 5, keď je kolík P0 stlačený, použijeme nasledujúci blok kódu:



Obrázok 17 Príkaz "kolík P0 je stlačený"

Po spustení tohto kódu sa na LED matici BBC micro:bitu zobrazí číslo 5, keď je kolík P0 stlačený alebo k nemu pripojený nejaký komponent.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Vyberte príkaz "kolík P0 je stlačený" a vložte ho do svojho programu.
3. Do príkazu vložte akciu, ktorú chcete vykonať, napríklad zobrazenie čísla.
4. Spustíte svoj program a stlačte kolík P0 na BBC micro:bitu alebo k nemu pripojte nejaký komponent, aby ste videli, ako sa akcia vykoná!

Tento príkaz je veľmi užitočný na vytváranie interaktívnych programov, kde micro:bit reaguje na rôzne externé vstupy.

Príkaz "úroveň osvetlenia"

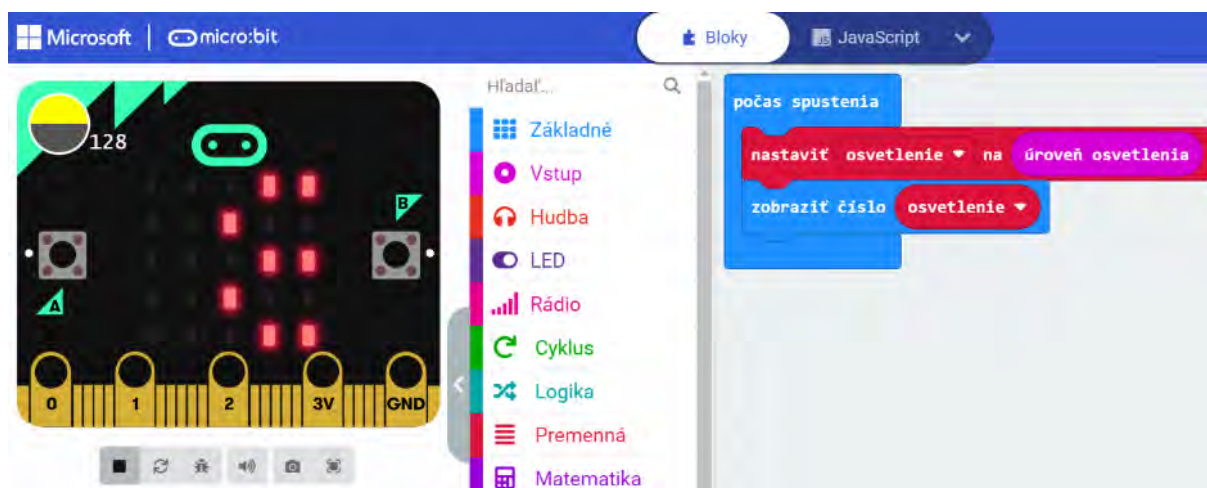
Príkaz "úroveň osvetlenia" slúži na meranie intenzity svetla, ktoré zachytí micro:bit. Micro:bit má zabudovaný svetelný senzor, ktorý dokáže merať úroveň okolitého osvetlenia v luxoch.

Ako to funguje?

1. Meranie osvetlenia: Svetelný senzor na micro:bitu detekuje intenzitu svetla v okolí a vráti hodnotu v luxoch.
2. Použitie hodnoty: Hodnotu osvetlenia môžeme použiť v rôznych projektoch, napríklad na automatické zapnutie svetla pri tme alebo na meranie denného svetla.

Príklad:

Ak chceme zobraziť úroveň osvetlenia na LED matici BBC micro:bitu, použijeme nasledujúci blok kódu:



Obrázok 18 Príkaz "úroveň osvetlenia"

Po spustení tohto kódu sa na LED matici BBC micro:bitu zobrazí aktuálna úroveň osvetlenia v luxoch.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Vyberte príkaz "úroveň osvetlenia" a vložte ho do svojho programu.
3. Uložte hodnotu osvetlenia do premennej a zobrazte ju na LED matici.
4. Spustíte svoj program a sledujte, ako sa úroveň osvetlenia zobrazuje na vašom BBC micro:bite!

Príkaz "smerovanie kompasu"

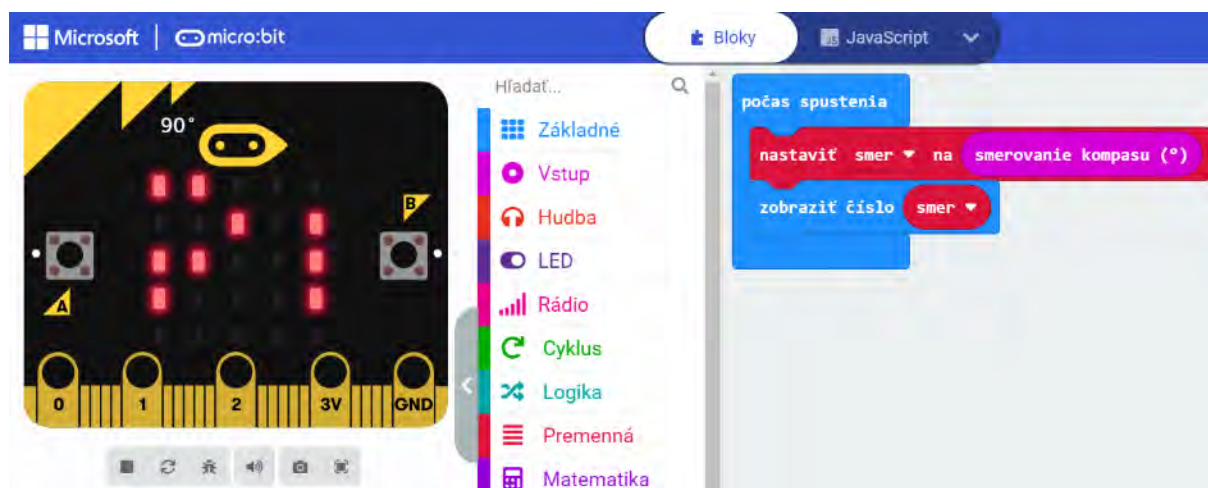
Príkaz "smerovanie kompasu" slúži na meranie smeru pomocou zabudovaného kompasu v BBC micro:bitu. Kompas dokáže určiť, ktorým smerom je zariadenie orientované vzhľadom na magnetický sever.

Ako to funguje?

1. Detekcia smeru: Kompas v BBC micro:bitu deteguje smer a určuje hodnotu uhla v stupňoch, kde 0° predstavuje sever.
2. Návrat hodnôt: Príkaz vráti hodnotu uhla, ktorý ukazuje smerovanie BBC micro:bitu v stupňoch.

Príklad:

Ak chceme zobrazíť smerovanie kompasu na LED matici BBC micro:bitu, použijeme nasledujúci blok kódu:



Obrázok 19 Príkaz "smerovanie kompasu"

Po spustení tohto kódu sa na LED matici micro:bitu zobrazí hodnota uhla v stupňoch, ktorá ukazuje aktuálny smer kompasu.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Vyberte príkaz "smerovanie kompasu" a vložte ho do svojho programu.
3. Uložte hodnotu smerovania do premennej a zobrazte ju na LED matici.
4. Spustite svoj program a sledujte, ako sa hodnota uhla zobrazuje na vašom BBC micro:bite!

Tento príkaz je veľmi užitočný na meranie smeru a tvorbu projektov, ktoré využívajú orientáciu v priestore.

Príkaz "teplota"

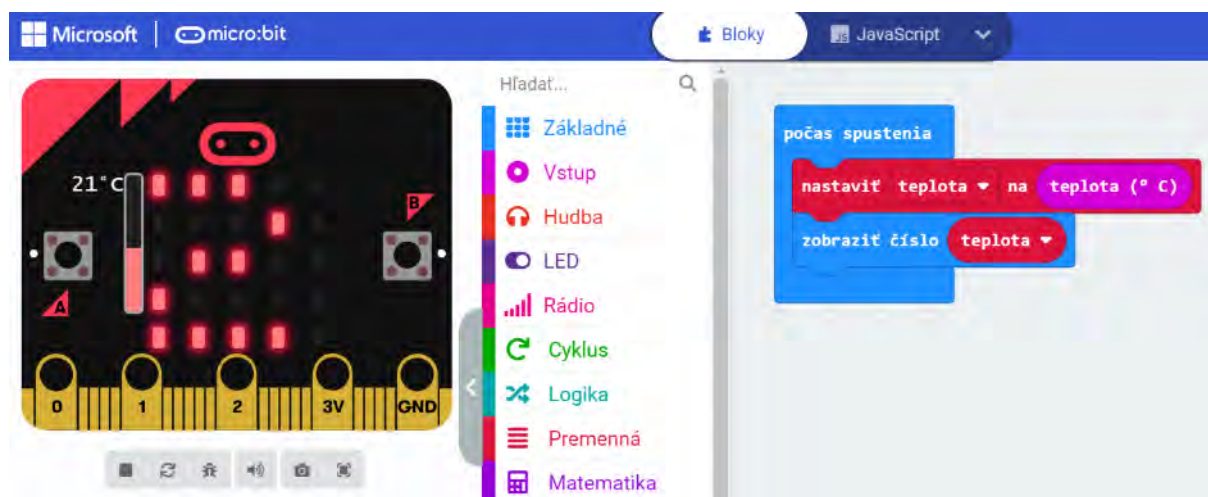
Príkaz "teplota" slúži na meranie aktuálnej teploty okolitého prostredia pomocou zabudovaného teplotného senzora v BBC micro:bitu. Táto hodnota sa udáva v stupňoch Celzia.

Ako to funguje?

1. Meranie teploty: Teplotný senzor na BBC micro:bitu deteguje aktuálnu teplotu okolitého prostredia.
2. Návrat hodnôt: Príkaz vráti hodnotu teploty v stupňoch Celzia, ktorú môžeme použiť v rôznych projektoch, napríklad na zobrazenie aktuálnej teploty na LED matici.

Príklad:

Ak chceme zobraziť aktuálnu teplotu na LED matici micro:bitu, použijeme nasledujúci blok kódu:



Obrázok 20 Príkaz "teplota"

Po spustení tohto kódu sa na LED matici BBC micro:bitu zobrazí aktuálna teplota v stupňoch Celzia.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Vyberte príkaz "teplota" a vložte ho do svojho programu.
3. Uložte hodnotu teploty do premennej a zobrazte ju na LED matici.
4. Spustite svoj program a sledujte, ako sa aktuálna teplota zobrazuje na vašom BBC micro:bite!

Tento príkaz je veľmi užitočný na meranie teplotných podmienok a tvorbu projektov, ktoré reagujú na zmeny teploty.

Príkaz "je gesto potrasenie"

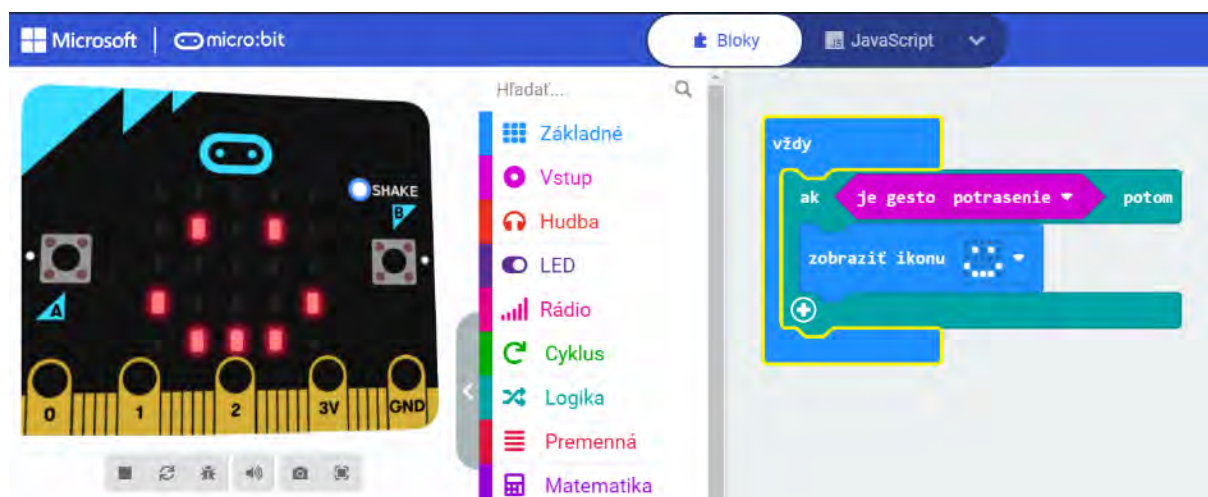
Príkaz "je gesto potrasenie" slúži na zistenie, či bolo BBC micro:bitom potrasené. BBC micro:bit má zabudovaný akcelerometer, ktorý dokáže detekovať rôzne gestá, vrátane trasenia.

Ako to funguje?

1. Detekcia gestá: Akonáhle BBC micro:bit deteguje gesto potrasenia, príkaz vráti hodnotu true (pravda), inak vráti hodnotu false (nepravda).
2. Vykonanie kódu: Tento príkaz sa často používa v podmienkových blokoch, aby sa určité akcie vykonali len vtedy, keď je gesto potrasenie detegované.

Príklad:

Ak chceme zobrazit' smajlík, keď BBC micro:bit deteguje gesto potrasenie, použijeme nasledujúci blok kódu:



Obrázok 21 Príkaz "je gesto potrasenie"

Po spustení tohto kódu sa na LED matici BBC micro:bitu zobrazí smajlík vždy, keď deteguje gesto potrasenie.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Vyberte príkaz "je gesto potrasenie" a vložte ho do svojho programu.
3. Do príkazu vložte akciu, ktorú chcete vykonať, napríklad zobrazenie ikony smajlíka.
4. Spustíte svoj program a potraсте BBC micro:bitom, aby ste videli, ako sa akcia vykoná!

Tento príkaz je veľmi užitočný na vytváranie interaktívnych programov, kde micro:bit reaguje na rôzne gestá.

Príkaz "keď hlasný zvuk"

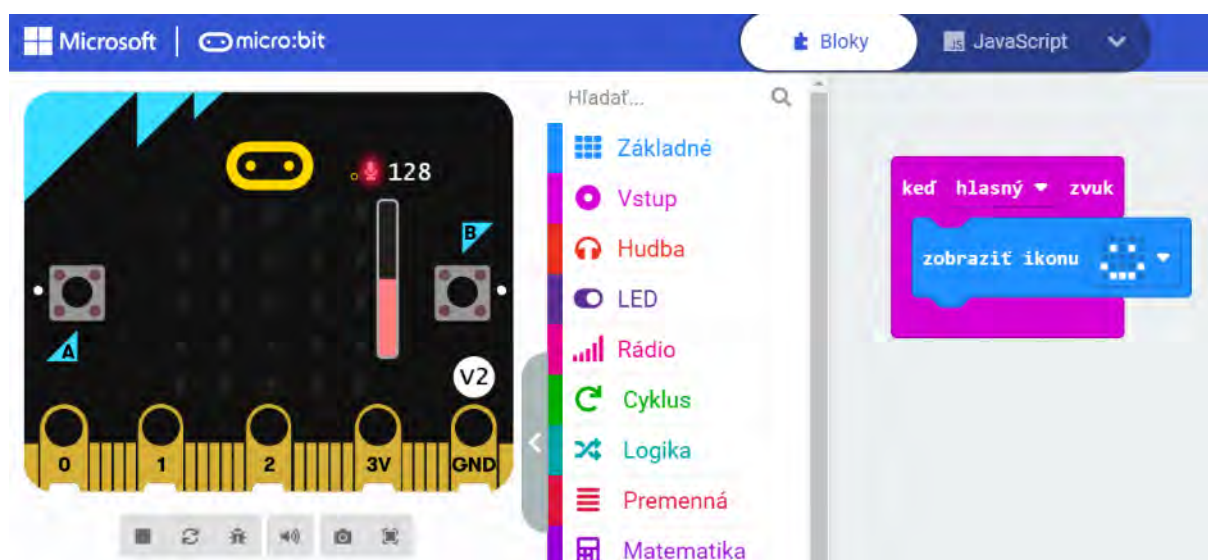
Príkaz "keď hlasný zvuk" slúži na vykonanie určitej akcie, keď BBC micro:bit deteguje hlasný zvuk. BBC micro:bit má zabudovaný mikrofón, ktorý dokáže zachytiť zvuky z prostredia.

Ako to funguje?

1. Detekcia zvuku: Akonáhle mikrofón na BBC micro:bitu deteguje hlasný zvuk, spustí sa kód vložený do tohto príkazu.
2. Vykonanie kódu: Po detekcii hlasného zvuku sa vykoná akcia, ktorú sme do príkazu vložili.

Príklad:

Ak chceme zobraziť hviezdičku, keď BBC micro:bit deteguje hlasný zvuk, použijeme nasledujúci blok kódu:



Obrázok 22 Príkaz "keď hlasný zvuk"

Po spustení tohto kódu sa na LED matici BBC micro:bitu zobrazí hviezdička vždy, keď deteguje hlasný zvuk.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Vyberte príkaz "keď hlasný zvuk" a vložte ho do svojho programu.
3. Do príkazu vložte akciu, ktorú chcete vykonať, napríklad zobrazenie ikony hviezdičky.
4. Spustíte svoj program a urobte hlasný zvuk blízko BBC micro:bitu, aby ste videli, ako sa akcia vykoná!

Tento príkaz je veľmi užitočný na vytváranie interaktívnych projektov, kde BBC micro:bit reaguje na zvukové podnety z prostredia.

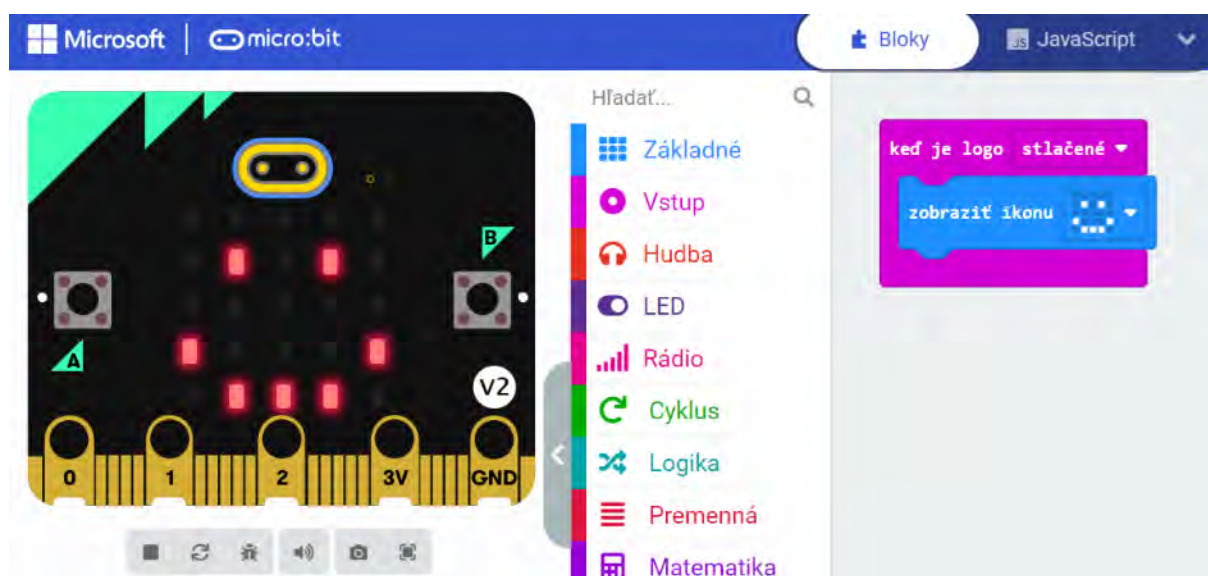
Príkaz "keď je logo stlačené"

Príkaz "keď je logo stlačené" slúži na vykonanie určitej akcie, keď sa stlačí logo na BBC micro:bite. Logo na BBC micro:bite je dotyková plocha, ktorá funguje podobne ako tlačidlo.

Ako to funguje?

1. Detekcia stlačenia: Akonáhle sa logo na BBC micro:bite stlačí, spustí sa kód vložený do tohto príkazu.
2. Vykonanie kódu: Po detekcii stlačenia loga sa vykoná akcia, ktorú sme do príkazu vložili.

Príklad: Ak chceme zobrazíť úsmev, keď sa logo na BBC micro:bite stlačí, použijeme nasledujúci blok kódu:



Obrázok 23 Príkaz "keď je logo stlačené"

Po spustení tohto kódu sa na LED matici BBC micro:bitu zobrazí úsmev vždy, keď sa stlačí logo.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Vyberte príkaz "keď logo stlačené" a vložte ho do svojho programu.
3. Do príkazu vložte akciu, ktorú chcete vykonať, napríklad zobrazenie ikony úsmevu.
4. Spustíte svoj program a stlačíte logo na BBC micro:bite, aby ste videli, ako sa akcia vykoná!

Tento príkaz je veľmi užitočný na vytváranie interaktívnych projektov, kde BBC micro:bit reaguje na dotykové podnety.

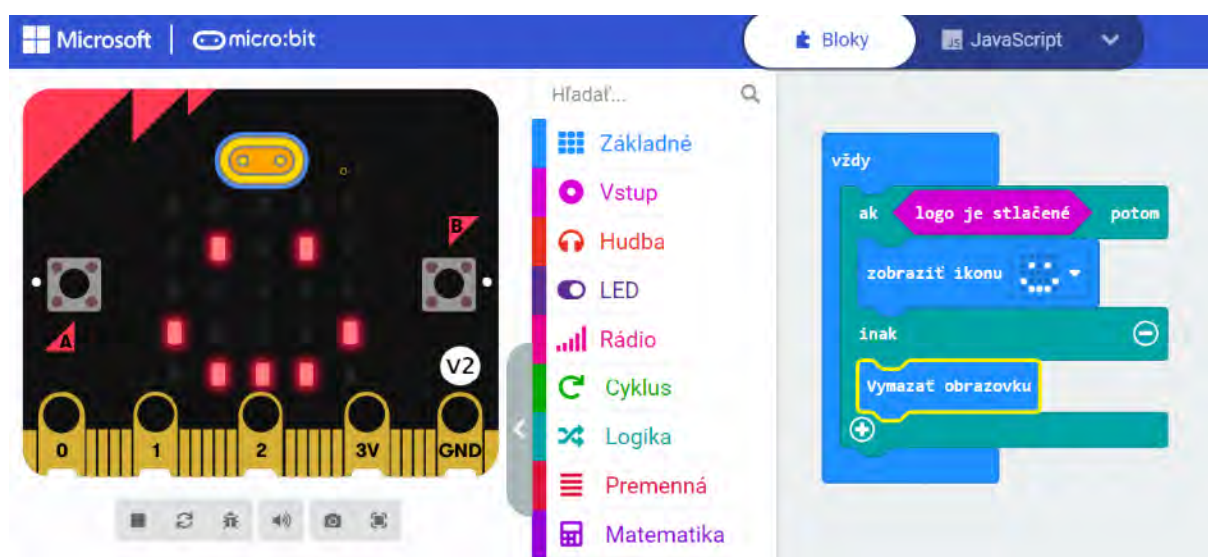
Príkaz "logo je stlačené"

Príkaz "logo je stlačené" slúži na kontrolu, či je práve logo na BBC micro:bit stlačené. Ak je logo stlačené vykoná sa jeden príkaz, ak stlačené nie je vykoná sa iný príkaz.

Ako to funguje?

1. Detekcia stlačenia: Mikrokontrolér deteguje nepretržité stlačenie loga na BBC micro:bit.
2. Vykonanie kódu: Pokiaľ je logo stlačené, príkaz vykonáva zadaný kód.

Príklad: Ak chceme, aby BBC micro:bit zobrazuje srdce počas toho, ako je logo stlačené a čistú obrazovku pokiaľ logo stlačené nie je, použijeme nasledujúci blok kódu:



Obrázok 24 Príkaz "logo je stlačené"

Po spustení tohto kódu sa na LED matici BBC micro:bitu zobrazuje srdce vždy, keď je logo stlačené a čistá obrazovka ak logo stlačené nie je.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Vyberte príkaz "logo je stlačené" a vložte ho do svojho programu.
3. Do príkazu vložte akciu, ktorú chcete vykonať, aj je logo stlačené napríklad zobrazenie ikony úsmevu.
4. Do príkazu vložte akciu, ktorú chcete vykonať, ak nie je logo stlačené napríklad vymazanie obrazovky.
5. Spustíte svoj program a stlačte logo na BBC micro:bit, aby ste videli, ako sa akcia vykoná!

Tento príkaz je veľmi užitočný na vytváranie projektov, kde BBC micro:bit reaguje na dotykové podnety v reálnom čase.

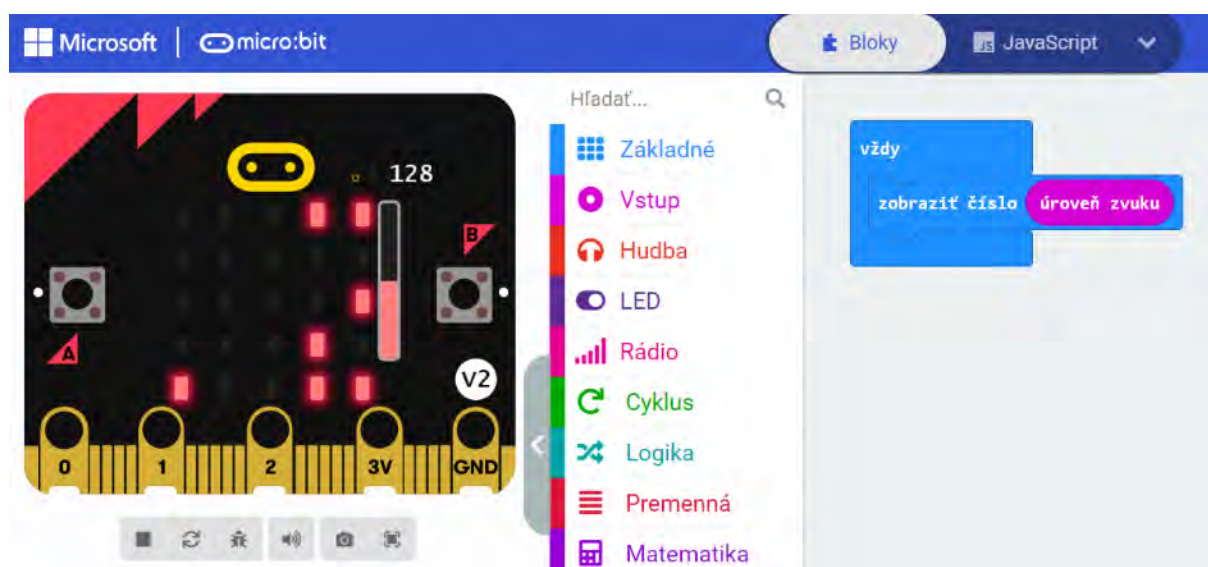
Príkaz "úroveň zvuku"

Príkaz "úroveň zvuku" slúži na meranie intenzity zvuku v prostredí, v ktorom sa BBC micro:bit nachádza. Tento príkaz využíva zabudovaný mikrofón BBC micro:bitu na detekciu úrovne zvuku a následné vykonanie akcie založenej na nameranej hodnote.

Ako to funguje?

1. Meranie zvuku: Mikrofón na BBC micro:bite neustále sníma úroveň zvuku v okolí.
2. Vykonanie kódu: Na základe nameranej úrovne zvuku môže BBC micro:bit vykonať akciu, ktorú sme zadali do príkazu.

Príklad: Ak chceme, aby BBC micro:bit zobrazoval číslo, ktoré predstavuje úroveň zvuku, použijeme nasledujúci blok kódu:



Obrázok 25 Príkaz "úroveň zvuku"

Po spustení tohto kódu bude BBC micro:bit nepretržite zobrazovať aktuálnu úroveň zvuku na LED matici.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Použite príkaz "úroveň zvuku" v rámci svojho programu.
3. Do príkazu vložte akciu, ktorú chcete vykonať, napríklad zobrazenie nameranej hodnoty.
4. Spustite svoj program a sledujte, ako BBC micro:bit zobrazuje úroveň zvuku vo vašom okolí!

Tento príkaz je veľmi užitočný na vytváranie projektov, kde BBC micro:bit reaguje na zmeny v prostredí na základe úrovne zvuku.

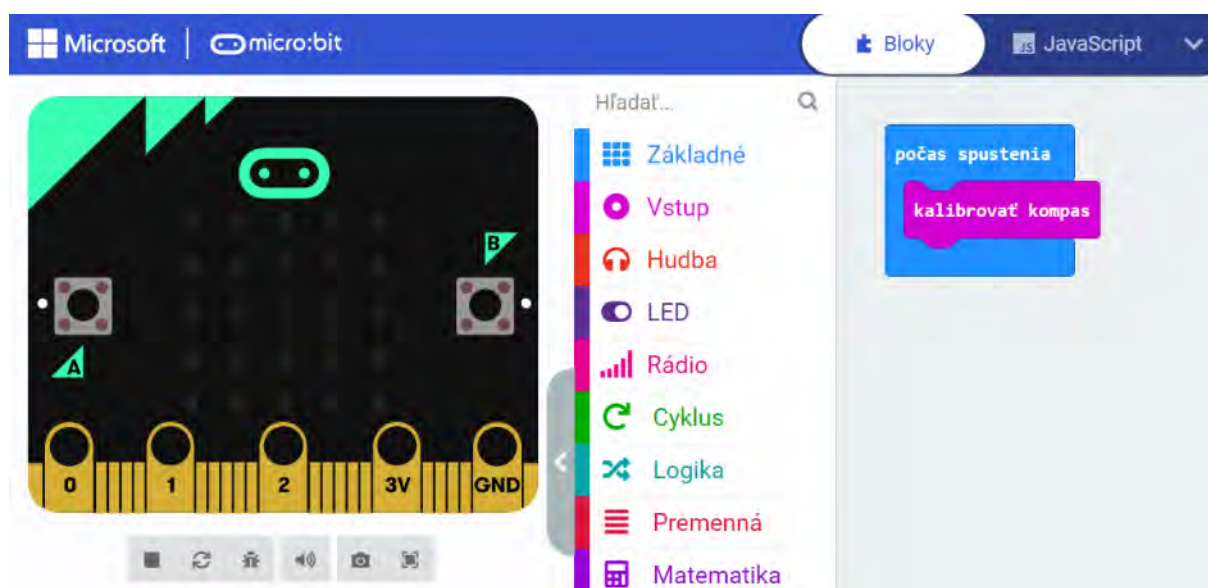
Príkaz "kalibrovať kompas"

Príkaz "kalibrovať kompas" slúži na kalibráciu kompasu na BBC micro:bit. Kompas je senzor, ktorý dokáže detegovať magnetické pole Zeme a určiť smer severu. Pred použitím kompasu je potrebné ho kalibrovať, aby poskytoval presné údaje.

Ako to funguje?

1. Spustenie kalibrácie: Kalibrácia začína zadaním príkazu "kalibrovať kompas". BBC micro:bit vás vyzve, aby ste ním pohybovali v určitých smeroch.
2. Kalibračný proces: Počas kalibrácie musíte pohybovať BBC micro:bitom, aby ste vyplnili všetky body na obrazovke. Tento proces pomáha kompasu získať presné údaje o okolitom magnetickom poli.
3. Ukončenie kalibrácie: Keď sú všetky body na obrazovke vyplnené, kalibrácia je dokončená a kompas je pripravený na použitie.

Príklad: Ak chcete začať kalibráciu kompasu, použijeme nasledujúci blok kódu:



Obrázok 26 Príkaz "kalibrovať kompas"

Tento príkaz spustí kalibračný proces na BBC micro:bit.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Použite príkaz "kalibrovať kompas" v rámci svojho programu.
3. Sledujte inštrukcie na obrazovke BBC micro:bitu a pohybujte ním podľa potreby.
4. Po dokončení kalibrácie je kompas pripravený na použitie!

Tento príkaz je nevyhnutný na zabezpečenie presnosti údajov kompasu a je veľmi užitočný pri projektoch, ktoré vyžadujú orientáciu na základe smeru.

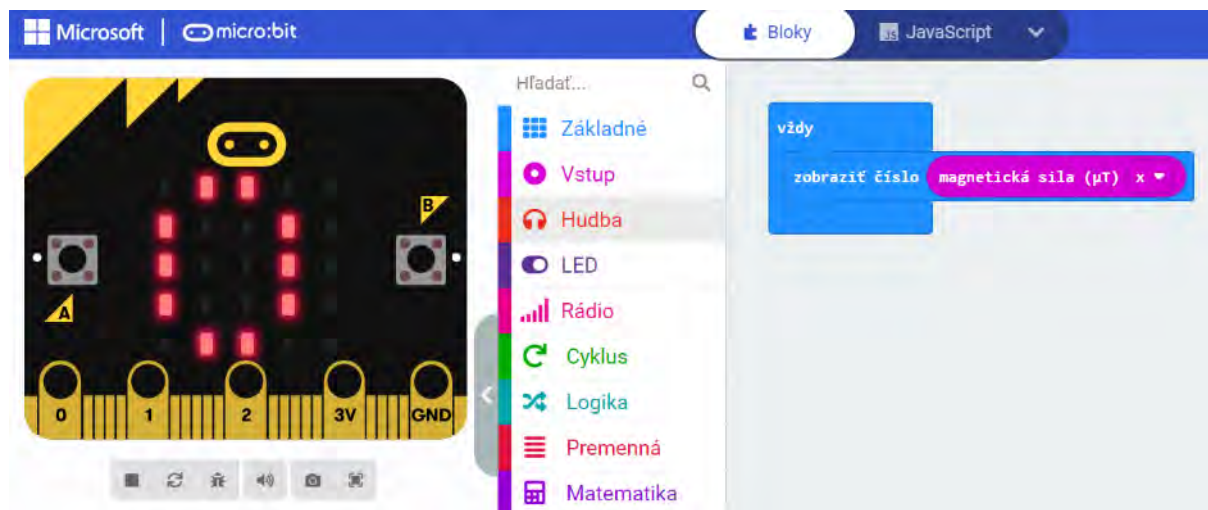
Príkaz "magnetická sila (μT) x"

Príkaz "magnetická sila (μT) x" slúži na meranie komponentu magnetického poľa pozdĺž osi X v mikrotésle (μT). BBC micro:bit má zabudovaný magnetometer, ktorý umožňuje meranie intenzity magnetického poľa v troch osiach: X, Y a Z.

Ako to funguje?

1. Meranie magnetického poľa: Magnetometer na BBC micro:bite neustále sníma intenzitu magnetického poľa pozdĺž osi X.
2. Vykonanie kódu: Na základe nameranej hodnoty magnetického poľa môže micro:bit vykonať akciu, ktorú sme zadali do príkazu.

Príklad: Ak chceme zobraziť hodnotu intenzity magnetického poľa pozdĺž osi X, použijeme nasledujúci blok kódu:



Obrázok 27 Príkaz "magnetická sila (μT) x"

Po spustení tohto kódu bude BBC micro:bit nepretržite zobrazovať aktuálnu intenzitu magnetického poľa pozdĺž osi X na LED matici.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Použite príkaz "magnetická sila (μT) x" v rámci svojho programu.
3. Do príkazu vložte akciu, ktorú chcete vykonať, napríklad zobrazenie nameranej hodnoty.
4. Spustíte svoj program a sledujte, ako BBC micro:bit meria intenzitu magnetického poľa pozdĺž osi X!

Tento príkaz je veľmi užitočný na vytváranie projektov, kde BBC micro:bit reaguje na zmeny v magnetickom poli.

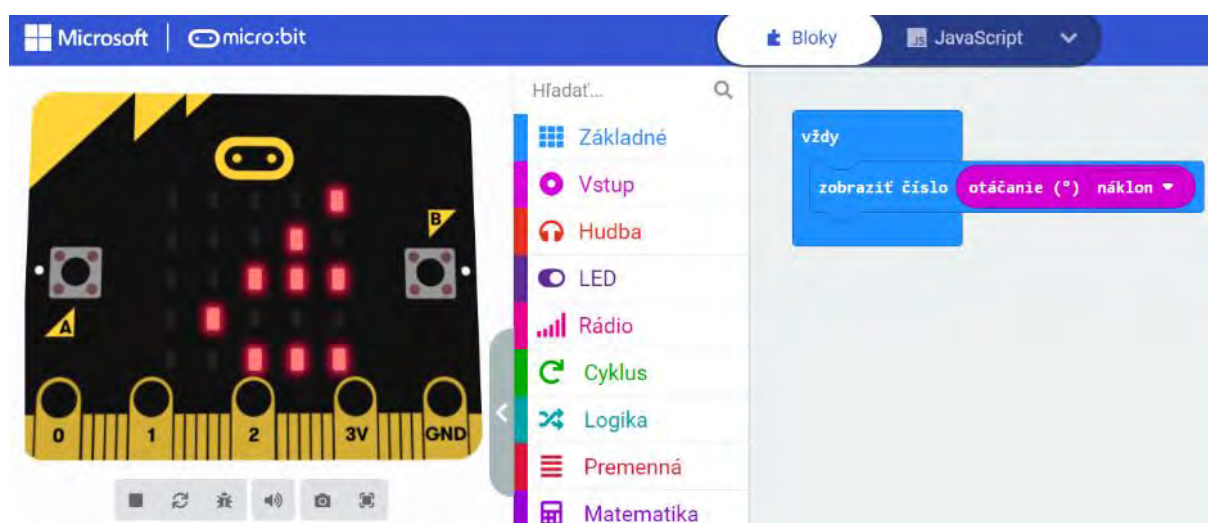
Príkaz "otáčania (°) náklon"

Príkaz "otáčania (°) náklon" slúži na meranie uhla náklonu BBC micro:bita. Tento príkaz umožňuje detekovať, ako je BBC micro:bit naklonený pozdĺž osi X alebo Y v stupňoch (°). Je to užitočné na meranie pohybu a orientácie zariadenia.

Ako to funguje?

1. Meranie náklonu: BBC micro:bit neustále sníma náklon pozdĺž zvolenej osi (X alebo Y).
2. Vykonanie kódu: Na základe nameraného uhla náklonu môže BBC micro:bit vykonať akciu, ktorú sme zadali do príkazu.

Príklad: Ak chceme zobrazit' uhol náklonu pozdĺž osi X, použijeme nasledujúci blok kódu:



Obrázok 28 Príkaz "otáčania (°) náklon"

Po spustení tohto kódu bude BBC micro:bit nepretržite zobrazovať aktuálny uhol náklonu pozdĺž osi X na LED matici.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Použite príkaz "otáčania (°) náklon" v rámci svojho programu.
3. Do príkazu vložte akciu, ktorú chcete vykonať, napríklad zobrazenie nameranej hodnoty.
4. Spustite svoj program a sledujte, ako BBC micro:bit meria náklon vo vašom okolí!

Tento príkaz je veľmi užitočný na vytváranie projektov, kde micro:bit reaguje na zmeny v orientácii a pohybe.

Príkaz "keď sa kolík P0 uvoľní"

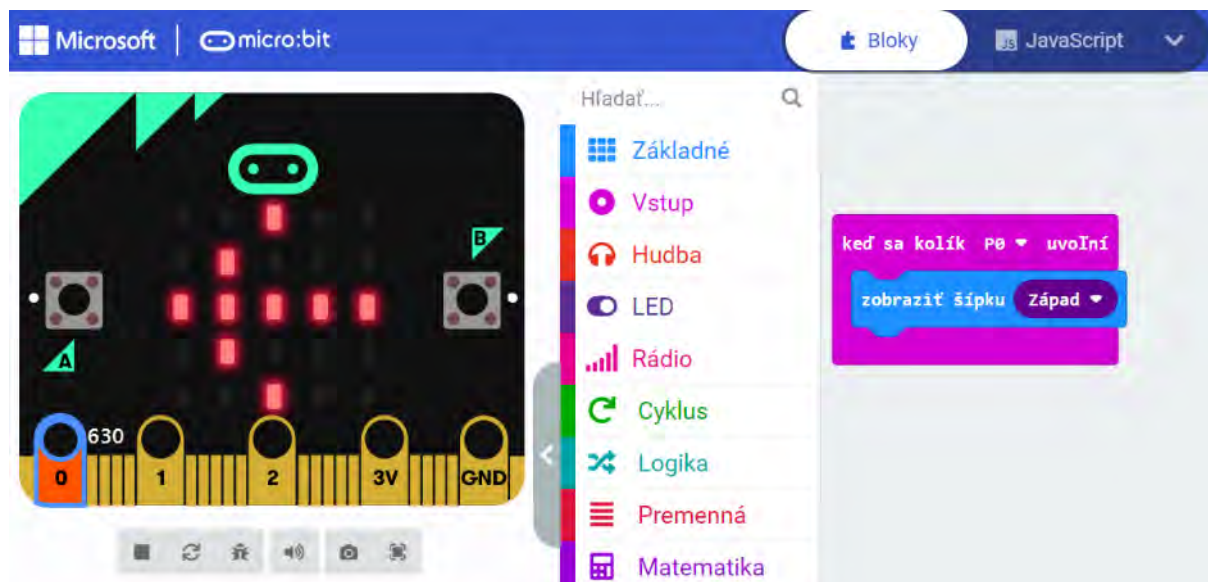
Príkaz "keď sa kolík P0 uvoľní" slúži na vykonanie určitej akcie, keď sa odpojí vstupný kolík P0 na micro:bit. Tento príkaz reaguje na zmenu stavu z pripojeného na odpojený.

Ako to funguje?

1. Detekcia odpojenia: Micro:bit neustále monitoruje stav kolíka P0.
2. Vykonanie kódu: Keď sa kolík P0 uvoľní, spustí sa kód vložený do tohto príkazu.

Príklad:

Ak chceme, aby sa pri uvoľnení kolíka P0 zobrazila na micro:bit ikona šípu, použijeme nasledujúci blok kódu:



Obrázok 29 Príkaz "keď sa kolík P0 uvoľní"

Po spustení tohto kódu sa na LED matici micro:bitu zobrazí ikona šípu vždy, keď sa kolík P0 uvoľní.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Použite príkaz "keď sa kolík P0 uvoľní" vo svojom programe.
3. Do príkazu vložte akciu, ktorú chcete vykonať, napríklad zobrazenie ikony šípu.
4. Spustíte svoj program a skúste uvoľniť kolík P0, aby ste videli, ako sa akcia vykoná!

Tento príkaz je veľmi užitočný na vytváranie projektov, kde micro:bit reaguje na zmeny v stave pripojenia periférnych zariadení.

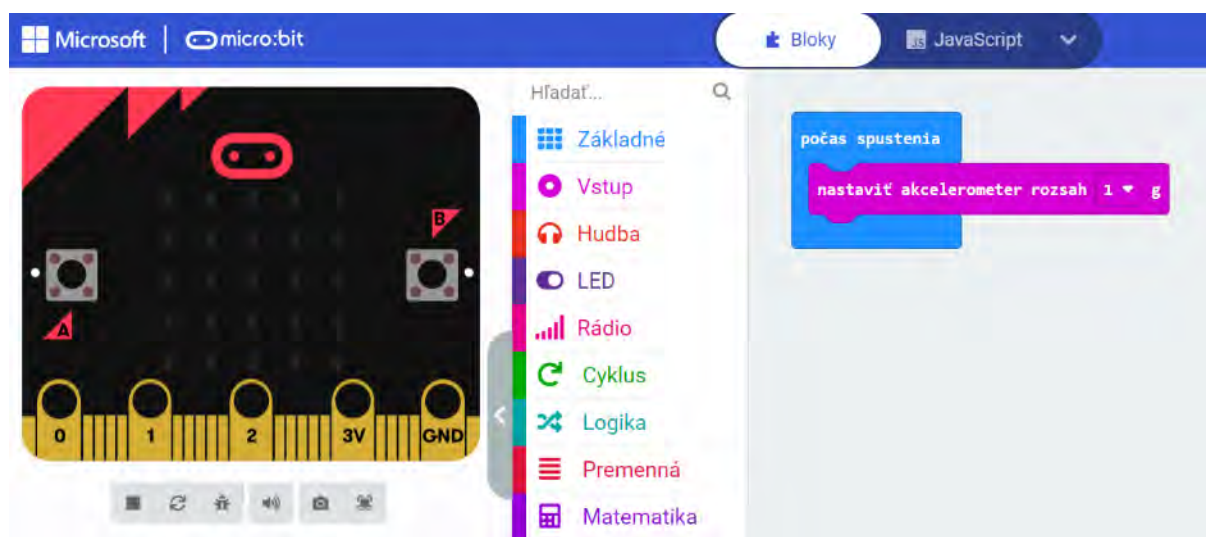
Príkaz "nastaviť akcelerometer rozsah 1g"

Príkaz "nastaviť akcelerometer rozsah 1g" slúži na nastavenie citlivosti akcelerometra na BBC micro:bit na rozsah 1g. Akcelerometer je senzor, ktorý meria zrýchlenie pozdĺž všetkých troch osí (X, Y, Z).

Ako to funguje?

1. Nastavenie rozsahu: Tento príkaz nastaví citlivosť akcelerometra na 1g ($9,8 \text{ m/s}^2$), čo znamená, že akcelerometer bude detegovať zrýchlenie až do hodnoty 1g.
2. Presné merania: Nastavením menšieho rozsahu, ako je 1g, dosiahnete vyššiu citlivosť a presnosť merania pre menšie zrýchlenia.

Príklad: Ak chcete nastaviť rozsah akcelerometra na 1g, použijete nasledujúci blok kódu:



Obrázok 30 Príkaz "nastaviť akcelerometer rozsah 1g"

Po spustení tohto kódu bude akcelerometer na micro:bit nastavený na meranie zrýchlenia v rozsahu 1g.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Použite príkaz "nastaviť akcelerometer rozsah 1g" vo svojom programe.
3. Spustíte svoj program a sledovať, ako BBC micro:bit meria zrýchlenie vo vašom prostredí s nastaveným rozsahom 1g!

Tento príkaz je veľmi užitočný na presné meranie menších zrýchlení a je ideálny na projekty, kde je dôležitá citlivosť na menšie pohyby.

Príkaz "nastaviť pre hlasný zvuk prahovú hodnotu na 128"

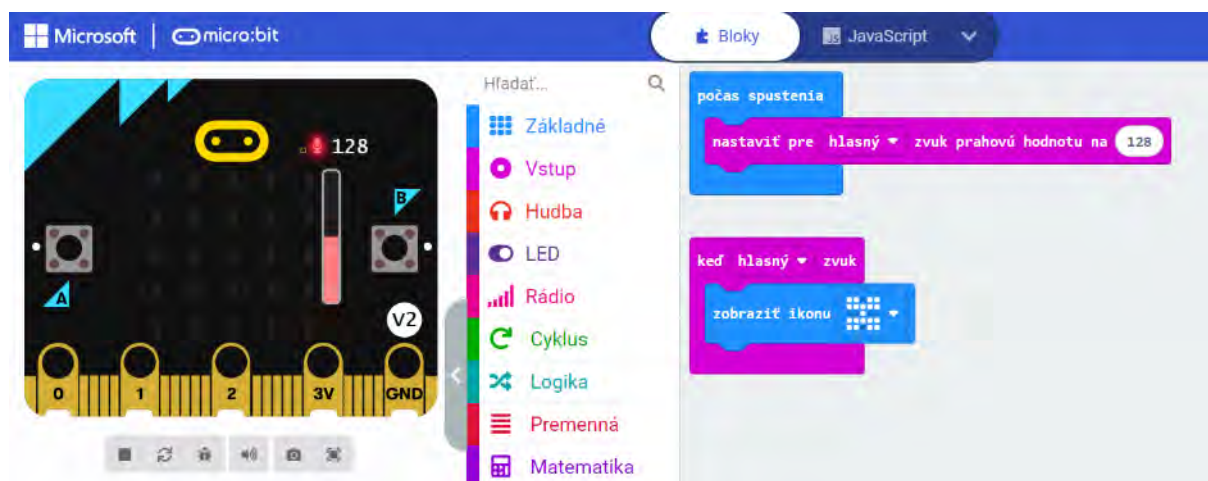
Príkaz "nastaviť pre hlasný zvuk prahovú hodnotu na 128" slúži na nastavenie citlivosti mikrofónu na BBC micro:bit tak, aby detegoval hlasný zvuk iba vtedy, keď úroveň zvuku prekročí stanovenú prahovú hodnotu. V tomto prípade je prahová hodnota nastavená na 128.

Ako to funguje?

1. Nastavenie prahu: Tento príkaz nastaví úroveň zvuku, ktorá musí byť prekročená, aby bol detegovaný hlasný zvuk.
2. Detekcia hlasného zvuku: Keď mikrofón na BBC micro:bit zaznamená zvukovú úroveň vyššiu ako 128, vykoná sa akcia vložená do príslušného príkazu.

Príklad:

Ak chceme, aby BBC micro:bit zobrazil ikonu hviezdčky, keď úroveň zvuku prekročí 128, použijeme nasledujúci blok kódu:



Obrázok 31 Príkaz "nastaviť pre hlasný zvuk prahovú hodnotu na 128"

Po spustení tohto kódu sa na LED matici micro:bitu zobrazí motýľ, keď úroveň zvuku prekročí prahovú hodnotu 128.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Použite príkaz "nastaviť pre hlasný zvuk prahovú hodnotu na 128" vo svojom programe.
3. Nastavte akciu, ktorú chcete vykonať pri detekcii hlasného zvuku, napríklad zobrazenie ikony motýľa.
4. Spustíte svoj program a vytvorte hlasný zvuk blízko BBC micro:bitu, aby ste videli, ako sa akcia vykoná!

Tento príkaz je užitočný na vytváranie projektov, kde chcete, aby BBC micro:bit reagoval na hlasné zvuky z prostredia.

4 Hudba

Príkaz "play melody at tempo (bpm) until done"

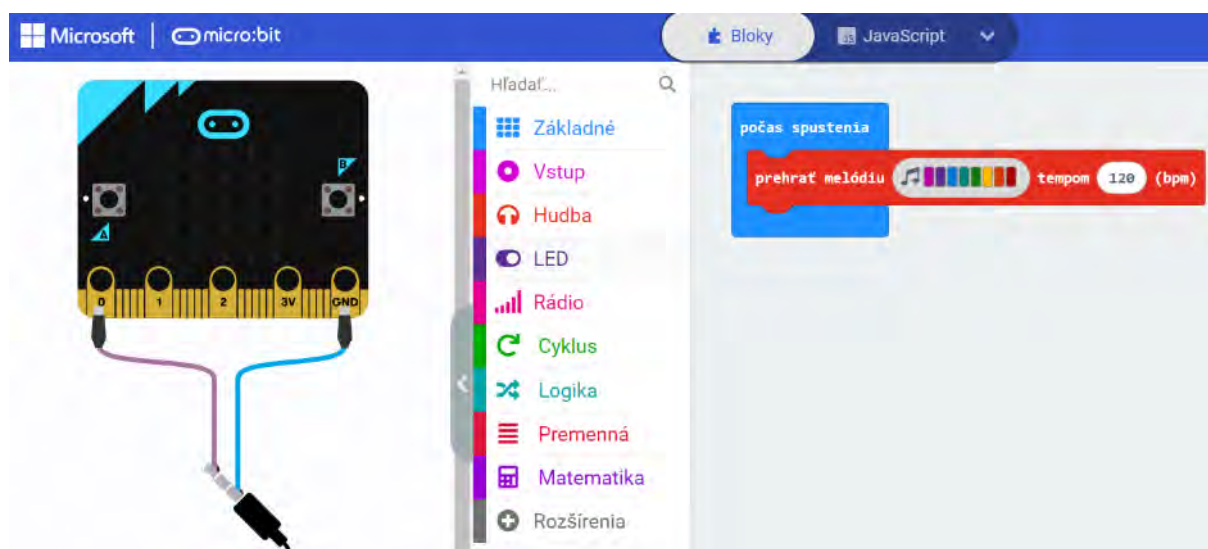
Príkaz "play melody at tempo (bpm) until done" slúži na prehranie melódie pri zadanom tempe (v úderoch za minútu - BPM) až do dokončenia. Tento príkaz využíva vstavaný piezoreproduktor micro:bita na prehrávanie hudby.

Ako to funguje?

1. Nastavenie tempa: Tento príkaz umožňuje nastaviť tempo melódie v BPM, ktoré určuje rýchlosť prehrávania melódie.
2. Prehrávanie melódie: Melódia sa prehrá pri zadanom tempe, a to až do jej dokončenia.

Príklad:

Ak chceme prehrávať jednoduchú melódiu pri tempe 120 BPM, použijeme nasledujúci blok kódu:



Obrázok 32 Príkaz "play melody at tempo (bpm) until done"

Po spustení tohto kódu bude BBC micro:bit prehrávať melódiu pri zadanom tempe 120 BPM, až kým sa melódia neukončí.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Použite príkaz "play melody at tempo (bpm) until done" vo svojom programe.
3. Nastavte melódiu a tempo, ktoré chcete použiť.
4. Spustite svoj program a sledujte, ako micro:bit prehráva melódiu pri zadanom tempe!

Tento príkaz je veľmi užitočný na vytváranie hudobných projektov, kde micro:bit prehráva melódie pri rôznych tempách.

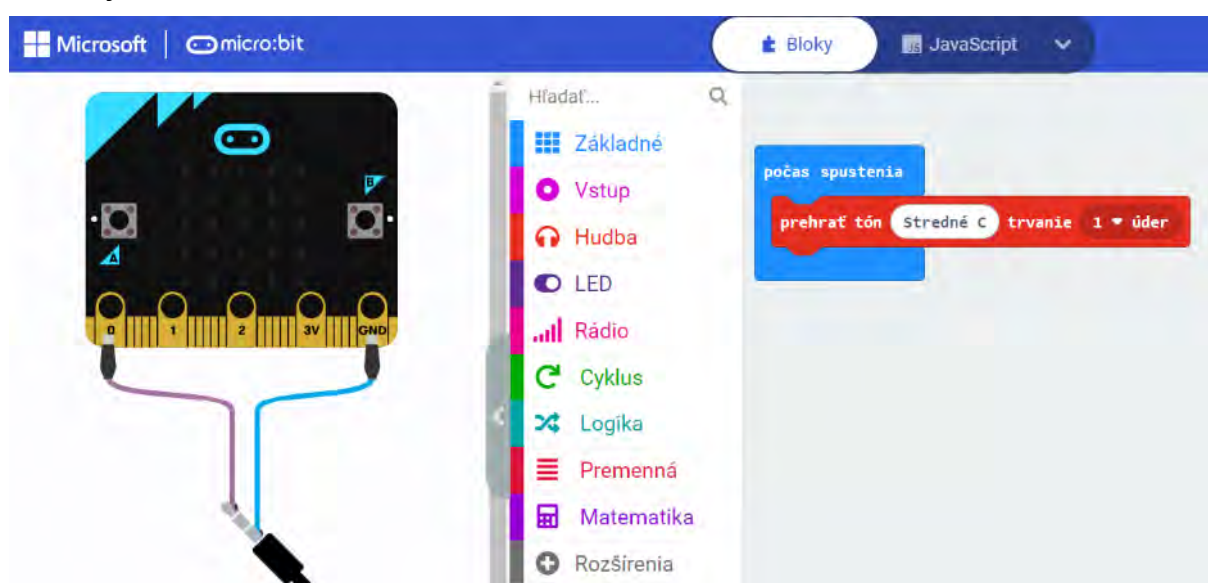
Príkaz "play tone stredné C for 1 úder until done"

Príkaz "play tone stredné C for 1 úder until done" slúži na prehranie tónu stredného C (C4) počas jednej doby úderu, až kým sa prehrávanie nedokončí. Tento príkaz využíva vstavaný piezoreproduktor micro:bita na prehrávanie tónov.

Ako to funguje?

1. Nastavenie tónu a dĺžky: Tento príkaz umožňuje nastaviť konkrétny tón (v tomto prípade stredné C) a dĺžku prehrávania (1 úder).
2. Prehrávanie tónu: Po nastavení tónu a dĺžky príkaz prehrá tón až do jeho dokončenia.

Príklad: Ak chceme prehrávať tón stredného C počas jednej doby úderu, použijeme nasledujúci blok kódu:



Obrázok 33 Príkaz "play tone stredné C for 1 úder until done"

Po spustení tohto kódu bude micro:bit prehrávať tón stredného C počas jednej doby úderu, až kým sa prehrávanie nedokončí.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Použite príkaz "play tone stredné C for 1 úder until done" vo svojom programe.
3. Nastavte tón a dĺžku, ktoré chcete použiť.
4. Spustite svoj program a sledujte, ako micro:bit prehráva tón stredného C!

Tento príkaz je veľmi užitočný na vytváranie hudobných projektov, kde BBC micro:bit prehráva konkrétne tóny pri určitých úderoch.

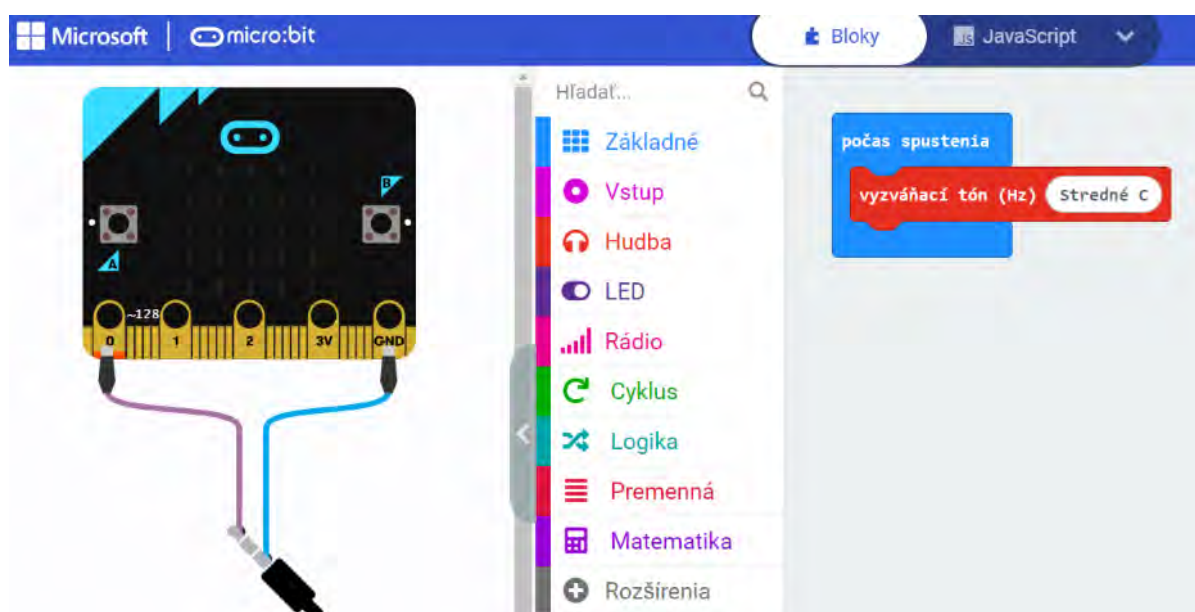
Príkaz "vyzváňací tón (Hz) stredné C"

Príkaz "vyzváňací tón (Hz) stredné C" slúži na prehranie tónu stredného C (C4) pri určitej frekvencii v Hertzoch (Hz). Tento príkaz využíva vstavaný piezoreproduktor BBC micro:bitu na prehrávanie tónov na konkrétnej frekvencii.

Ako to funguje?

1. Nastavenie frekvencie: Tento príkaz umožňuje nastaviť konkrétnu frekvenciu, v tomto prípade pre tón stredného C (C4).
2. Prehrávanie tónu: Po nastavení frekvencie príkaz prehrá zvolený tón.

Príklad: Ak chcete prehrávať tón stredného C pri frekvencii 261.63 Hz, použijete nasledujúci blok kódu:



Obrázok 34 Príkaz "vyzváňací tón (Hz) stredné C"

Po spustení tohto kódu bude BBC micro:bit prehrávať tón stredného C pri frekvencii 261.63 Hz.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Použite príkaz "vyzváňací tón (Hz) stredné C" vo svojom programe.
3. Nastavte frekvenciu pre tón, ktorý chcete použiť.
4. Spustíte svoj program a sledujte, ako BBC micro:bit prehráva tón stredného C!

Tento príkaz je veľmi užitočný na vytváranie hudobných projektov, kde BBC micro:bit prehráva konkrétne tóny pri určitých frekvenciách.

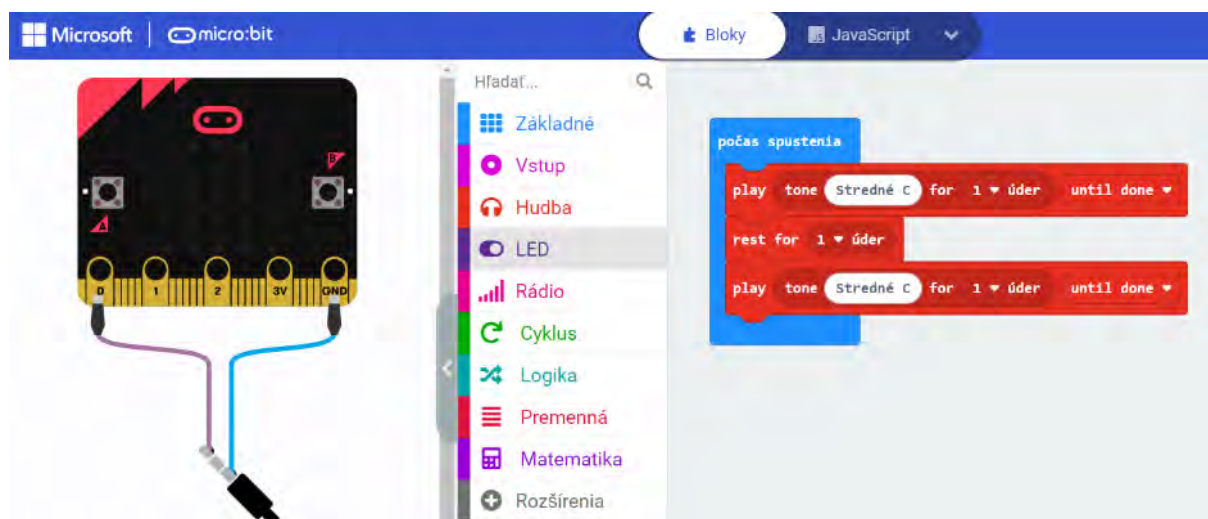
Príkaz "rest for 1 úder"

Príkaz "rest for 1 úder" slúži na vloženie pauzy (ticha) do prehrávania hudby na BBC micro:bit. Táto pauza trvá presne jeden úder, čo znamená, že počas tejto doby nebude BBC micro:bit prehrávať žiadne tóny ani melódie.

Ako to funguje?

1. Nastavenie pauzy: Tento príkaz umožňuje vložiť prestávku v prehrávaní hudby na dobu jedného úderu.
2. Prehrávanie hudby: Počas pauzy sa nebude prehrávať žiadny tón, čím sa vytvorí ticho v melódii.

Príklad: Ak chceme pridať pauzu trvajúcu jeden úder medzi prehrávanie tónov, použijeme nasledujúci blok kódu:



Obrázok 35 Príkaz "rest for 1 úder"

Po spustení tohto kódu BBC micro:bit prehrá jeden tón (stredné C) počas jedného úderu, potom bude BBC micro:bit ticho počas jedného úderu a následne znovu BBC micro:bit prehrá jeden tón (stredné C) počas jedného úderu.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Použite príkaz "rest for 1 úder" vo svojom programe.
3. Nastavte pauzu v melódii, ktorú chcete použiť.
4. Spustite svoj program a sledujte, ako BBC micro:bit pridáva ticho do prehrávania melódie!

Tento príkaz je veľmi užitočný na vytváranie hudobných projektov, kde chcete vložiť pauzy do melódie na vytvorenie dynamiky a rytmu.

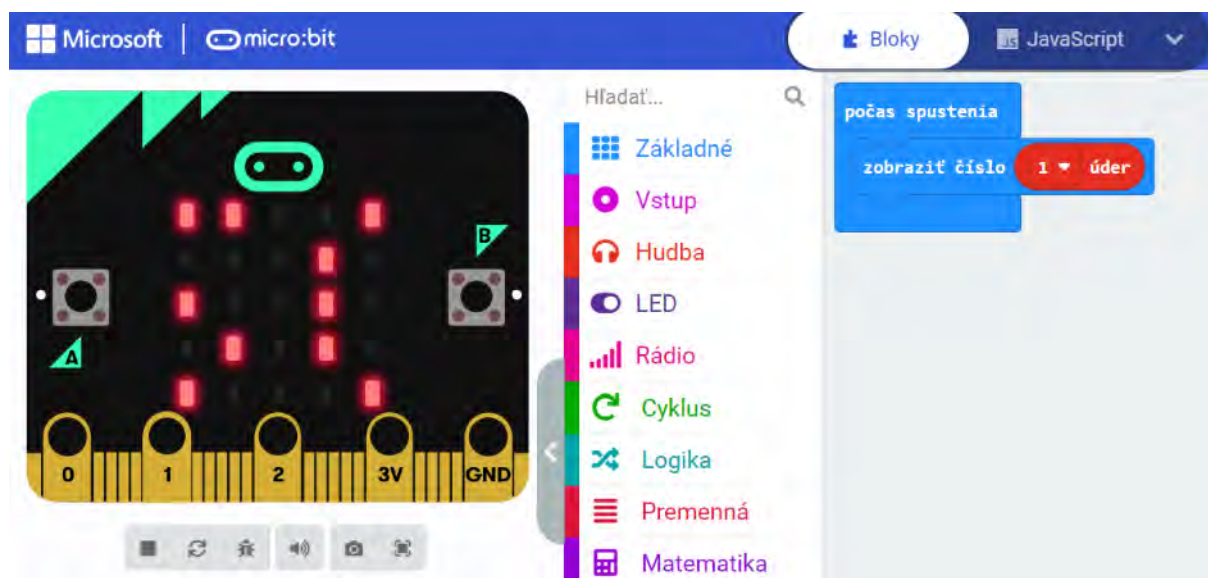
Príkaz "1 úder"

Príkaz "1 úder" slúži na nastavenie trvania jedného úderu v kontexte prehrávania hudby na micro:bite. Tento príkaz určuje, ako dlho trvá jeden úder, a na základe toho sa riadi prehrávanie melódií a tónov.

Ako to funguje?

1. Nastavenie trvania: Tento príkaz nastaví trvanie jedného úderu v milisekundách, na základe aktuálneho tempa (BPM).
2. Prehrávanie hudby: Všetky tóny a melódie sa budú prehrávať podľa nastaveného trvania jedného úderu.

Príklad: Ak chceme zobrazit' aktuálne trvanie jedného úderu, použijeme nasledujúci blok kódu:



Obrázok 36 Príkaz "1 úder"

Po spustení tohto kódu bude micro:bit zobrazovať, ako dlho trvá jeden úder v milisekundách na základe aktuálneho tempa.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Použite príkaz "1 úder" vo svojom programe.
3. Nastavte tempo a pozorujte, ako sa mení trvanie jedného úderu.
4. Spustíte svoj program a sledujte, ako BBC micro:bit zobrazí aktuálne trvanie jedného úderu!

Tento príkaz je veľmi užitočný na presné nastavenie rytmu a trvania tónov pri prehrávaní hudby na micro:bite.

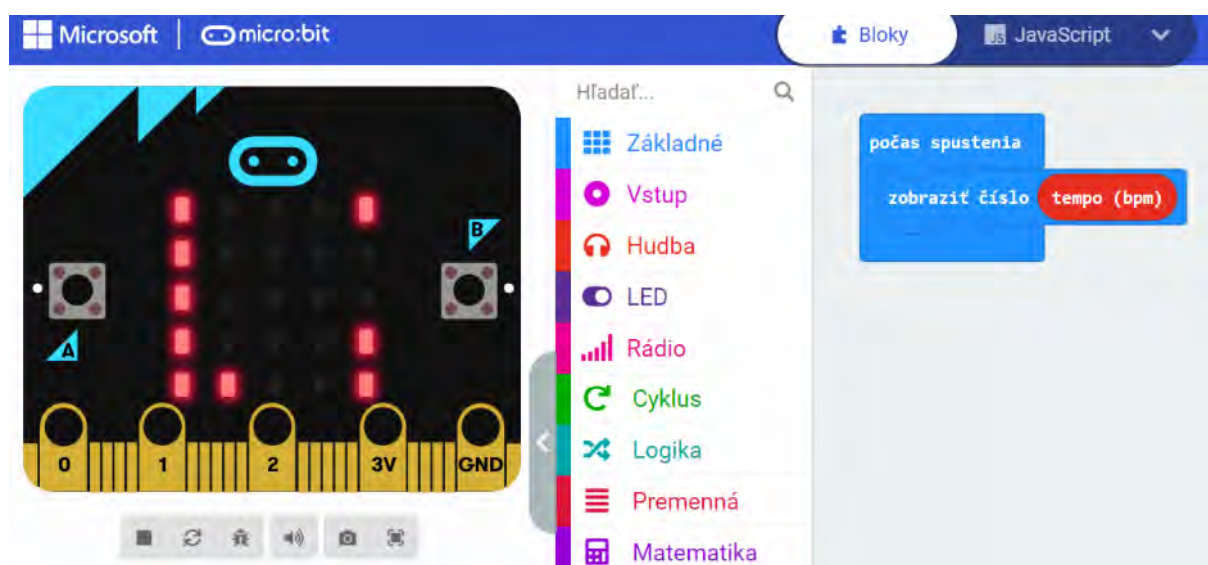
Príkaz "tempo (bpm)"

Príkaz "tempo (bpm)" slúži na získanie aktuálneho tempa prehrávania hudby na micro:bite v úderoch za minútu (BPM). Tento príkaz je užitočný, ak chcete vedieť, aké je momentálne nastavené tempo, pri ktorom sa prehrávajú všetky tóny a melódie.

Ako to funguje?

1. Získanie tempa: Tento príkaz vám umožňuje zistiť aktuálne nastavené tempo prehrávania v BPM.
2. Prehrávanie hudby: Prehrávanie hudby pokračuje pri aktuálnom tempe, ktoré bolo nastavené.

Príklad: Ak chcete zobrazit' aktuálne tempo prehrávania v BPM, použijete nasledujúci blok kódu:



Obrázok 37 Príkaz "tempo (bpm)"

Po spustení tohto kódu bude micro:bit zobrazovať aktuálnu hodnotu tempa v BPM.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Použite príkaz "tempo (bpm)" vo svojom programe.
3. Spustíte svoj program a sledujte, ako micro:bit zobrazuje aktuálne nastavené tempo!

Tento príkaz je veľmi užitočný na kontrolu a zistenie tempa hudby a melódii prehrávaných na micro:bite.

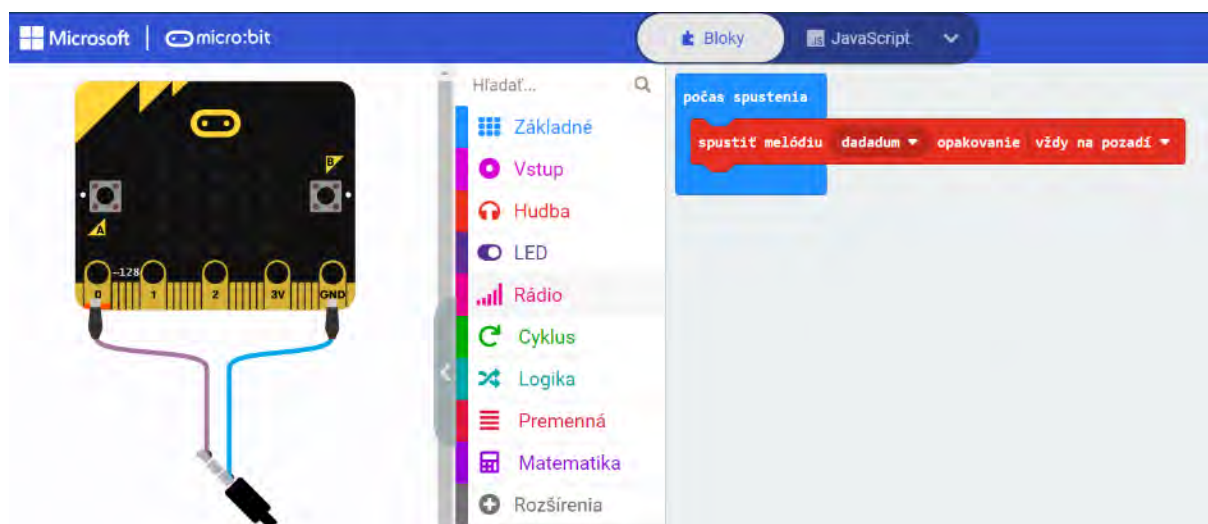
Príkaz "play melody dadadum in background"

Príkaz "play melody dadadum in background" slúži na prehranie melódie "dadadum" na pozadí, čo znamená, že zatiaľ čo sa melódia prehráva, micro:bit môže súčasne vykonávať ďalšie operácie. Tento príkaz využíva vstavaný piezoreproduktor micro:bita na prehrávanie hudby.

Ako to funguje?

1. Výber melódie: Príkaz zvolí melódiu "dadadum".
2. Prehrávanie na pozadí: Melódia sa prehráva na pozadí, čo znamená, že micro:bit môže súčasne vykonávať ďalšie úlohy.

Príklad: Ak chcete prehrávať melódiu "dadadum" na pozadí, použijete nasledujúci blok kódu:



Obrázok 38 Príkaz "play melody dadadum in background"

Po spustení tohto kódu bude micro:bit prehrávať melódiu "dadadum" na pozadí.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Použite príkaz "play melody dadadum in background" vo svojom programe.
3. Spustíte svoj program a sledujte, ako micro:bit prehráva melódiu "dadadum" na pozadí, zatiaľ čo môže vykonávať ďalšie úlohy!

Tento príkaz je veľmi užitočný na vytváranie hudobných projektov, kde micro:bit prehráva melódie na pozadí a súčasne vykonáva ďalšie operácie.

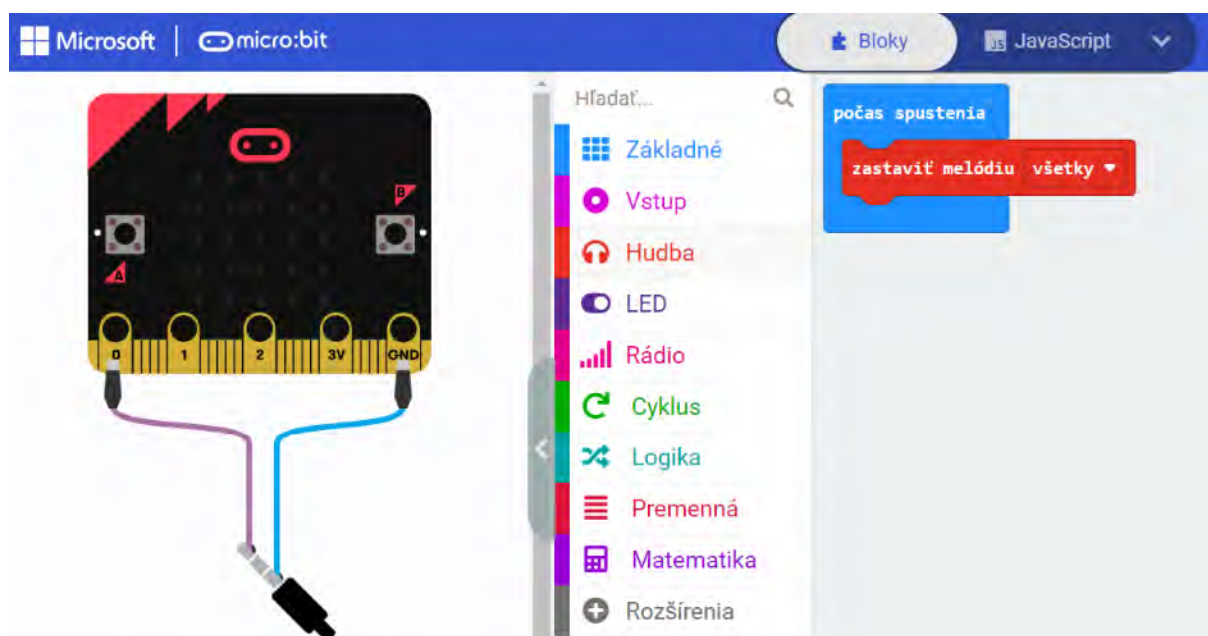
Príkaz "zastaviť melódiu všetky"

Príkaz "zastaviť melódiu všetky" slúži na okamžité zastavenie všetkých práve prehrávaných melódií na micro:bite. Tento príkaz zastaví každú melódiu, ktorá bola spustená a hrá na pozadí alebo v popredí.

Ako to funguje?

1. Zastavenie melódií: Tento príkaz okamžite zastaví všetky aktuálne prehrávané melódie.
2. Vykonanie kódu: Po zadaní príkazu sa okamžite ukončí prehrávanie všetkých melódií.

Príklad: Ak chceme zastaviť všetky prehrávané melódie, použijeme nasledujúci blok kódu:



Obrázok 39 Príkaz "zastaviť melódiu všetky"

Po spustení tohto kódu BBC micro:bit okamžite zastaví prehrávanie všetkých melódií.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Použite príkaz "zastaviť melódiu všetky" vo svojom programe.
3. Spustíte svoj program a sledujte, ako BBC micro:bit okamžite zastaví všetky prehrávané melódie pri vykonaní príkazu!

Tento príkaz je veľmi užitočný na rýchle ukončenie prehrávania hudby alebo melódií na micro:bite.

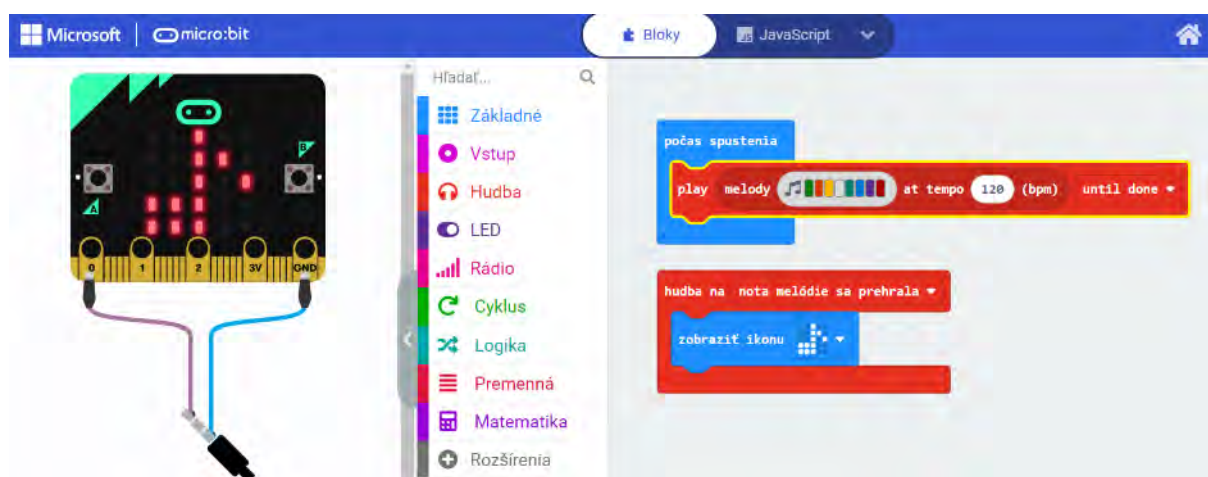
Príkaz "hudba na nota melódie sa prehrala"

Príkaz "hudba na nota melódie sa prehrala" slúži na zistenie, či sa prehrala konkrétna nota v melódii na micro:bit. Tento príkaz je užitočný na kontrolu priebehu prehrávania hudby a vykonanie akcií po skončení noty.

Ako to funguje?

1. Detekcia prehrania noty: Tento príkaz sleduje, kedy sa prehrá konkrétna nota v melódii.
2. Vykonanie kódu: Po zistení, že sa nota prehrala, môžete vykonať ďalšiu akciu podľa zadaného kódu.

Príklad: Ak chceme zobrazit' ikonu noty po prehraní noty v melódii, použijeme nasledujúci blok kódu:



Obrázok 40 Príkaz "hudba na nota melódie sa prehrala"

Po spustení tohto kódu bude BBC micro:bit zobrazovať úsmev po prehraní každej noty v melódii.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Použite príkaz "hudba na nota melódie sa prehrala" vo svojom programe.
3. Nastavte akciu, ktorú chcete vykonať po prehraní noty.
4. Spustite svoj program a sledujte, ako BBC micro:bit vykonáva zadanú akciu po prehraní každej noty!

Tento príkaz je veľmi užitočný na vytváranie hudobných projektov, kde chcete reagovať na prehratie konkrétnych tónov v melódii.

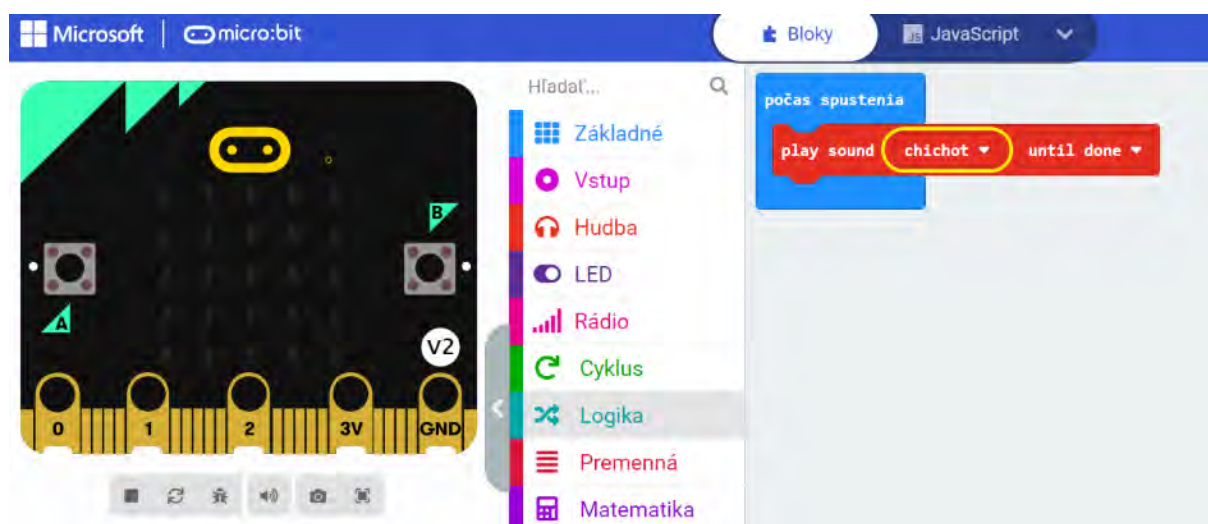
Príkaz "play chichot until done"

Príkaz "play chichot until done" slúži na prehranie zvuku "chichot" (smiech) až do jeho úplného dokončenia. Tento príkaz využíva vstavaný piezoreproduktor micro:bita na prehrávanie tohto konkrétneho zvuku.

Ako to funguje?

1. Výber zvuku: Tento príkaz zvolí zvuk "chichot".
2. Prehrávanie zvuku: Zvuk "chichot" sa prehráva až do jeho dokončenia, bez ohľadu na iné operácie.

Príklad: Ak chcete prehrávať zvuk "chichot" až do jeho dokončenia, použijete nasledujúci blok kódu:



Obrázok 41 Príkaz "play chichot until done"

Po spustení tohto kódu bude BBC micro:bit prehrávať zvuk "chichot" až do jeho úplného dokončenia.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Použite príkaz "play chichot until done" vo svojom programe.
3. Spustíte svoj program a sledujte, ako micro:bit prehráva zvuk "chichot" až do konca!

Tento príkaz je veľmi užitočný na vytváranie zábavných projektov, kde micro:bit prehráva zvuky.

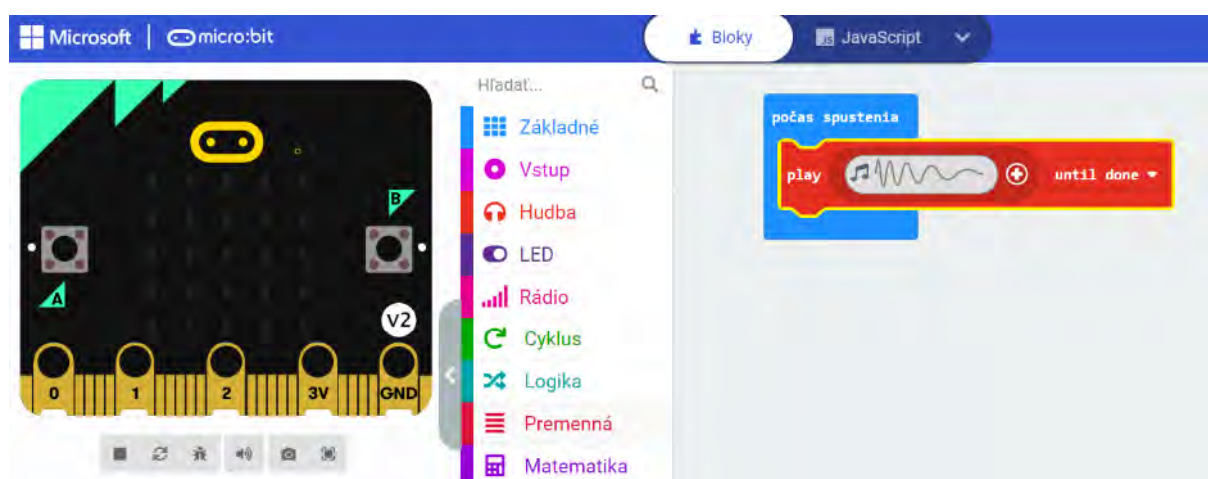
Príkaz "play zvuková vlna"

Príkaz "play zvuková vlna" slúži na prehranie konkrétnej zvukovej vlny na micro:bit. Tento príkaz využíva vstavaný piezoreproduktor micro:bita na prehrávanie zvukových vln, ktoré môžu byť použité v rôznych projektoch.

Ako to funguje?

1. Výber zvukovej vlny: Tento príkaz vám umožní zvoliť konkrétnu zvukovú vlnu z knižnice zvukových prejavov.
2. Zvukovú vlnu si môžete ľubovoľne prispôbovať jednoduchým kliknutím na ňu a zmenou parametrov.
3. Prehrávanie zvuku: Po zvolení zvukovej vlny sa táto prehrá na micro:bit.

Príklad: Ak chcete prehrávať zvukovú vlnu, použijete nasledujúci blok kódu:



Obrázok 42 Príkaz "play zvuková vlna"

Po spustení tohto kódu bude micro:bit prehrávať vybranú zvukovú vlnu až do jej dokončenia.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Použite príkaz "play zvuková vlna" vo svojom programe.
3. Vyberte zvukovú vlnu, ktorú chcete použiť.
4. Upravte jej parametre kliknutím na ňu, alebo kliknutím na tlačidlo +
5. Spustite svoj program a sledujte, ako BBC micro:bit prehráva zvolenú zvukovú vlnu!

Tento príkaz je veľmi užitočný na vytváranie zábavných a interaktívnych projektov, kde BBC micro:bit prehráva rôzne zvukové vlny.

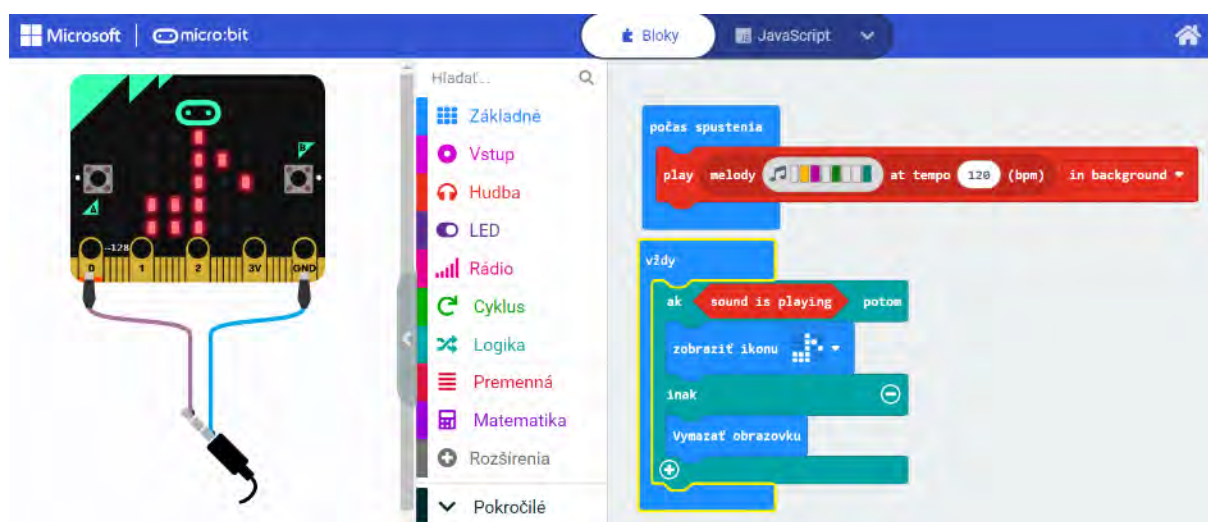
Príkaz "sound is playing"

Príkaz "sound is playing" slúži na zistenie, či BBC micro:bit aktuálne prehráva nejaký zvuk alebo nie. Tento príkaz je užitočný na kontrolu stavu prehrávania zvukov a rozhodnutie, či vykonať určitú akciu.

Ako to funguje?

1. Detekcia zvuku: Tento príkaz sleduje, či BBC micro:bit momentálne prehráva zvuk.
2. Vykonanie kódu: Na základe zistenia, či zvuk hrá alebo nie, môžete vykonať ďalšiu akciu podľa zadaného kódu.

Príklad: Ak chceme zobraziť ikonu hudby, keď BBC micro:bit prehráva zvuk, použijeme nasledujúci blok kódu:



Obrázok 43 Príkaz "sound is playing"

Po spustení tohto kódu bude BBC micro:bit zobrazovať ikonu hudby, keď prehráva zvuk, a vymaže obrazovku, keď zvuk nehrá.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Použite príkaz "sound is playing" vo svojom programe.
3. Nastavte akciu, ktorú chcete vykonať, keď BBC micro:bit prehráva zvuk, napríklad zobrazenie ikony hudby.
4. Spustite svoj program a sledujte, ako BBC micro:bit reaguje na prehrávanie zvuku!

Tento príkaz je veľmi užitočný na vytváranie interaktívnych projektov, kde chcete, aby BBC micro:bit reagoval na aktuálny stav prehrávania zvukov.

5 LED

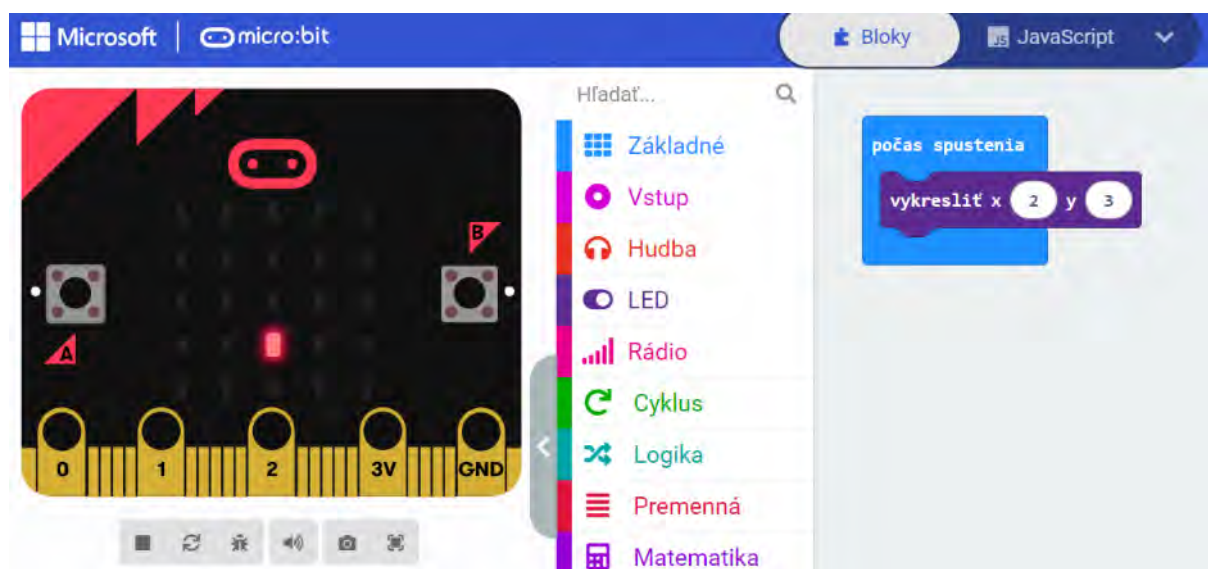
Príkaz "vykresliť X Y"

Príkaz "vykresliť X Y" slúži na vykreslenie konkrétneho bodu na LED matici micro:bita na základe súradníc X a Y. Tento príkaz umožňuje kontrolu a nastavovanie jednotlivých LED diód na displeji.

Ako to funguje?

1. Nastavenie súradníc: Tento príkaz vyžaduje zadanie súradníc X (horizontálna os) a Y (vertikálna os), ktoré určujú presnú polohu bodu na LED matici.
2. Vykreslenie bodu: Po zadaní súradníc sa na tejto pozícii na LED matici rozsvieti LED dióda.

Príklad: Ak chceme vykresliť bod na súradniciach X=2 a Y=3, použijeme nasledujúci blok kódu:



Obrázok 44 Príkaz "vykresliť X Y"

Po spustení tohto kódu sa na LED matici BBC micro:bitu rozsvieti LED dióda na pozícii X=2, Y=3.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Použite príkaz "vykresliť X Y" vo svojom programe.
3. Nastavte súradnice, na ktorých chcete vykresliť bod.
4. Spustíte svoj program a sledujte, ako BBC micro:bit vykresľuje bod na zadaných súradniciach!

Tento príkaz je veľmi užitočný na vytváranie grafických obrazcov a interaktívnych vizuálnych projektov na LED matici BBC micro:bitu.

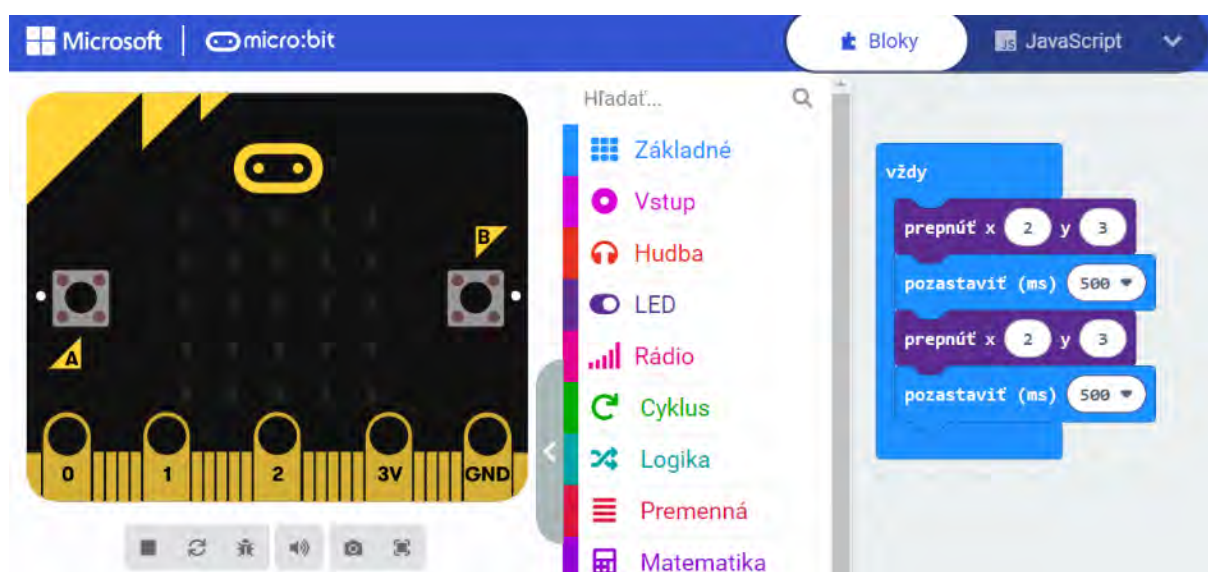
Príkaz "prepnúť X Y"

Príkaz "prepnúť X Y" slúži na zmenu stavu LED diódy na daných súradniciach X a Y na LED matici BBC micro:bita. Tento príkaz prepne stav LED diódy zo zapnutého na vypnutý alebo z vypnutého na zapnutý.

Ako to funguje?

1. Nastavenie súradníc: Tento príkaz vyžaduje zadanie súradníc X (horizontálna os) a Y (vertikálna os), ktoré určujú presnú polohu LED diódy na LED matici.
2. Prepnutie stavu: Po zadaní súradníc sa na tejto pozícii LED dióda rozsvieti, ak bola vypnutá, alebo zhasne, ak bola zapnutá.

Príklad: Ak chceme prepnúť stav LED diódy na súradniciach X=2 a Y=3, použijeme nasledujúci blok kódu:



Obrázok 45 Príkaz "prepnúť X Y"

Po spustení tohto kódu sa na LED matici BBC micro:bitu prepne stav LED diódy na pozícii X=2, Y=3, počká pól sekundy a znova prepne stav LED diódy na pozícii X=2, Y=3 a znova počká pól sekundy. Týmto docielime, že LED dióda na pozícii X=2, Y=3 bliká.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Použite príkaz "prepnúť X Y" vo svojom programe.
3. Nastavte súradnice, na ktorých chcete prepnúť stav LED diódy.
4. Spustíte svoj program a sledujte, ako BBC micro:bit prepína stav LED diódy na zadaných súradniciach!

Tento príkaz je veľmi užitočný na vytváranie grafických obrazcov a interaktívnych vizuálnych projektov na LED matici BBC micro:bitu.

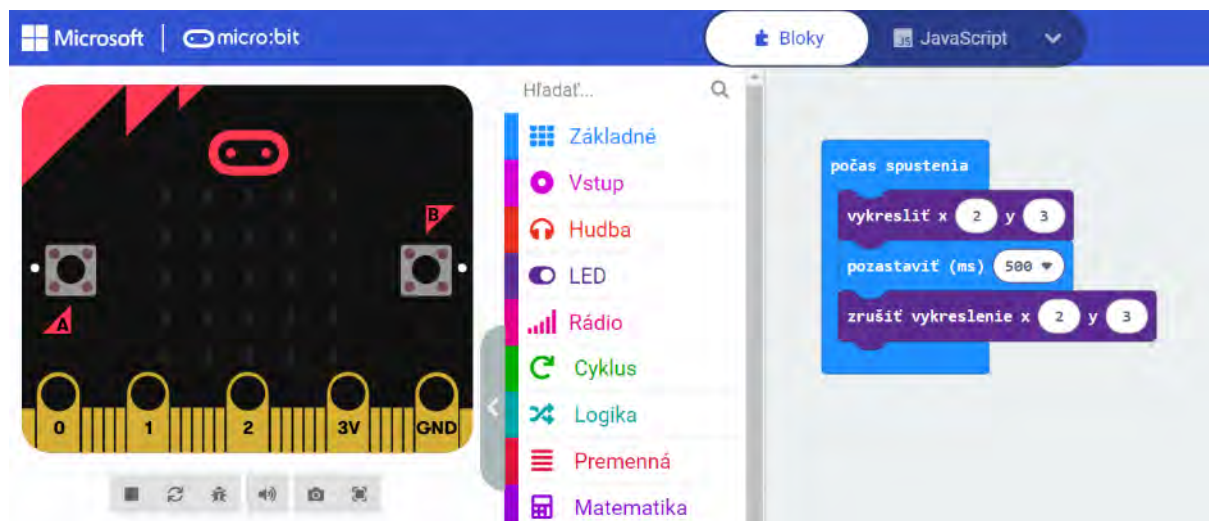
Príkaz "zrušiť vykreslenie X Y"

Príkaz "zrušiť vykreslenie X Y" slúži na vypnutie konkrétneho bodu na LED matici BBC micro:bita na základe súradníc X a Y. Tento príkaz vypne LED diódu na určených súradniciach.

Ako to funguje?

1. Nastavenie súradníc: Tento príkaz vyžaduje zadanie súradníc X (horizontálna os) a Y (vertikálna os), ktoré určujú presnú polohu LED diódy na LED matici.
2. Vypnutie bodu: Po zadaní súradníc sa na tejto pozícii LED dióda zhasne.

Príklad: Ak chceme vypnúť bod na súradniciach X=2 a Y=3, použijeme nasledujúci blok kódu:



Obrázok 46 Príkaz "zrušiť vykreslenie X Y"

Po spustení tohto kódu sa na LED matici BBC micro:bitu zhasne LED dióda na pozícii X=2, Y=3.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Použite príkaz "zrušiť vykreslenie X Y" vo svojom programe.
3. Nastavte súradnice, na ktorých chcete vypnúť bod.
4. Spustite svoj program a sledujte, ako micro:bit zhasína bod na zadaných súradniciach!

Tento príkaz je veľmi užitočný na vytváranie dynamických grafických obrazcov a interaktívnych vizuálnych projektov na LED matici micro:bitu.

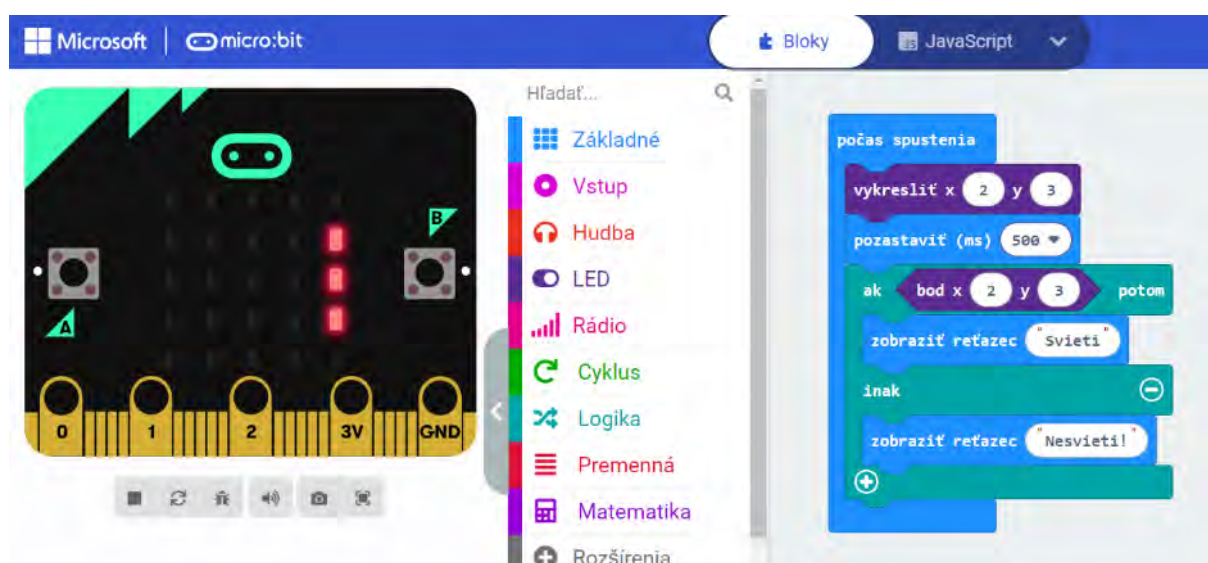
Príkaz "bod X Y"

Príkaz "bod X Y" slúži na zistenie stavu konkrétneho bodu na LED matici micro:bita na základe súradníc X a Y. Tento príkaz umožňuje kontrolovať, či je LED dióda na daných súradniciach zapnutá alebo vypnutá.

Ako to funguje?

1. Nastavenie súradníc: Tento príkaz vyžaduje zadanie súradníc X (horizontálna os) a Y (vertikálna os), ktoré určujú presnú polohu LED diódy na LED matici.
2. Kontrola stavu: Po zadaní súradníc príkaz zistí, či je LED dióda na tejto pozícii zapnutá alebo vypnutá. Výsledkom je buď hodnota true (zapnuté), alebo false (vypnuté).

Príklad: Ak chceme zistiť stav LED diódy na súradniciach X=2 a Y=3, použijeme nasledujúci blok kódu:



Obrázok 47 Príkaz "bod X Y"

Po spustení tohto kódu bude micro:bit zobrazovať, či je LED dióda na pozícii X=2, Y=3 „Svieti“ alebo „Nesvieti“. Zmenou hodnoty príkazu na „Vykresliť X 1 Y 2“ dostaneme odpoveď „Nesvieti“.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Použite príkaz "bod X Y" vo svojom programe.
3. Vložte podmienku "AK" tak aby sa vaše súradnice zhodovali s príkazom "vykresliť". A pridajte príkaz "Zobrazíť reťazec s textom "Svieti"
4. "Inak" nech sa "zobrazí reťazec" s textom "Nesvieti"
5. Nastavte súradnice, ktorých stav chcete skontrolovať.
6. Spustíte svoj program a sledujte, ako micro:bit zobrazuje stav bodu na zadanych súradniciach!

Tento príkaz je veľmi užitočný na kontrolu stavu konkrétnych LED diód na LED matici micro:bitu.

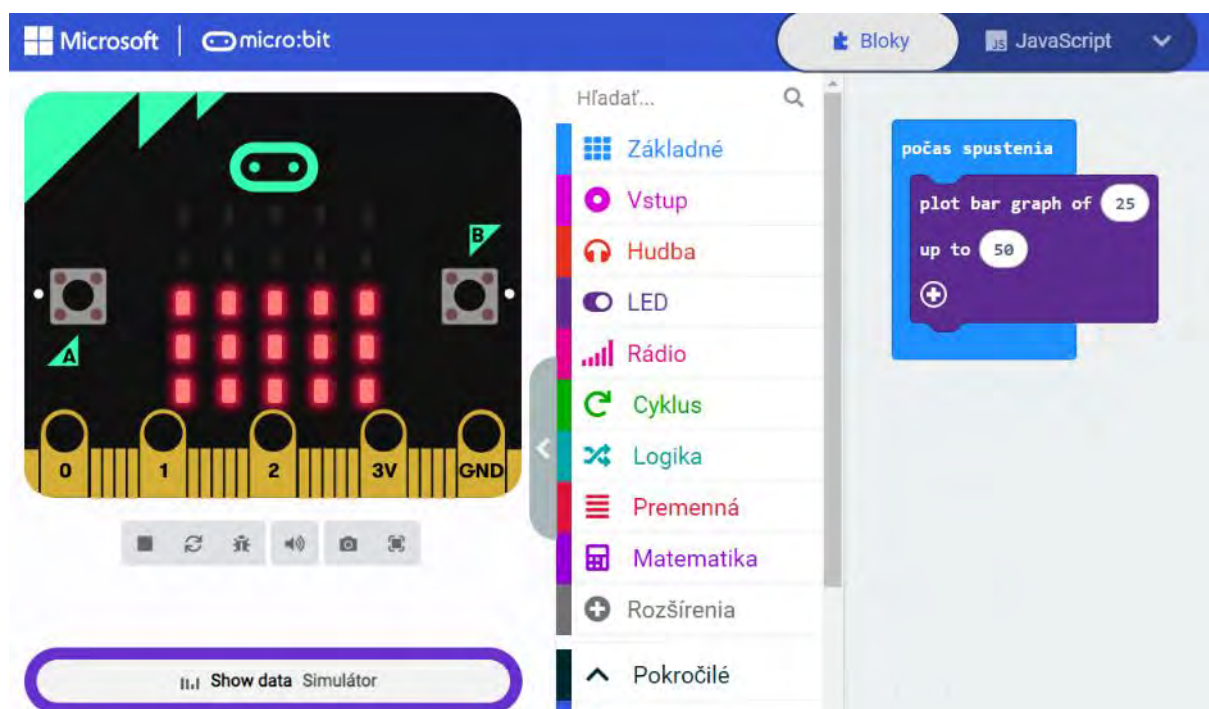
Príkaz "plot bar graph of 0 up to 0"

Príkaz "plot bar graph of 0 up to 50" slúži na vykreslenie stĺpcového grafu na LED matici micro:bita. Tento príkaz zobrazí hodnotu v stĺpcovom grafe, pričom minimálna hodnota je 0 a maximálna hodnota je 50 (číslo 50 je len príklad, môže byť podstatne vyššie). To je užitočné pri vizualizácii číselných hodnôt, ako sú senzorové údaje.

Ako to funguje?

1. Nastavenie hodnôt: Tento príkaz zobrazuje stĺpcový graf na LED matici s minimálnou hodnotou 0 a maximálnou hodnotou 50.
2. Vykreslenie grafu: Micro:bit zobrazí stĺpec na základe aktuálnej hodnoty v zadanom rozsahu (0 až 50).

Príklad: Ak chcete vykresliť stĺpcový graf s hodnotou 25 v rozsahu 0 až 50, použijete nasledujúci blok kódu:



Obrázok 48 Príkaz "plot bar graph of 0 up to 0"

Po spustení tohto kódu bude micro:bit zobrazovať stĺpcový graf so zadanými hodnotami.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Použite príkaz "plot bar graph of 0 up to 50" vo svojom programe.
3. Nastavte hodnoty, ktoré chcete vizualizovať.
4. Spustite svoj program a sledujte, ako micro:bit zobrazuje stĺpcový graf na LED matici!

Tento príkaz je veľmi užitočný na vizualizáciu číselných údajov v reálnom čase na LED matici micro:bita.

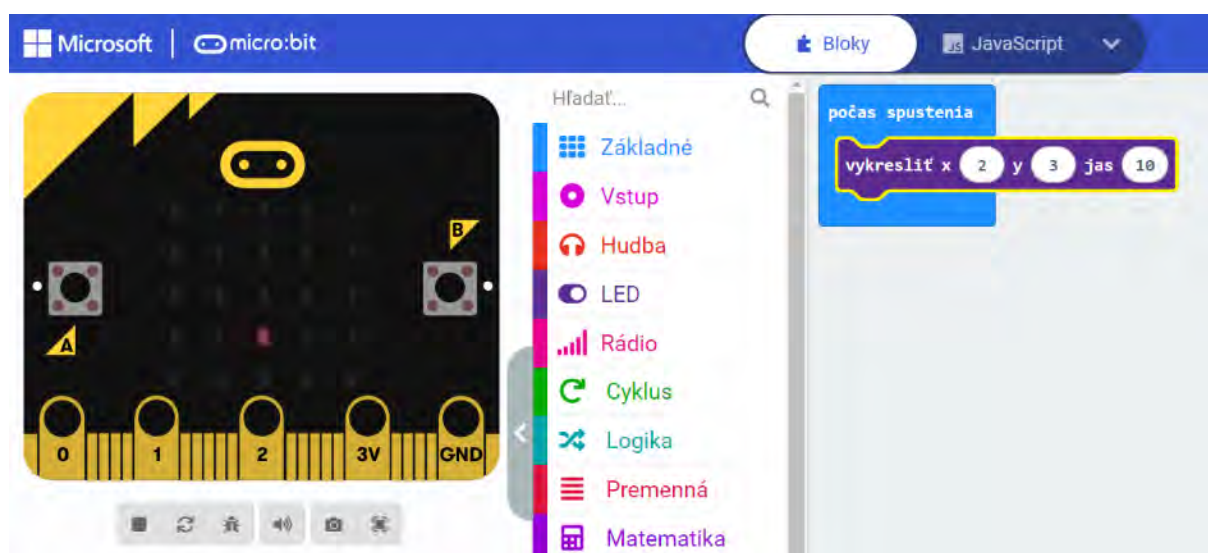
Príkaz "vykresliť X Y jas 255"

Príkaz "vykresliť X Y jas 255" slúži na vykreslenie konkrétneho bodu na LED matici micro:bita na základe súradníc X a Y, a nastavenie jasnosti tohto bodu na maximálnu hodnotu 255. Tento príkaz umožňuje kontrolovať jas jednotlivých LED diód na displeji.

Ako to funguje?

1. Nastavenie súradníc a jasnosti: Tento príkaz vyžaduje zadanie súradníc X (horizontálna os) a Y (vertikálna os), ktoré určujú presnú polohu bodu na LED matici, a tiež nastavenie jasnosti na hodnotu 255.
2. Vykreslenie bodu: Po zadání súradníc a jasnosti sa na tejto pozícii na LED matici rozsvieti LED dióda s maximálnym jasom 255.

Príklad: Ak chceme vykresliť bod na súradniciach X=2 a Y=3 s jasom 10, použijeme nasledujúci blok kódu:



Obrázok 49 Príkaz "vykresliť X Y jas 255"

Po spustení tohto kódu sa na LED matici micro:bitu rozsvieti LED dióda na pozícii X=2, Y=3 s jasom 10.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Použite príkaz "vykresliť X Y jas 10" vo svojom programe.
3. Nastavte súradnice a jas, ktoré chcete použiť.
4. Spustite svoj program a sledujte, ako micro:bit vykresľuje bod na zadanych súradniciach s maximálnym jasom!

Tento príkaz je veľmi užitočný na vytváranie grafických obrazcov a interaktívnych vizuálnych projektov na LED matici micro:bitu.

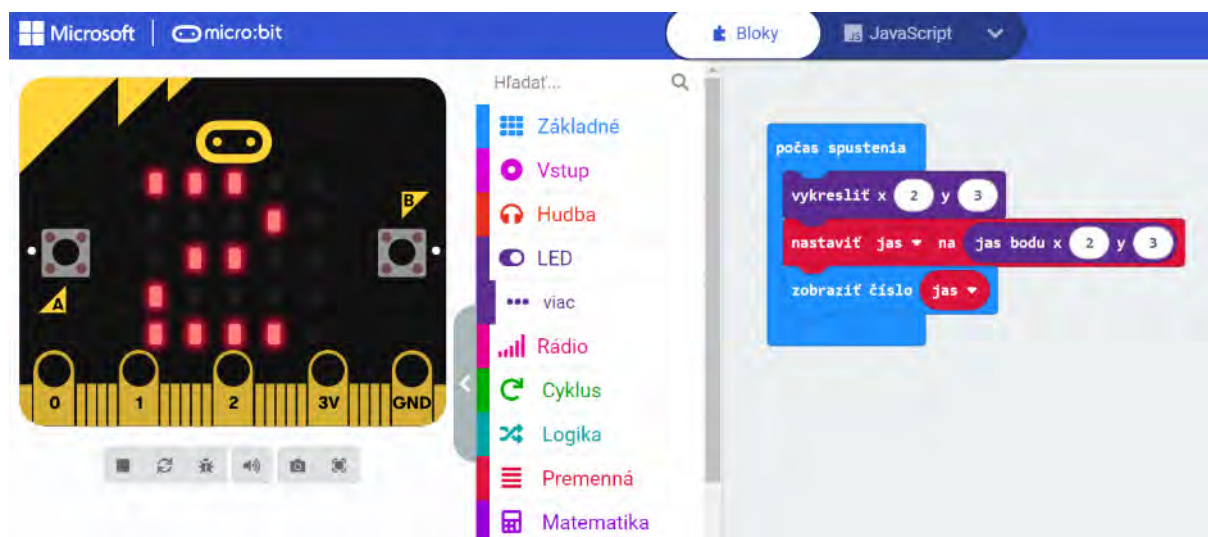
Príkaz "jas bodu X Y"

Príkaz "jas bodu X Y" slúži na zistenie aktuálnej úrovne jasu konkrétneho bodu na LED matici micro:bita na základe súradníc X a Y. Tento príkaz umožňuje skontrolovať, aký jas má daná LED dióda.

Ako to funguje?

1. Nastavenie súradníc: Tento príkaz vyžaduje zadanie súradníc X (horizontálna os) a Y (vertikálna os), ktoré určujú presnú polohu bodu na LED matici.
2. Kontrola jasu: Po zadaní súradníc príkaz vráti aktuálnu hodnotu jasu LED diódy na tejto pozícii. Jas je udávaný v hodnotách od 0 (vypnuté) do 255 (maximálny jas).

Príklad: Ak chceme zistiť jas bodu na súradniciach X=2 a Y=3, použijeme nasledujúci blok kódu:



Obrázok 50 Príkaz "jas bodu X Y"

Po spustení tohto kódu bude micro:bit zobrazovať aktuálnu hodnotu jasu bodu na pozícii X=2, Y=3.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Použite príkaz "jas bodu X Y" vo svojom programe.
3. Nastavte súradnice, ktorých jas chcete skontrolovať.
4. Spustíte svoj program a sledujte, ako micro:bit zobrazí aktuálnu hodnotu jasu bodu na zadaných súradniciach!

Tento príkaz je veľmi užitočný na kontrolu a analýzu stavu jednotlivých LED diód na LED matici micro:bita.

Príkaz "jas"

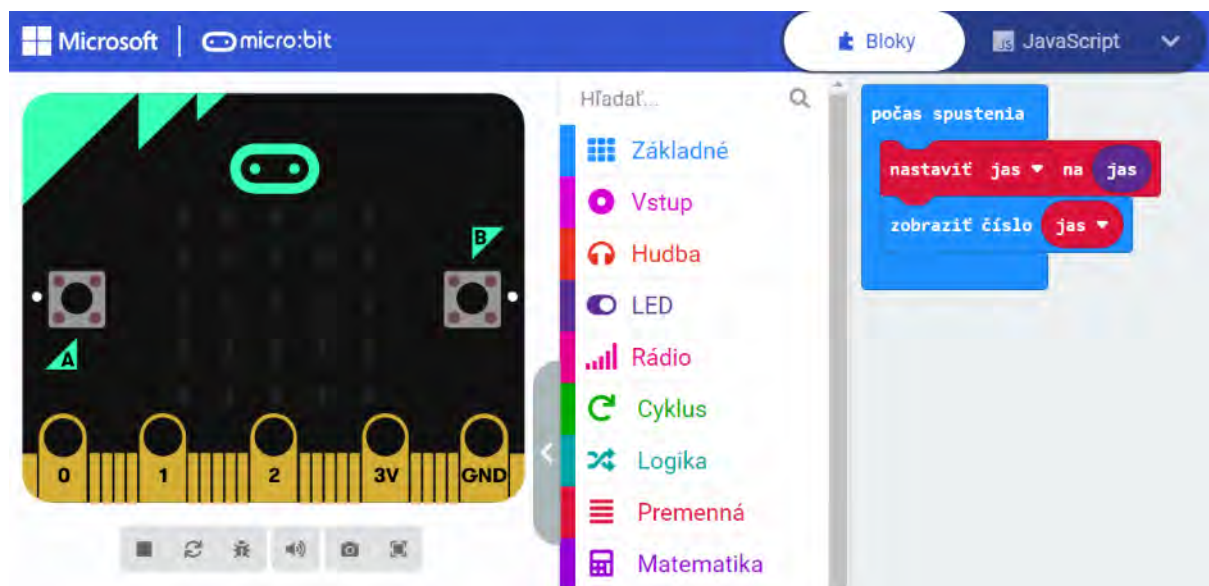
Príkaz "jas" slúži na zistenie aktuálnej úrovne jasu pre LED diódy na LED matici micro:bitu. Tento príkaz umožňuje zistiť aktuálnu nastavenú úroveň jasu.

Ako to funguje?

- Zistenie jasu: Tento príkaz umožňuje zistiť aktuálne nastavenú úroveň jasu.

Príklad:

Ak chceme zistiť aktuálnu úroveň jasu, použijeme nasledujúci blok kódu:



Obrázok 51 Príkaz "jas"

Po spustení tohto kódu bude micro:bit zobrazovať aktuálnu úroveň jasu.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Použite príkaz "jas" na zistenie aktuálnej úrovne jasu vo svojom programe.
3. Spustíte svoj program a sledujte, ako micro:bit zobrazuje aktuálnu hodnotu!

Tento príkaz je veľmi užitočný na kontrolu a úpravu jasu LED matice na micro:bite.

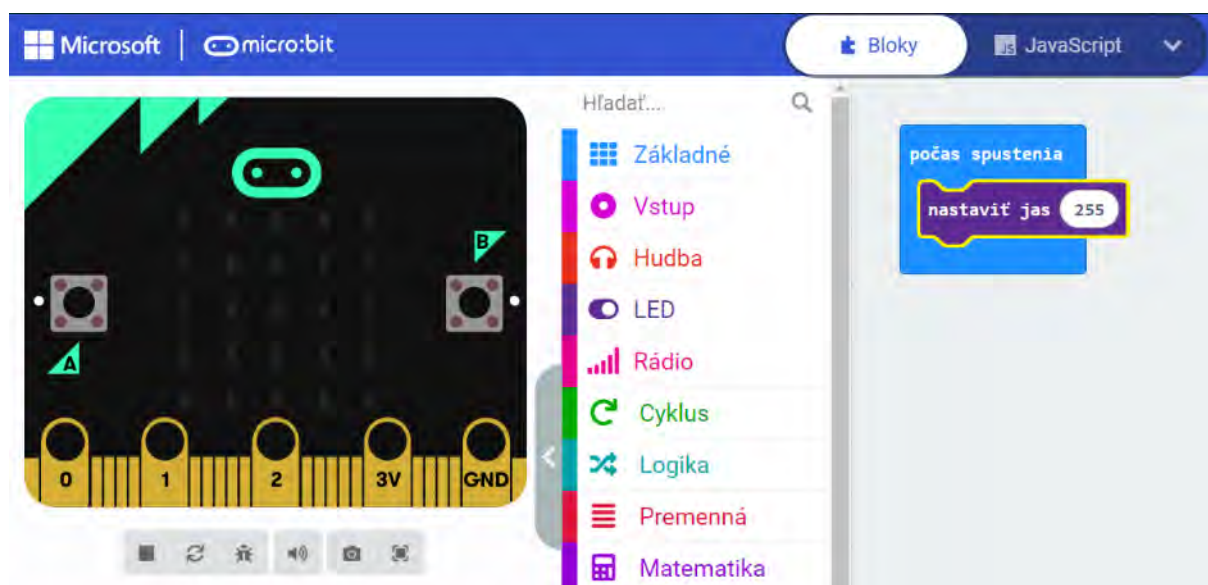
Príkaz "nastaviť jas 255"

Príkaz "nastaviť jas 255" slúži na nastavenie úrovne jasu pre všetky LED diódy na LED matici micro:bitu na maximálnu hodnotu 255. Tento príkaz umožňuje upraviť jas všetkých LED diód naraz.

Ako to funguje?

1. Nastavenie jasu: Tento príkaz nastaví úroveň jasu na hodnotu 255, čo je maximálny jas.
2. Prehrávanie: Všetky LED diódy na LED matici budú svietiť maximálnou intenzitou.

Príklad: Ak chcete nastaviť jas na 255, použijete nasledujúci blok kódu:



Obrázok 52 Príkaz "nastaviť jas 255"

Po spustení tohto kódu bude micro:bit nastavovať jas všetkých LED diód na maximálnu hodnotu 255.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Použite príkaz "nastaviť jas 255" vo svojom programe.
3. Spustite svoj program a sledujte, ako micro:bit nastaví jas všetkých LED diód na maximálnu hodnotu!

Tento príkaz je veľmi užitočný na maximálne osvetlenie LED matice na micro:bite.

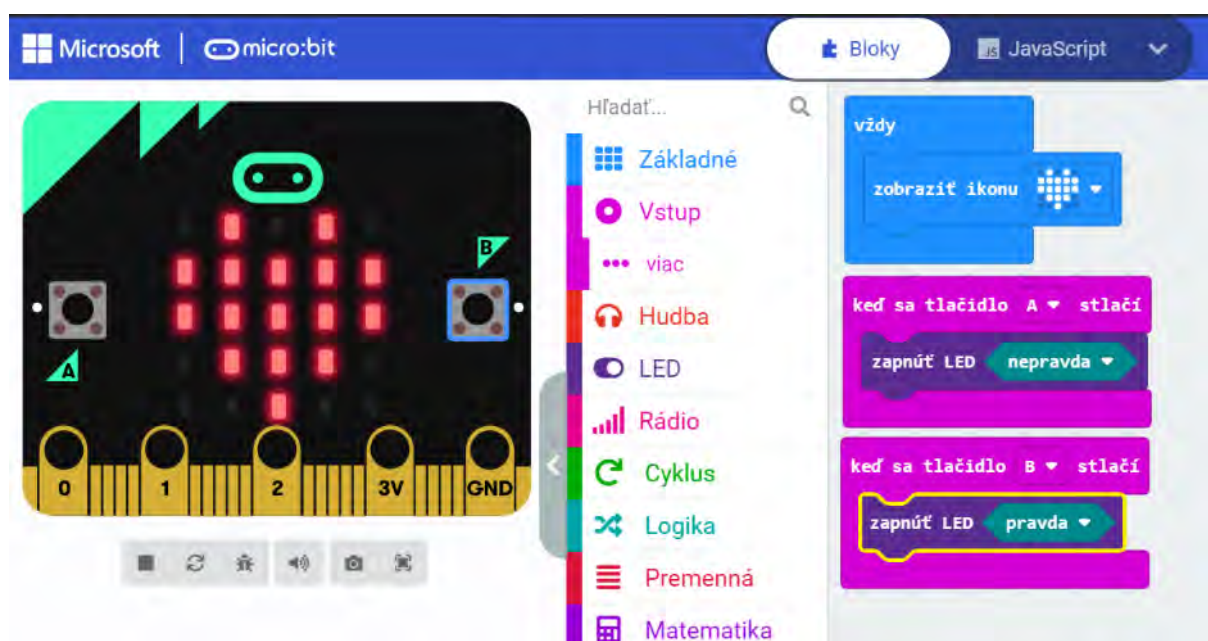
Príkaz "zapnúť LED nepravda"

Príkaz "zapnúť LED nepravda" v skutočnosti nastaví LED na vypnutý stav na LED matici micro:bita. Tento príkaz využíva logické hodnoty na ovládanie stavu LED diód.

Ako to funguje?

1. Nastavenie stavu: Tento príkaz vyžaduje zadanie hodnoty "nepravda", čo znamená, že LED displej bude vypnutý. Príkaz funguje aj na jednotlivé súradnice, nielen na celý displej.
2. Prepnutie stavu: LED displej bude vypnutý, ak je hodnota "nepravda".
3. Prepnutie stavu: LED displej bude zapnutý, ak je hodnota "pravda".

Príklad: Ak chceme vypnúť LED displej, použijeme nasledujúci blok kódu:



Obrázok 53 Príkaz "zapnúť LED nepravda"

Po spustení tohto kódu bude na LED displeji svietiť ikona srdca, po stlačení tlačidla A sa displej vypne a po stlačení tlačidla B sa znovu zapne.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Použite príkaz "zapnúť LED nepravda" vo svojom programe.
3. Nastavte súradnice a stav "nepravda", aby ste vyplli LED.
4. Spustíte svoj program a sledujte, ako micro:bit vypína LED na zadaných súradniciach!

Tento príkaz je veľmi užitočný na kontrolu stavu LED diód na LED matici micro:bitu.

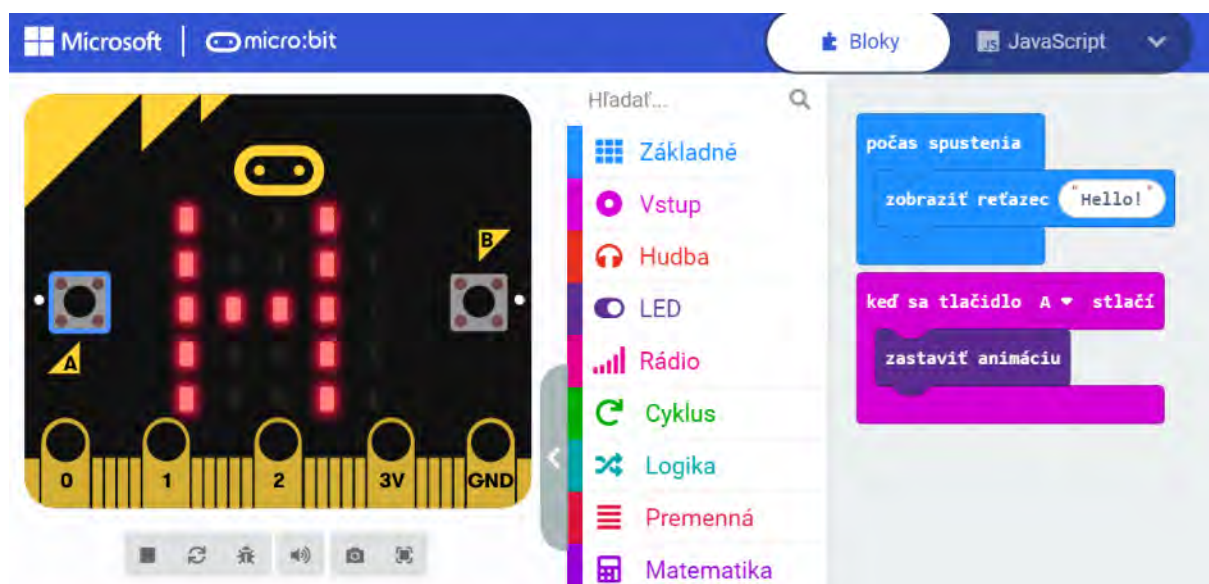
Príkaz "zastaviť animáciu"

Príkaz "zastaviť animáciu" slúži na okamžité zastavenie prebiehajúcej animácie na LED matici micro:bitu. Tento príkaz zastaví prehrávanie.

Ako to funguje?

- Zastavenie animácie: Tento príkaz zastaví akúkoľvek aktuálne prebiehajúcu animáciu na LED matici.

Príklad: Ak chcete zastaviť prebiehajúcu animáciu, použijete nasledujúci blok kódu:



Obrázok 54 Príkaz "zastaviť animáciu"

Po spustení tohto kódu bude micro:bit prehrávať animáciu pohybujúceho sa textu. Ak stlačíme tlačidlo A na micro:bite animácia sa zastaví.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Použite príkaz "zastaviť animáciu" vo svojom programe.
3. Spustíte svoj program a sledujte, ako micro:bit okamžite zastaví všetky animácie!

Tento príkaz je veľmi užitočný na ukončenie prehrávania animácií a resetovanie LED matice na micro:bite.

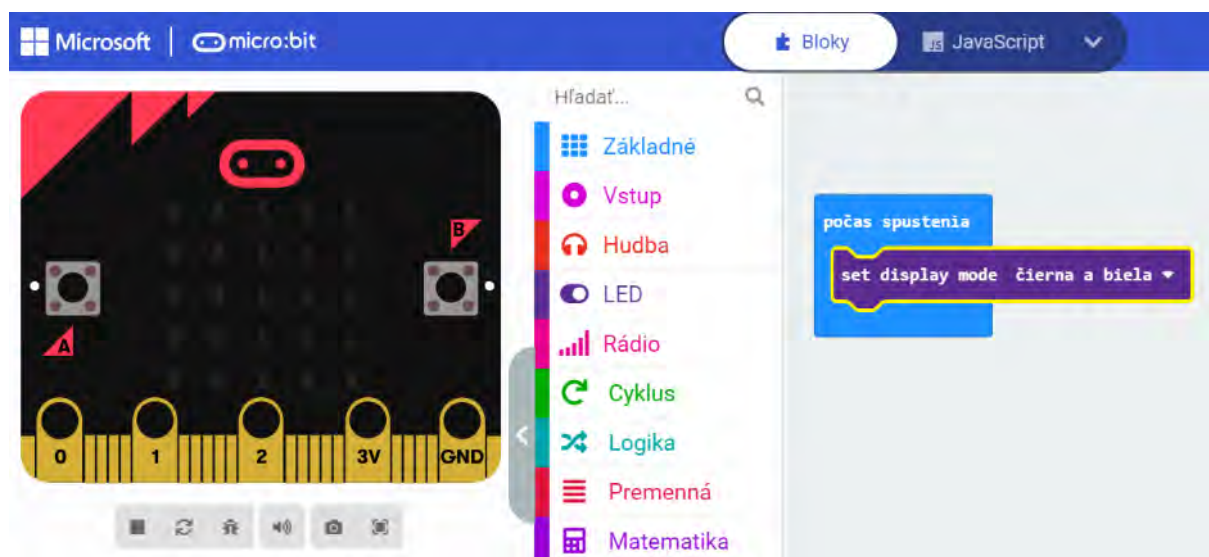
Príkaz "set display mode čierna a biela"

Príkaz "set display mode čierna a biela" slúži na nastavenie zobrazenia displeja micro:bita do režimu čierna a biela. Tento príkaz upraví zobrazenie tak, aby LED diódy na displeji micro:bita zobrazovali len dve farby: čiernu (vypnuté) a bielu (zapnuté).

Ako to funguje?

1. Nastavenie režimu: Tento príkaz nastaví zobrazenie displeja micro:bita do režimu, kde sa používajú len dve farby.
2. Zobrazenie: Všetky LED diódy na LED matici budú zobrazovať len stavy zapnuté (biela) alebo vypnuté (čierna).

Príklad: Ak chcete nastaviť displej do režimu čierna a biela, použijete nasledujúci blok kódu:



Obrázok 55 Príkaz "set display mode čierna a biela"

Po spustení tohto kódu bude micro:bit zobrazuje iba čierne a biele stavy na LED matici.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Použite príkaz "set display mode čierna a biela" vo svojom programe.
3. Spustíte svoj program a sledujte, ako micro:bit nastaví zobrazenie do režimu čierna a biela!

Tento príkaz je veľmi užitočný na vytváranie vizuálnych efektov, kde chcete zobraziť len jednoduché stavy zapnuté/vypnuté na LED matici micro:bita.

6 Rádio

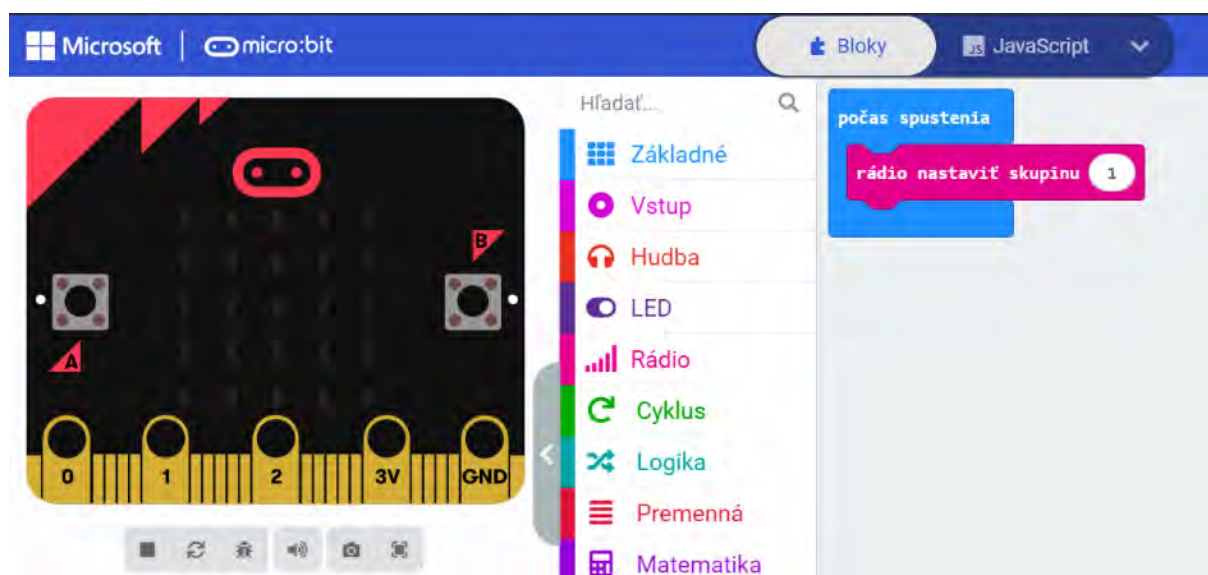
Príkaz "rádio nastaviť skupinu 1"

Príkaz "rádio nastaviť skupinu 1" slúži na nastavenie micro:bitu do konkrétnej skupiny pre rádiovú komunikáciu. Tento príkaz umožňuje micro:bitom, ktoré sú v rovnakej skupine, komunikovať medzi sebou cez rádiové signály.

Ako to funguje?

1. Nastavenie skupiny: Tento príkaz nastaví micro:bit na rádiovú skupinu s číslom 1. Micro:bity, ktoré sú nastavené na rovnakú skupinu, môžu medzi sebou posielat' a prijímať rádiové správy.
2. Rádiová komunikácia: Po nastavení skupiny môže micro:bit komunikovať s inými micro:bitmi v rovnakej skupine pomocou rádiových príkazov.

Príklad: Ak chcete nastaviť micro:bit do rádiovkej skupiny 1, použijete nasledujúci blok kódu:



Obrázok 56 Príkaz "rádio nastaviť skupinu 1"

Po spustení tohto kódu bude micro:bit nastavený na rádiovú skupinu 1 a bude schopný komunikovať s inými micro:bitmi v tejto skupine.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Použite príkaz "rádio nastaviť skupinu 1" vo svojom programe.
3. Spustíte svoj program a nastavíte micro:bit na rádiovú skupinu 1.
4. Sledujte, ako micro:bit komunikuje s inými micro:bitmi v rovnakej skupine!

Tento príkaz je veľmi užitočný na vytváranie projektov, kde micro:bit musí komunikovať s inými micro:bitmi cez rádiové signály.

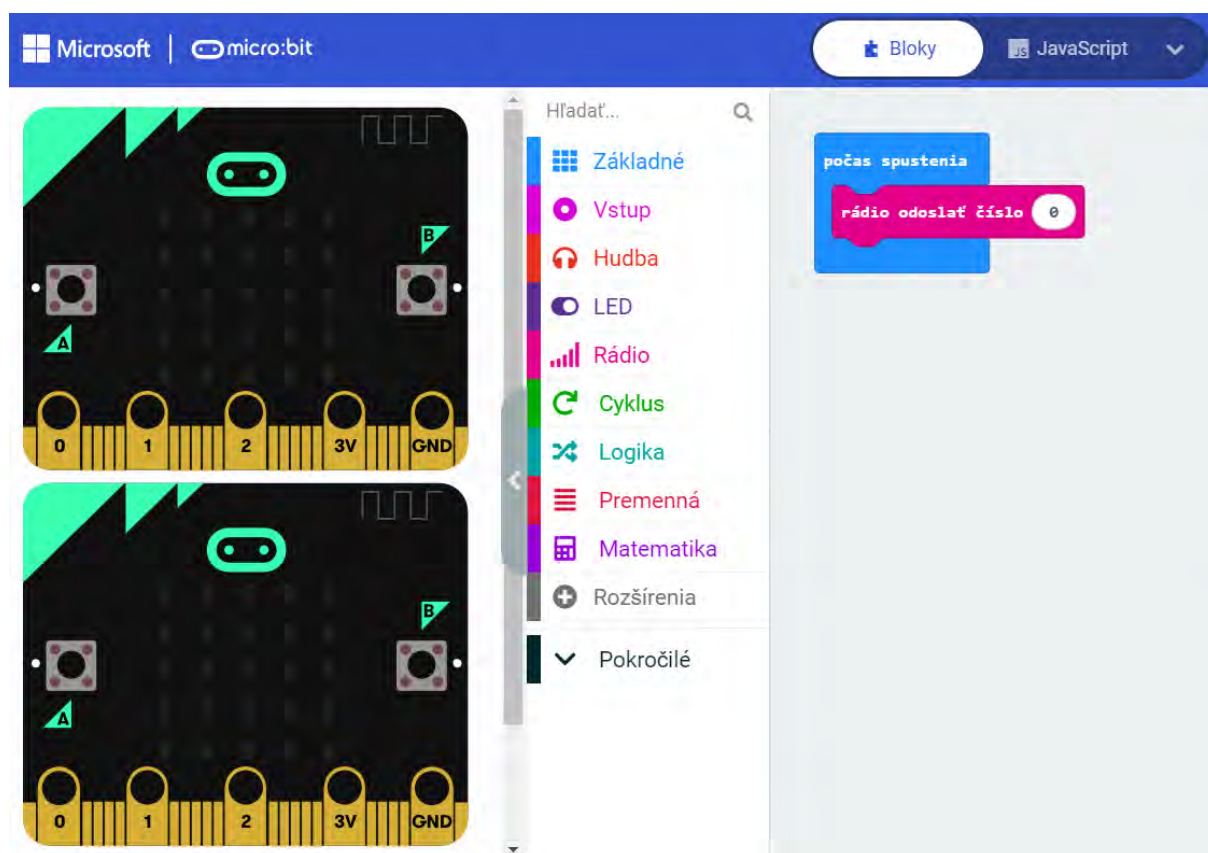
Príkaz "rádio odoslať číslo 0"

Príkaz "rádio odoslať číslo 0" slúži na odoslanie číselnej správy s hodnotou 0 pomocou rádiového signálu na micro:bite. Tento príkaz umožňuje komunikovať medzi micro:bitmi v rovnakej rádiovkej skupine.

Ako to funguje?

1. Odoslanie číselnej správy: Tento príkaz odošle rádiovú správu s číselnou hodnotou 0 do ostatných micro:bitov v rovnakej skupine.
2. Rádiová komunikácia: Micro:bit odošle hodnotu 0 pomocou rádiového signálu a ostatné micro:bity v rovnakej skupine môžu túto správu prijať.

Príklad: Ak chceme odoslať číslo 0 pomocou rádiového signálu, použijeme nasledujúci blok kódu:



Obrázok 57 Príkaz "rádio odoslať číslo 0"

Po spustení tohto kódu bude micro:bit odosielať číselnú hodnotu 0 cez rádiový signál do ostatných micro:bitov v rovnakej skupine.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Použite príkaz "rádio odoslať číslo 0" vo svojom programe.
3. Spustíte svoj program a sledujte, ako micro:bit odosiela hodnotu 0 cez rádiový signál!

Tento príkaz je veľmi užitočný na komunikáciu medzi micro:bitmi a na vytváranie interaktívnych projektov, kde micro:bit musí posielať a prijímať číselné hodnoty cez rádiové signály.

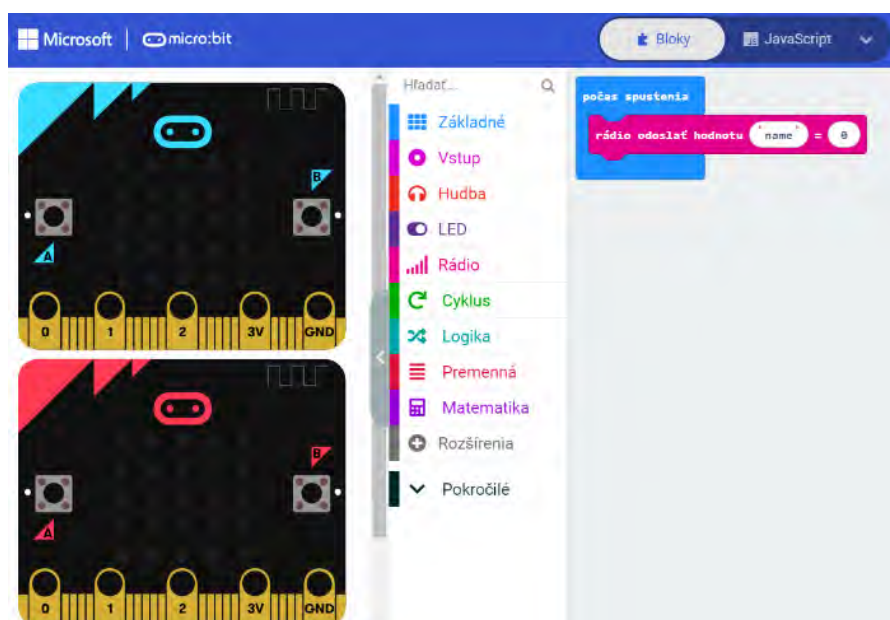
Príkaz "rádio odoslať hodnotu name = 0"

Príkaz "rádio odoslať hodnotu name = 0" slúži na odoslanie správy s pomenovanou hodnotou prostredníctvom rádiového signálu na micro:bit. Tento príkaz umožňuje posielať hodnoty pod konkrétnym názvom, čo je užitočné pre rozlíšenie rôznych údajov v rádiovéj komunikácii.

Ako to funguje?

1. Odoslanie pomenovanej hodnoty: Tento príkaz odošle rádiovú správu s hodnotou 0, ktorá je označená názvom "name".
2. Rádiová komunikácia: Micro:bit odošle hodnotu 0 s názvom "name" pomocou rádiového signálu a ostatné micro:bity v rovnakej skupine môžu túto správu prijať a rozpoznať podľa názvu.

Príklad: Ak chceme odoslať hodnotu 0 s názvom "name" pomocou rádiového signálu, použijeme nasledujúci blok kódu:



Obrázok 58 Príkaz "rádio odoslať hodnotu name = 0"

Po spustení tohto kódu bude micro:bit odosielať hodnotu 0 s názvom "name" cez rádiový signál do ostatných micro:bitov v rovnakej skupine.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Použite príkaz "rádio odoslať hodnotu name = 0" vo svojom programe.
3. Nastavte hodnotu a názov, ktoré chcete odoslať.
4. Spustíte svoj program a sledujte, ako micro:bit odosiela hodnotu 0 s názvom "name" cez rádiový signál!

Tento príkaz je veľmi užitočný na komunikáciu medzi micro:bitmi, kde je potrebné odosielať a prijímať rôzne údaje rozpoznateľné podľa názvov.

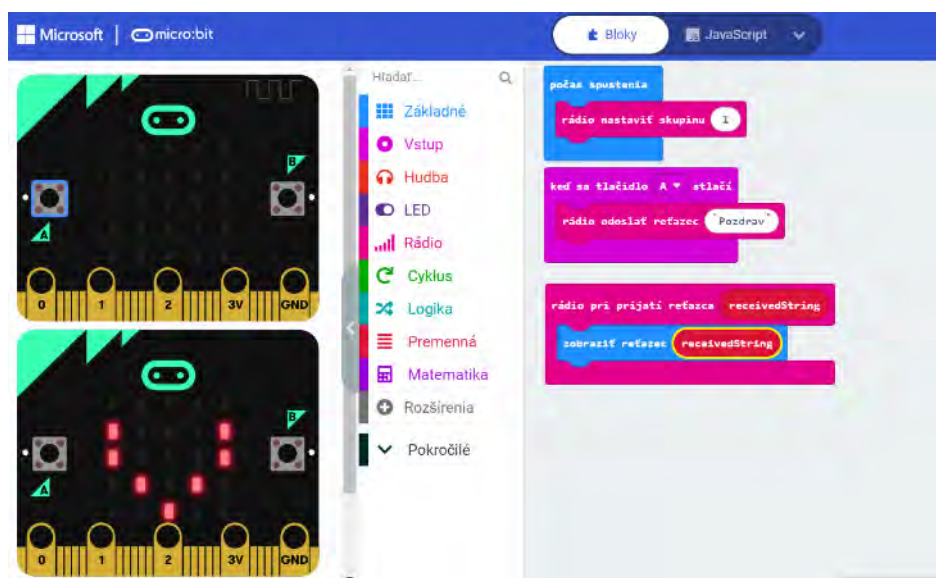
Príkaz "rádio odoslať reťazec"

Príkaz "rádio odoslať reťazec" slúži na odoslanie textovej správy pomocou rádiového signálu na micro:bite. Tento príkaz umožňuje komunikáciu medzi micro:bitmi v rovnakej rádiovkej skupine pomocou textových reťazcov.

Ako to funguje?

1. Odoslanie textovej správy: Tento príkaz odošle rádiovú správu s textovým reťazcom do ostatných micro:bitov v rovnakej skupine.
2. Rádiová komunikácia: Micro:bit odošle zadaný textový reťazec pomocou rádiového signálu a ostatné micro:bity v rovnakej skupine môžu túto správu prijať.

Príklad: Ak chceme odoslať textový reťazec "Pozdrav" pomocou rádiového signálu, použijeme nasledujúci blok kódu:



Obrázok 59 Príkaz "rádio odoslať reťazec"

Po spustení tohto kódu bude micro:bit odosielať textový reťazec "Pozdrav" cez rádiový signál do ostatných micro:bitov v rovnakej skupine.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Použite príkaz "rádio odoslať reťazec" vo svojom programe.
3. Nastavte textový reťazec, ktorý chcete odoslať.
4. Spustíte svoj program a sledujte, ako micro:bit odosiela textový reťazec cez rádiový signál!

Tento príkaz je veľmi užitočný na komunikáciu medzi micro:bitmi pomocou textových správ.

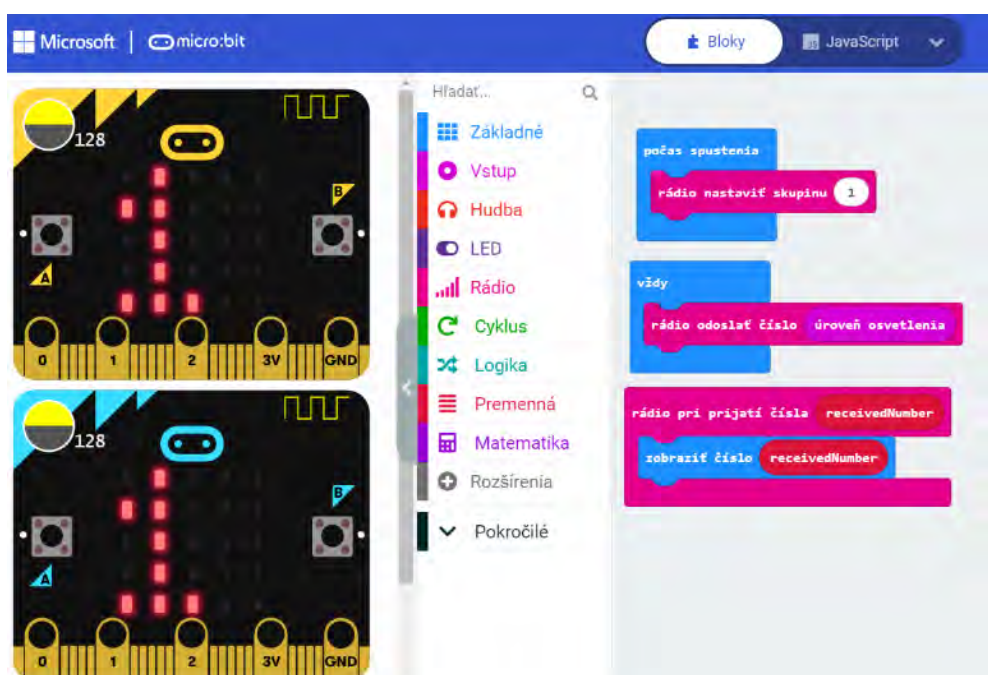
Príkaz "rádio pri prijatí čísla"

Príkaz "rádio pri prijatí čísla" slúži na vykonanie určitej akcie, keď micro:bit prijme číselnú správu prostredníctvom rádiového signálu. Tento príkaz je užitočný na reakciu na prijaté číselné hodnoty od iných micro:bitov v rovnakej rádiovkej skupine.

Ako to funguje?

1. Prijatie číselnej správy: Tento príkaz deteguje, keď micro:bit prijme rádiovú správu s číselnou hodnotou.
2. Vykonanie kódu: Po prijatí číselnej správy sa vykoná akcia, ktorú sme do príkazu vložili.

Príklad: Ak chceme zobrazit' prijaté číslo úrovně osvetlenia na LED matici, keď micro:bit prijme rádiovú správu s číselnou hodnotou, použijeme nasledujúci blok kódu:



Obrázok 60 Príkaz "rádio pri prijatí čísla"

Po spustení tohto kódu bude micro:bit zobrazovať prijatú číselnú hodnotu úrovně osvetlenia na LED matici.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Použite príkaz "rádio pri prijatí čísla" vo svojom programe.
3. Do príkazu vložte akciu, ktorú chcete vykonať, napríklad zobrazenie prijatého čísla.
4. Spustite svoj program a sledujte, ako micro:bit reaguje na prijaté číselné správy!

Tento príkaz je veľmi užitočný na vytváranie interaktívnych projektov, kde micro:bit reaguje na prijaté číselné hodnoty od iných micro:bitov.

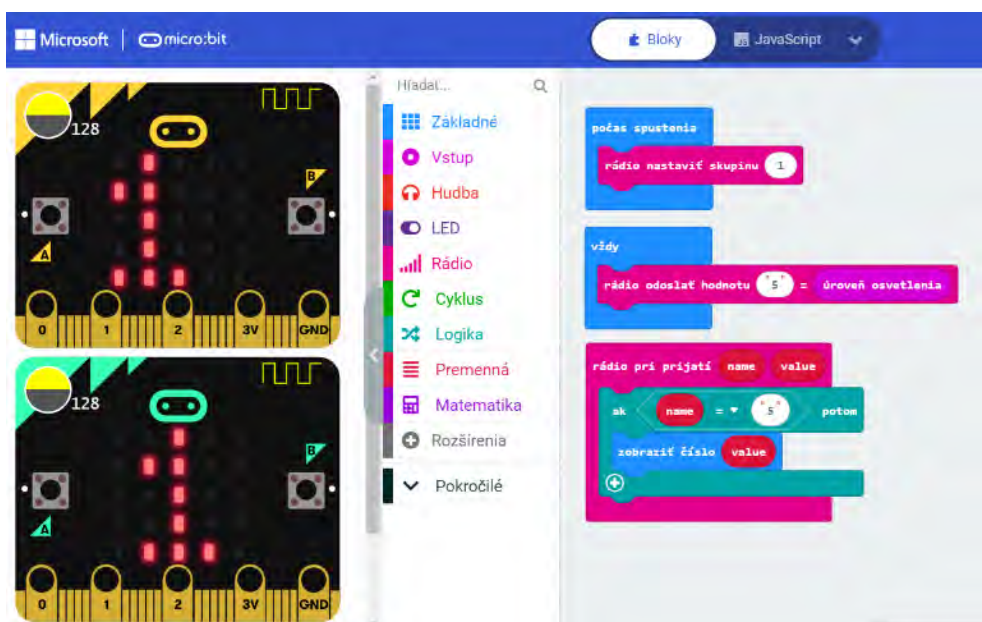
Príkaz "rádio pri prijatí name value"

Príkaz "rádio pri prijatí name value" slúži na vykonanie určitej akcie, keď micro:bit prijme správu s pomenovanou hodnotou prostredníctvom rádiového signálu. Tento príkaz umožňuje reagovať na prijaté hodnoty, ktoré sú označené konkrétnym názvom, od iných micro:bitov v rovnakej rádiovkej skupine.

Ako to funguje?

1. Prijatie pomenovanej hodnoty: Tento príkaz deteguje, keď micro:bit prijme rádiovú správu s hodnotou označenou konkrétnym názvom.
2. Vykonanie kódu: Po prijatí pomenovanej hodnoty sa vykoná akcia, ktorú sme do príkazu vložili.

Príklad: Ak chceme zobraziť prijatú hodnotu úrovne osvetlenia na LED matici, keď micro:bit prijme správu s názvom "5", použijeme nasledujúci blok kódu:



Obrázok 61 Príkaz "rádio pri prijatí name value"

Po spustení tohto kódu bude micro:bit zobrazovať hodnotu úrovne osvetlenia prijatú pod názvom "5" na LED matici.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Použite príkaz "rádio pri prijatí name value" vo svojom programe.
3. Do príkazu vložte akciu, ktorú chcete vykonať pri prijatí pomenovanej hodnoty, napríklad zobrazenie prijatej hodnoty.
4. Spustíte svoj program a sledujte, ako micro:bit reaguje na prijaté pomenované hodnoty!

Tento príkaz je veľmi užitočný na vytváranie interaktívnych projektov, kde micro:bit reaguje na prijaté hodnoty označené konkrétnymi názvami od iných micro:bitov.

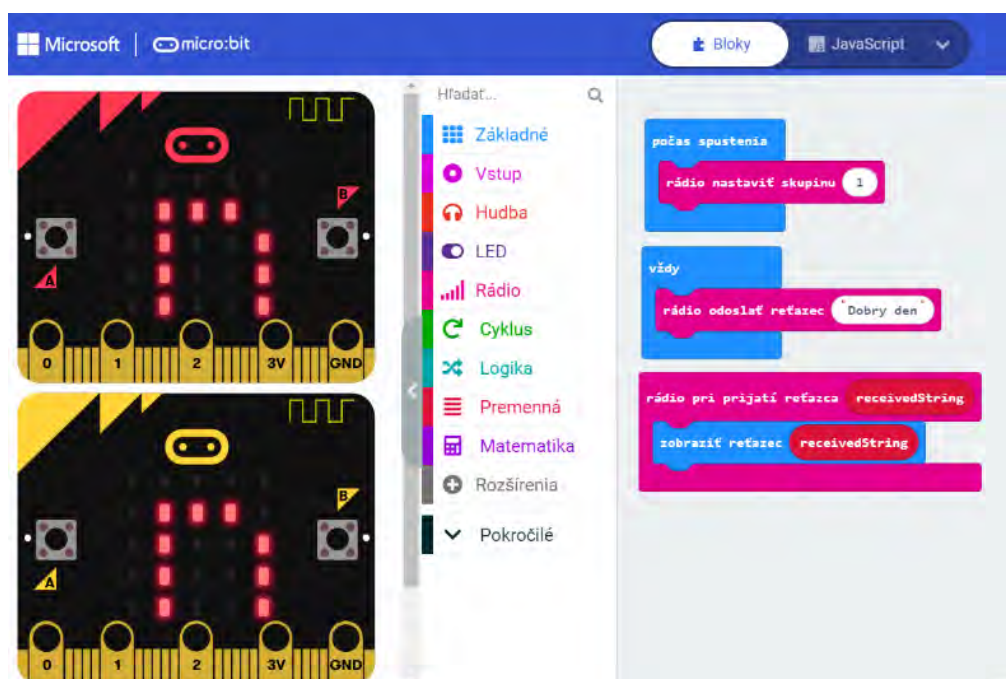
Príkaz "rádio pri prijatí reťazca receiveString"

Príkaz "rádio pri prijatí reťazca receiveString" slúži na vykonanie určitej akcie, keď micro:bit prijme textovú správu prostredníctvom rádiového signálu. Tento príkaz umožňuje reagovať na prijaté textové reťazce od iných micro:bitov v rovnakej rádiovkej skupine.

Ako to funguje?

1. Prijatie textovej správy: Tento príkaz deteguje, keď micro:bit prijme rádiovú správu s textovým reťazcom.
2. Vykonanie kódu: Po prijatí textovej správy sa vykoná akcia, ktorú sme do príkazu vložili.

Príklad: Ak chceme zobraziť prijatý reťazec na LED matici, keď micro:bit prijme rádiovú správu s textovým reťazcom, použijeme nasledujúci blok kódu:



Obrázok 62 Príkaz "rádio pri prijatí reťazca receiveString"

Po spustení tohto kódu bude micro:bit zobrazovať prijatý textový reťazec na LED matici.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Použite príkaz "rádio pri prijatí reťazca receiveString" vo svojom programe.
3. Do príkazu vložte akciu, ktorú chcete vykonať pri prijatí textovej správy, napríklad zobrazenie prijatého reťazca.
4. Spustite svoj program a sledujte, ako micro:bit reaguje na prijaté textové reťazce!

Tento príkaz je veľmi užitočný na vytváranie interaktívnych projektov, kde micro:bit reaguje na prijaté textové správy od iných micro:bitov.

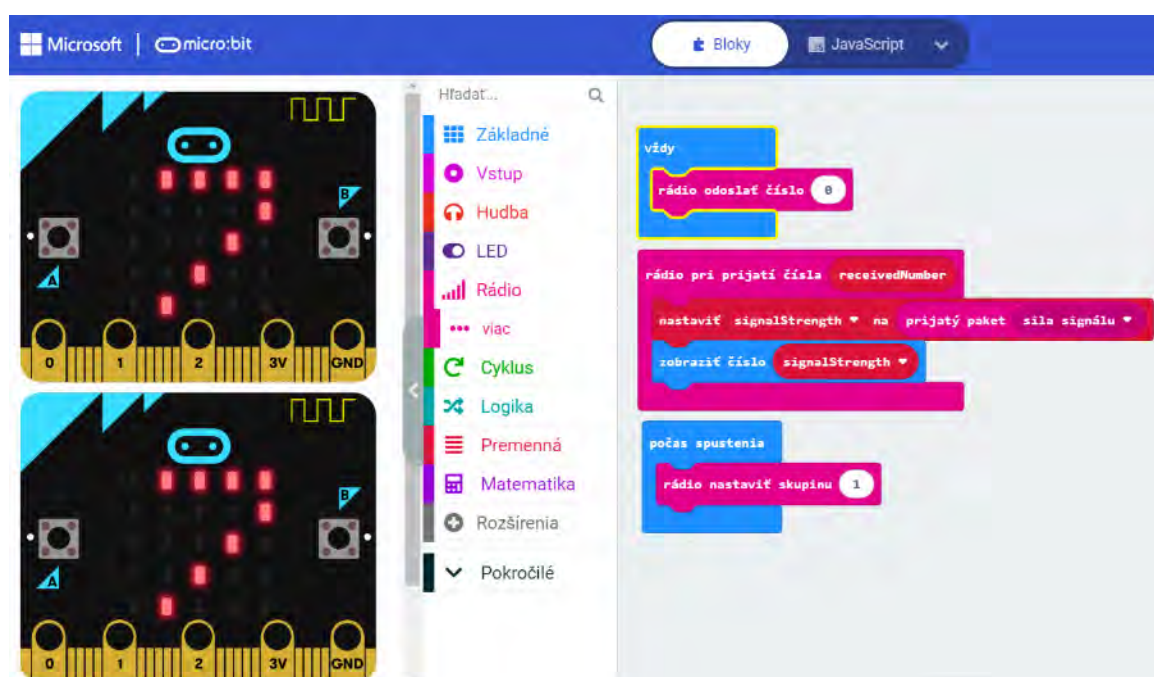
Príkaz "prijatý paket sila signálu"

Príkaz "prijatý paket sila signálu" slúži na získanie informácie o sile signálu prijatej rádiovej správy na micro:bit. Tento príkaz je užitočný na sledovanie kvality rádiovej komunikácie medzi micro:bitmi.

Ako to funguje?

1. Získanie sily signálu: Tento príkaz deteguje silu signálu prijatej rádiovej správy v dB (decibel).
2. Použitie sily signálu: Po prijatí rádiovej správy sa hodnota sily signálu môže použiť na rôzne akcie alebo analýzy.

Príklad: Ak chceme zobrazit' silu signálu prijatej rádiovej správy na LED matici, použijeme nasledujúci blok kódu:



Obrázok 63 Príkaz "prijatý paket sila signálu"

Po spustení tohto kódu bude micro:bit zobrazovať silu signálu prijatej rádiovej správy na LED matici.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Použite príkaz "prijatý paket sila signálu" vo svojom programe.
3. Do príkazu vložte akciu, ktorú chcete vykonať s hodnotou sily signálu, napríklad zobrazenie tejto hodnoty.
4. Spustíte svoj program a sledujte, ako micro:bit zobrazí silu signálu prijatej rádiovej správy!

Tento príkaz je veľmi užitočný na monitorovanie kvality rádiovej komunikácie a analýzu sily prijatého signálu od iných micro:bitov.

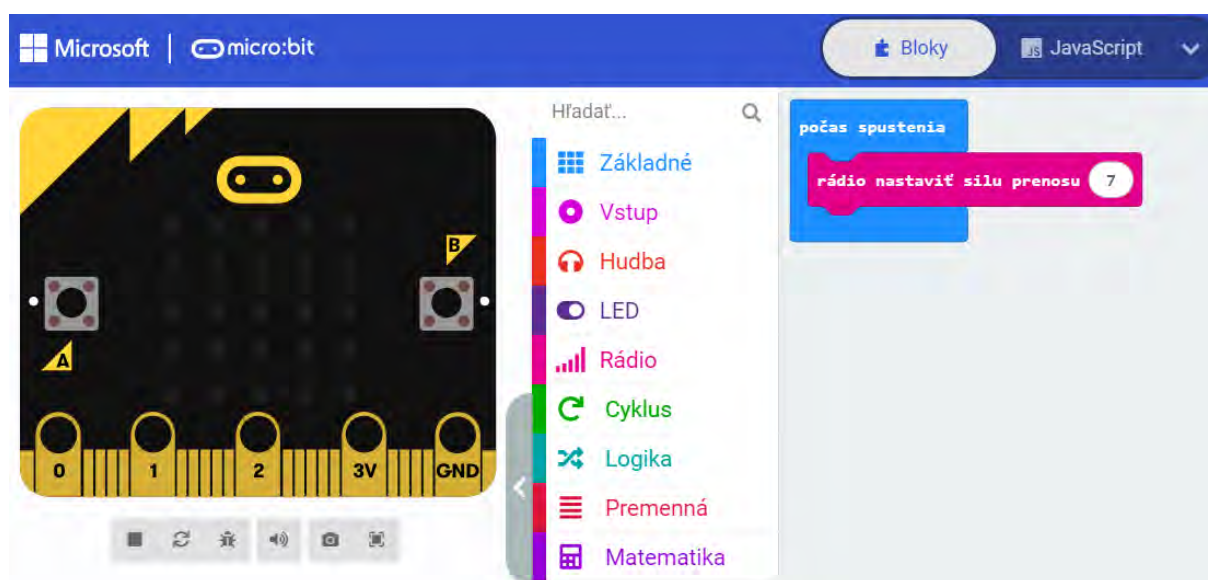
Príkaz "rádio nastaviť silu prenosu"

Príkaz "rádio nastaviť silu prenosu" slúži na nastavenie vysielacieho výkonu (sily) rádiového signálu na micro:bit. Tento príkaz umožňuje kontrolovať silu prenosu signálu pri rádiovkej komunikácii medzi micro:bitmi, čo môže ovplyvniť dosah a kvalitu komunikácie.

Ako to funguje?

1. Nastavenie sily prenosu: Tento príkaz vyžaduje zadanie hodnoty, ktorá určuje silu rádiového signálu. Hodnota môže byť v rozsahu od 0 (najnižšia sila) do 7 (najvyššia sila).
2. Vysielací výkon: Po nastavení sily prenosu bude micro:bit používať zadaný výkon pri vysielaní rádiových správ.

Príklad: Ak chcete nastaviť silu prenosu na maximálnu hodnotu 7, použijete nasledujúci blok kódu:



Obrázok 64 Príkaz "rádio nastaviť silu prenosu"

Po spustení tohto kódu bude micro:bit vysielat' rádiový signál s maximálnou silou.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Použite príkaz "radio nastaviť silu prenosu" vo svojom programe.
3. Nastavte silu prenosu na požadovanú hodnotu.
4. Spustite svoj program a sledujte, ako micro:bit vysiela rádiový signál s nastavenou silou!

Tento príkaz je veľmi užitočný na kontrolu dosahu a kvality rádiovkej komunikácie medzi micro:bitmi.

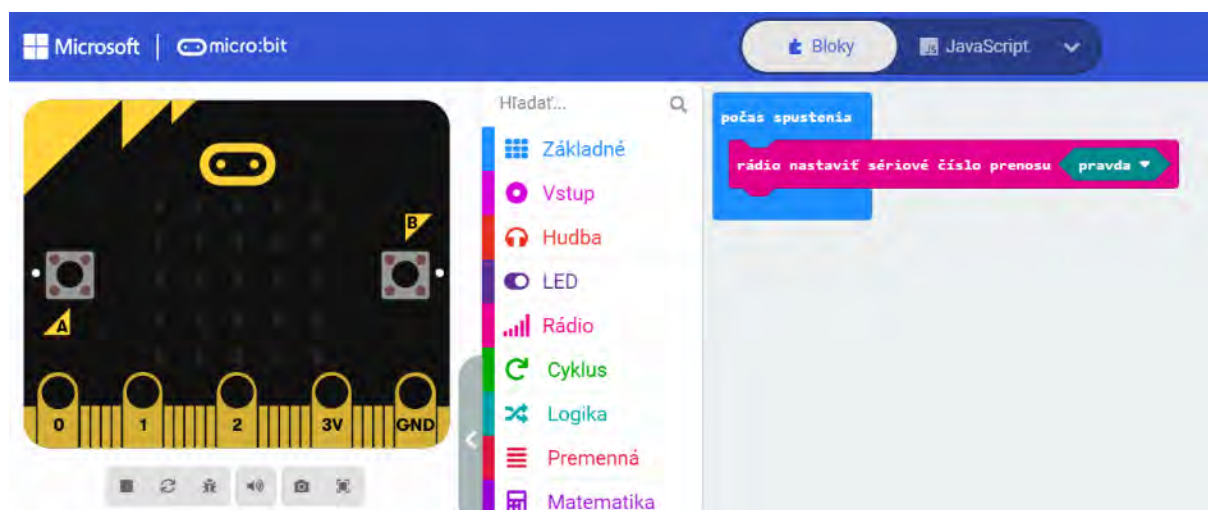
Príkaz "rádio nastaviť sériové číslo prenosu pravda"

Príkaz "rádio nastaviť sériové číslo prenosu pravda" slúži na povolenie alebo zakázanie prenosu sériového čísla micro:bita spolu s každou rádiovou správou. Tento príkaz umožňuje odosielať identifikačné informácie o konkrétnom micro:bita spolu s dátami, čo môže byť užitočné pre identifikáciu a autentifikáciu zariadení v rádiovéj sieti.

Ako to funguje?

1. Nastavenie sériového čísla: Tento príkaz umožňuje povoliť (pravda) alebo zakázať (nepravda) prenos sériového čísla micro:bita pri každej rádiovéj komunikácii.
2. Rádiová komunikácia: Po povolení prenosu sériového čísla sa identifikačné informácie micro:bita odosielať spolu s každou rádiovou správou.

Príklad: Ak chcete povoliť prenos sériového čísla, použijete nasledujúci blok kódu:



Obrázok 65 Príkaz "rádio nastaviť sériové číslo prenosu pravda"

Po spustení tohto kódu bude micro:bit odosielať sériové číslo spolu s každou rádiovou správou.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Použite príkaz "rádio nastaviť sériové číslo prenosu pravda" vo svojom programe.
3. Spustíte svoj program a sledujte, ako micro:bit odosiela sériové číslo spolu s rádiovými správami!

Tento príkaz je veľmi užitočný na zabezpečenie identifikácie zariadení v rádiovéj sieti a na vytváranie pokročilých interaktívnych projektov, kde je potrebné vedieť, ktorý konkrétny micro:bit odoslal správu.

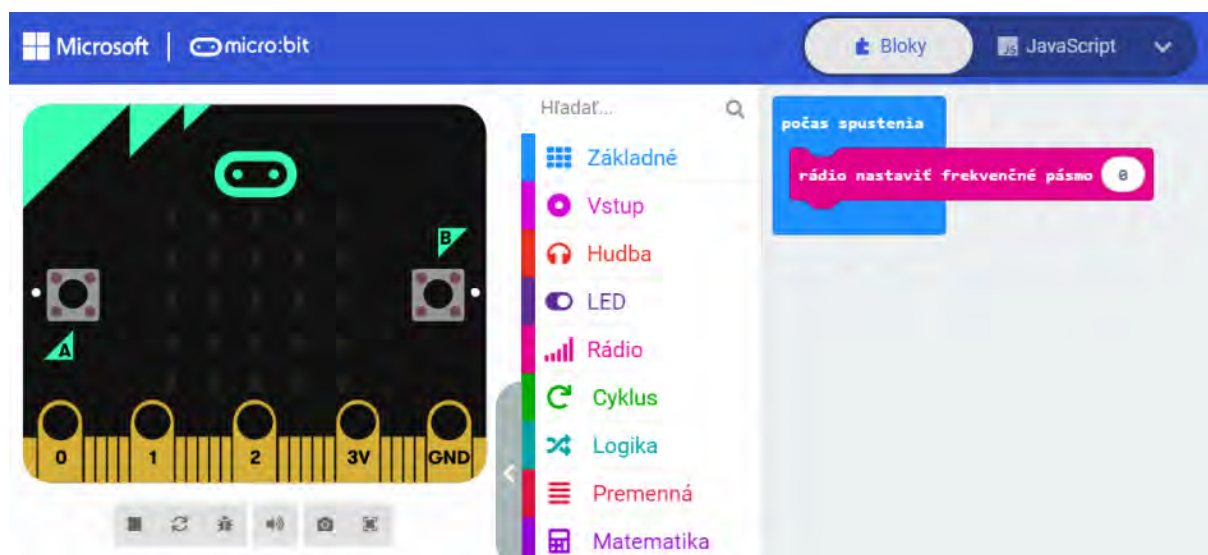
Príkaz "rádio nastaviť frekvenčné pásmo 0"

Príkaz "rádio nastaviť frekvenčné pásmo 0" slúži na nastavenie micro:bita na konkrétne frekvenčné pásmo pre rádiovú komunikáciu. Tento príkaz umožňuje určiť, na akom frekvenčnom pásme bude micro:bit komunikovať, čo je užitočné pre optimalizáciu a riadenie rádiových prenosov.

Ako to funguje?

1. Nastavenie frekvenčného pásma: Tento príkaz vyžaduje zadanie hodnoty, ktorá určuje frekvenčné pásmo. Frekvenčné pásmo 0 je predvolené pásmo používané micro:bitom.
2. Rádiová komunikácia: Po nastavení frekvenčného pásma bude micro:bit komunikovať na zvolenom pásme, čo môže ovplyvniť dosah a kvalitu rádiových prenosov.

Príklad: Ak chcete nastaviť micro:bit na frekvenčné pásmo 0, použijete nasledujúci blok kódu:



Obrázok 66 Príkaz "rádio nastaviť frekvenčné pásmo 0"

Po spustení tohto kódu bude micro:bit nastavený na predvolené frekvenčné pásmo 0 pre rádiovú komunikáciu.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Použite príkaz "rádio nastaviť frekvenčné pásmo 0" vo svojom programe.
3. Nastavte frekvenčné pásmo na požadovanú hodnotu.
4. Spustíte svoj program a sledujte, ako micro:bit komunikuje na zvolenom frekvenčnom pásme!

Tento príkaz je veľmi užitočný na kontrolu frekvenčných pásiem používaných pri rádiovéj komunikácii medzi micro:bitmi.

Príkaz "rádio volať udalosť"

Príkaz "rádio volať udalosť" slúži na vykonanie konkrétnej akcie pri prijatí rádiovej správy na micro:bit. Tento príkaz umožňuje micro:bitu reagovať na prijaté rádiové správy tým, že spustí určenú udalosť alebo akciu.

Ako to funguje?

1. Stlačením tlačidla A na prvom micro:bit sa odošle text "piskaj".
2. Druhý micro:bit správu prijme, rozpozná text "piskaj" a spustí príkaz "rádio vyvolať udalosť".
3. Micro:bit zareaguje na túto udalosť tým, že zahrá tón.

Príklad: Ak chceme aby micro:bit po prijatí textu „píska“ zahrá tón, použijeme nasledujúci blok kódu:



Obrázok 67 Príkaz "rádio volať udalosť"

Po spustení tohto kódu micro:bit zahrá tón Stredné C

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Vytvorte dva programy – pre odosielateľa a prijímateľa.
3. Na prijímacom micro:bit definujte udalosť a použite príkaz *rádio volať udalosť*.
4. Na odosielačom micro:bit použite *rádio poslať reťazec*.
5. Nahrajte programy do dvoch micro:bitov a stlačte tlačidlo A – micro:bit 2 zahrá tón.

Tento príkaz je veľmi užitočný na vytváranie interaktívnych projektov, kde micro:bit reaguje na prijaté rádiové správy

7 Cyklus

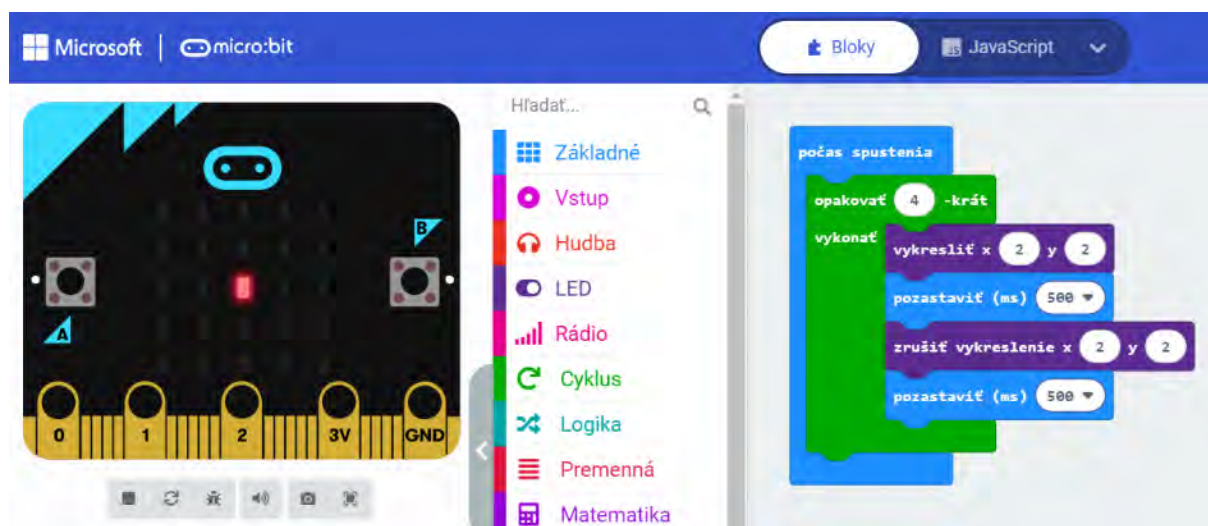
Príkaz "opakovať 4 krát"

Príkaz "opakovať 4 krát" slúži na vykonanie určitej akcie alebo súboru akcií štyrikrát za sebou. Tento príkaz umožňuje opakovať blok kódu stanovený počet krát, čo je užitočné na automatizáciu opakujúcich sa úloh.

Ako to funguje?

1. Nastavenie opakovania: Tento príkaz určuje, že nasledujúci blok kódu sa má vykonať štyrikrát.
2. Vykonanie kódu: Počet opakovaní je pevne stanovený na štyri, takže nasledujúci kód sa vykoná presne štyrikrát.

Príklad: Ak chceme rozsvietiť a zhasnúť LED na pozícii (2, 2) štyrikrát za sebou, použijeme nasledujúci blok kódu:



Obrázok 68 Príkaz "opakovať 4 krát"

Po spustení tohto kódu bude LED na pozícii (2, 2) štyrikrát bliknúť s pauzou medzi zapnutím a vypnutím.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Použite príkaz "opakovať 4 krát" vo svojom programe.
3. Vložte akcie, ktoré chcete vykonať opakovane, napríklad rozsvietenie a zhasnutie LED.
4. Spustite svoj program a sledujte, ako micro:bit vykoná akcie presne štyrikrát!

Tento príkaz je veľmi užitočný na automatizáciu opakujúcich sa úloh a vytváranie interaktívnych projektov na micro:bite.

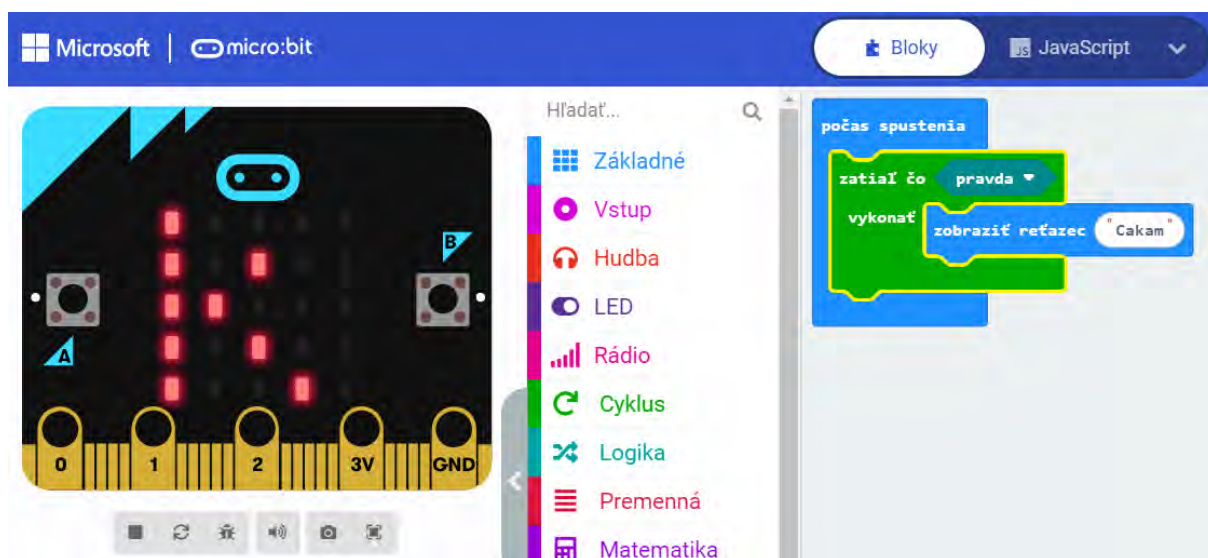
Príkaz "zatiaľ čo pravda"

Príkaz "zatiaľ čo pravda" slúži na vytvorenie cyklu, ktorý sa opakuje, pokiaľ je podmienka pravdivá. Tento príkaz umožňuje opakovane vykonávať blok kódu, kým sa nezmení stav podmienky na nepravdivý.

Ako to funguje?

1. Nastavenie podmienky: Tento príkaz obsahuje podmienku, ktorá je spoiatku pravdivá (true).
2. Opakovanie kódu: Blok kódu sa bude opakovane vykonávať, pokiaľ je podmienka pravdivá. Cyklus sa zastaví, keď sa podmienka zmení na nepravdivú (false).

Príklad: Ak chceme neustále zobrazit' text "Cakam", pokiaľ je podmienka pravdivá, použijeme nasledujúci blok kódu:



Obrázok 69 Príkaz "zatiaľ čo pravda"

Po spustení tohto kódu sa blok kódu nebude vykonávať, pretože podmienka je vždy nepravdivá.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Použite príkaz "zatiaľ čo pravda" vo svojom programe.
3. Do príkazu vložte akcie, ktoré chcete vykonať opakovane, kým je podmienka pravdivá.
4. Spustite svoj program a sledujte, ako micro:bit vykonáva akcie podľa podmienky!

Tento príkaz je veľmi užitočný na vytváranie opakujúcich sa úloh, ktoré sa vykonávajú, kým sa nesplní určitá podmienka.

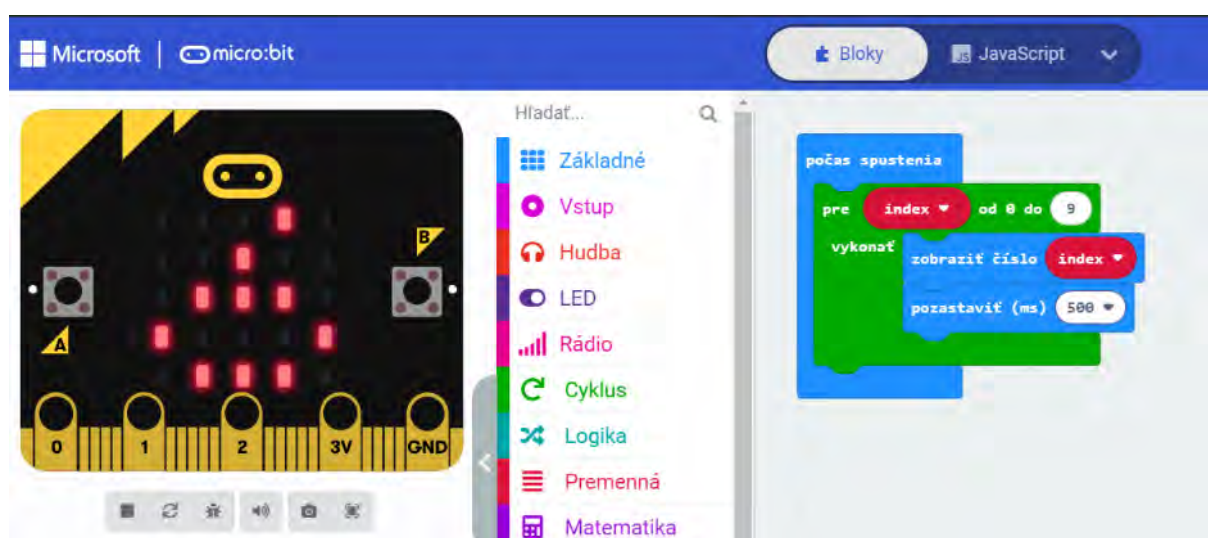
Príkaz "pre index od 0 do 9"

Príkaz "pre index od 0 do 9" slúži na vytvorenie cyklu, ktorý postupne vykonáva určitý blok kódu s indexom premennej od hodnoty 0 do hodnoty 9. Tento príkaz umožňuje opakovanie akcie konkrétny počet krát, pričom hodnota premennej sa postupne mení v každom opakovaní.

Ako to funguje?

1. Nastavenie rozsahu: Tento príkaz definuje rozsah indexu premennej od 0 do 9.
2. Vykonanie kódu: Blok kódu sa vykoná 10 krát, pričom hodnota premennej index sa v každom opakovaní postupne zvyšuje od 0 do 9.

Príklad: Ak chceme zobrazit' hodnotu premennej index na LED matici, použijeme nasledujúci blok kódu:



Obrázok 70 Príkaz "pre index od 0 do 9"

Po spustení tohto kódu bude micro:bit zobrazovať hodnoty od 0 do 9, pričom každá hodnota bude zobrazená na 500 milisekúnd.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Použite príkaz "pre index od 0 do 9" vo svojom programe.
3. Vložte akcie, ktoré chcete vykonať opakovane, napríklad zobrazenie hodnoty premennej index.
4. Spustíte svoj program a sledujte, ako micro:bit vykonáva akcie pre každý index od 0 do 9!

Tento príkaz je veľmi užitočný na automatizáciu opakujúcich sa úloh a na prácu s postupne sa meniacimi hodnotami premennej.

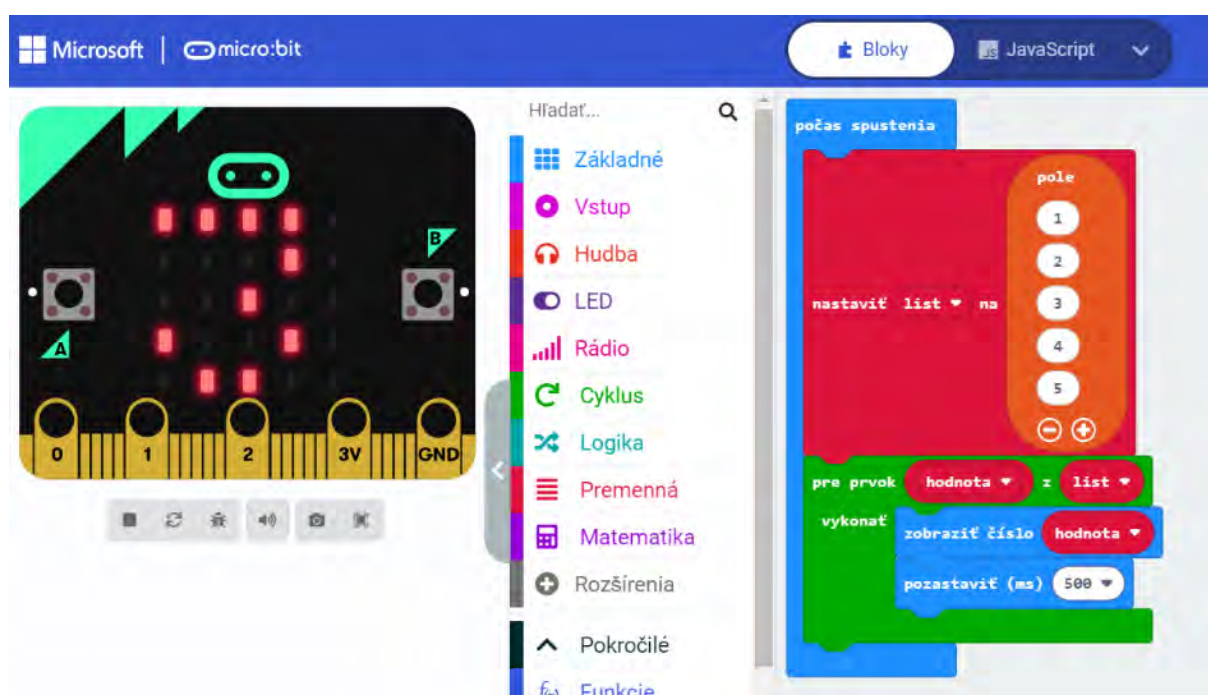
Príkaz "pre prvok hodnota z list"

Príkaz "pre prvok hodnota z list" slúži na iterovanie cez každý prvok v zozname a vykonanie určitej akcie pre každý prvok. Tento príkaz je užitočný na prácu s kolekciami údajov, kde chcete spracovať každý prvok v zozname postupne.

Ako to funguje?

1. Nastavenie iterácie: Tento príkaz definuje iteráciu cez každý prvok v zozname.
2. Vykonanie kódu: Blok kódu sa vykoná pre každý prvok v zozname, pričom premenná hodnota predstavuje aktuálny prvok.

Príklad: Ak máme zoznam čísel a chceme zobrazit' každé číslo na LED matici, použijeme nasledujúci blok kódu:



Obrázok 71 Príkaz "pre prvok hodnota z list"

Po spustení tohto kódu bude micro:bit zobrazovať každé číslo zo zoznamu s pauzou medzi zobrazeniami.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Použite príkaz "pre prvok hodnota z list" vo svojom programe.
3. Definujte zoznam a vložte akcie, ktoré chcete vykonať pre každý prvok.
4. Spustíte svoj program a sledujte, ako micro:bit vykonáva akcie pre každý prvok zoznamu!

Tento príkaz je veľmi užitočný na prácu s kolekciami údajov a na automatizáciu opakujúcich sa úloh pre každý prvok zoznamu.

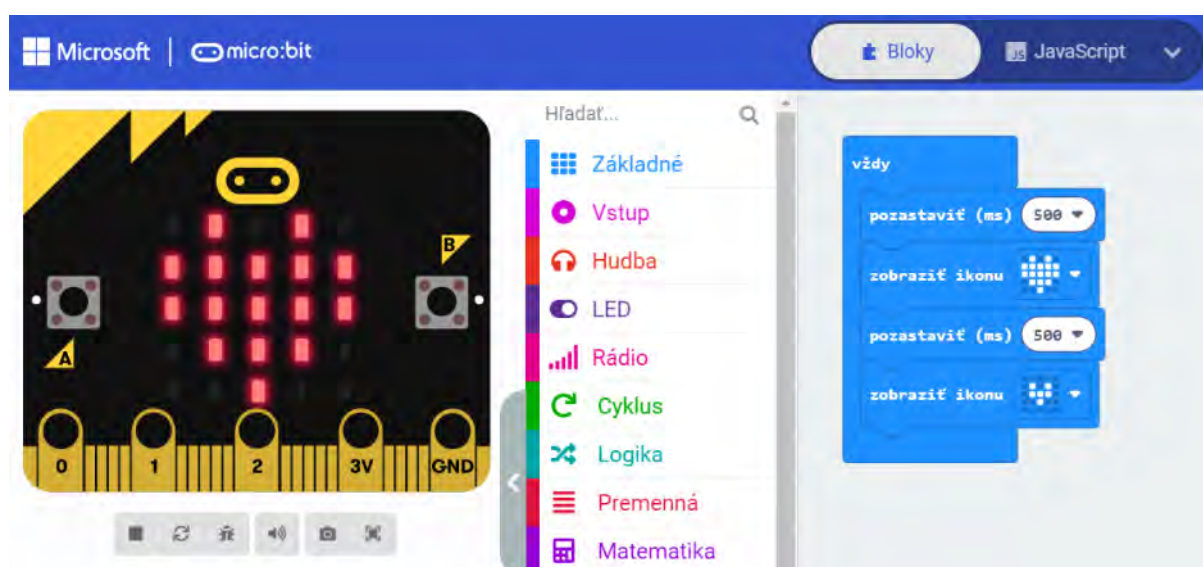
Príkaz "každých 500 ms"

Príkaz "každých 500 ms" slúži na opakované vykonanie určitej akcie každých 500 milisekúnd (ms). Tento príkaz umožňuje vytvoriť časovú slučku, ktorá vykonáva blok kódu v pravidelných intervaloch.

Ako to funguje?

1. Nastavenie intervalu: Tento príkaz nastaví interval na 500 ms, teda pol sekundy.
2. Vykonanie kódu: Blok kódu sa bude opakovane vykonávať každých 500 ms, kým sa program nezastaví alebo nezmení podmienky.

Príklad: Ak chceme každých 500 ms zmeniť zobrazenie ikony na LED matici, použijeme nasledujúci blok kódu:



Obrázok 72 Príkaz "každých 500 ms"

Po spustení tohto kódu bude micro:bit každých 500 ms prepínať medzi ikonami srdca a malého srdca.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Použite príkaz "každých 500 ms" vo svojom programe.
3. Do príkazu vložte akcie, ktoré chcete vykonať v pravidelných intervaloch.
4. Spustite svoj program a sledujte, ako micro:bit vykonáva akcie každých 500 ms!

Tento príkaz je veľmi užitočný na vytváranie opakujúcich sa úloh a interaktívnych projektov s pravidelnými časovými intervalmi.

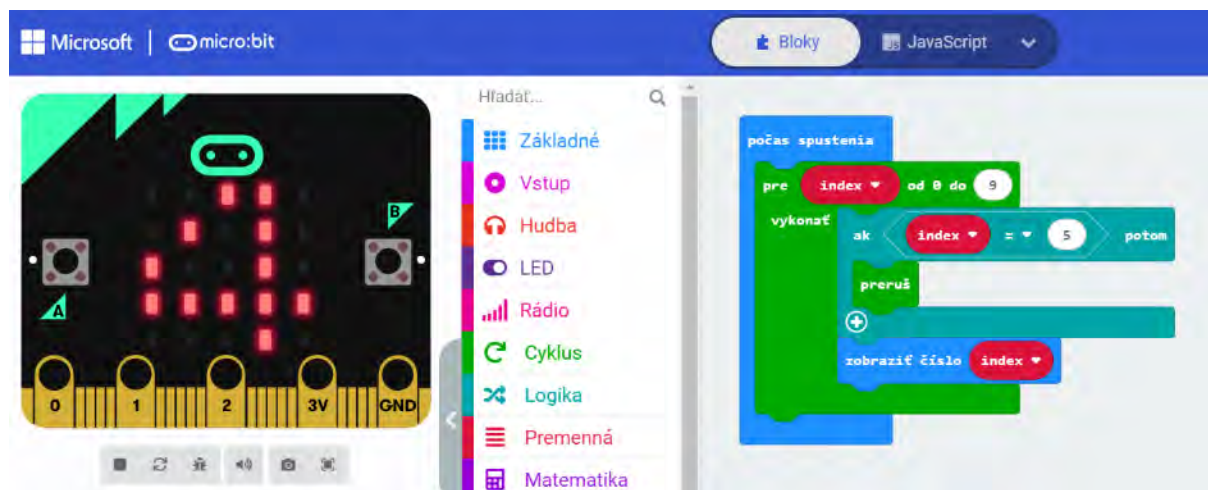
Príkaz "preruš"

Príkaz "preruš" slúži na okamžité ukončenie vykonávania aktuálneho cyklu alebo opakujúceho sa bloku kódu. Tento príkaz je užitočný, ak chcete predčasne ukončiť slučku na základe určitej podmienky.

Ako to funguje?

1. Detekcia príkazu: Keď príkaz "preruš" narazí vo vnútri cyklu, okamžite ukončí vykonávanie tohto cyklu.
2. Ukončenie slučky: Po ukončení cyklu sa vykonávanie programu presunie za tento cyklus.

Príklad: Ak chceme ukončiť slučku, keď sa hodnota index rovná 5, použijeme nasledujúci blok kódu:



Obrázok 73 Príkaz "preruš"

Po spustení tohto kódu bude micro:bit zobrazovať hodnoty od 0 do 4 a ukončí slučku, keď sa hodnota index stane 5.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Použite príkaz "preruš" vo svojom programe.
3. Nastavte podmienku, kedy chcete ukončiť vykonávanie slučky.
4. Spustíte svoj program a sledujte, ako micro:bit ukončí slučku na základe zadaných podmienok!

Tento príkaz je veľmi užitočný na kontrolu toku vykonávania programu a na predčasné ukončenie opakujúcich sa úloh.

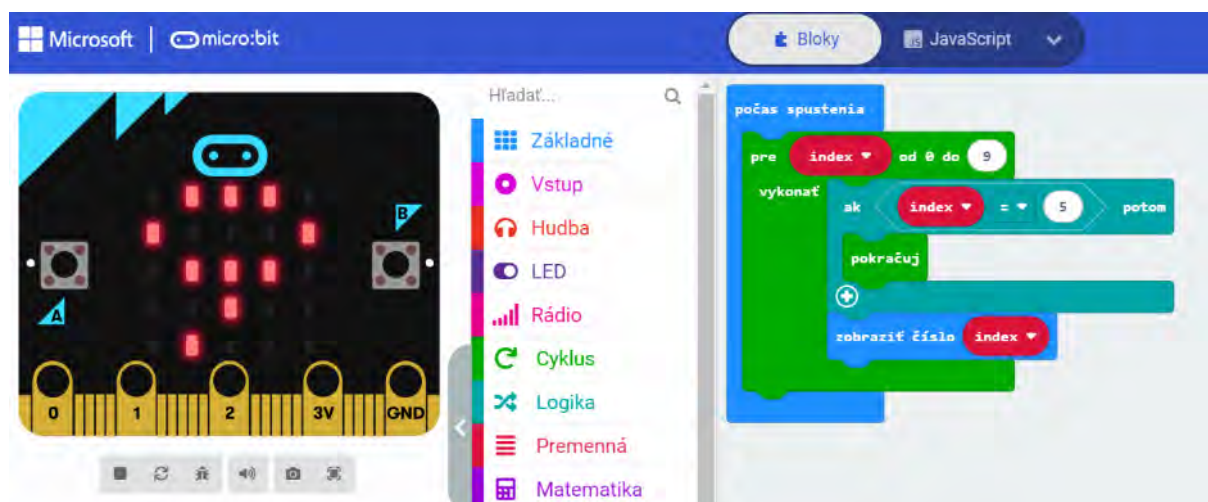
Príkaz "pokračuj"

Príkaz "pokračuj" slúži na preskočenie zvyšných príkazov v aktuálnom opakovaní cyklu a pokračovanie s ďalším opakovaním. Tento príkaz je užitočný, ak chcete niektoré iterácie cyklu vynechať na základe určitej podmienky.

Ako to funguje?

1. Detekcia príkazu: Keď príkaz "pokračuj" narazí vo vnútri cyklu, okamžite preskočí zvyšné príkazy v aktuálnom opakovaní.
2. Prechod na ďalšie opakovanie: Po preskočení sa vykonávanie programu presunie na ďalšie opakovanie cyklu.

Príklad: Ak chceme preskočiť zobrazenie čísla 5 a pokračovať s ďalšími číslami, použijeme nasledujúci blok kódu:



Obrázok 74 Príkaz "pokračuj"

Po spustení tohto kódu bude micro:bit zobrazovať hodnoty od 0 do 9, okrem čísla 5, ktoré bude preskočené.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Použite príkaz "pokračuj" vo svojom programe.
3. Nastavte podmienku, kedy chcete preskočiť vykonávanie zvyšných príkazov v cykle.
4. Spustíte svoj program a sledujte, ako micro:bit preskočí zvyšné príkazy a pokračuje s ďalším opakovaním!

Tento príkaz je veľmi užitočný na kontrolu toku vykonávania programu a na vynechanie určitých iterácií cyklu.

8 Logika

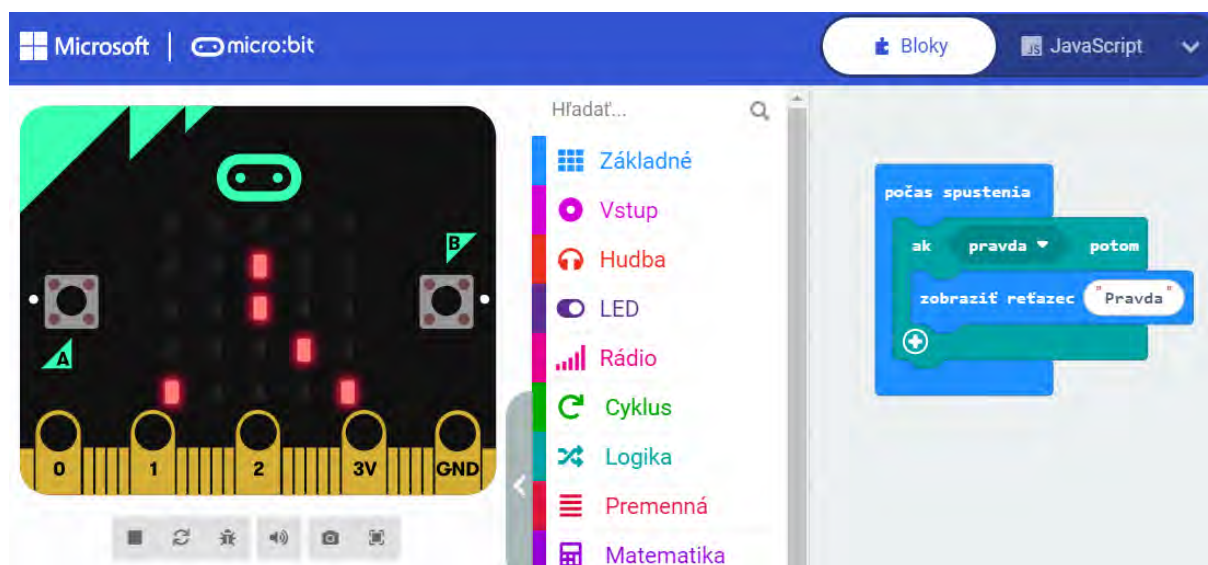
Príkaz "ak pravda potom"

Príkaz "ak pravda potom" slúži na vykonanie určitej akcie alebo bloku kódu, ak je zadaná podmienka pravdivá. Tento príkaz umožňuje kontrolovať tok vykonávania programu na základe podmienky.

Ako to funguje?

1. Kontrola podmienky: Tento príkaz vyžaduje zadanie podmienky, ktorá bude vyhodnotená ako pravdivá alebo nepravdivá.
2. Vykonanie kódu: Ak je podmienka pravdivá, vykoná sa blok kódu uvedený v príkaze "ak pravda potom".

Príklad: Ak chceme zobraziť text "Pravda" na LED matici, keď je podmienka true, použijeme nasledujúci blok kódu:



Obrázok 75 Príkaz "ak pravda potom"

Po spustení tohto kódu bude micro:bit zobrazovať text "Pravda", pretože podmienka je pravdivá.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Použite príkaz "ak pravda potom" vo svojom programe.
3. Nastavte podmienku a akciu, ktorú chcete vykonať, keď je podmienka pravdivá.
4. Spustíte svoj program a sledujte, ako micro:bit vykonáva akciu, keď je podmienka pravdivá!

Tento príkaz je veľmi užitočný na kontrolu toku vykonávania programu a na rozhodovanie na základe podmienok.

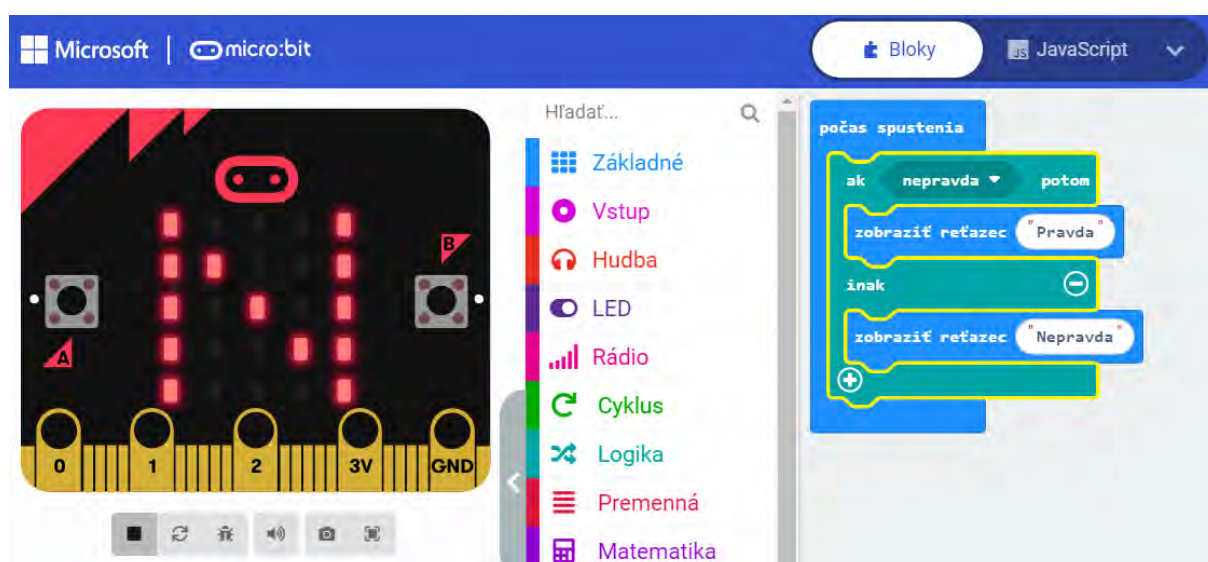
Príkaz "ak pravda potom inak"

Príkaz "ak pravda potom inak" slúži na vykonanie určitej akcie alebo bloku kódu, ak je zadaná podmienka pravdivá, a na vykonanie inej akcie, ak je podmienka nepravdivá. Tento príkaz umožňuje kontrolovať tok vykonávania programu na základe podmienky a zároveň poskytuje alternatívnu akciu pre prípad, že podmienka nie je splnená.

Ako to funguje?

1. Kontrola podmienky: Tento príkaz vyžaduje zadanie podmienky, ktorá bude vyhodnotená ako pravdivá alebo nepravdivá.
2. Vykonanie kódu: Ak je podmienka pravdivá, vykoná sa blok kódu uvedený v príkaze "ak pravda". Ak je podmienka nepravdivá, vykoná sa alternatívny blok kódu uvedený v príkaze "inak".

Príklad: Ak chceme zobrazit' text "Pravda" na LED matici, keď je podmienka true, a text "Nepravda" keď je podmienka false, použijeme nasledujúci blok kódu:



Obrázok 76 Príkaz "ak pravda potom inak"

Po spustení tohto kódu bude micro:bit zobrazovať text "Nepravda", pretože podmienka je nepravdivá. Ak by bola podmienka pravdivá, zobrazoval by text "Pravda".

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Použite príkaz "ak pravda potom inak" vo svojom programe.
3. Nastavte podmienku, akciu pre pravdivú podmienku, a alternatívnu akciu pre nepravdivú podmienku.
4. Spustite svoj program a sledujte, ako micro:bit vykonáva akcie podľa zadaných podmienok!

Tento príkaz je veľmi užitočný na kontrolu toku vykonávania programu a na rozhodovanie na základe podmienok, s alternatívnym riešením pre nepravdivé podmienky.

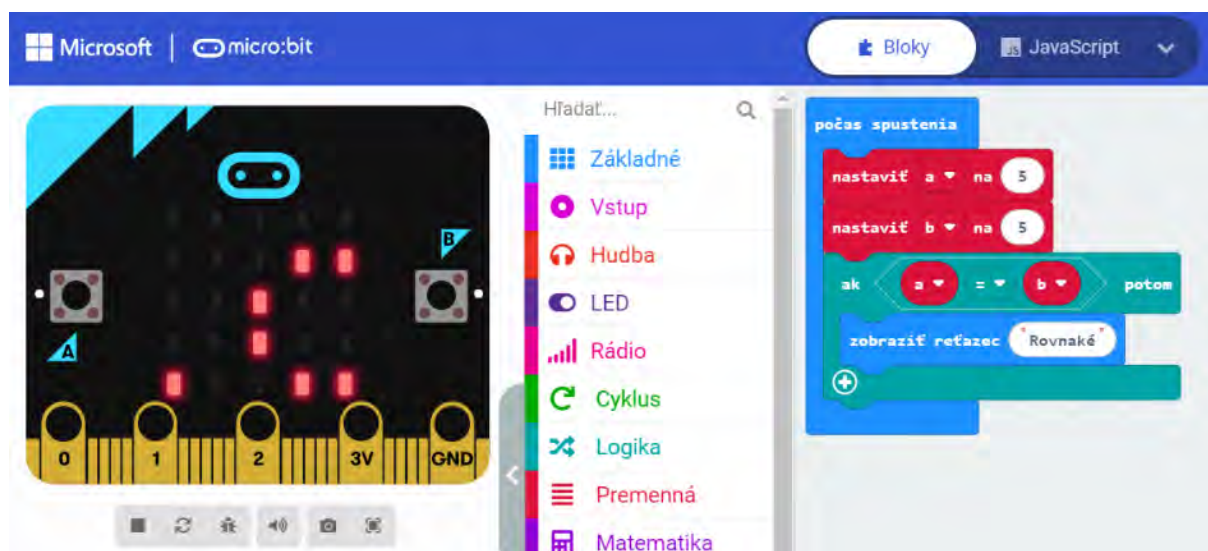
Príkaz "ak s porovnávacou logikou"

Príkaz "ak s porovnávacou logikou" slúži na vykonanie určitej akcie alebo bloku kódu, ak je zadaná podmienka pravdivá na základe porovnávackej logiky. Tento príkaz umožňuje kontrolovať tok vykonávania programu na základe výsledkov logického porovnania medzi dvoma alebo viacerými hodnotami.

Ako to funguje?

1. Porovnanie hodnôt: Tento príkaz vyžaduje zadanie porovnávackej logiky medzi hodnotami (napr. `==`, `!=`, `<`, `>`, `<=`, `>=`).
2. Kontrola podmienky: Po vyhodnotení porovnávackej logiky sa zistí, či je podmienka pravdivá alebo nepravdivá.
3. Vykonanie kódu: Ak je podmienka pravdivá, vykoná sa blok kódu uvedený v príkaze "ak".

Príklad: Ak chceme zobrazit' text "Rovnaké", keď sa hodnota premennej a rovná hodnote premennej b, použijeme nasledujúci blok kódu:



Obrázok 77 Príkaz "ak s porovnávacou logikou"

Po spustení tohto kódu bude micro:bit zobrazovať text "Rovnaké", pretože hodnoty a a b sú rovnaké.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Použite príkaz "ak s porovnávacou logikou" vo svojom programe.
3. Nastavte podmienku porovnávackej logiky a akciu, ktorú chcete vykonať, keď je podmienka pravdivá.
4. Spustíte svoj program a sledujte, ako micro:bit vykonáva akcie na základe porovnávackej logiky!

Tento príkaz je veľmi užitočný na kontrolu toku vykonávania programu a na rozhodovanie na základe logických porovnaní medzi hodnotami.

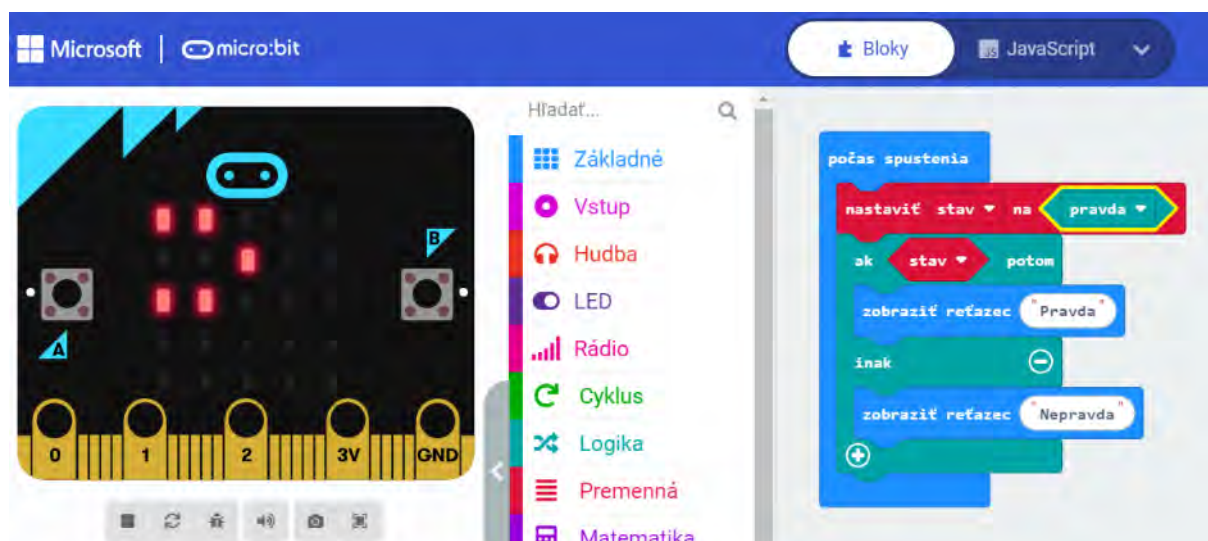
Príkaz "ak s logickými premennými"

Príkaz "ak s logickými premennými" slúži na vykonanie určitej akcie alebo bloku kódu na základe hodnoty logickej premennej. Logická premenná môže mať hodnotu true (pravda) alebo false (nepravda), a príkaz sa vykoná iba vtedy, keď je podmienka splnená.

Ako to funguje?

1. Kontrola logickej premennej: Tento príkaz vyžaduje logickú premennú, ktorá bude vyhodnotená ako pravdivá alebo nepravdivá.
2. Vykonanie kódu: Ak je logická premenná true, vykoná sa blok kódu uvedený v príkaze "ak".

Príklad: Ak chceme zobrazit' text "Pravda" na LED matici, keď je logická premenná stav true, použijeme nasledujúci blok kódu:



Obrázok 78 Príkaz "ak s logickými premennými"

Po spustení tohto kódu bude micro:bit zobrazovať text "Pravda", pretože premenná stav je true. Ak by premenná stav bola false, zobrazoval by text "Nepravda".

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Použite príkaz "ak s logickými premennými" vo svojom programe.
3. Nastavte logickú premennú a použite ju v podmienke príkazu "ak".
4. Spustíte svoj program a sledujte, ako micro:bit vykonáva akcie na základe hodnôt logických premenných!

Tento príkaz je veľmi užitočný na kontrolu toku vykonávania programu a na rozhodovanie na základe hodnôt logických premenných.

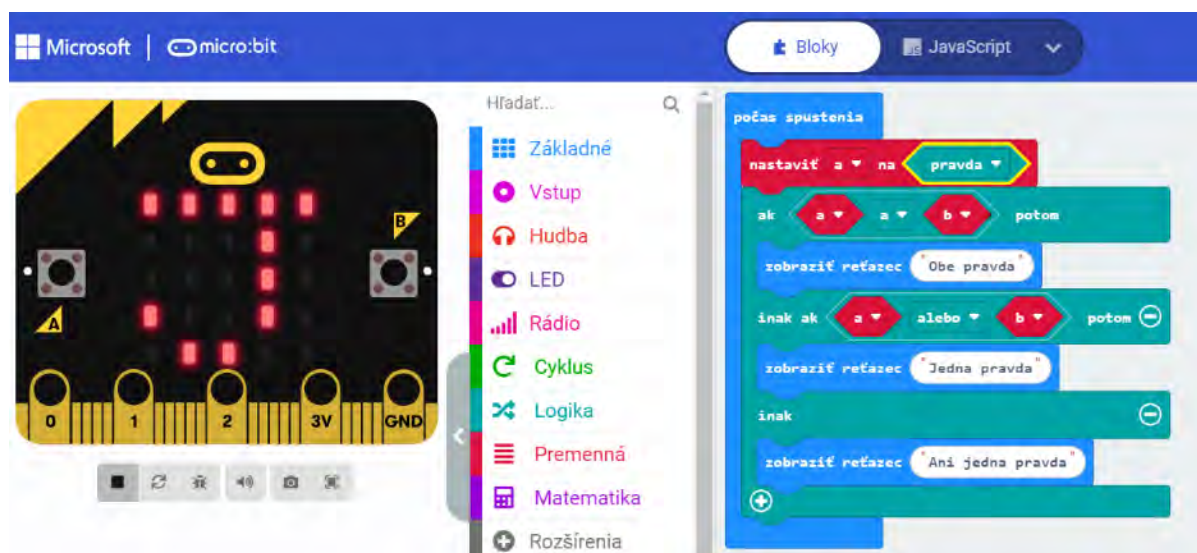
Príkaz "ak s logickými premennými a/alebo"

Príkaz "ak s logickými premennými a/alebo" slúži na vykonanie určitej akcie alebo bloku kódu na základe hodnoty viacerých logických premenných pomocou operátorov a (AND) a alebo (OR). Tento príkaz umožňuje kontrolovať tok vykonávania programu na základe kombinovaných logických podmienok.

Ako to funguje?

1. Kontrola logických premenných: Tento príkaz vyžaduje logické premenné, ktoré budú vyhodnotené ako pravdivé (true) alebo nepravdivé (false).
2. Použitie operátorov a a alebo: Operátor a vyžaduje, aby obe podmienky boli pravdivé, zatiaľ čo operátor alebo vyžaduje, aby aspoň jedna z podmienok bola pravdivá.
3. Vykonanie kódu: Ak sú podmienky pravdivé na základe kombinácie logických premenných a operátorov, vykoná sa blok kódu uvedený v príkaze "ak".

Príklad: Ak chceme zobraziť text "Pravda", keď sú obidve logické premenné a a b pravdivé, alebo aspoň jedna z nich je pravdivá, použijeme nasledujúci blok kódu:



Obrázok 79 Príkaz "ak s logickými premennými a/alebo"

Po spustení tohto kódu bude micro:bit zobrazovať text "Jedna pravda", pretože logická premenná **a** je true a **b** je false. Ak by boli obe pravdivé, zobrazoval by text "Obe pravda".

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Použite príkaz "ak s logickými premennými a/alebo" vo svojom programe.
3. Nastavte logické premenné a použite ich v podmienke príkazu "ak" s operátormi **a** a **alebo**.
4. Spustite svoj program a sledujte, ako micro:bit vykonáva akcie na základe kombinovaných logických podmienok!

Tento príkaz je veľmi užitočný na kontrolu toku vykonávania programu a na rozhodovanie na základe kombinovaných logických premenných.

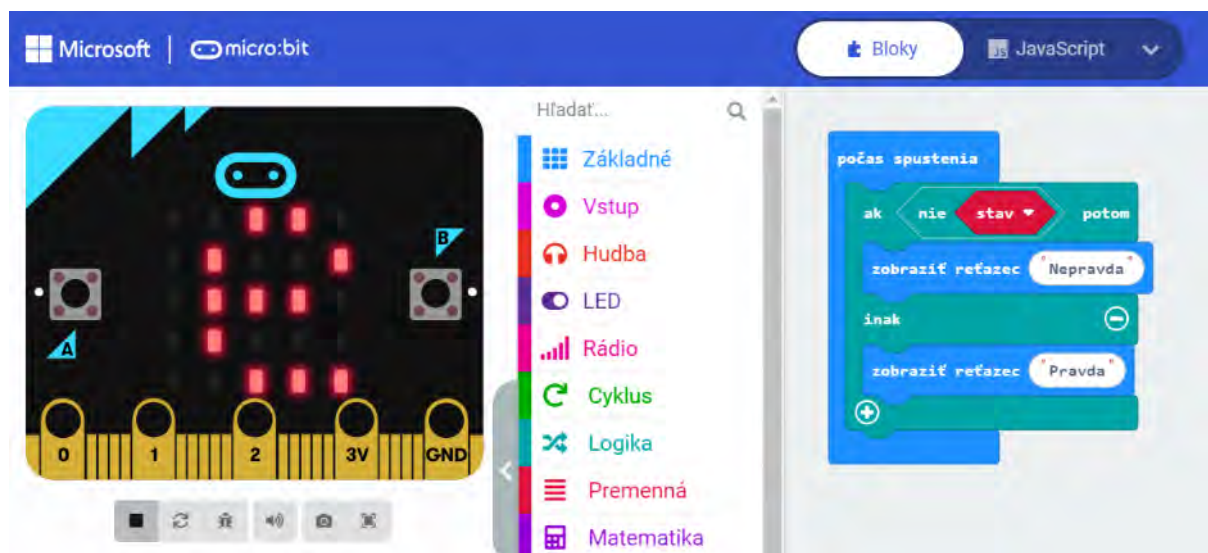
Použite príkaz "ak s logickou premennou nie"

Príkaz "ak s logickou premennou **nie**" slúži na vykonanie určitej akcie alebo bloku kódu, keď je hodnota logickej premennej nepravdivá (inverzná hodnota). Tento príkaz umožňuje kontrolovať tok vykonávania programu na základe negácie logických hodnôt.

Ako to funguje?

1. Negácia hodnoty: Tento príkaz použije logickú premennú s inverznou hodnotou nie, ktorá zmení hodnotu true na false a naopak.
2. Kontrola podmienky: Po negovaní sa hodnota vyhodnotí ako pravdivá alebo nepravdivá.
3. Vykonanie kódu: Ak je negovaná podmienka pravdivá, vykoná sa blok kódu uvedený v príkaze "ak".

Príklad: Ak chceme zobrazit' text "Nepravda", keď je logická premenná stav nepravdivá (teda false), použijeme nasledujúci blok kódu:



Obrázok 80 Použite príkaz "ak s logickou premennou nie"

Po spustení tohto kódu bude micro:bit zobrazovať text "Nepravda", pretože premenná stav je false. Ak by premenná stav bola true, zobrazoval by text "Pravda".

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Použite príkaz "ak s logickou premennou nie" vo svojom programe.
3. Nastavte logickú premennú a použite ju s negáciou (nie) v podmienke príkazu "ak".
4. Spustíte svoj program a sledujte, ako micro:bit vykonáva akcie na základe negovanej logickej hodnoty!

Tento príkaz je veľmi užitočný na kontrolu toku vykonávania programu a na rozhodovanie na základe inverzných logických premenných.

9 Premenné

Premenné v MakeCode sú ako schránky, do ktorých môžete uložiť rôzne typy hodnôt.

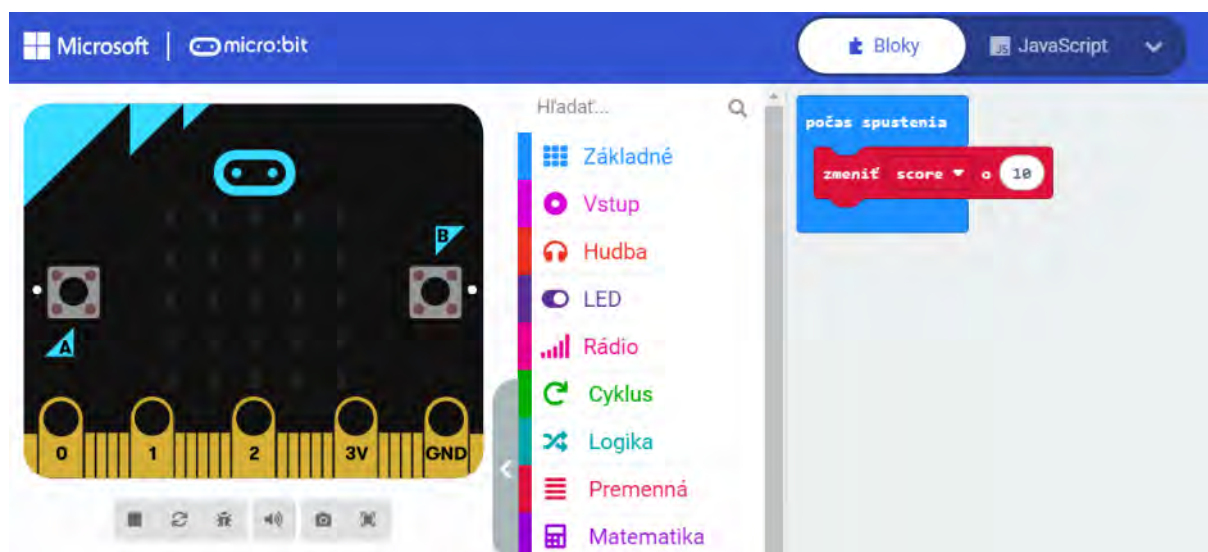
Existuje niekoľko typov premenných:

1. Číselné premenné: Ukladajú číselné hodnoty.
2. Textové premenné: Ukladajú textové reťazce, napríklad slová alebo vety.
3. Premenné typu boolean: Ukladajú hodnoty "pravda" alebo "nepravda".
4. Array (Pole): Ukladajú množiny hodnôt v jednej premennej.
5. Premenné typu enum: Definované zoznamy hodnôt.

Premenné vám umožňujú efektívne spravovať a manipulovať s údajmi vo vašom kóde. Je to ako mať organizovaný pracovný stôl, kde presne viete, kde je čo uložené.

Číselné premenné v MakeCode sú určené na uloženie číselných hodnôt, či už celých čísel alebo desatinných čísel. Môžu obsahovať hodnoty ako 7, -13, alebo 5.25. Tieto premenné sa často používajú na matematické výpočty, ukladanie výsledkov, počítanie a podobne.

Napríklad, ak chcete vytvoriť premennú pre ukladanie skóre v hre, použijete číselnú premennú. V kóde to môže vyzeráť takto:

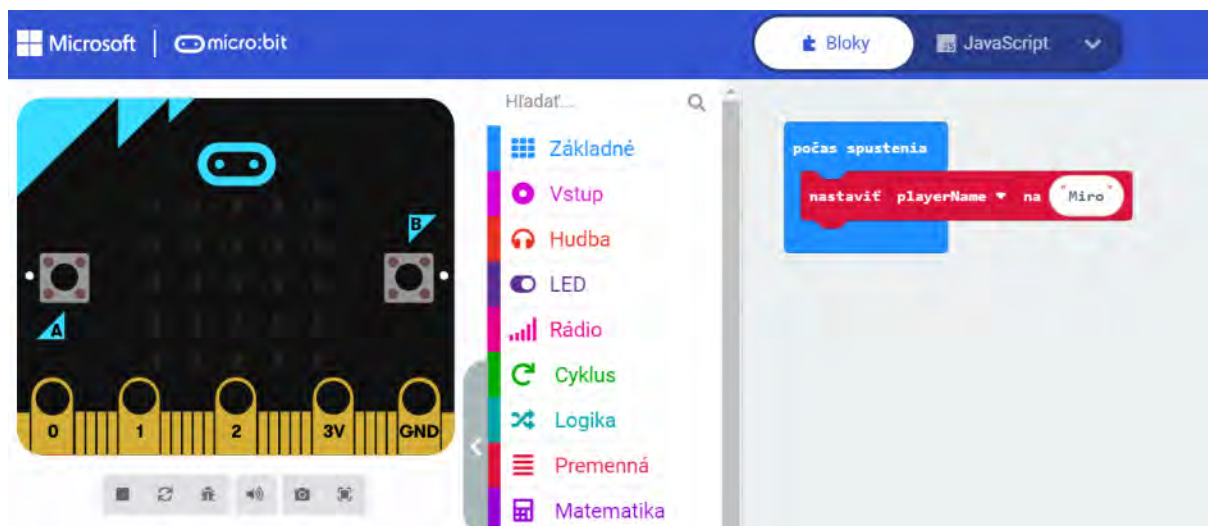


Obrázok 81 Číselné premenné

Premenné umožňujú sledovať a manipulovať hodnoty v rámci vášho programu efektívne a flexibilne.

Textové premenné v MakeCode slúžia na uloženie textových reťazcov. Môžu obsahovať slová, vety alebo aj celé odseky textu. Tieto premenné sú užitočné, keď potrebujete pracovať s textovými údajmi, napríklad pri zobrazovaní správ, ukladaní používateľských mien alebo spracovávaní vstupných údajov od používateľa.

Napríklad, ak chcete vytvoriť premennú pre ukladanie mena hráča, použijete textovú premennú. V kóde to môže vyzeráť takto:

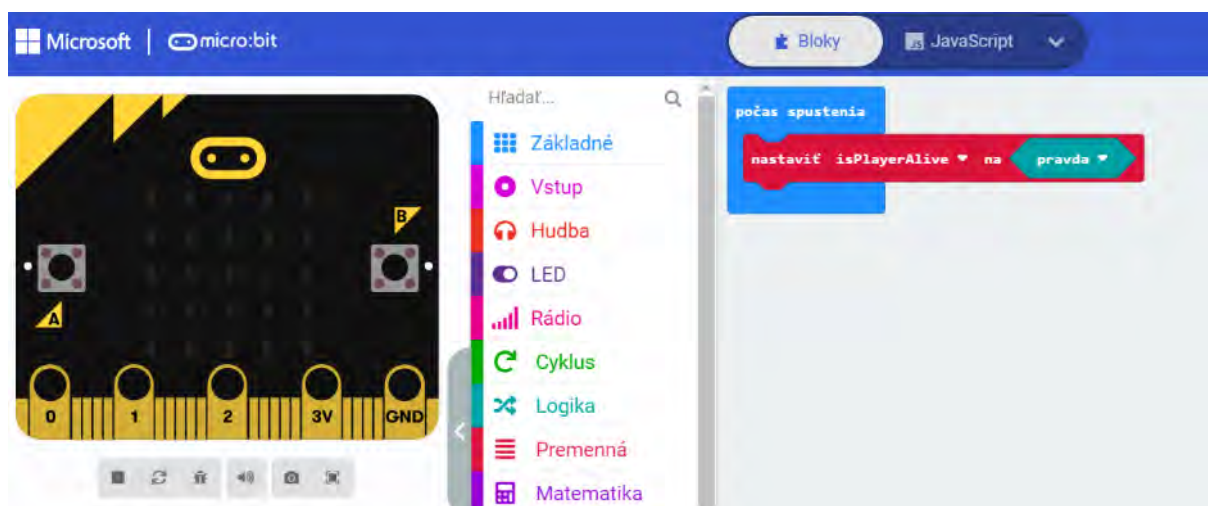


Obrázok 82 Textové premenné

Textové premenné umožňujú flexibilne pracovať s textovými údajmi a sú neoddeliteľnou súčasťou mnohých aplikácií a hier.

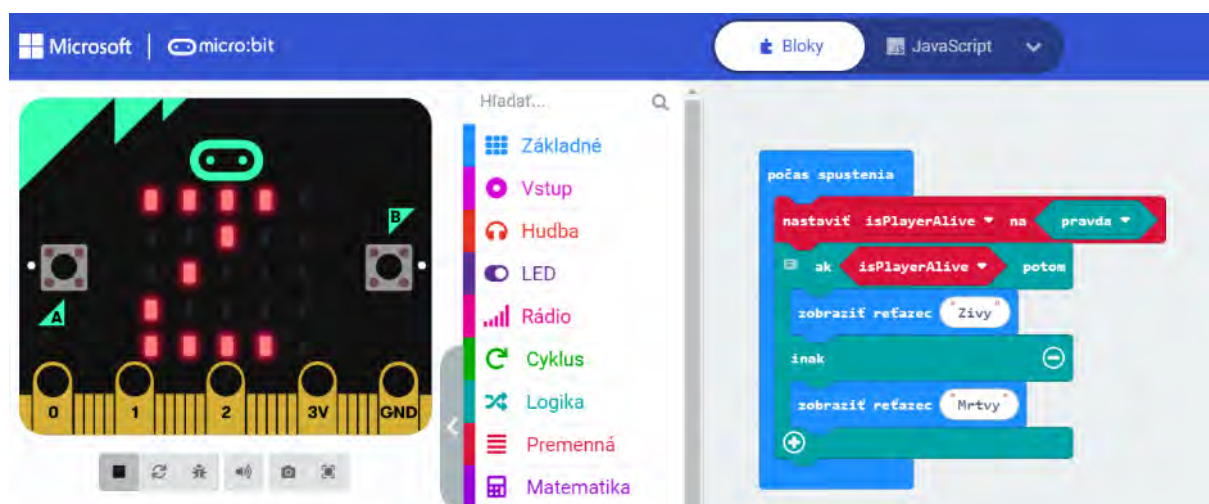
Premenné typu boolean v MakeCode slúžia na uloženie hodnôt pravda alebo nepravda, ktoré sú užitočné pri podmienkových rozhodnutiach v programe. Boolean premenné môžu obsahovať len dve hodnoty: true (pravda) alebo false (nepravda).

Napríklad, ak chcete sledovať, či je hráč v hre stále nažive, môžete použiť boolean premennú. V kóde to môže vyzeráť takto:



Obrázok 83 Premenné typu boolean

Potom môžete túto premennú použiť v podmienke:

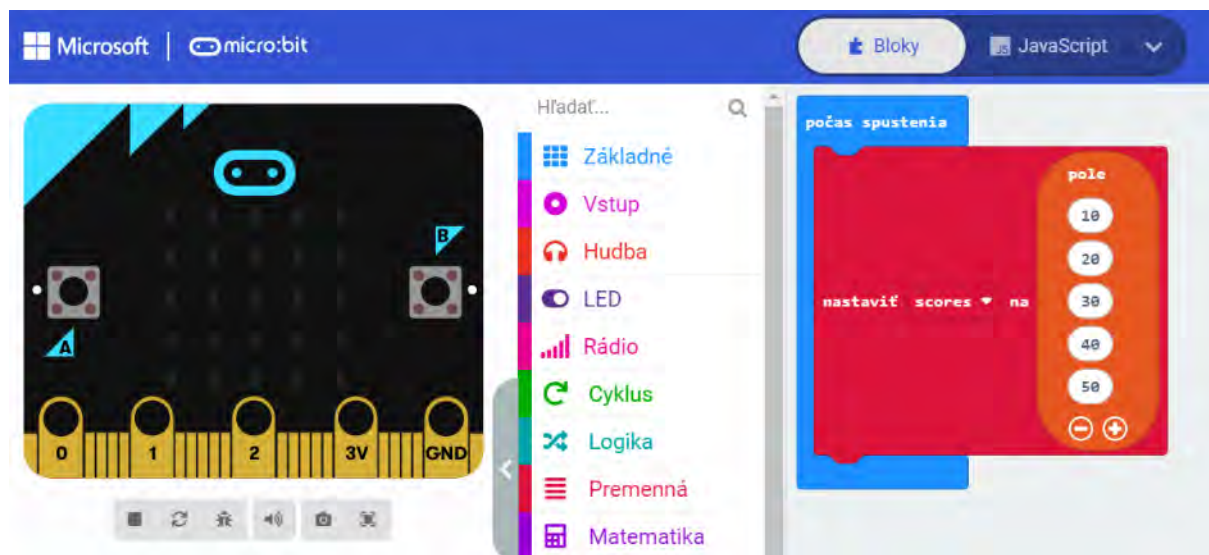


Obrázok 84 príklad premennej typu boolean

Boolean premenné sú neoddeliteľnou súčasťou logiky programu, keď potrebujete robiť rozhodnutia na základe jednoduchých áno/nie otázok.

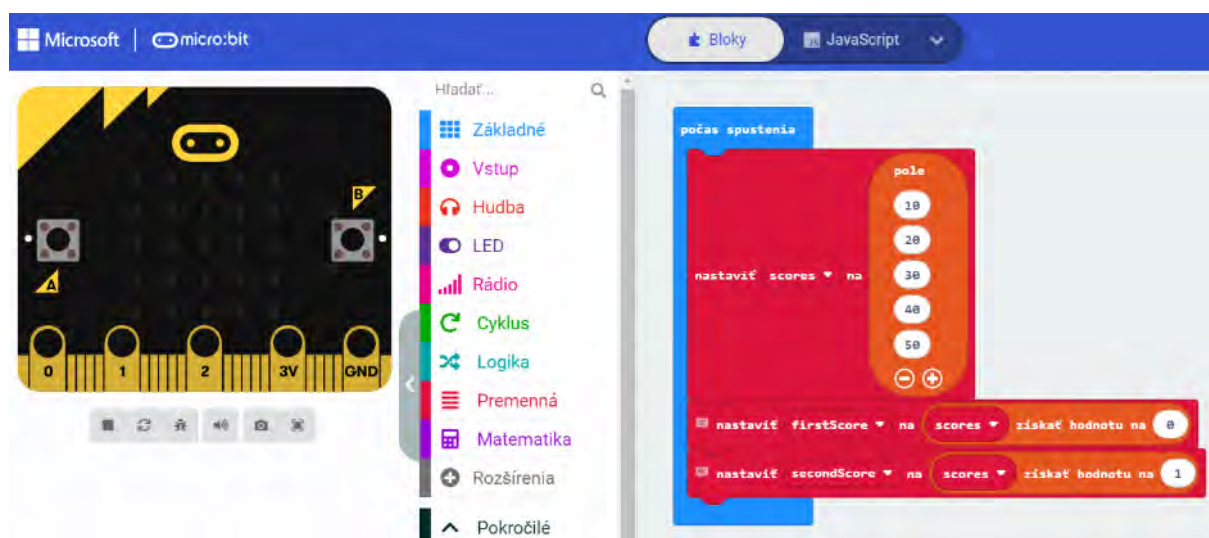
Pole (Array) v MakeCode sú premenné, ktoré dokážu uložiť množinu hodnôt pod jedným názvom. Hodnoty v poli môžu byť akéhokoľvek typu, ako čísla, texty, boolean hodnoty, alebo dokonca ďalšie polia. To je užitočné, keď potrebujete spravovať viacero hodnôt súčasne a prístup k nim podľa indexu.

Napríklad, ak chcete uložiť skóre piatich hráčov, použijete pole. V kóde to môže vyzeráť takto:



Obrázok 85 Pole (Array)

Prístup k jednotlivým hodnotám v poli je pomocou indexu:

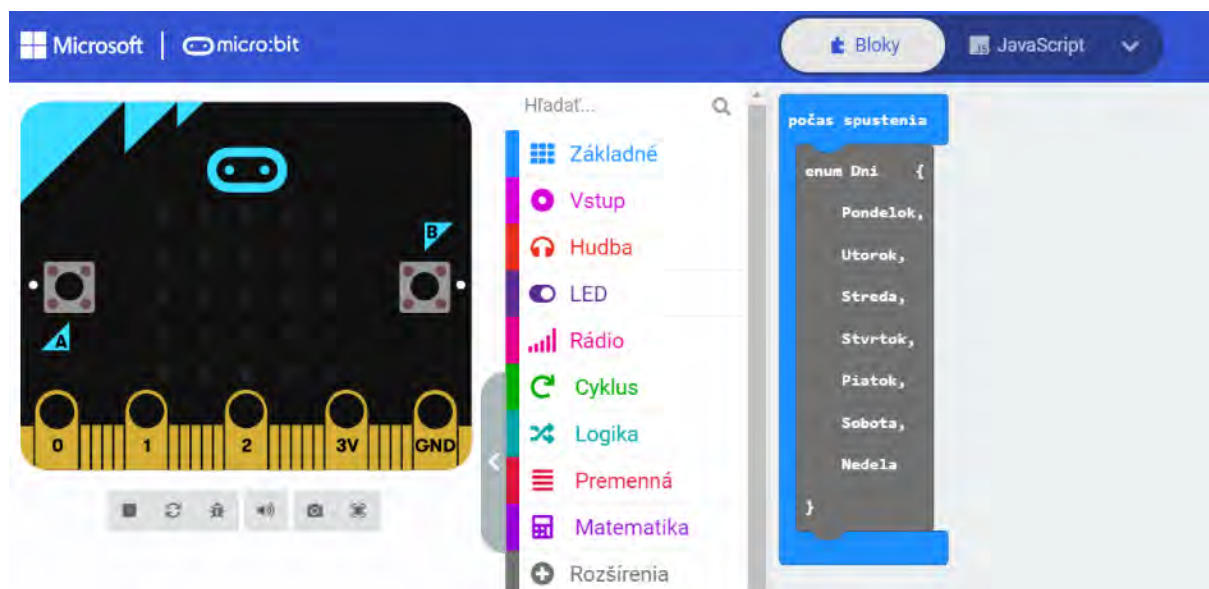


Obrázok 86 Príklad prístupu k jednotlivým hodnotám v poli

Polia umožňujú efektívnu prácu s veľkým množstvom údajov a sú neoddeliteľnou súčasťou mnohých programovacích úloh.

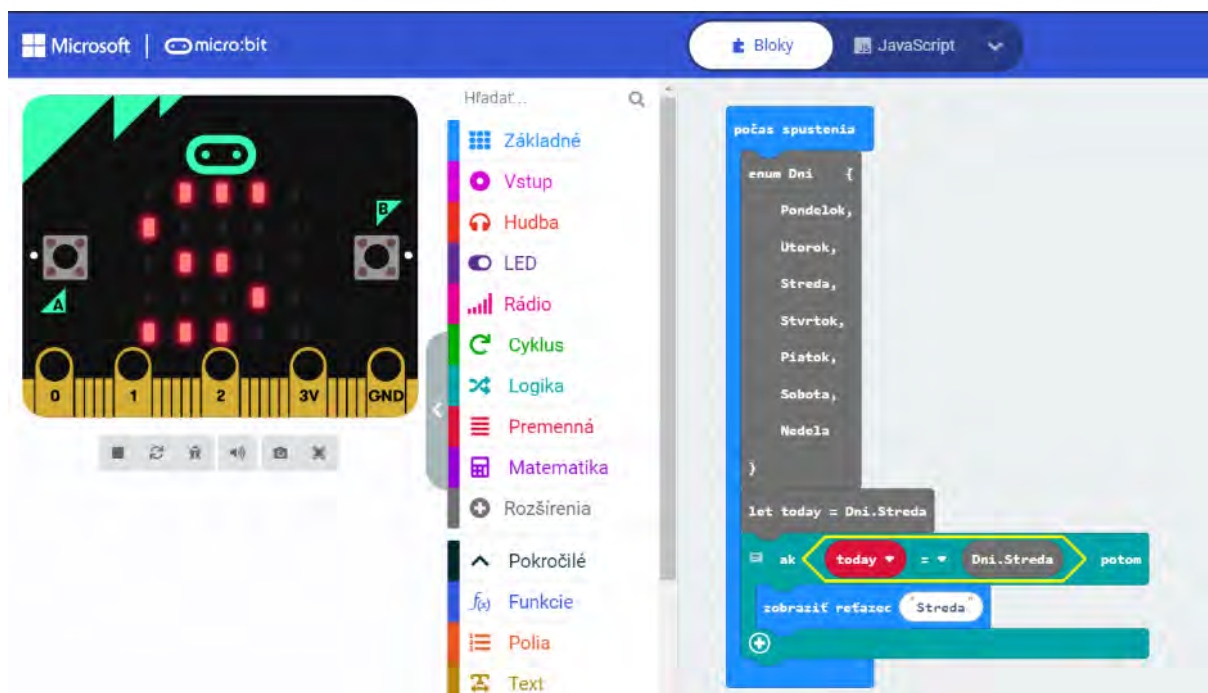
Premenné typu enum v MakeCode slúžia na definovanie zoznamu hodnôt, ktoré môžu byť použité v programe pod jedným názvom. Enum premenné sú užitočné pri práci s údajmi, ktoré majú obmedzený počet možných hodnôt.

Napríklad, ak chcete definovať dni v týždni, použijete **enum premennú**. V kóde to môže vyzeráť takto:



Obrázok 87 Premenné typu enum

Potom môžete použiť tieto hodnoty v programe:



Obrázok 88 Příklad premennej typu enum

Enum premenné vám pomáhajú sprístupniť a pracovať so zoznamami, kde sú hodnoty preddefinované, čo zvyšuje čitateľnosť a spoľahlivosť kódu.

10 Matematika

Funkcia sčítania

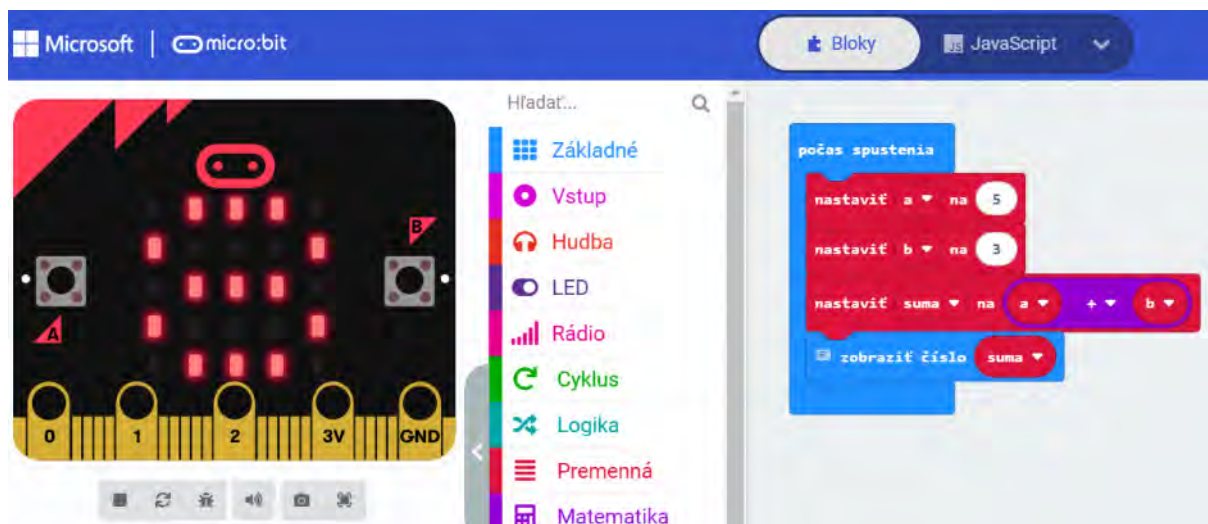
Funkcia sčítania v MakeCode slúži na vykonanie aritmetickej operácie sčítania medzi dvoma číslami. Táto operácia je veľmi užitočná na rôzne výpočty a je často využívaná v rôznych projektoch s BBC micro:bitom.

Ako to funguje?

1. Definovanie premenných: Najprv definujeme dve premenné, ktorých hodnoty chceme sčítať.
2. Vykonanie sčítania: Potom použijeme príkaz na sčítanie týchto dvoch premenných a uloženie výsledku do novej premennej alebo priamo zobrazenie výsledku.

Príklad:

Tu je jednoduchý príklad, ako použiť sčítanie v MakeCode:



Obrázok 89 Funkcia sčítania

V tomto príklade:

1. Definujeme premennú a s hodnotou 5.
2. Definujeme premennú b s hodnotou 3.
3. Sčítame tieto dve premenné a výsledok uložíme do premennej suma.
4. Na LED matici micro:bitu zobrazíme hodnotu premennej suma, čo je 8.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Vytvorte nový projekt a vložte doň uvedený blok kódu.
3. Nahrajte program do micro:bitu a pozorujte, ako sa na LED matici zobrazí výsledok sčítania.

Táto funkcia je veľmi užitočná pri rôznych výpočtoch a môže byť súčasťou komplexnejších programov.

Funkcia odčítania

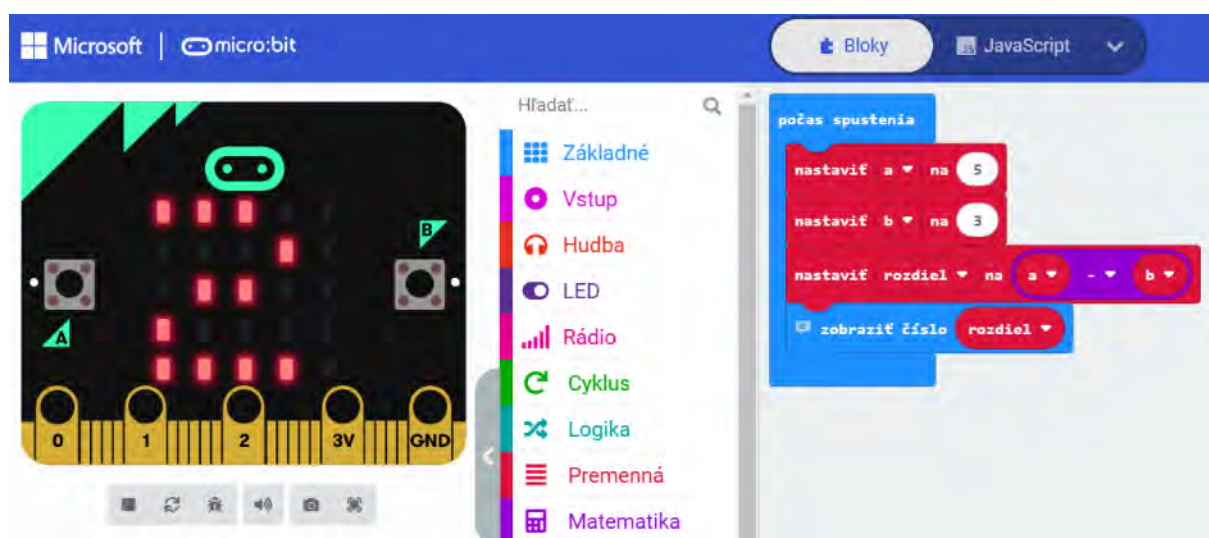
Funkcia odčítania v MakeCode slúži na vykonanie aritmetickej operácie odčítania medzi dvoma číslami. Táto operácia je veľmi užitočná na rôzne výpočty a je často využívaná v rôznych projektoch s micro:bitom.

Ako to funguje?

1. Definovanie premenných: Najprv definujeme dve premenné, ktorých hodnoty chceme odčítať.
2. Vykonanie odčítania: Potom použijeme príkaz na odčítanie týchto dvoch premenných a uloženie výsledku do novej premennej alebo priamo zobrazenie výsledku.

Príklad:

Tu je jednoduchý príklad, ako použiť odčítanie v MakeCode:



Obrázok 90 Funkcia odčítania

V tomto príklade:

1. Definujeme premennú a s hodnotou 5.
2. Definujeme premennú b s hodnotou 3.
3. Odčítame premennú b od premennej a a výsledok uložíme do premennej rozdiel.
4. Na LED matici micro:bitu zobrazíme hodnotu premennej rozdiel, čo je 2.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Vytvorte nový projekt a vložte doň uvedený blok kódu.
3. Nahrajte program do micro:bitu a pozorujte, ako sa na LED matici zobrazí výsledok odčítania.

Táto funkcia je veľmi užitočná pri rôznych výpočtoch a môže byť súčasťou komplexnejších programov.

Funkcia násobenia

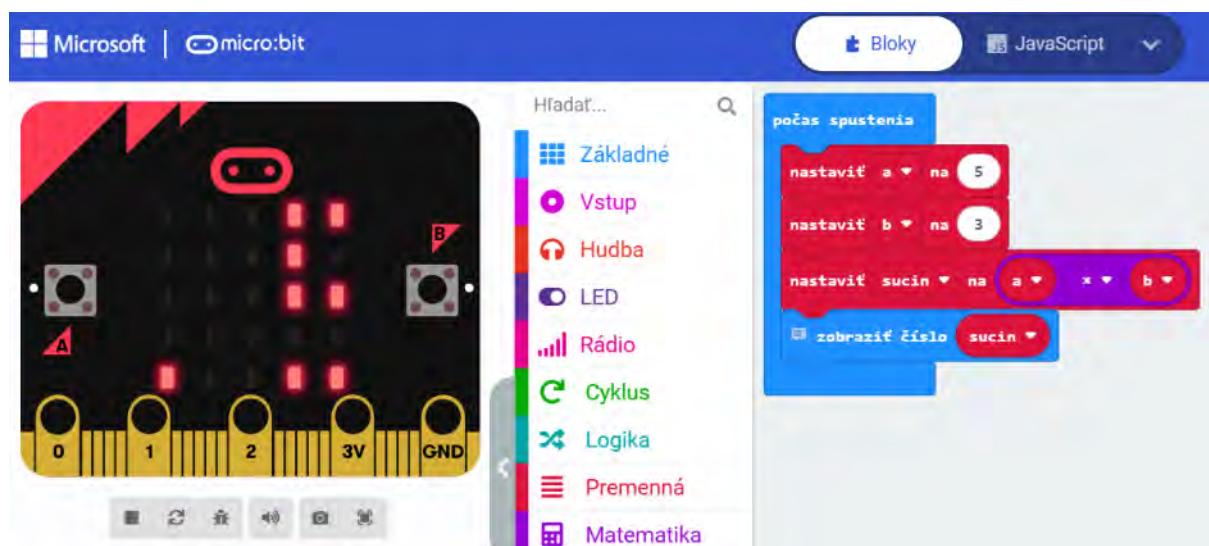
Funkcia násobenia v MakeCode slúži na vykonanie aritmetickej operácie násobenia medzi dvoma číslami. Táto operácia je veľmi užitočná na rôzne výpočty a je často využívaná v rôznych projektoch s micro:bitom.

Ako to funguje?

1. Definovanie premenných: Najprv definujeme dve premenné, ktorých hodnoty chceme násobiť.
2. Vykonanie násobenia: Potom použijeme príkaz na násobenie týchto dvoch premenných a uloženie výsledku do novej premennej alebo priamo zobrazenie výsledku.

Príklad:

Tu je jednoduchý príklad, ako použiť násobenie v MakeCode:



Obrázok 91 Funkcia násobenia

V tomto príklade:

1. Definujeme premennú a s hodnotou 5.
2. Definujeme premennú b s hodnotou 3.
3. Násobíme tieto dve premenné a výsledok uložíme do premennej sucin.
4. Na LED matici micro:bitu zobrazíme hodnotu premennej sucin, čo je 15.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Vytvorte nový projekt a vložte doň uvedený blok kódu.
3. Nahrajte program do micro:bitu a pozorujte, ako sa na LED matici zobrazí výsledok násobenia.

Táto funkcia je veľmi užitočná pri rôznych výpočtoch a môže byť súčasťou komplexnejších programov.

Funkcia delenia

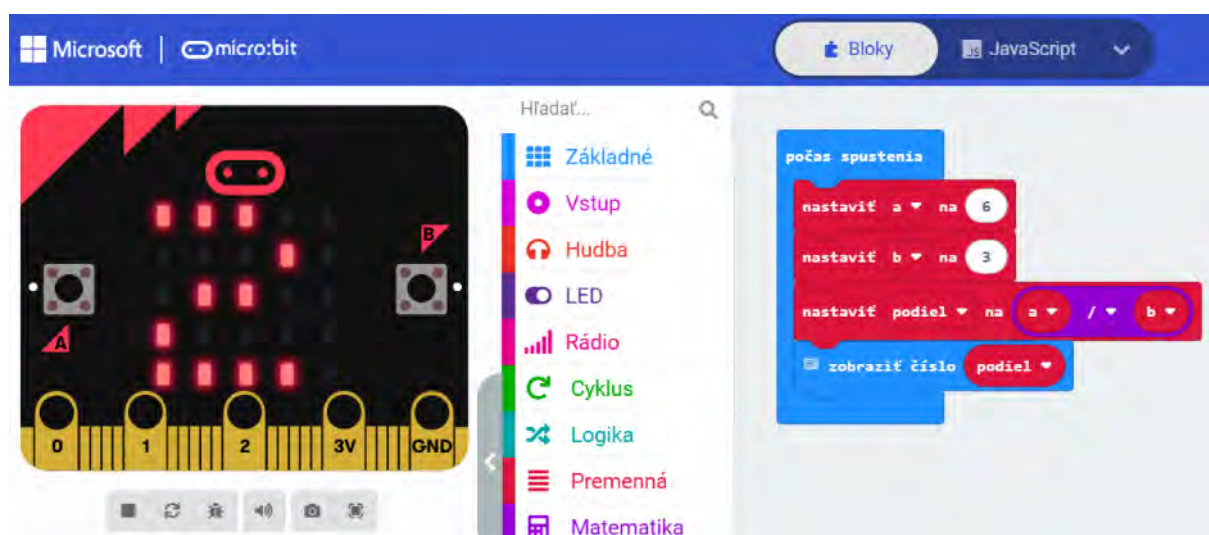
Funkcia delenia v MakeCode slúži na vykonanie aritmetickej operácie delenia medzi dvoma číslami. Táto operácia je veľmi užitočná na rôzne výpočty a je často využívaná v rôznych projektoch s micro:bitom.

Ako to funguje?

1. Definovanie premenných: Najprv definujeme dve premenné, ktorých hodnoty chceme deliť.
2. Vykonanie delenia: Potom použijeme príkaz na delenie týchto dvoch premenných a uloženie výsledku do novej premennej alebo priamo zobrazenie výsledku.

Príklad:

Tu je jednoduchý príklad, ako použiť delenie v MakeCode:



Obrázok 92 Funkcia delenia

V tomto príklade:

1. Definujeme premennú a s hodnotou 6.
2. Definujeme premennú b s hodnotou 3.
3. Delíme premennú a premennou b a výsledok uložíme do premennej podiel.
4. Na LED matici micro:bitu zobrazíme hodnotu premennej podiel, čo je 2.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Vytvorte nový projekt a vložte doň uvedený blok kódu.
3. Nahrajte program do micro:bitu a pozorujte, ako sa na LED matici zobrazí výsledok delenia.

Táto funkcia je veľmi užitočná pri rôznych výpočtoch a môže byť súčasťou komplexnejších programov.

Funkcia zvyšku po delení (remainder of)

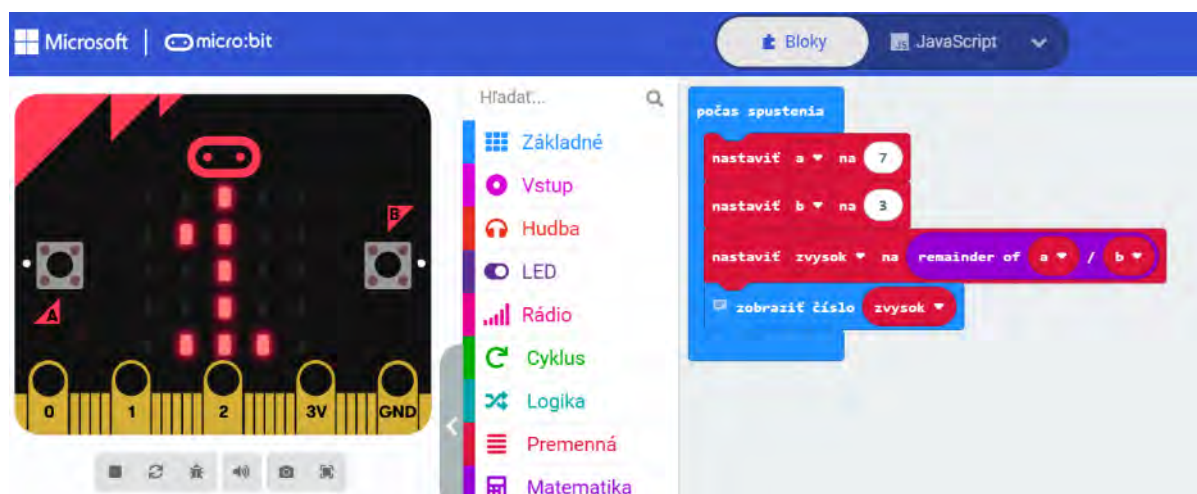
Funkcia zvyšku po delení v MakeCode slúži na vykonanie aritmetickej operácie zistenia zvyšku po delení medzi dvoma číslami. Táto operácia je veľmi užitočná na rôzne výpočty a je často využívaná v rôznych projektoch s micro:bitom.

Ako to funguje?

1. Definovanie premenných: Najprv definujeme dve premenné, ktorých hodnoty chceme deliť a zistiť zvyšok.
2. Vykonanie operácie: Potom použijeme príkaz na zistenie zvyšku po delení týchto dvoch premenných a uloženie výsledku do novej premennej alebo priamo zobrazenie výsledku.

Príklad:

Tu je jednoduchý príklad, ako použiť funkciu zvyšku po delení v MakeCode:



Obrázok 93 Funkcia zvyšku po delení (remainder of)

V tomto príklade:

1. Definujeme premennú a s hodnotou 7.
2. Definujeme premennú b s hodnotou 3.
3. Zistíme zvyšok po delení premennej a premennou b a výsledok uložíme do premennej zvysok.
4. Na LED matici micro:bitu zobrazíme hodnotu premennej zvysok, čo je 1.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Vytvorte nový projekt a vložte doň uvedený blok kódu.
3. Nahrajte program do micro:bitu a pozorujte, ako sa na LED matici zobrazí výsledok zvyšku po delení.

Táto funkcia je veľmi užitočná pri rôznych výpočtoch a môže byť súčasťou komplexnejších programov.

Funkcia minimálnej hodnoty (min)

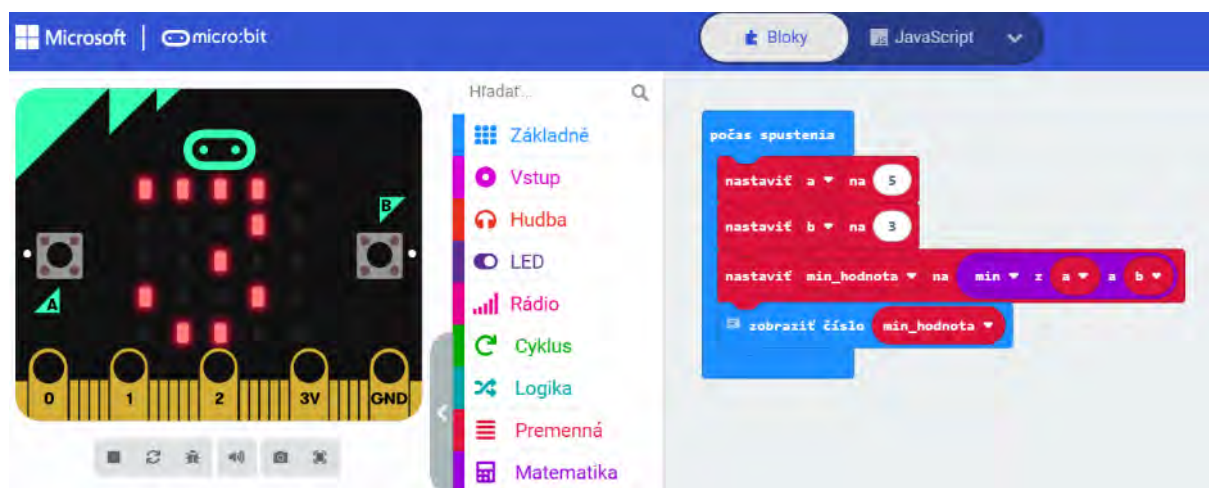
Funkcia minimálnej hodnoty (min) v MakeCode slúži na zistenie menšej z dvoch hodnôt. Táto operácia je veľmi užitočná pri rôznych výpočtoch a porovnávaniach a je často využívaná v rôznych projektoch s micro:bitom.

Ako to funguje?

1. Definovanie premenných: Najprv definujeme dve premenné, ktorých hodnoty chceme porovnať.
2. Zistenie minimálnej hodnoty: Potom použijeme príkaz na zistenie menšej z týchto dvoch premenných a uloženie výsledku do novej premennej alebo priamo zobrazenie výsledku.

Príklad:

Tu je jednoduchý príklad, ako použiť funkciu minimálnej hodnoty v MakeCode:



Obrázok 94 Funkcia minimálnej hodnoty (min)

V tomto príklade:

1. Definujeme premennú a s hodnotou 5.
2. Definujeme premennú b s hodnotou 3.
3. Pomocou funkcie Math.min zistíme menšiu hodnotu medzi premennými a a b a výsledok uložíme do premennej min_hodnota.
4. Na LED matici micro:bitu zobrazíme hodnotu premennej min_hodnota, čo je 3.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Vytvorte nový projekt a vložte doň uvedený blok kódu.
3. Nahrajte program do micro:bitu a pozorujte, ako sa na LED matici zobrazí minimálna hodnota z dvoch zadáných hodnôt.

Táto funkcia je veľmi užitočná pri rôznych výpočtoch a porovnávaniach a môže byť súčasťou komplexnejších programov.

Funkcia maximálnej hodnoty (max)

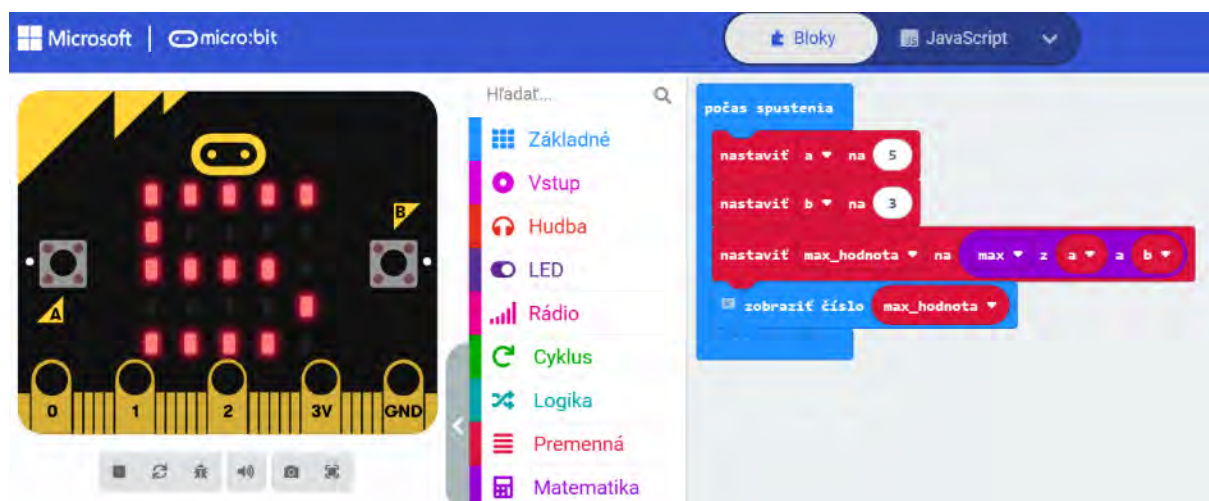
Funkcia maximálnej hodnoty (max) v MakeCode slúži na zistenie väčšej z dvoch hodnôt. Táto operácia je veľmi užitočná pri rôznych výpočtoch a porovnávaníach a je často využívaná v rôznych projektoch s micro:bitom.

Ako to funguje?

1. Definovanie premenných: Najprv definujeme dve premenné, ktorých hodnoty chceme porovnať.
2. Zistenie maximálnej hodnoty: Potom použijeme príkaz na zistenie väčšej z týchto dvoch premenných a uloženie výsledku do novej premennej alebo priamo zobrazenie výsledku.

Príklad:

Tu je jednoduchý príklad, ako použiť funkciu maximálnej hodnoty v MakeCode:



Obrázok 95 Funkcia maximálnej hodnoty (max)

V tomto príklade:

1. Definujeme premennú a s hodnotou 5.
2. Definujeme premennú b s hodnotou 3.
3. Pomocou funkcie Math.max zistíme väčšiu hodnotu medzi premennými **a** a **b** a výsledok uložíme do premennej max_hodnota.
4. Na LED matici micro:bitu zobrazíme hodnotu premennej max_hodnota, čo je 5.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Vytvorte nový projekt a vložte doň uvedený blok kódu.
3. Nahrajte program do micro:bitu a pozorujte, ako sa na LED matici zobrazí maximálna hodnota z dvoch zadaných hodnôt.

Táto funkcia je veľmi užitočná pri rôznych výpočtoch a porovnávaníach a môže byť súčasťou komplexnejších programov.

Funkcia absolútnej hodnoty

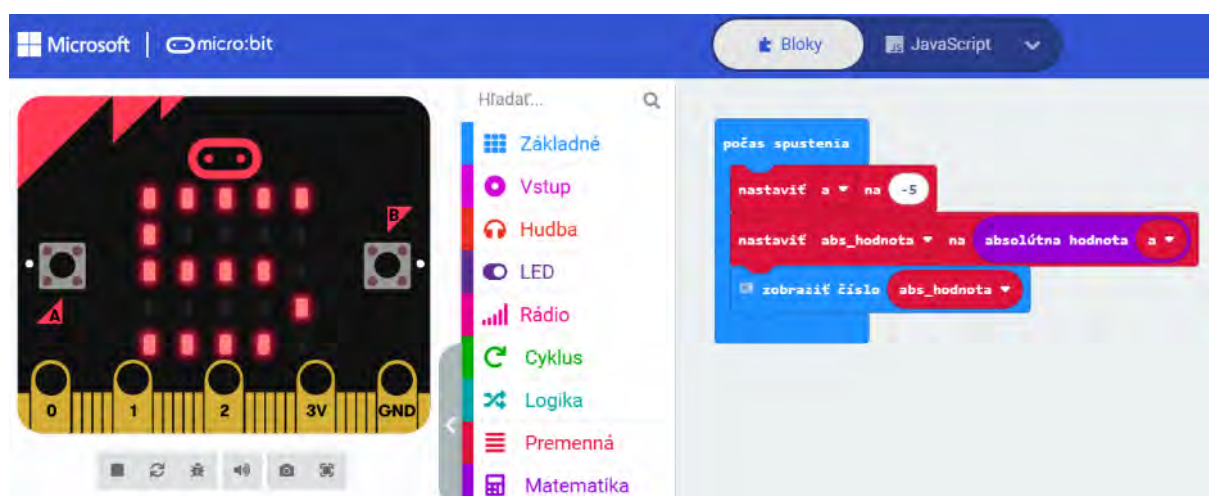
Funkcia absolútnej hodnoty v MakeCode slúži na zistenie absolútnej (kladnej) hodnoty zadaného čísla. Táto operácia je veľmi užitočná pri rôznych výpočtoch, kde je potrebné získať nezápornú hodnotu čísla, a je často využívaná v rôznych projektoch s micro:bitom.

Ako to funguje?

1. Definovanie premennej: Najprv definujeme premennú, ktorej absolútnu hodnotu chceme zistiť.
2. Vykonanie operácie: Potom použijeme príkaz na zistenie absolútnej hodnoty tejto premennej a uloženie výsledku do novej premennej alebo priamo zobrazenie výsledku.

Príklad:

Tu je jednoduchý príklad, ako použiť funkciu absolútnej hodnoty v MakeCode:



Obrázok 96 Funkcia absolútnej hodnoty

V tomto príklade:

1. Definujeme premennú `a` s hodnotou `-5`.
2. Pomocou funkcie `Math.abs` zistíme absolútnu hodnotu premennej `a` a výsledok uložíme do premennej `abs_hodnota`.
3. Na LED matici micro:bitu zobrazíme hodnotu premennej `abs_hodnota`, čo je `5`.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Vytvorte nový projekt a vložte doň uvedený blok kódu.
3. Nahrajte program do micro:bitu a pozorujte, ako sa na LED matici zobrazí absolútna hodnota zadaného čísla.

Táto funkcia je veľmi užitočná pri rôznych výpočtoch a môže byť súčasťou komplexnejších programov.

Funkcia druhej odmocniny

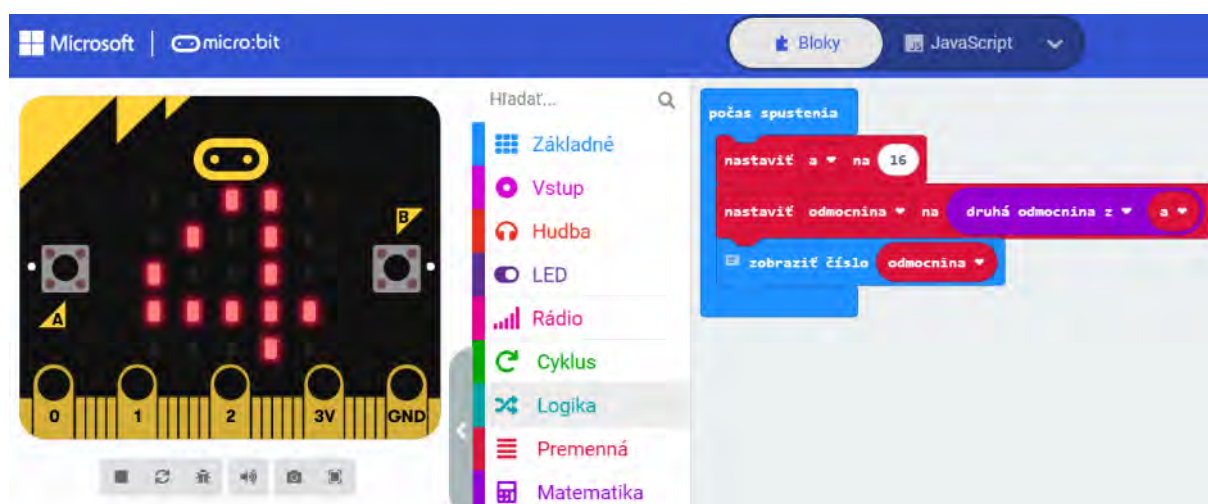
Funkcia druhej odmocniny v MakeCode slúži na zistenie druhej odmocniny zadaného čísla. Táto operácia je veľmi užitočná pri rôznych výpočtoch a je často využívaná v rôznych projektoch s micro:bitom.

Ako to funguje?

1. Definovanie premennej: Najprv definujeme premennú, ktorej druhú odmocninu chceme zistiť.
2. Vykonanie operácie: Potom použijeme príkaz na zistenie druhej odmocniny tejto premennej a uloženie výsledku do novej premennej alebo priamo zobrazenie výsledku.

Príklad:

Tu je jednoduchý príklad, ako použiť funkciu druhej odmocniny v MakeCode:



Obrázok 97 Funkcia druhej odmocniny

V tomto príklade:

1. Definujeme premennú *a* s hodnotou 16.
2. Pomocou funkcie `Math.sqrt` zistíme druhú odmocninu premennej *a* a výsledok uložíme do premennej *odmocnina*.
3. Na LED matici micro:bitu zobrazíme hodnotu premennej *odmocnina*, čo je 4.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Vytvorte nový projekt a vložte doň uvedený blok kódu.
3. Nahrajte program do micro:bitu a pozorujte, ako sa na LED matici zobrazí druhá odmocnina zadaného čísla.

Táto funkcia je veľmi užitočná pri rôznych výpočtoch a môže byť súčasťou komplexnejších programov.

Funkcia zaokrúhľovania

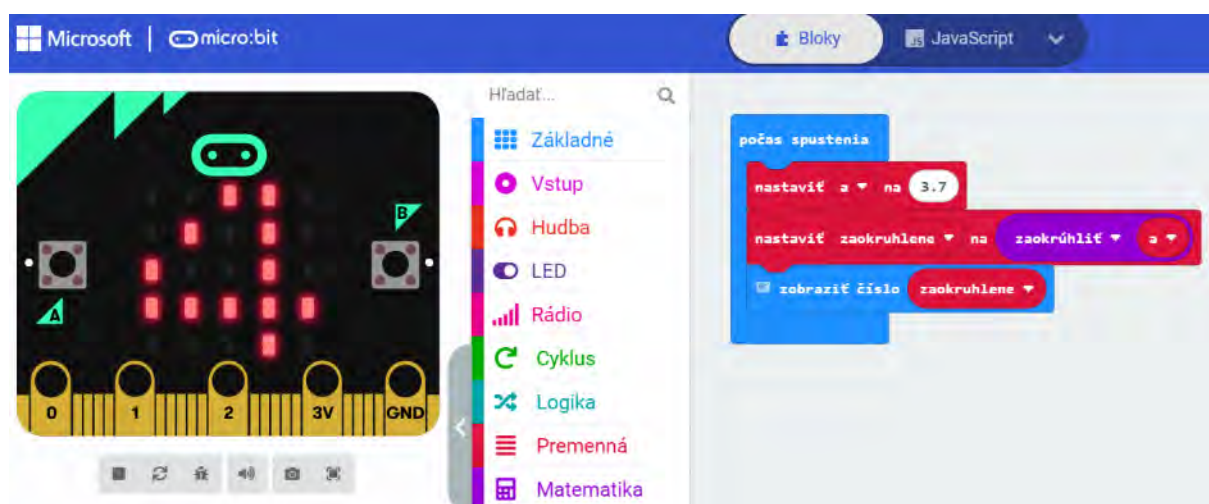
Funkcia zaokrúhľovania v MakeCode slúži na zaokrúhlenie zadaného čísla na najbližšie celé číslo. Táto operácia je veľmi užitočná pri rôznych výpočtoch a je často využívaná v rôznych projektoch s micro:bitom.

Ako to funguje?

1. Definovanie premennej: Najprv definujeme premennú, ktorej hodnotu chceme zaokrúhliť.
2. Vykonanie operácie: Potom použijeme príkaz na zaokrúhlenie tejto premennej a uloženie výsledku do novej premennej alebo priamo zobrazenie výsledku.

Príklad:

Tu je jednoduchý príklad, ako použiť funkciu zaokrúhľovania v MakeCode:



Obrázok 98 Funkcia zaokrúhľovania

V tomto príklade:

1. Definujeme premennú a s hodnotou 3.7.
2. Pomocou funkcie Math.round zaokrúhlime hodnotu premennej a a výsledok uložíme do premennej zaokruhlene.
3. Na LED matici micro:bitu zobrazíme hodnotu premennej zaokruhlene, čo je 4.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Vytvorte nový projekt a vložte doň uvedený blok kódu.
3. Nahrajte program do micro:bitu a pozorujte, ako sa na LED matici zobrazí zaokrúhlená hodnota zadaného čísla.

Táto funkcia je veľmi užitočná pri rôznych výpočtoch a môže byť súčasťou komplexnejších programov.

Funkcia výberu náhodného čísla

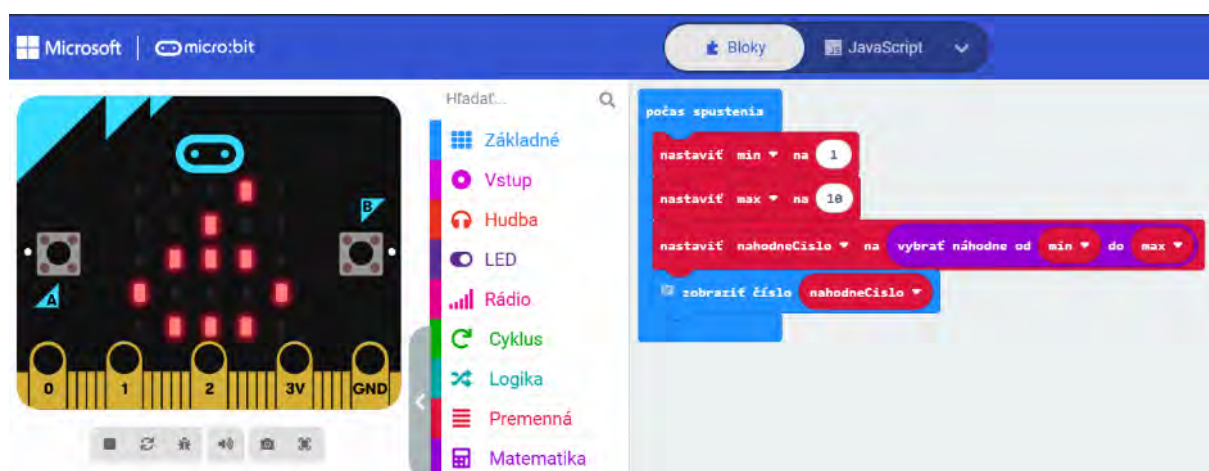
Funkcia výberu náhodného čísla v MakeCode slúži na generovanie náhodného čísla v zadanom rozsahu. Táto operácia je veľmi užitočná pri rôznych výpočtoch a hrách a je často využívaná v rôznych projektoch s micro:bitom.

Ako to funguje?

1. Definovanie rozsahu: Najprv definujeme rozsah, z ktorého chceme vybrať náhodné číslo.
2. Vykonanie operácie: Potom použijeme príkaz na výber náhodného čísla v tomto rozsahu a uloženie výsledku do premennej alebo priamo zobrazenie výsledku.

Príklad:

Tu je jednoduchý príklad, ako použiť funkciu výberu náhodného čísla v MakeCode:



Obrázok 99 Funkcia výberu náhodného čísla

V tomto príklade:

1. Definujeme premennú min s hodnotou 1.
2. Definujeme premennú max s hodnotou 10.
3. Pomocou funkcie Math.randomRange generujeme náhodné číslo medzi hodnotami min a max a výsledok uložíme do premennej nahodneCislo.
4. Na LED matici micro:bitu zobrazíme hodnotu premennej nahodneCislo, čo bude náhodné číslo v rozsahu od 1 do 10.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Vytvorte nový projekt a vložte doň uvedený blok kódu.
3. Nahrajte program do micro:bitu a pozorujte, ako sa na LED matici zobrazí náhodne vybrané číslo.

Táto funkcia je veľmi užitočná pri rôznych výpočtoch a hrách a môže byť súčasťou komplexnejších programov.

Funkcia obmedzenia hodnoty na interval

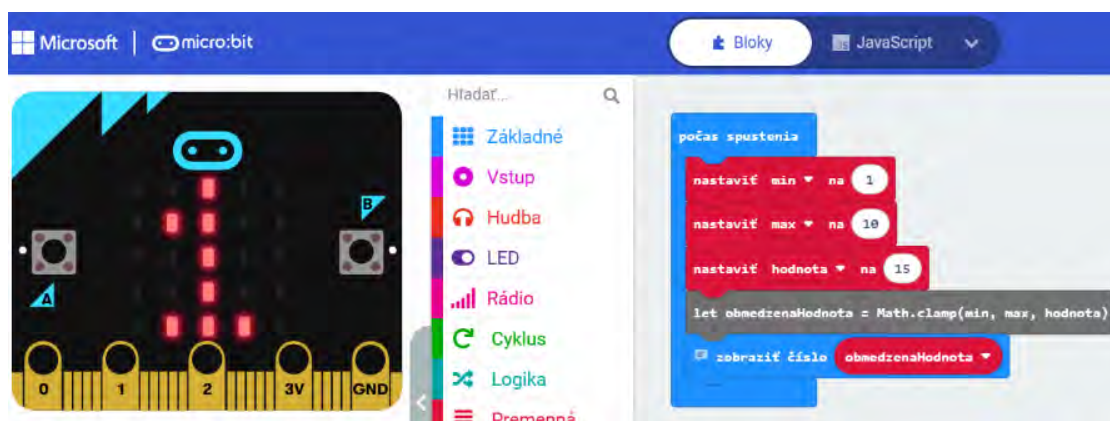
Funkcia obmedzenia hodnoty na interval v MakeCode slúži na obmedzenie zadaného čísla do zadaného rozsahu. Táto operácia je veľmi užitočná pri rôznych výpočtoch a kontrolách hodnôt a je často využívaná v rôznych projektoch s micro:bitom.

Ako to funguje?

1. Definovanie rozsahu: Najprv definujeme rozsah, v ktorom chceme obmedziť hodnotu.
2. Definovanie premennej: Definujeme premennú, ktorú chceme obmedziť do zadaného rozsahu.
3. Vykonanie operácie: Potom použijeme príkaz na obmedzenie tejto premennej do zadaného rozsahu a uloženie výsledku do novej premennej alebo priamo zobrazenie výsledku.

Príklad:

Tu je jednoduchý príklad, ako použiť funkciu obmedzenia hodnoty na interval v MakeCode:



Obrázok 100 Funkcia obmedzenia hodnoty na interval

V tomto príklade:

1. Definujeme premennú min s hodnotou 1.
2. Definujeme premennú max s hodnotou 10.
3. Definujeme premennú hodnota s hodnotou 15.
4. Pomocou funkcie Math.clamp obmedzíme hodnotu premennej hodnota do rozsahu medzi min a max a výsledok uložíme do premennej obmedzenaHodnota.
5. Na LED matici micro:bitu zobrazíme hodnotu premennej obmedzenaHodnota, čo bude 10 (pretože 15 je mimo rozsahu 1 až 10 a je obmedzená na maximálnu hodnotu 10).

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Vytvorte nový projekt a vložte doň uvedený blok kódu.
3. Nahrajte program do micro:bitu a pozorujte, ako sa na LED matici zobrazí obmedzená hodnota zadaného čísla.

Táto funkcia je veľmi užitočná pri rôznych výpočtoch a kontrolách hodnôt a môže byť súčasťou komplexnejších programov.

Funkcia mapovania z intervalu od-do do iného intervalu

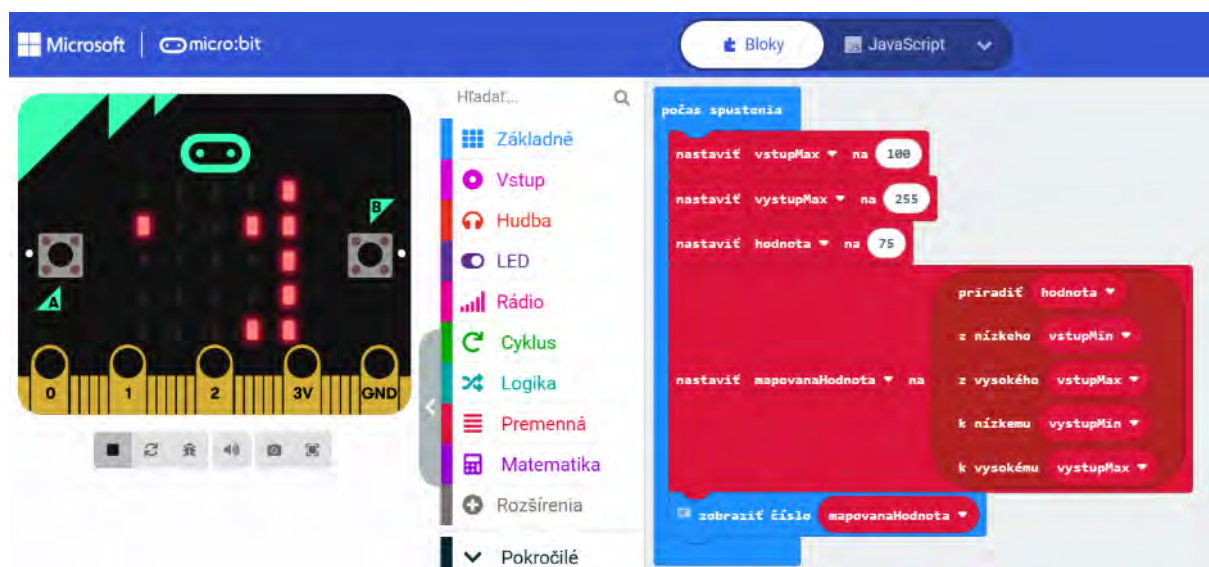
Funkcia mapovania v MakeCode slúži na prepočítanie hodnoty z jedného intervalu na zodpovedajúcu hodnotu v inom intervale. Táto operácia je veľmi užitočná pri rôznych výpočtoch a škálovaní hodnôt a je často využívaná v rôznych projektoch s micro:bitom.

Ako to funguje?

1. Definovanie vstupného a výstupného intervalu: Najprv definujeme vstupný interval, z ktorého chceme mapovať hodnotu, a výstupný interval, do ktorého chceme túto hodnotu mapovať.
2. Definovanie premennej: Definujeme premennú, ktorú chceme mapovať z jedného intervalu do druhého.
3. Vykonanie operácie: Potom použijeme príkaz na mapovanie tejto premennej a uloženie výsledku do novej premennej alebo priamo zobrazenie výsledku.

Príklad:

Tu je jednoduchý príklad, ako použiť funkciu mapovania v MakeCode:



Obrázok 101 Funkcia mapovania z intervalu od-do do iného intervalu

V tomto príklade:

1. Definujeme premennú vstupMin s hodnotou 0.
2. Definujeme premennú vstupMax s hodnotou 100.
3. Definujeme premennú vystupMin s hodnotou 0.
4. Definujeme premennú vystupMax s hodnotou 255.
5. Definujeme premennú hodnota s hodnotou 75.
6. Pomocou funkcie pins.map mapujeme hodnotu premennej hodnota z intervalu 0-100 do intervalu 0-255 a výsledok uložíme do premennej mapovanaHodnota.
7. Na LED matici micro:bitu zobrazíme hodnotu premennej mapovanaHodnota, čo bude 191.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Vytvorte nový projekt a vložte doň uvedený blok kódu.
3. Nahrajte program do micro:bitu a pozorujte, ako sa na LED matici zobrazí mapovaná hodnota zadaného čísla.

Táto funkcia je veľmi užitočná pri rôznych výpočtoch a škálovaní hodnôt a môže byť súčasťou komplexnejších programov.

Funkcia náhodného výberu medzi pravdou a nepravdou

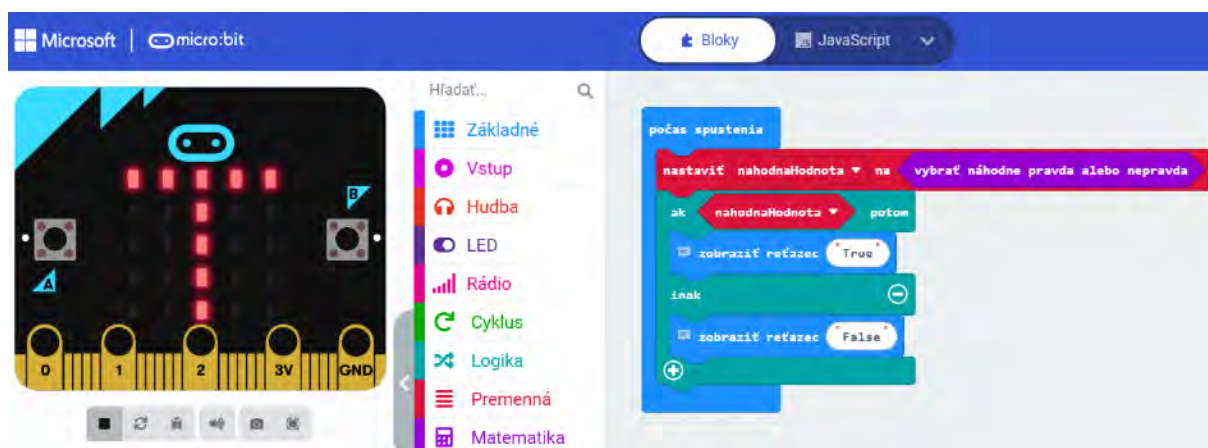
Funkcia náhodného výberu medzi pravdou a nepravdou v MakeCode slúži na generovanie náhodnej hodnoty true (pravda) alebo false (nepravda). Táto operácia je veľmi užitočná pri rôznych hrách, rozhodovaní a náhodných výberoch v projektoch s micro:bitom.

Ako to funguje?

- Vykonanie operácie: Použijeme príkaz na generovanie náhodnej hodnoty true alebo false a výsledok uložíme do premennej alebo priamo zobrazíme výsledok.

Príklad:

Tu je jednoduchý príklad, ako použiť funkciu náhodného výberu medzi pravdou a nepravdou v MakeCode:



Obrázok 102 Funkcia náhodného výberu medzi pravdou a nepravdou

V tomto príklade:

1. Pomocou funkcie `Math.randomBoolean` generujeme náhodnú hodnotu true alebo false a výsledok uložíme do premennej `nahodnaHodnota`.
2. Ak je náhodná hodnota true, na LED matici micro:bitu zobrazíme text "True".
3. Ak je náhodná hodnota false, na LED matici micro:bitu zobrazíme text "False".

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Vytvorte nový projekt a vložte doň uvedený blok kódu.
3. Nahrajte program do micro:bitu a pozorujte, ako sa na LED matici zobrazí náhodne vybraná hodnota pravda alebo nepravda.

Táto funkcia je veľmi užitočná pri rôznych hrách, rozhodovaní a náhodných výberoch a môže byť súčasťou komplexnejších programov.

11 Funkcie

Funkcia $f(x)$ s parametrom text

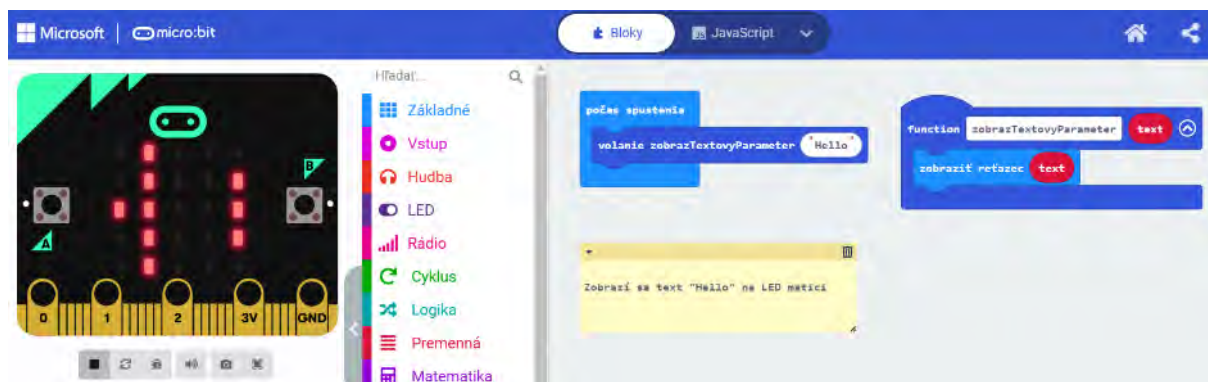
Funkcia $f(x)$ s parametrom text v MakeCode slúži na vykonanie rôznych operácií na základe textového reťazca zadaného ako parameter. Táto operácia je veľmi užitočná pri práci s textovými údajmi a umožňuje vytvárať dynamické a flexibilné programy.

Ako to funguje?

1. Definovanie funkcie: Najprv definujeme funkciu, ktorá prijíma textový reťazec ako parameter.
2. Vykonanie operácie: Potom použijeme príkaz na vykonanie operácie na tomto textovom reťazci a výsledok uložíme do premennej alebo priamo zobrazíme výsledok.

Príklad:

Tu je jednoduchý príklad, ako použiť funkciu $f(x)$ s parametrom text v MakeCode:



Obrázok 103 Funkcia $f(x)$ s parametrom text

V tomto príklade:

1. Definujeme funkciu zobrazTextovyParameter, ktorá prijíma textový reťazec text ako parameter.
2. Vo vnútri funkcie použijeme príkaz basic.showString na zobrazenie textu na LED matici micro:bitu.
3. Zavoláme funkciu zobrazTextovyParameter s parametrom "Hello", čo spôsobí zobrazenie textu "Hello" na LED matici.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Vytvorte nový projekt a vložte doň uvedený blok kódu.
3. Nahrajte program do micro:bitu a pozorujte, ako sa na LED matici zobrazí text zadaný ako parameter funkcie.

Táto funkcia je veľmi užitočná pri práci s textovými údajmi a umožňuje vytvárať dynamické a flexibilné programy.

Funkcia f(x) s parametrom logické premenné

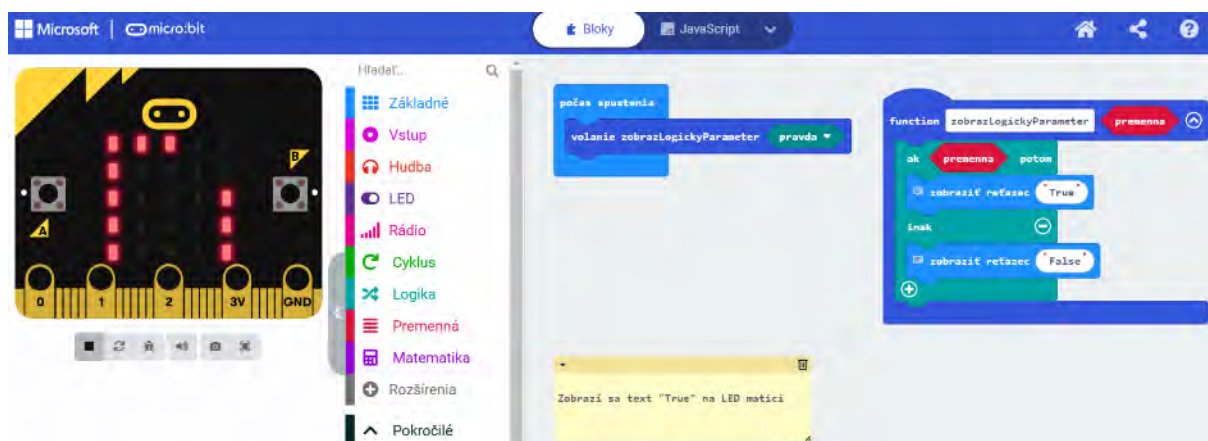
Funkcia f(x) s parametrom logické premenné v MakeCode slúži na vykonanie rôznych operácií na základe logických hodnôt (true/false) zadaných ako parameter. Táto operácia je veľmi užitočná pri práci s logickými údajmi a umožňuje vytvárať dynamické a flexibilné programy.

Ako to funguje?

1. Definovanie funkcie: Najprv definujeme funkciu, ktorá prijíma logickú premennú ako parameter.
2. Vykonanie operácie: Potom použijeme príkaz na vykonanie operácie na tejto logickej premennej a výsledok uložíme do premennej alebo priamo zobrazíme výsledok.

Príklad:

Tu je jednoduchý príklad, ako použiť funkciu f(x) s parametrom logické premenné v MakeCode:



Obrázok 104 Funkcia f(x) s parametrom logické premenné

V tomto príklade:

1. Definujeme funkciu zobrazLogickyParameter, ktorá prijíma logickú premennú premenna ako parameter.
2. Vo vnútri funkcie použijeme príkaz if na kontrolu hodnoty premennej a podľa toho zobrazíme text "True" alebo "False" na LED matici micro:bitu.
3. Zavoláme funkciu zobrazLogickyParameter s parametrom true, čo spôsobí zobrazenie textu "True" na LED matici.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Vytvorte nový projekt a vložte doň uvedený blok kódu.
3. Nahrajte program do micro:bitu a pozorujte, ako sa na LED matici zobrazí text podľa hodnoty logického parametra.

Táto funkcia je veľmi užitočná pri práci s logickými údajmi a umožňuje vytvárať dynamické a flexibilné programy.

Funkcia $f(x)$ s parametrom číslo

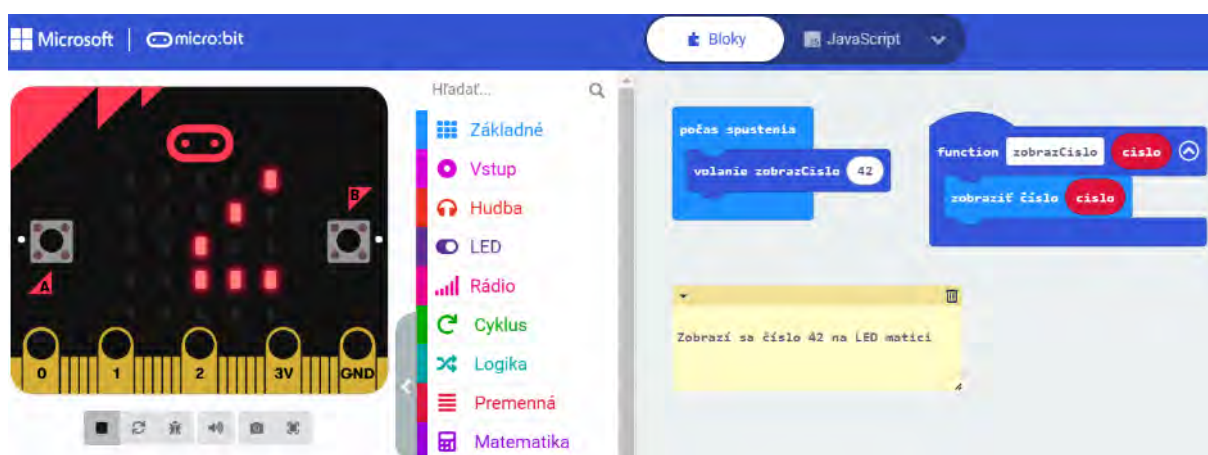
Funkcia $f(x)$ s parametrom číslo v MakeCode slúži na vykonanie rôznych operácií na základe číselnej hodnoty zadaného ako parameter. Táto operácia je veľmi užitočná pri práci s číselnými údajmi a umožňuje vytvárať dynamické a flexibilné programy.

Ako to funguje?

1. Definovanie funkcie: Najprv definujeme funkciu, ktorá prijíma číselnú hodnotu ako parameter.
2. Vykonanie operácie: Potom použijeme príkaz na vykonanie operácie na tejto číselnej hodnote a výsledok uložíme do premennej alebo priamo zobrazíme výsledok.

Príklad:

Tu je jednoduchý príklad, ako použiť funkciu $f(x)$ s parametrom číslo v MakeCode:



Obrázok 105 Funkcia $f(x)$ s parametrom číslo

V tomto príklade:

1. Definujeme funkciu zobrazCislo, ktorá prijíma číselnú hodnotu cislo ako parameter.
2. Vo vnútri funkcie použijeme príkaz `basic.showNumber` na zobrazenie čísla na LED matici micro:bitu.
3. Zavoláme funkciu zobrazCislo s parametrom 42, čo spôsobí zobrazenie čísla 42 na LED matici.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Vytvorte nový projekt a vložte doň uvedený blok kódu.
3. Nahrajte program do micro:bitu a pozorujte, ako sa na LED matici zobrazí číslo zadané ako parameter funkcie.

Táto funkcia je veľmi užitočná pri práci s číselnými údajmi a umožňuje vytvárať dynamické a flexibilné programy.

Funkcia $f(x)$ s parametrom pole

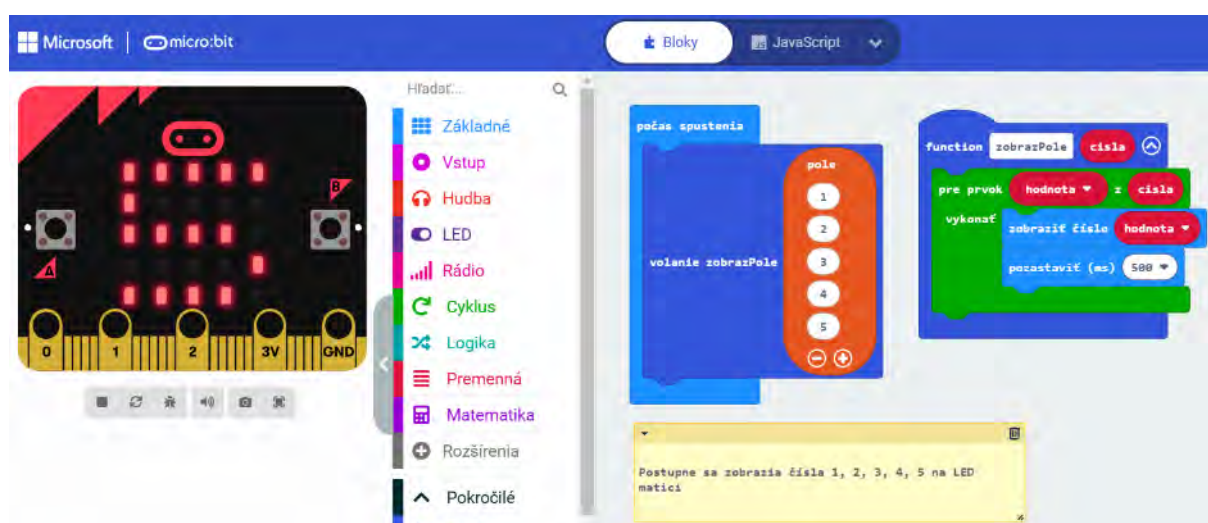
Funkcia $f(x)$ s parametrom pole v MakeCode slúži na vykonanie rôznych operácií na základe hodnôt zadaných v poli. Táto operácia je veľmi užitočná pri práci s viacerými údajmi naraz a umožňuje vytvárať dynamické a flexibilné programy.

Ako to funguje?

1. Definovanie funkcie: Najprv definujeme funkciu, ktorá prijíma pole ako parameter.
2. Vykonanie operácie: Potom použijeme príkaz na vykonanie operácie na hodnotách v tomto poli a výsledok uložíme do premennej alebo priamo zobrazíme výsledok.

Príklad:

Tu je jednoduchý príklad, ako použiť funkciu $f(x)$ s parametrom pole v MakeCode:



Obrázok 106 Funkcia $f(x)$ s parametrom pole

V tomto príklade:

1. Definujeme funkciu zobrazPole, ktorá prijíma pole cisla ako parameter.
2. Vo vnútri funkcie použijeme cyklus for...of na iteráciu cez hodnoty v poli a zobrazíme každú hodnotu na LED matici micro:bitu s krátkou pauzou medzi jednotlivými hodnotami.
3. Zavoláme funkciu zobrazPole s parametrom [1, 2, 3, 4, 5], čo spôsobí postupné zobrazenie čísel 1, 2, 3, 4 a 5 na LED matici.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Vytvorte nový projekt a vložte doň uvedený blok kódu.
3. Nahrajte program do micro:bitu a pozorujte, ako sa na LED matici zobrazia hodnoty zadaného poľa.

Táto funkcia je veľmi užitočná pri práci s viacerými údajmi naraz a umožňuje vytvárať dynamické a flexibilné programy.

Funkcia f(x) s parametrom LED sprite

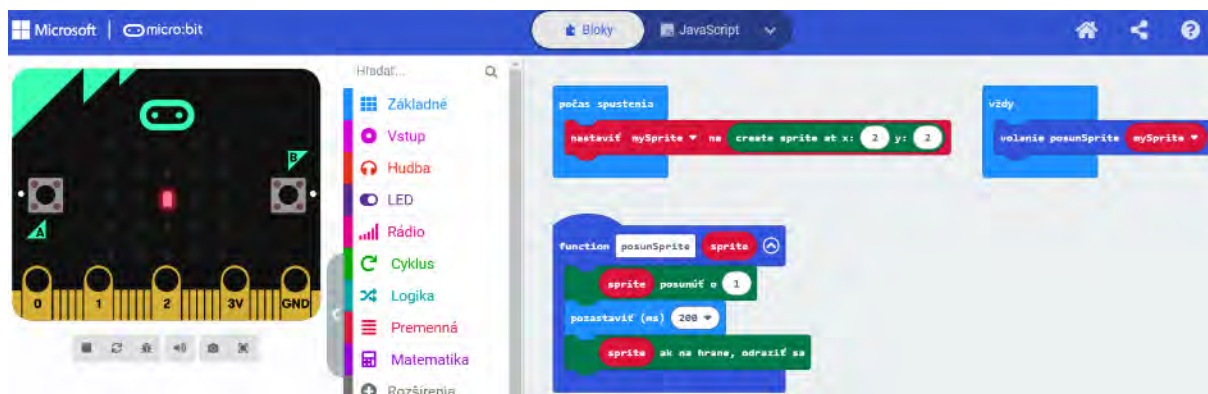
Funkcia f(x) s parametrom LED sprite v MakeCode slúži na vykonanie rôznych operácií na základe zadaného LED sprite (bodky na LED matici). Táto operácia je veľmi užitočná pri práci s grafickými objektmi na micro:bit a umožňuje vytvárať dynamické a interaktívne programy.

Ako to funguje?

1. Definovanie funkcie: Najprv definujeme funkciu, ktorá prijíma LED sprite ako parameter.
2. Vykonanie operácie: Potom použijeme príkaz na vykonanie operácie na tomto LED sprite a výsledok uložíme do premennej alebo priamo zobrazíme výsledok.

Príklad:

Tu je jednoduchý príklad, ako použiť funkciu f(x) s parametrom LED sprite v MakeCode:



Obrázok 107 Funkcia f(x) s parametrom LED sprite

V tomto príklade:

1. Definujeme funkciu posunSprite, ktorá prijíma LED sprite ako parameter.
2. Vo vnútri funkcie použijeme príkaz sprite.move(1) na posunutie sprite o jednu pozíciu a príkaz sprite.ifOnEdgeBounce() na odrazenie sprite od okraja LED matice.
3. Vytvoríme LED sprite mySprite na pozícii (2, 2).
4. Použijeme cyklus basic.forever na nepretržité volanie funkcie posunSprite, čo spôsobí, že sprite sa bude neustále pohybovať a odrážať od okrajov LED matice.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Vytvorte nový projekt a vložte doň uvedený blok kódu.
3. Nahrajte program do micro:bitu a pozorujte, ako sa sprite pohybuje a odráža od okrajov LED matice.

Táto funkcia je veľmi užitočná pri práci s grafickými objektmi na micro:bit a umožňuje vytvárať dynamické a interaktívne programy.

Funkcia $f(x)$ s parametrom image

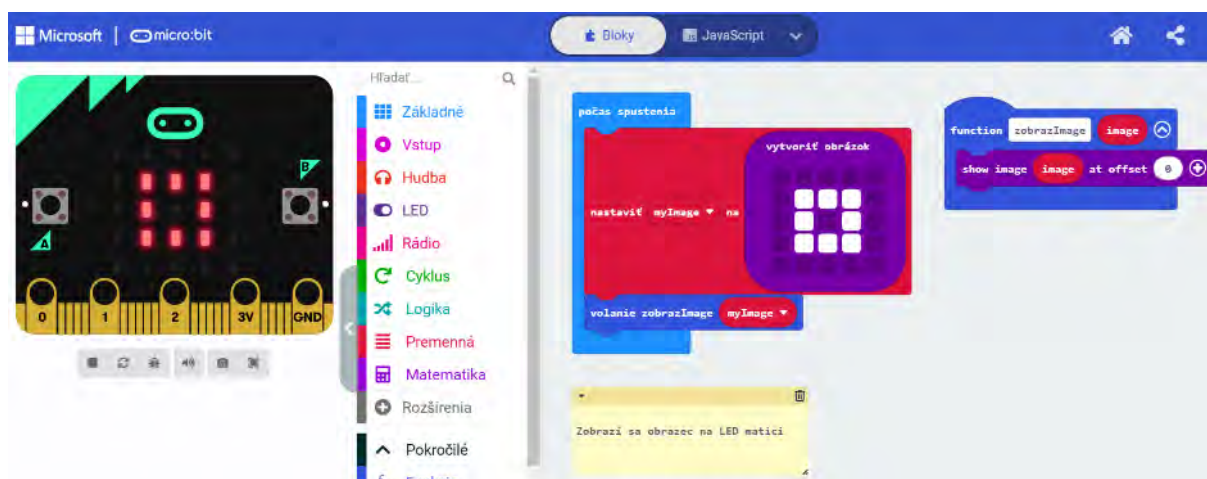
Funkcia $f(x)$ s parametrom image v MakeCode slúži na vykonanie rôznych operácií na základe zadaného obrazového objektu (image). Táto operácia je veľmi užitočná pri práci s grafickými objektmi na micro:bit a umožňuje vytvárať dynamické a interaktívne programy.

Ako to funguje?

1. Definovanie funkcie: Najprv definujeme funkciu, ktorá prijíma obrazový objekt (image) ako parameter.
2. Vykonanie operácie: Potom použijeme príkaz na vykonanie operácie na tomto obrazovom objekte a výsledok uložíme do premennej alebo priamo zobrazíme výsledok.

Príklad:

Tu je jednoduchý príklad, ako použiť funkciu $f(x)$ s parametrom image v MakeCode:



Obrázok 108 Funkcia $f(x)$ s parametrom image

V tomto príklade:

1. Definujeme funkciu zobrazImage, ktorá prijíma obrazový objekt image ako parameter.
2. Vo vnútri funkcie použijeme príkaz image.showImage(0) na zobrazenie obrazca na LED matici micro:bitu.
3. Vytvoríme obrazový objekt myImage so vzorom a zavoláme funkciu zobrazImage s týmto obrazovým objektom, čo spôsobí zobrazenie obrazca na LED matici.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Vytvorte nový projekt a vložte doň uvedený blok kódu.
3. Nahrajte program do micro:bitu a pozorujte, ako sa na LED matici zobrazí obrazový objekt.

Táto funkcia je veľmi užitočná pri práci s grafickými objektmi na micro:bit a umožňuje vytvárať dynamické a interaktívne programy.

12 Polia

Funkcia "nastaviť zoznam na pole"

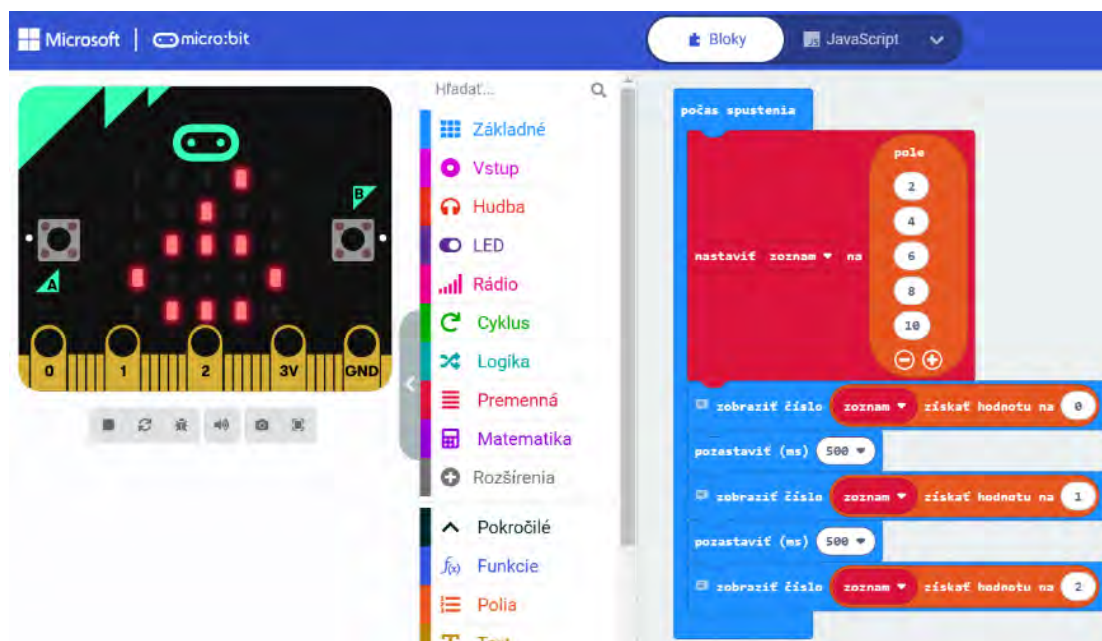
Táto funkcia v MakeCode slúži na inicializáciu zoznamu (array) s určitými hodnotami. Táto operácia je veľmi užitočná pri práci s poľami a manipulácii s viacerými údajmi naraz, čo umožňuje vytvárať dynamické a flexibilné programy.

Ako to funguje?

1. Definovanie premennej: Najprv definujeme premennú typu zoznam (array), ktorú chceme inicializovať.
2. Nastavenie hodnôt: Potom použijeme príkaz na nastavenie hodnôt v tomto zozname na požadované hodnoty.

Príklad:

Tu je jednoduchý príklad, ako použiť funkciu "nastaviť zoznam na pole" v MakeCode:



Obrázok 109 Funkcia "nastaviť zoznam na pole"

V tomto príklade:

1. Definujeme premennú zoznam, ktorá je zoznamom čísel (array) a nastavíme jej hodnoty na [2, 4, 6, 8, 10].
2. Na LED matici micro:bitu postupne zobrazíme hodnoty zoznamu: prvá hodnota (zoznam[0]) je 2, druhá hodnota (zoznam[1]) je 4 a tretia hodnota (zoznam[2]) je 6, s krátkou pauzou medzi zobrazeniami.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Vytvorte nový projekt a vložte doň uvedený blok kódu.
3. Nahrajte program do micro:bitu a pozorujte, ako sa na LED matici zobrazia hodnoty zoznamu.

Táto funkcia je veľmi užitočná pri práci s poľami a manipulácii s viacerými údajmi naraz.

Funkcia "nastaviť textový zoznam na pole"

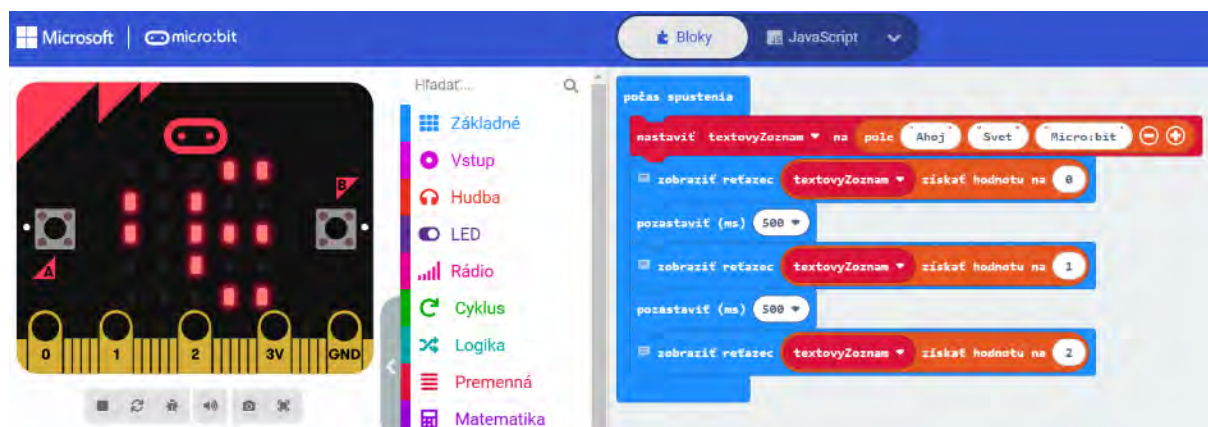
Táto funkcia v MakeCode slúži na inicializáciu zoznamu (array) s určitými textovými hodnotami. Táto operácia je veľmi užitočná pri práci s poľami a manipulácii s viacerými textovými údajmi naraz, čo umožňuje vytvárať dynamické a flexibilné programy.

Ako to funguje?

1. Definovanie premennej: Najprv definujeme premennú typu zoznam (array), ktorú chceme inicializovať textovými hodnotami.
2. Nastavenie hodnôt: Potom použijeme príkaz na nastavenie hodnôt v tomto zozname na požadované textové hodnoty.

Príklad:

Tu je jednoduchý príklad, ako použiť funkciu "nastaviť textový zoznam na pole" v MakeCode:



Obrázok 110 Funkcia "nastaviť textový zoznam na pole"

V tomto príklade:

1. Definujeme premennú textovyZoznam, ktorá je zoznamom textových hodnôt (array) a nastavíme jej hodnoty na ["Ahoj", "Svet", "Micro:bit"].
2. Na LED matici micro:bitu postupne zobrazíme hodnoty zoznamu: prvá hodnota (textovyZoznam[0]) je "Ahoj", druhá hodnota (textovyZoznam[1]) je "Svet" a tretia hodnota (textovyZoznam[2]) je "Micro:bit", s krátkou pauzou medzi zobrazovaniami.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Vytvorte nový projekt a vložte doň uvedený blok kódu.
3. Nahrajte program do micro:bitu a pozorujte, ako sa na LED matici zobrazia textové hodnoty zoznamu.

Táto funkcia je veľmi užitočná pri práci s poľami a manipulácii s viacerými textovými údajmi naraz.

Funkcia "prázdné pole"

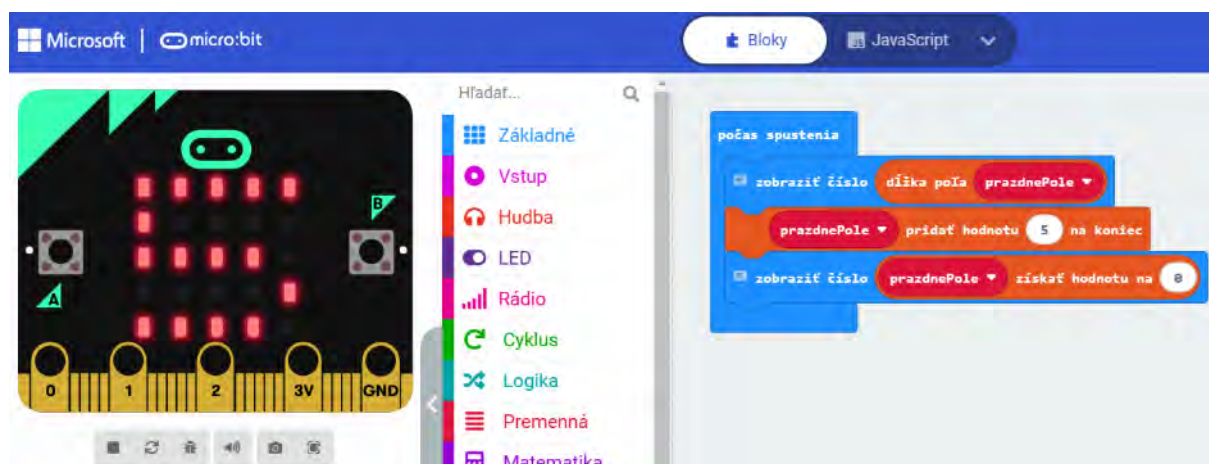
Táto funkcia v MakeCode slúži na vytvorenie a inicializáciu prázdneho zoznamu (array). Táto operácia je veľmi užitočná pri práci s poľami, kde potrebujeme postupne pridávať hodnoty do zoznamu a začíname s prázdny poľom.

Ako to funguje?

1. Definovanie premennej: Najprv definujeme premennú typu zoznam (array), ktorú chceme inicializovať ako prázdne pole.
2. Použitie prázdneho poľa: Môžeme následne pridávať hodnoty do tohto poľa podľa potreby.

Príklad:

Tu je jednoduchý príklad, ako použiť funkciu "prázdné pole" v MakeCode:



Obrázok 111 Funkcia "prázdné pole"

V tomto príklade:

1. Definujeme premennú prazdnePole, ktorá je prázdne pole (array) typu číslo.
2. Na LED matici micro:bitu zobrazíme dĺžku prázdneho poľa (`prazdnePole.length`), čo je 0.
3. Pridáme hodnotu 5 do prázdneho poľa pomocou príkazu `prazdnePole.push(5)`.
4. Na LED matici micro:bitu zobrazíme prvú hodnotu v poli (`prazdnePole[0]`), čo je 5.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Vytvorte nový projekt a vložte doň uvedený blok kódu.
3. Nahrajte program do micro:bitu a pozorujte, ako sa na LED matici zobrazia hodnoty prázdneho poľa po pridaní novej hodnoty.

Táto funkcia je veľmi užitočná pri práci s poľami, kde potrebujeme postupne pridávať hodnoty do zoznamu.

Funkcia "dĺžka poľa (zoznam)"

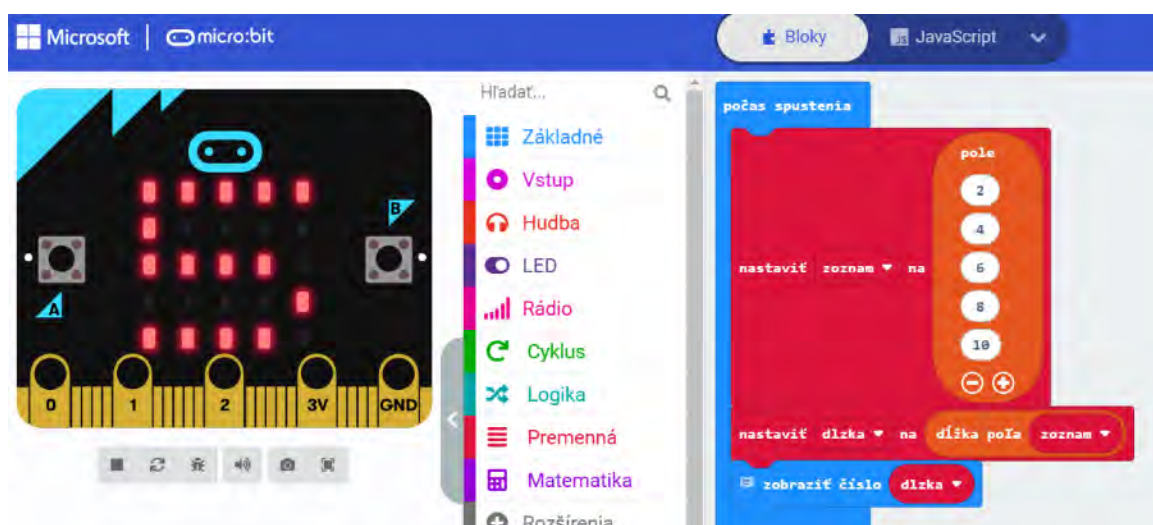
Táto funkcia v MakeCode slúži na získanie dĺžky (počet prvkov) v zozname (array). Táto operácia je veľmi užitočná pri práci s poľami, kde potrebujeme zistiť, koľko prvkov zoznam obsahuje, čo umožňuje lepšiu manipuláciu a kontrolu údajov.

Ako to funguje?

1. Definovanie premennej: Najprv definujeme premennú typu zoznam (array), ktorú chceme inicializovať s určitými hodnotami.
2. Zistenie dĺžky: Potom použijeme príkaz na zistenie dĺžky tohto zoznamu a uloženie výsledku do premennej alebo priamo zobrazenie výsledku.

Príklad:

Tu je jednoduchý príklad, ako použiť funkciu "dĺžka poľa (zoznam)" v MakeCode:



Obrázok 112 Funkcia "dĺžka poľa (zoznam)"

V tomto príklade:

1. Definujeme premennú zoznam, ktorá je zoznamom čísel (array) a nastavíme jej hodnoty na [2, 4, 6, 8, 10].
2. Pomocou príkazu `zoznam.length` zistíme dĺžku zoznamu a výsledok uložíme do premennej `dlzka`.
3. Na LED matici micro:bitu zobrazíme hodnotu premennej `dlzka`, čo je 5 (počet prvkov v zozname).

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Vytvorte nový projekt a vložte doň uvedený blok kódu.
3. Nahrajte program do micro:bitu a pozorujte, ako sa na LED matici zobrazí dĺžka zoznamu.

Táto funkcia je veľmi užitočná pri práci s poľami, kde potrebujeme zistiť počet prvkov v zozname.

Funkcia "získať hodnotu na"

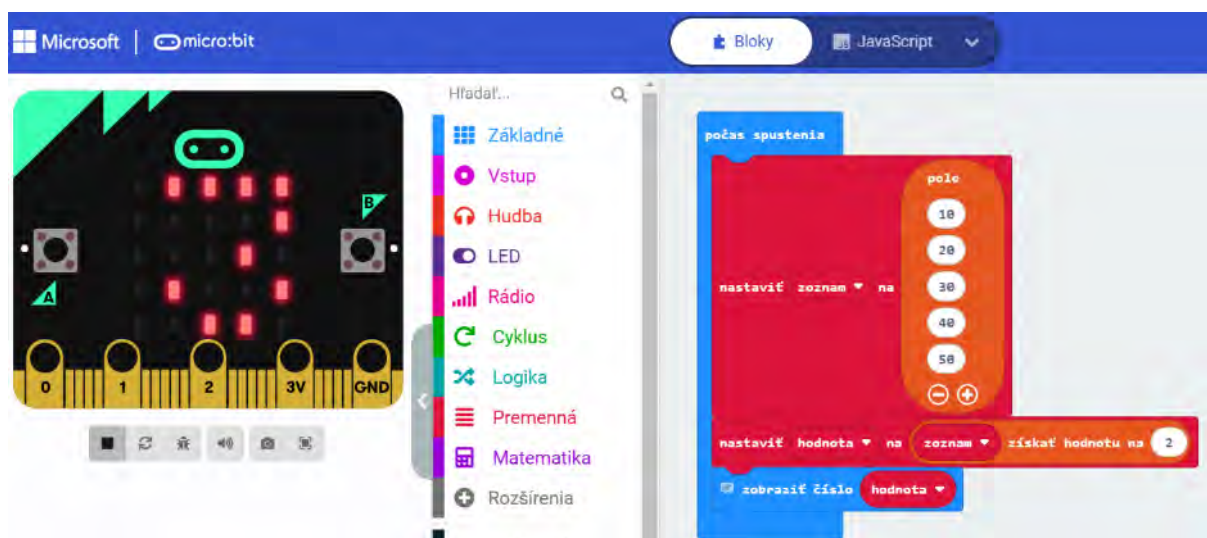
Táto funkcia v MakeCode slúži na získanie hodnoty na určitom indexe v zozname (array). Táto operácia je veľmi užitočná pri práci s poľami, kde potrebujeme prístup k konkrétnym prvkom zoznamu na základe ich pozície.

Ako to funguje?

1. Definovanie premennej: Najprv definujeme premennú typu zoznam (array), ktorú chceme inicializovať s určitými hodnotami.
2. Získanie hodnoty: Potom použijeme príkaz na získanie hodnoty na určitom indexe v tomto zozname a uloženie výsledku do premennej alebo priamo zobrazenie výsledku.

Príklad:

Tu je jednoduchý príklad, ako použiť funkciu "získať hodnotu na" v MakeCode:



Obrázok 113 Funkcia "získať hodnotu na"

V tomto príklade:

1. Definujeme premennú zoznam, ktorá je zoznamom čísel (array) a nastavíme jej hodnoty na [10, 20, 30, 40, 50].
2. Pomocou príkazu `zoznam[2]` získame hodnotu na indexe 2 (tretí prvok v zozname) a výsledok uložíme do premennej hodnota.
3. Na LED matici micro:bitu zobrazíme hodnotu premennej hodnota, čo je 30.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Vytvorte nový projekt a vložte doň uvedený blok kódu.
3. Nahrajte program do micro:bitu a pozorujte, ako sa na LED matici zobrazí hodnota na konkrétnom indexe zoznamu.

Táto funkcia je veľmi užitočná pri práci s poľami, kde potrebujeme prístup k jednotlivým hodnotám na základe ich pozície.

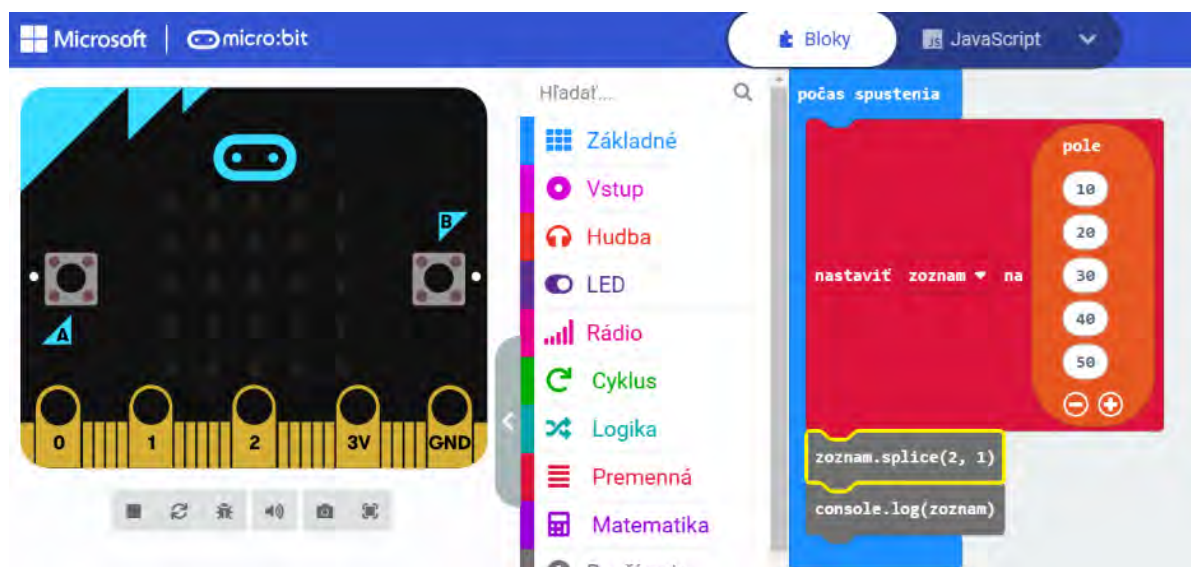
Príkaz "odobrať hodnotu na indexe"

Táto funkcia v MakeCode slúži na odstránenie hodnoty na určitom indexe v zozname (array). Tento príkaz je užitočný, keď potrebujeme z nášho zoznamu odstrániť konkrétnu položku.

Ako to funguje?

1. Definovanie premennej: Najprv definujeme premennú typu zoznam (array) a inicializujeme ju s určitými hodnotami.
2. Odstránenie hodnoty: Potom použijeme príkaz na odstránenie hodnoty na určitom indexe v tomto zozname. Tým sa táto hodnota odstráni zo zoznamu a všetky nasledujúce prvky sa posunú o jedno miesto dozadu, aby sa zaplnilo uvoľnené miesto.

Príklad: Pozrime sa na jednoduchý príklad použitia funkcie "odobrať hodnotu na indexe" v MakeCode:



Obrázok 114 Príkaz "odobrať hodnotu na indexe"

V tomto príklade:

1. Definujeme premennú zoznam, ktorá je zoznamom čísel (array) a nastavíme jej hodnoty na [10, 20, 30, 40, 50].
2. Pomocou príkazu `zoznam.splice(2, 1)` odstránime hodnotu na indexe 2 (tretí prvok v zozname). Výsledný zoznam bude [10, 20, 40, 50].

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Vytvorte nový projekt a vložte doň uvedený blok kódu.
3. Nahrajte program do micro:bitu a pozorujte, ako sa zoznam zmení po odstránení hodnoty na konkrétnom indexe.

Tento príkaz je veľmi užitočný pri práci s poľami, kde potrebujeme odstrániť jednotlivé hodnoty na základe ich pozície.

Funkcia "načítať a odobrať poslednú hodnotu zo zoznamu"

Táto funkcia v prostredí MakeCode sa používa na získanie posledného prvku zo zoznamu (list) a jeho odstránenie z tohto zoznamu.

Ako to funguje?

1. Definovanie premennej zoznam: Vytvorte premennú zoznam a inicializujte ju hodnotami [10, 20, 30, 40, 50].
2. Načítanie a odstránenie poslednej hodnoty: Použite príkaz na načítanie posledného prvku zo zoznamu a jeho uloženie do novej premennej poslednaHodnota. Potom odstráňte tento posledný prvok zo zoznamu.

Príklad: Pozrime sa na jednoduchý príklad použitia funkcie "načítať a odobrať poslednú hodnotu zo zoznamu" v MakeCode:



Obrázok 115 Funkcia "načítať a odobrať poslednú hodnotu zo zoznamu"

Tu je postup, ako to nastaviť v prostredí MakeCode pomocou blokov:

1. Inicializácia zoznamu:
 - a. Vytvorte nový zoznam zoznam a pridajte doň hodnoty [10, 20, 30, 40, 50].
2. Načítanie poslednej hodnoty:
 - a. Vytvorte premennú poslednaHodnota.
 - b. Použite blok "dĺžka zoznamu" na získanie dĺžky zoznamu a odčítajte od nej 1 na získanie indexu posledného prvku.
 - c. Použite blok "získať prvok zoznamu na" a nastavte ho na index posledného prvku (dĺžka zoznamu - 1).
 - d. Uložte túto hodnotu do premennej poslednaHodnota.
3. Odstránenie poslednej hodnoty:
 - a. Použite blok "odstrániť hodnotu zo zoznamu na" a nastavte ho na index posledného prvku (dĺžka zoznamu - 1).

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Vytvorte nový projekt a vložte doň uvedený blok kódu.
3. Nahrajte program do micro:bitu a pozorujte, ako sa zoznam zmení po načítaní a odstránení poslednej hodnoty.

Tento príkaz je veľmi užitočný pri práci s poľami, kde potrebujeme načítať a odstrániť posledný prvok zo zoznamu.

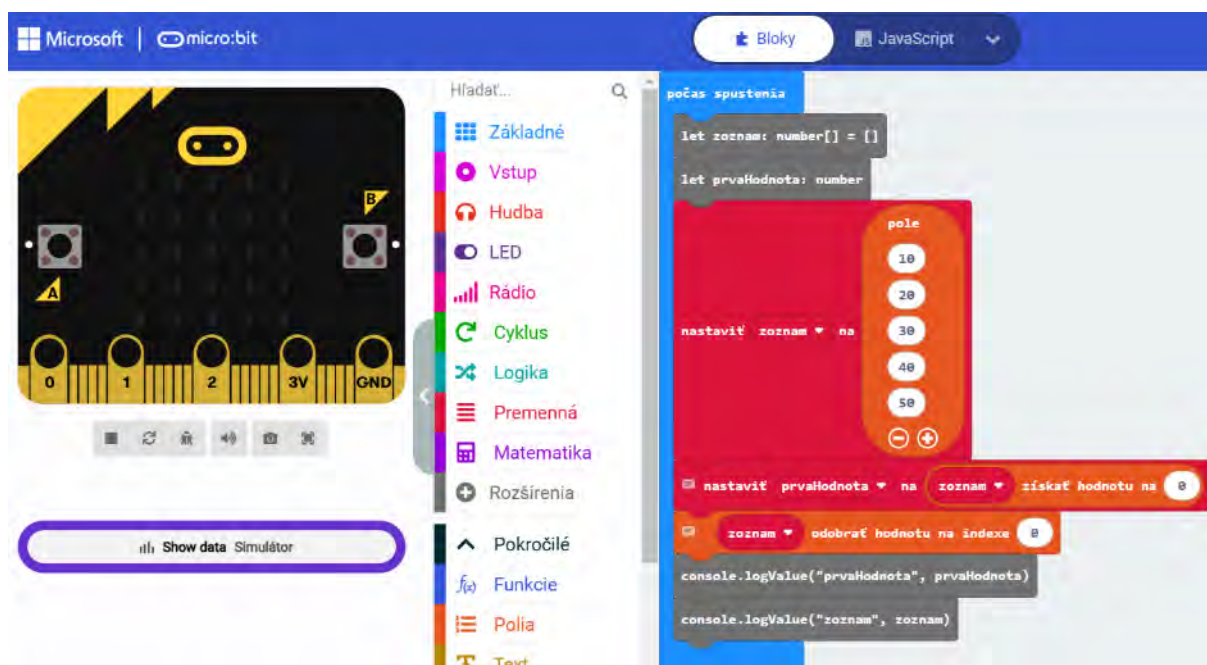
Príkaz "načítať a odobrať prvú hodnotu zo zoznamu"

Táto funkcia v MakeCode umožňuje načítať a odstrániť prvú hodnotu zo zoznamu (array). Tento príkaz je veľmi užitočný, keď potrebujeme získať prvú položku zo zoznamu a zároveň ju odstrániť.

Ako to funguje?

1. Definovanie premennej: Najprv definujeme premennú typu zoznam (array) a inicializujeme ju s určitými hodnotami.
2. Načítanie a odstránenie hodnoty: Potom použijeme príkaz na načítanie a odstránenie prvej hodnoty v tomto zozname. Hodnota sa načíta, odstráni zo zoznamu a môžeme ju uložiť do premennej alebo ju priamo zobrazíť.

Príklad: Tu je vizuálny príklad kódu v MakeCode pomocou blokov na načítanie a odstránenie prvej hodnoty zo zoznamu:



Obrázok 116 Príkaz "načítať a odobrať prvú hodnotu zo zoznamu"

Tu je postup, ako to nastaviť v prostredí MakeCode pomocou blokov:

1. Inicializácia zoznamu:
 - a. Vytvorte nový zoznam zoznam a pridajte doň hodnoty [10, 20, 30, 40, 50].
2. Načítanie prvej hodnoty:
 - a. Vytvorte premennú prvaHodnota.
 - b. Použite blok "získať prvok zoznamu na" a nastavte ho na index 0.
 - c. Uložte túto hodnotu do premennej prvaHodnota.
3. Odstránenie prvej hodnoty:
 - a. Použite blok "odstrániť hodnotu zo zoznamu na" a nastavte ho na index 0.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Vytvorte nový projekt a vložte doň uvedené bloky kódu.
3. Nahrajte program do micro:bitu a pozorujte, ako sa zoznam zmení po načítaní a odstránení prvej hodnoty.

Tento príkaz je veľmi užitočný pri práci s poľami, kde potrebujeme načítať a odstrániť prvý prvok zo zoznamu.

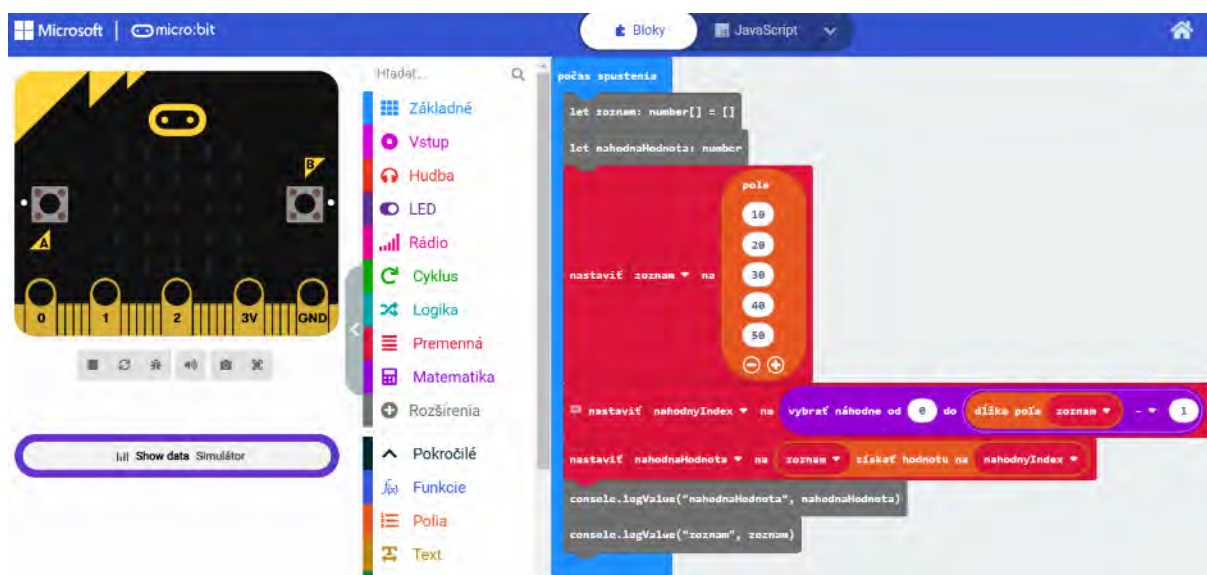
Príkaz "získať náhodnú hodnotu z"

Táto funkcia v MakeCode umožňuje získať náhodnú hodnotu zo zoznamu (array). Tento príkaz je veľmi užitočný, keď potrebujeme náhodne vybrať jeden prvok zo zoznamu.

Ako to funguje?

1. Definovanie premennej: Najprv definujeme premennú typu zoznam (array) a inicializujeme ju s určitými hodnotami.
2. Získanie náhodnej hodnoty: Potom použijeme príkaz na získanie náhodnej hodnoty z tohto zoznamu. Hodnota sa načíta a môžeme ju uložiť do premennej alebo ju priamo zobraziť.

Príklad: Tu je vizuálny príklad kódu v MakeCode pomocou blokov na získanie náhodnej hodnoty zo zoznamu:



Obrázok 117 Príkaz "získať náhodnú hodnotu z"

Tu je postup, ako to nastaviť v prostredí MakeCode pomocou blokov:

1. Inicializácia zoznamu:
 - a. Vytvorte nový zoznam zoznam a pridajte doň hodnoty [10, 20, 30, 40, 50].
2. Získanie náhodnej hodnoty:
 - a. Vytvorte premennú nahodnaHodnota.
 - b. Použite blok "náhodné číslo od 0 do" a nastavte ho na dĺžku zoznamu mínus 1 (dĺžka zoznamu - 1).
 - c. Použite blok "získať prvok zoznamu na" a nastavte ho na náhodný index, ktorý sme získali v predchádzajúcom kroku.
 - d. Uložte túto hodnotu do premennej nahodnaHodnota.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Vytvorte nový projekt a vložte doň uvedené bloky kódu.
3. Nahrajte program do micro:bitu a pozorujte, ako sa zobrazí náhodne vybraná hodnota zo zoznamu.

Tento príkaz je veľmi užitočný pri práci s poľami, kde potrebujeme náhodne vybrať jeden prvok zo zoznamu.

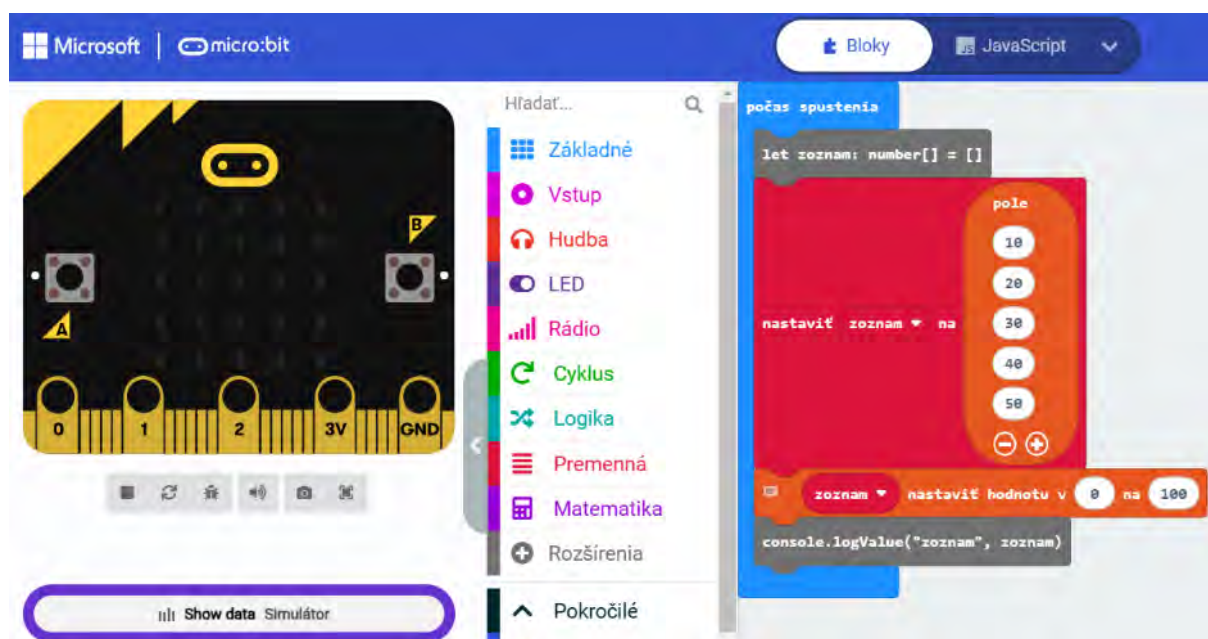
Príkaz "nastaviť hodnotu v 0 na"

Táto funkcia v MakeCode umožňuje nastaviť hodnotu na indexe 0 (prvého prvku) v zozname (array) na novú hodnotu. Tento príkaz je veľmi užitočný, keď potrebujeme aktualizovať hodnotu prvého prvku v zozname.

Ako to funguje?

1. Definovanie premennej: Najprv definujeme premennú typu zoznam (array) a inicializujeme ju s určitými hodnotami.
2. Nastavenie hodnoty: Potom použijeme príkaz na nastavenie hodnoty na indexe 0 (prvý prvok) v tomto zozname na novú hodnotu.

Príklad: Tu je vizuálny príklad kódu v MakeCode pomocou blokov na nastavenie hodnoty na indexe 0:



Obrázok 118 Príkaz "nastaviť hodnotu v 0 na"

Tu je postup, ako to nastaviť v prostredí MakeCode pomocou blokov:

1. Inicializácia zoznamu: Vytvorte nový zoznam zoznam a pridajte doň hodnoty [10, 20, 30, 40, 50].
2. Nastavenie hodnoty: Použite blok "nastaviť hodnotu v" a nastavte ho na index 0.
3. Nastavte novú hodnotu, ktorú chcete priradiť k prvému prvku zoznamu (napríklad 100).

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Vytvorte nový projekt a vložte doň uvedené bloky kódu.
3. Nahrajte program do micro:bitu a pozorujte, ako sa hodnota na indexe 0 zmení na novú hodnotu.

Tento príkaz je veľmi užitočný pri práci s poľami, kde potrebujeme aktualizovať hodnotu prvého prvku v zozname.

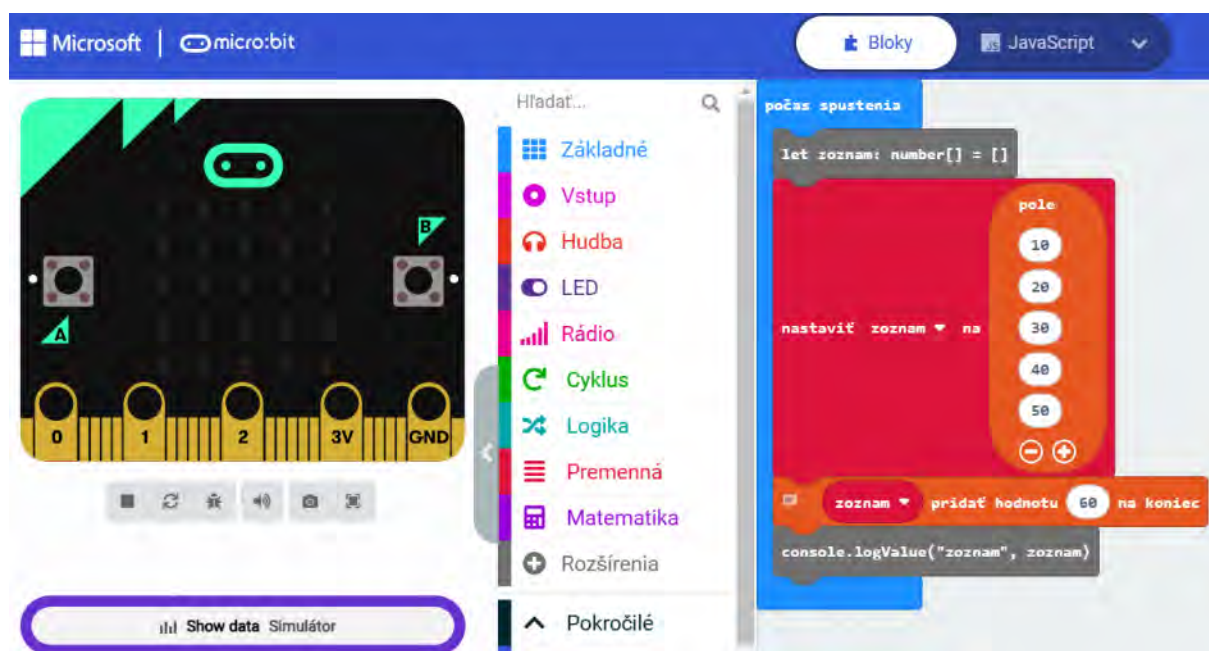
Príkaz "pridať hodnotu na koniec"

Táto funkcia v MakeCode umožňuje pridať novú hodnotu na koniec zoznamu (array). Tento príkaz je veľmi užitočný, keď potrebujeme rozšíriť zoznam o nový prvok.

Ako to funguje?

1. Definovanie premennej: Najprv definujeme premennú typu zoznam (array) a inicializujeme ju s určitými hodnotami.
2. Pridanie hodnoty: Potom použijeme príkaz na pridanie novej hodnoty na koniec tohto zoznamu.

Príklad: Tu je vizuálny príklad kódu v MakeCode pomocou blokov na pridanie hodnoty na koniec zoznamu:



Obrázok 119 Príkaz "pridať hodnotu na koniec"

Tu je postup, ako to nastaviť v prostredí MakeCode pomocou blokov:

1. Inicializácia zoznamu: Vytvorte nový zoznam zoznam a pridajte doň hodnoty [10, 20, 30, 40, 50].
2. Pridanie hodnoty na koniec: Použite blok "pridať hodnotu na koniec" a nastavte ho na novú hodnotu (napríklad 60).

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Vytvorte nový projekt a vložte doň uvedené bloky kódu.
3. Nahrajte program do micro:bitu a pozorujte, ako sa hodnota 60 pridá na koniec zoznamu.

Tento príkaz je veľmi užitočný pri práci s poľami, kde potrebujeme pridať nový prvok na koniec zoznamu.

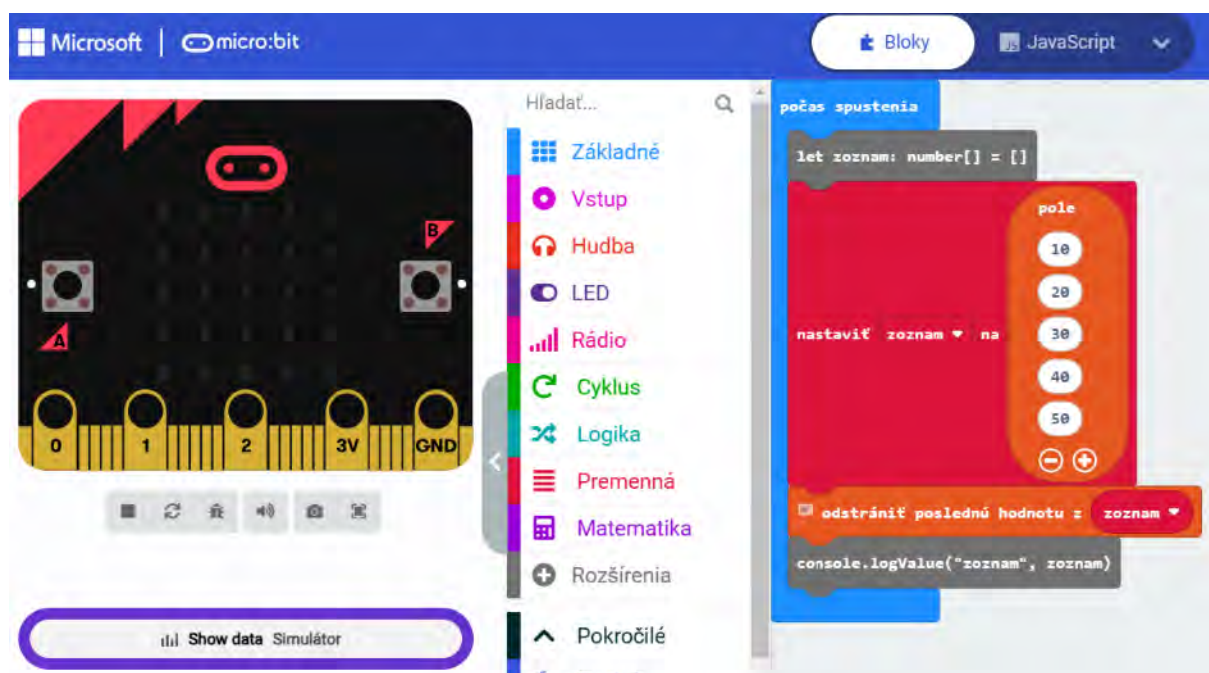
Príkaz "odstráň poslednú hodnotu z"

Táto funkcia v MakeCode umožňuje odstrániť poslednú hodnotu zo zoznamu (array). Tento príkaz je veľmi užitočný, keď potrebujeme odstrániť posledný prvok zo zoznamu.

Ako to funguje?

1. Definovanie premennej: Najprv definujeme premennú typu zoznam (array) a inicializujeme ju s určitými hodnotami.
2. Odstránenie poslednej hodnoty: Potom použijeme príkaz na odstránenie poslednej hodnoty v tomto zozname. Táto hodnota sa odstráni a zoznam sa skráti o jeden prvok.

Príklad: Tu je vizuálny príklad kódu v MakeCode pomocou blokov na odstránenie poslednej hodnoty zo zoznamu:



Obrázok 120 Príkaz "odstráň poslednú hodnotu z"

Tu je postup, ako to nastaviť v prostredí MakeCode pomocou blokov:

1. Inicializácia zoznamu: Vytvorte nový zoznam zoznam a pridajte doň hodnoty [10, 20, 30, 40, 50].
2. Odstránenie poslednej hodnoty: Použite blok "odstráň poslednú hodnotu z".

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Vytvorte nový projekt a vložte doň uvedené bloky kódu.
3. Nahrajte program do micro:bitu a pozorujte, ako sa hodnota odstráni z konca zoznamu.

Tento príkaz je veľmi užitočný pri práci s poľami, kde potrebujeme odstrániť posledný prvok zo zoznamu.

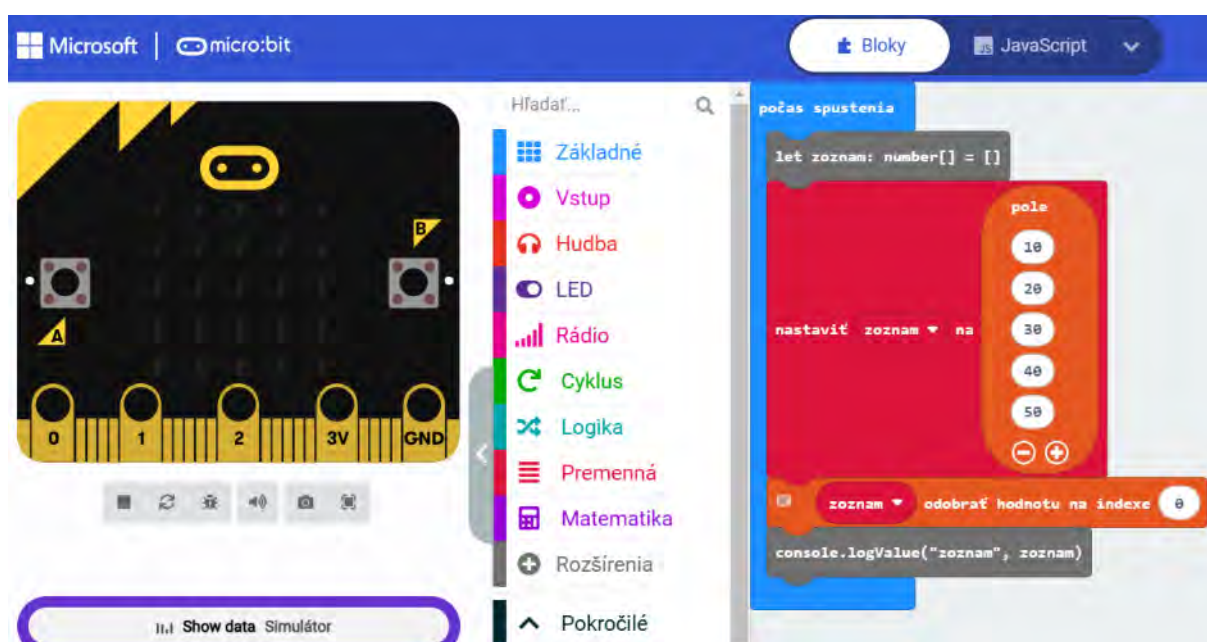
Príkaz "odstráň prvú hodnotu z"

Táto funkcia v MakeCode umožňuje odstrániť prvú hodnotu zo zoznamu (array). Tento príkaz je veľmi užitočný, keď potrebujeme odstrániť prvý prvok zo zoznamu.

Ako to funguje?

1. Definovanie premennej: Najprv definujeme premennú typu zoznam (array) a inicializujeme ju s určitými hodnotami.
2. Odstránenie prvej hodnoty: Potom použijeme príkaz na odstránenie prvej hodnoty v tomto zozname. Táto hodnota sa odstráni a všetky nasledujúce prvky sa posunú o jedno miesto dopredu, aby sa zaplnilo uvoľnené miesto.

Príklad: Tu je vizuálny príklad kódu v MakeCode pomocou blokov na odstránenie prvej hodnoty zo zoznamu:



Obrázok 121 Príkaz "odstráň prvú hodnotu z"

Tu je postup, ako to nastaviť v prostredí MakeCode pomocou blokov:

1. Inicializácia zoznamu: Vytvorte nový zoznam *zoznam* a pridajte doň hodnoty [10, 20, 30, 40, 50].
2. Odstránenie prvej hodnoty: Použite blok "odstrániť hodnotu zo zoznamu na" a nastavte ho na index 0.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Vytvorte nový projekt a vložte doň uvedené bloky kódu.
3. Nahrajte program do micro:bitu a pozorujte, ako sa hodnota odstráni z prvej pozície zoznamu.

Tento príkaz je veľmi užitočný pri práci s poľami, kde potrebujeme odstrániť prvý prvok zo zoznamu.

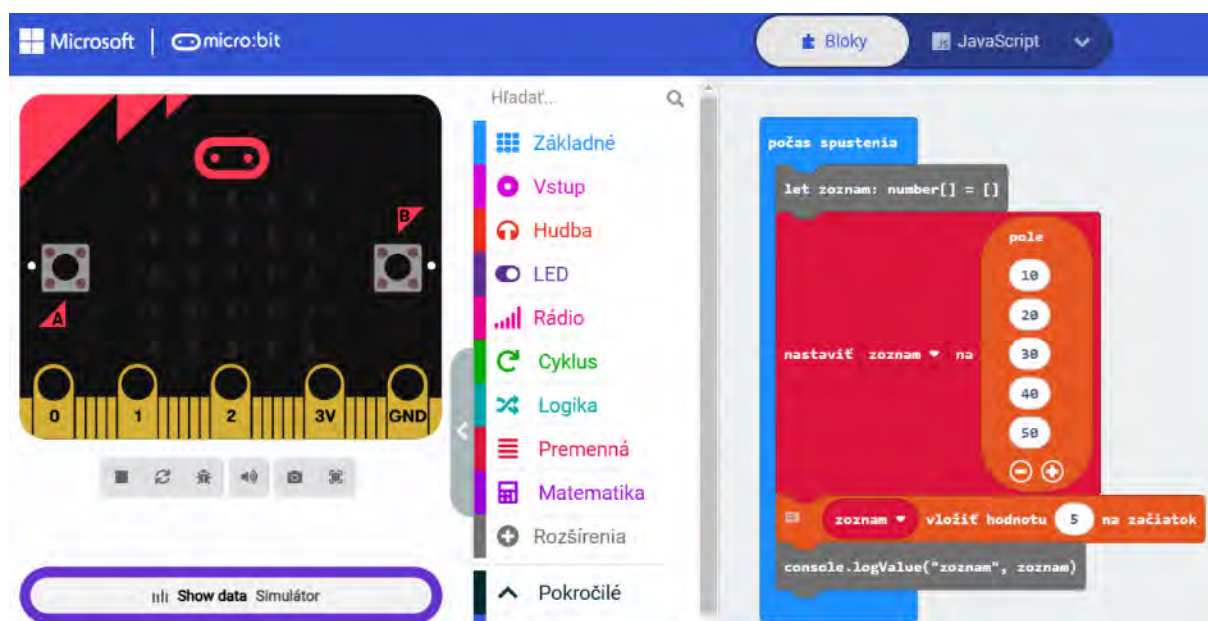
Príkaz "vložiť hodnotu na začiatok"

Táto funkcia v MakeCode umožňuje vložiť novú hodnotu na začiatok zoznamu (array). Tento príkaz je veľmi užitočný, keď potrebujeme pridať nový prvok na začiatok zoznamu.

Ako to funguje?

1. Definovanie premennej: Najprv definujeme premennú typu zoznam (array) a inicializujeme ju s určitými hodnotami.
2. Vloženie hodnoty: Potom použijeme príkaz na vloženie novej hodnoty na začiatok tohto zoznamu.

Príklad: Tu je vizuálny príklad kódu v MakeCode pomocou blokov na vloženie hodnoty na začiatok zoznamu:



Obrázok 122 Príkaz "vložiť hodnotu na začiatok"

Tu je postup, ako to nastaviť v prostredí MakeCode pomocou blokov:

1. Inicializácia zoznamu: Vytvorte nový zoznam *zoznam* a pridajte doň hodnoty [10, 20, 30, 40, 50].
2. Vloženie hodnoty na začiatok: Použite blok "vložiť hodnotu na začiatok" a nastavte ho na novú hodnotu (napríklad 5).

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Vytvorte nový projekt a vložte doň uvedené bloky kódu.
3. Nahrajte program do micro:bitu a pozorujte, ako sa hodnota 5 pridá na začiatok zoznamu.

Tento príkaz je veľmi užitočný pri práci s poľami, kde potrebujeme pridať nový prvok na začiatok zoznamu.

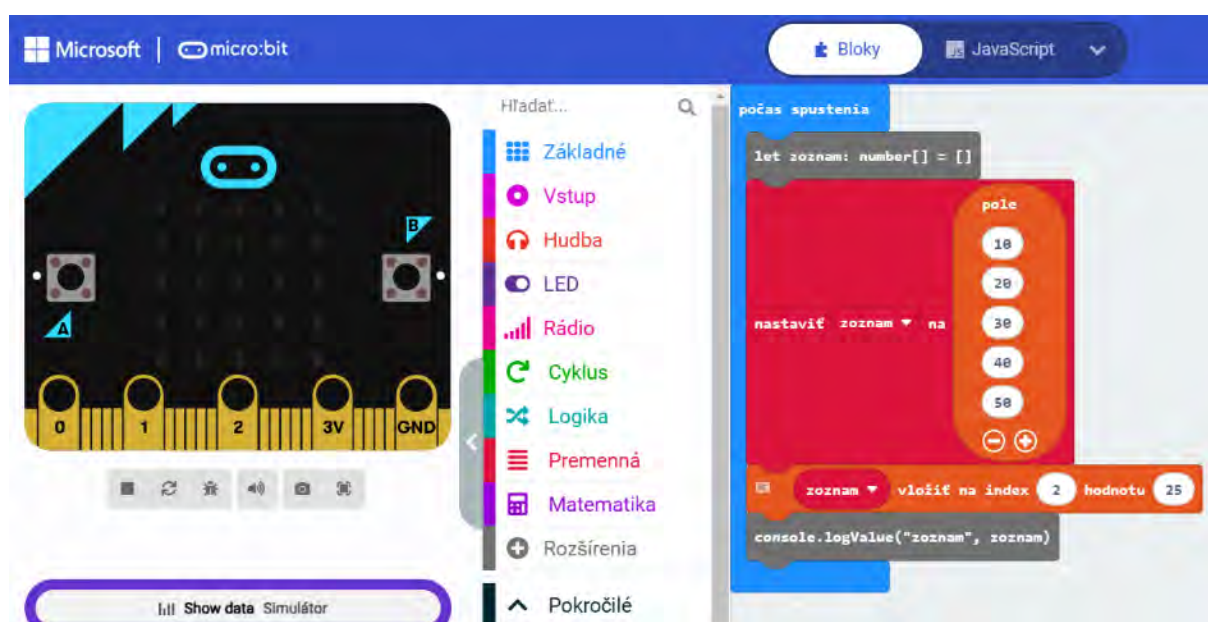
Príkaz "vložiť na index hodnotu"

Táto funkcia v MakeCode umožňuje vložiť novú hodnotu na konkrétny index v zozname (array). Tento príkaz je veľmi užitočný, keď potrebujeme pridať nový prvok na špecifickú pozíciu v zozname.

Ako to funguje?

1. Definovanie premennej: Najprv definujeme premennú typu zoznam (array) a inicializujeme ju s určitými hodnotami.
2. Vloženie hodnoty: Potom použijeme príkaz na vloženie novej hodnoty na konkrétny index v tomto zozname.

Príklad: Tu je vizuálny príklad kódu v MakeCode pomocou blokov na vloženie hodnoty na konkrétny index zoznamu:



Obrázok 123 Príkaz "vložiť na index hodnotu"

Tu je postup, ako to nastaviť v prostredí MakeCode pomocou blokov:

1. Inicializácia zoznamu: Vytvorte nový zoznam *zoznam* a pridajte doň hodnoty [10, 20, 30, 40, 50].
2. Vloženie hodnoty na konkrétny index: Použite blok "vložiť hodnotu na index" a nastavte index, kde chcete vložiť novú hodnotu (napríklad index 2). Nastavte novú hodnotu, ktorú chcete priradiť k danému indexu (napríklad 25).

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Vytvorte nový projekt a vložte doň uvedené bloky kódu.
3. Nahrajte program do micro:bitu a pozorujte, ako sa hodnota 25 vloží na index 2 v zozname.

Tento príkaz je veľmi užitočný pri práci s poľami, kde potrebujeme pridať nový prvok na konkrétnu pozíciu v zozname.

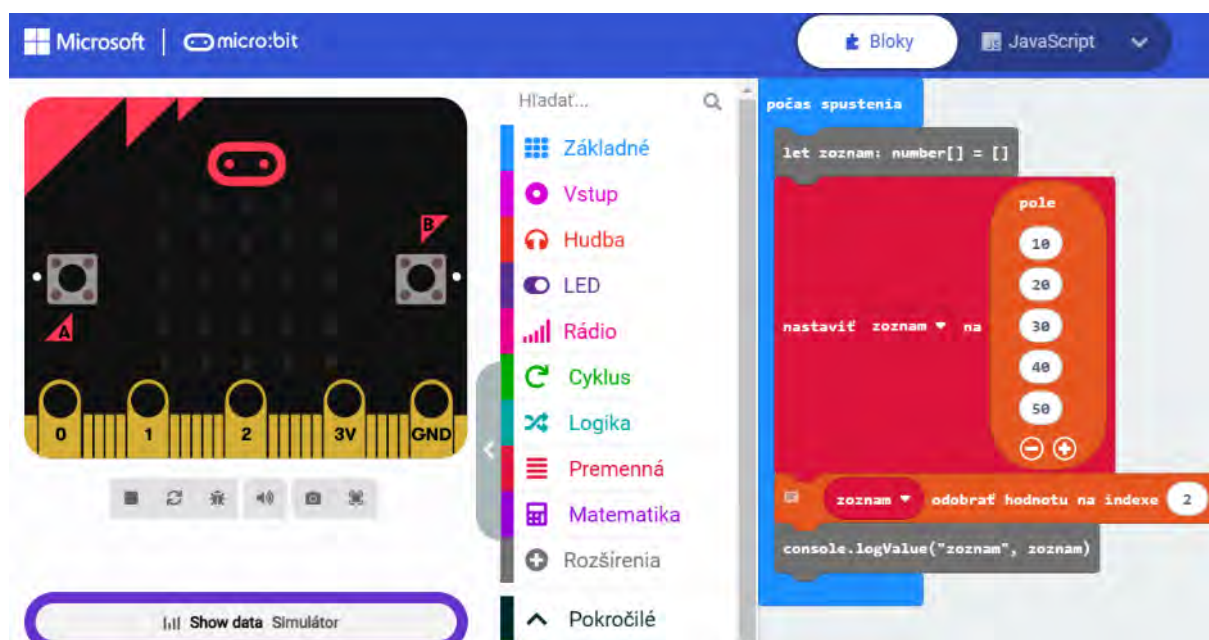
Príkaz "odobrať hodnotu na indexe"

Táto funkcia v MakeCode umožňuje odstrániť hodnotu na konkrétnom indexe v zozname (array). Tento príkaz je veľmi užitočný, keď potrebujeme odstrániť prvok zo zoznamu na špecifickej pozícii.

Ako to funguje?

1. Definovanie premennej: Najprv definujeme premennú typu zoznam (array) a inicializujeme ju s určitými hodnotami.
2. Odstránenie hodnoty: Potom použijeme príkaz na odstránenie hodnoty na konkrétnom indexe v tomto zozname. Táto hodnota sa odstráni a všetky nasledujúce prvky sa posunú o jedno miesto dopredu, aby sa zaplnilo uvoľnené miesto.

Príklad: Tu je vizuálny príklad kódu v MakeCode pomocou blokov na odstránenie hodnoty na konkrétnom indexe zoznamu:



Obrázok 124 Príkaz "odobrať hodnotu na indexe"

Tu je postup, ako to nastaviť v prostredí MakeCode pomocou blokov:

1. Inicializácia zoznamu: Vytvorte nový zoznam zoznam a pridajte doň hodnoty [10, 20, 30, 40, 50].
2. Odstránenie hodnoty na konkrétnom indexe: Použite blok "odstrániť hodnotu zo zoznamu na" a nastavte index, z ktorého chcete odstrániť hodnotu (napríklad index 2).

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Vytvorte nový projekt a vložte doň uvedené bloky kódu.
3. Nahrajte program do micro:bitu a pozorujte, ako sa hodnota odstráni z konkrétnej pozície zoznamu.

Tento príkaz je veľmi užitočný pri práci s poľami, kde potrebujeme odstrániť prvok zo zoznamu na konkrétnom indexe.

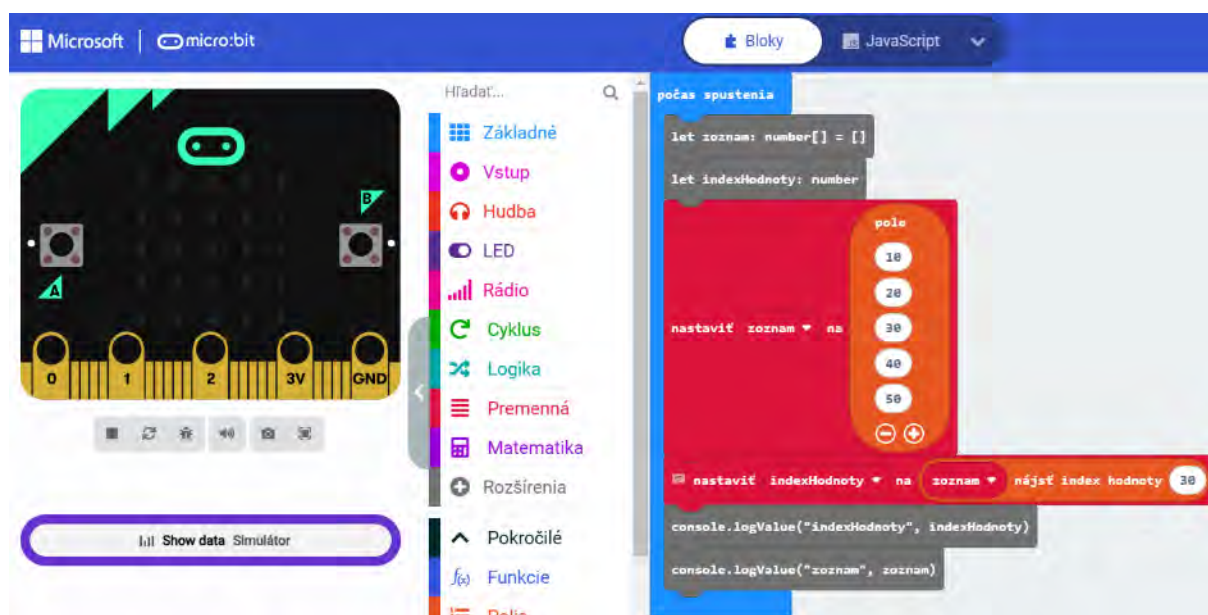
Príkaz "najdi index hodnoty"

Táto funkcia v MakeCode umožňuje nájsť index určitej hodnoty v zozname (array). Tento príkaz je veľmi užitočný, keď potrebujeme zistiť pozíciu konkrétnej hodnoty v zozname.

Ako to funguje?

1. Definovanie premennej: Najprv definujeme premennú typu zoznam (array) a inicializujeme ju s určitými hodnotami.
2. Nájdenie indexu: Potom použijeme príkaz na nájdenie indexu konkrétnej hodnoty v tomto zozname.

Príklad: Tu je vizuálny príklad kódu v MakeCode pomocou blokov na nájdenie indexu určitej hodnoty v zozname:



Obrázok 125 Príkaz "najdi index hodnoty"

Tu je postup, ako to nastaviť v prostredí MakeCode pomocou blokov:

1. Inicializácia zoznamu: Vytvorte nový zoznam *zoznam* a pridajte doň hodnoty [10, 20, 30, 40, 50].
2. Nájdenie indexu hodnoty: Vytvorte premennú *indexHodnoty*.
 - a. Použite blok "najdi index hodnoty" a nastavte hodnotu, ktorej index chcete nájsť (napríklad 30).
 - b. Uložte výsledný index do premennej *indexHodnoty*.

Vyskúšajte si to!

3. Otvorte MakeCode editor na stránke [micro:bit](https://microbit.org).
4. Vytvorte nový projekt a vložte doň uvedené bloky kódu.
5. Nahrajte program do micro:bitu a pozorujte, ako sa zobrazí index hodnoty 30 v zozname.

Tento príkaz je veľmi užitočný pri práci s poľami, kde potrebujeme zistiť pozíciu konkrétnej hodnoty v zozname.

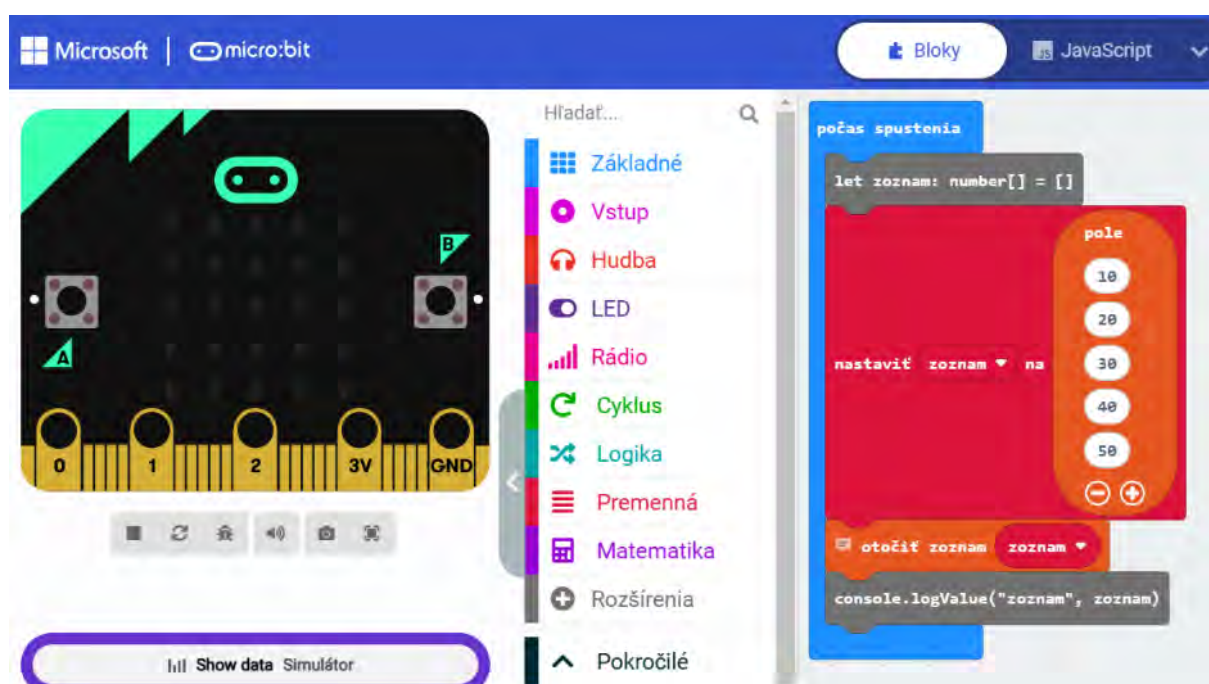
Príkaz "otočiť zoznam"

Táto funkcia v MakeCode umožňuje otočiť poradie prvkov v zozname (array). Tento príkaz je veľmi užitočný, keď potrebujeme zmeniť poradie prvkov v zozname na opačné.

Ako to funguje?

1. Definovanie premennej: Najprv definujeme premennú typu zoznam (array) a inicializujeme ju s určitými hodnotami.
2. Otočenie zoznamu: Potom použijeme príkaz na otočenie poradia prvkov v tomto zozname.

Príklad: Tu je vizuálny príklad kódu v MakeCode pomocou blokov na otočenie zoznamu:



Obrázok 126 Príkaz "otočiť zoznam"

Tu je postup, ako to nastaviť v prostredí MakeCode pomocou blokov:

1. Inicializácia zoznamu: Vytvorte nový zoznam *zoznam* a pridajte doň hodnoty [10, 20, 30, 40, 50].
2. Otočenie zoznamu: Použite blok "otočiť zoznam", ktorý otočí poradie prvkov v zozname.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Vytvorte nový projekt a vložte doň uvedené bloky kódu.
3. Nahrajte program do micro:bitu a pozorujte, ako sa poradie prvkov v zozname otočí.

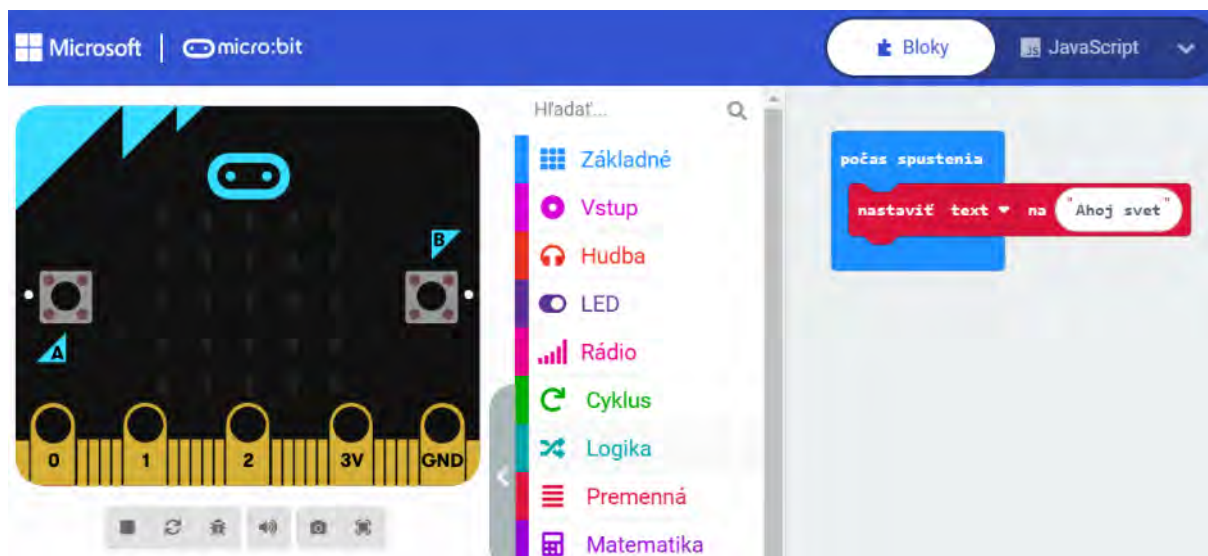
Tento príkaz je veľmi užitočný pri práci s poľami, kde potrebujeme zmeniť poradie prvkov na opačné.

13 Text

Vytvoriť textový reťazec:

- Tento príkaz vytvorí nový textový reťazec s určitou hodnotou.
- Použitie: Na vytvorenie nového textového reťazca.
- Výsledok: "Ahoj svet"

Príklad:

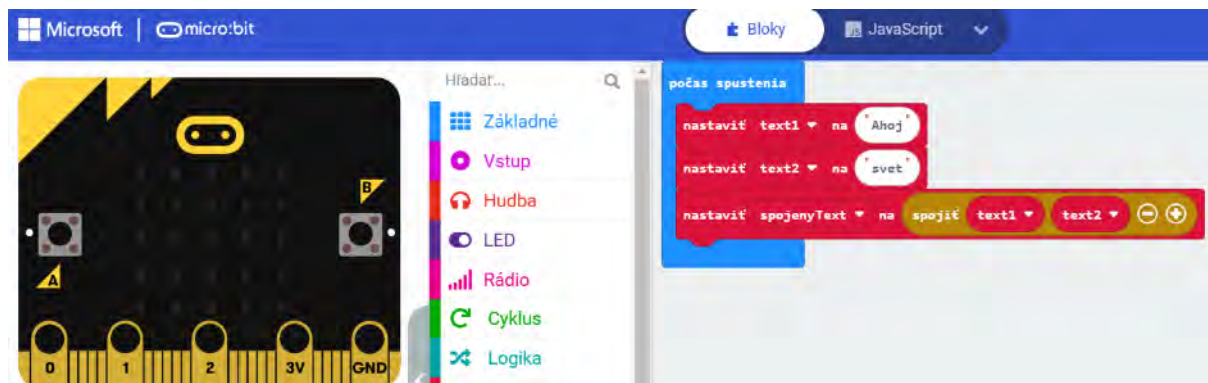


Obrázok 127 Príkaz "Vytvoriť textový reťazec"

Spojiť texty:

- Tento príkaz spojí dva textové reťazce do jedného.
- Použitie: Na zretazenie dvoch textových reťazcov.
- Výsledok: "Ahoj svet"

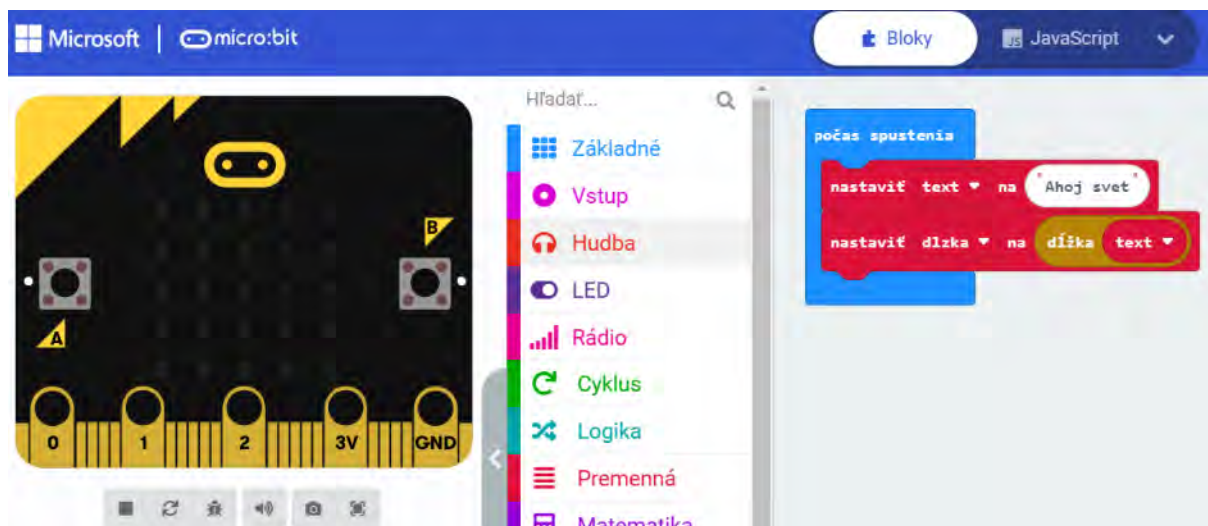
Príklad:



Obrázok 128 Príklad ako spojiť texty

Dĺžka textu:

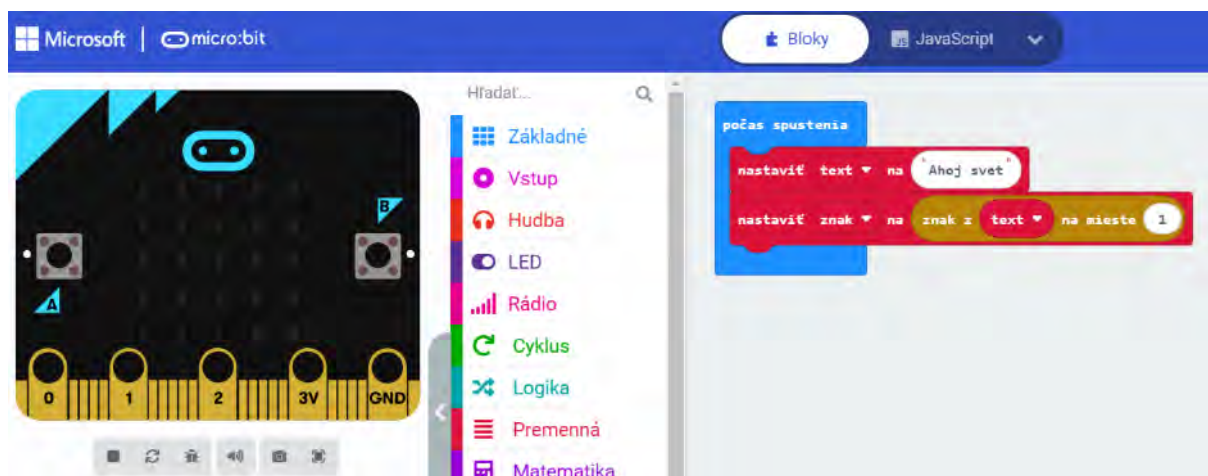
- Tento príkaz vráti počet znakov v textovom reťazci.
- Použitie: Na zistenie dĺžky textového reťazca.
- Výsledok: 9

Príklad:

Obrázok 129 Príklad na zistenie dĺžky textového reťazca

Získať znak na:

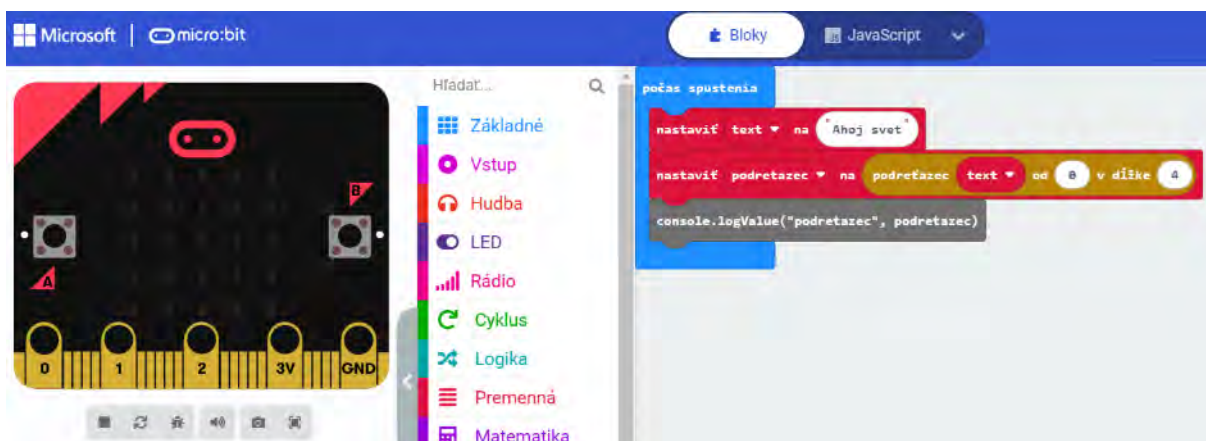
- Tento príkaz vráti znak na konkrétnom indexe v textovom reťazci.
- Použitie: Na získanie znaku z určitého miesta v texte.
- Výsledok: "h"

Príklad:

Obrázok 130 Príklad na získanie znaku z určitého miesta v texte

Podreťazec:

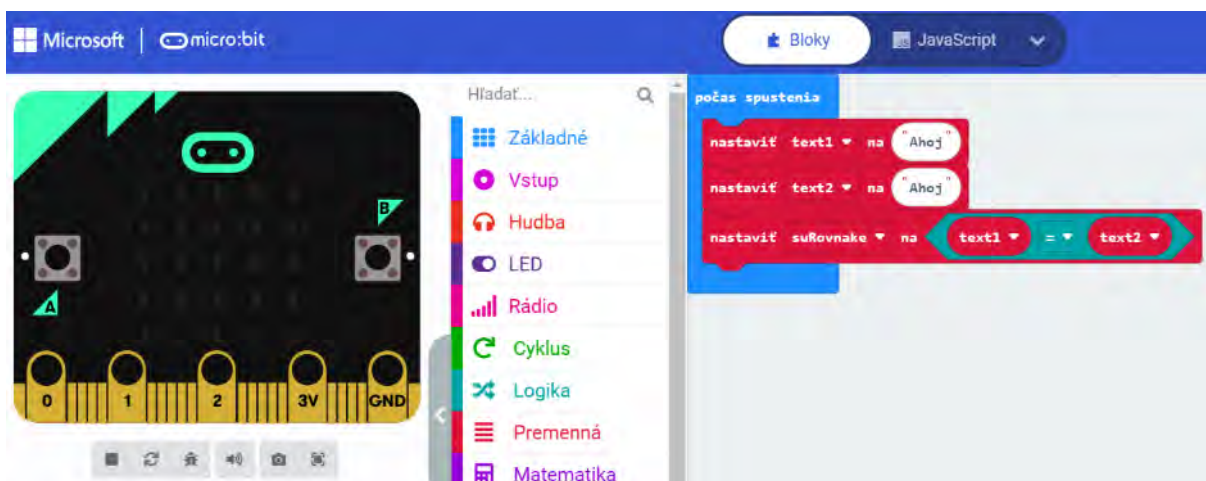
- Tento príkaz vráti časť textového reťazca medzi dvoma indexmi.
- Použitie: Na získanie časti textového reťazca.
 - Inicializácia textového reťazca: Vytvorte nový textový reťazec text a nastavte jeho hodnotu napríklad na "Ahoj svet".
 - Získanie podreťazca: Vytvorte premennú podreťazec.
 - Použite blok "podreťazec od v dĺžke" a nastavte počiatočný index, kde chcete začať (napríklad 0), a dĺžku podreťazca (napríklad 4).
- Výsledok: "Ahoj"

Príklad:

Obrázok 131 Príklad na získanie časti textového reťazca

Porovnať texty:

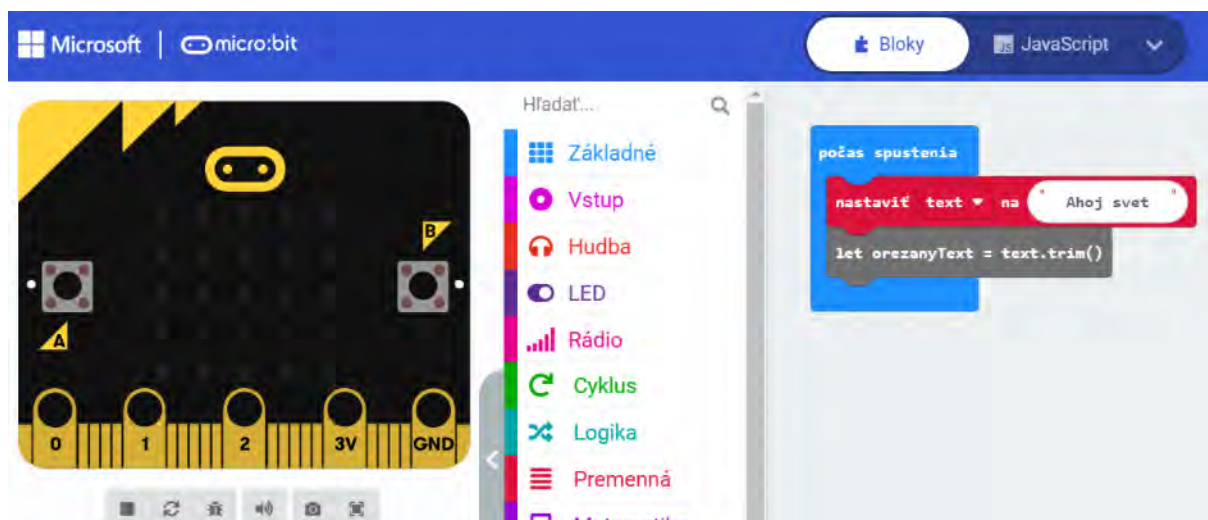
- Tento príkaz porovná dva textové reťazce a vráti hodnotu true alebo false.
- Použitie: Na porovnanie dvoch textových reťazcov.
- Výsledok: true

Príklad:

Obrázok 132 Príklad na porovnanie dvoch textových reťazcov

Trim textu:

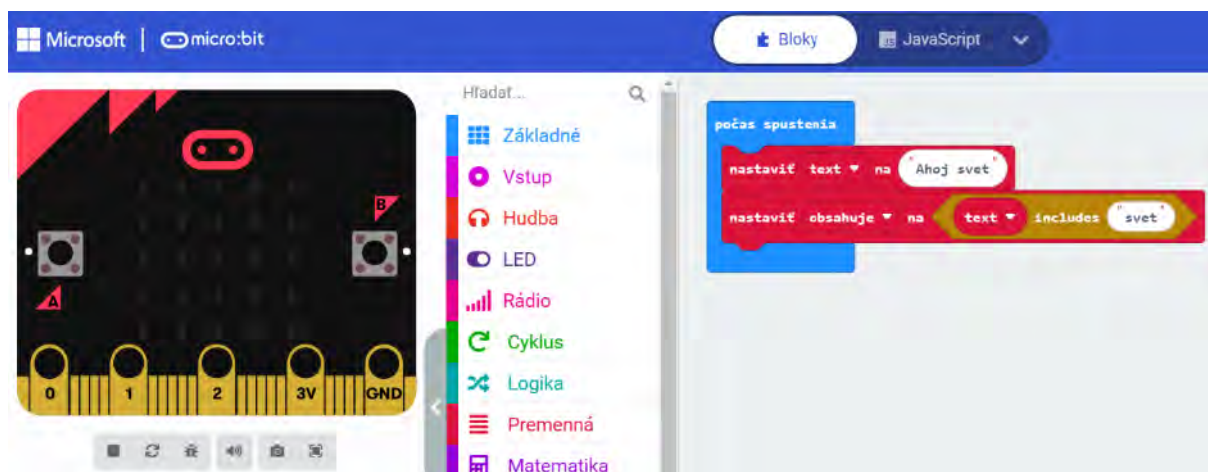
- Tento príkaz odstráni medzery z koncov textového reťazca.
- Použitie: Na odstránenie nepotrebných medzier z textu.
- Výsledok: "Ahoj svet"

Príklad:

Obrázok 133 Príklad na odstránenie nepotrebných medzier z textu

Kontrola, či text obsahuje:

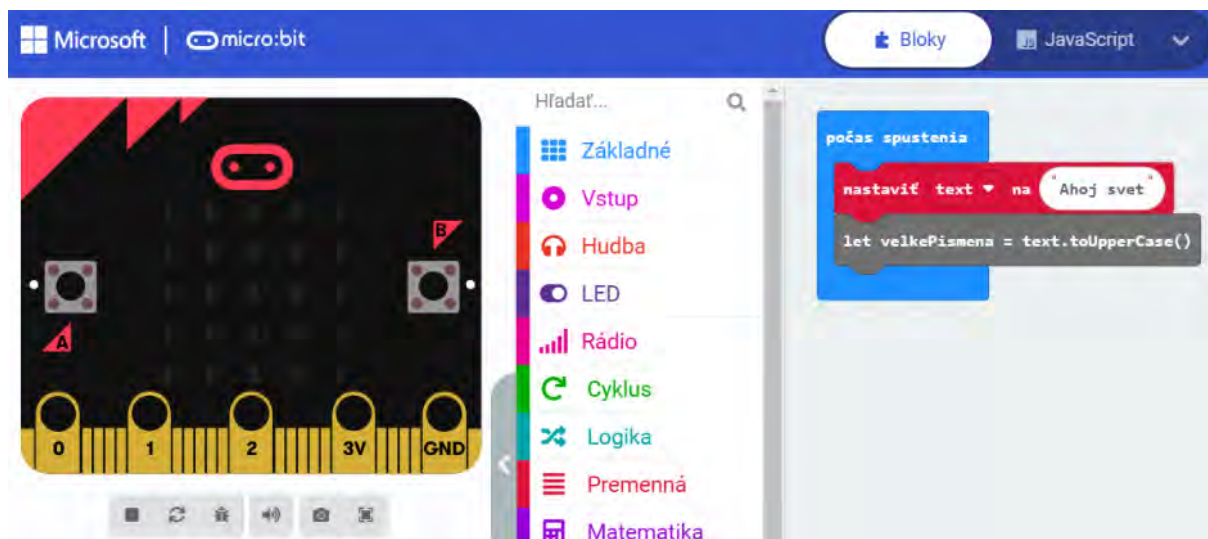
- Tento príkaz skontroluje, či textový reťazec obsahuje určitý podreťazec.
- Použitie: Na zistenie prítomnosti určitého textu v reťazci.
- Výsledok: true

Príklad:

Obrázok 134 Príklad na zistenie prítomnosti určitého textu v reťazci

Zmeniť na veľké písmená:

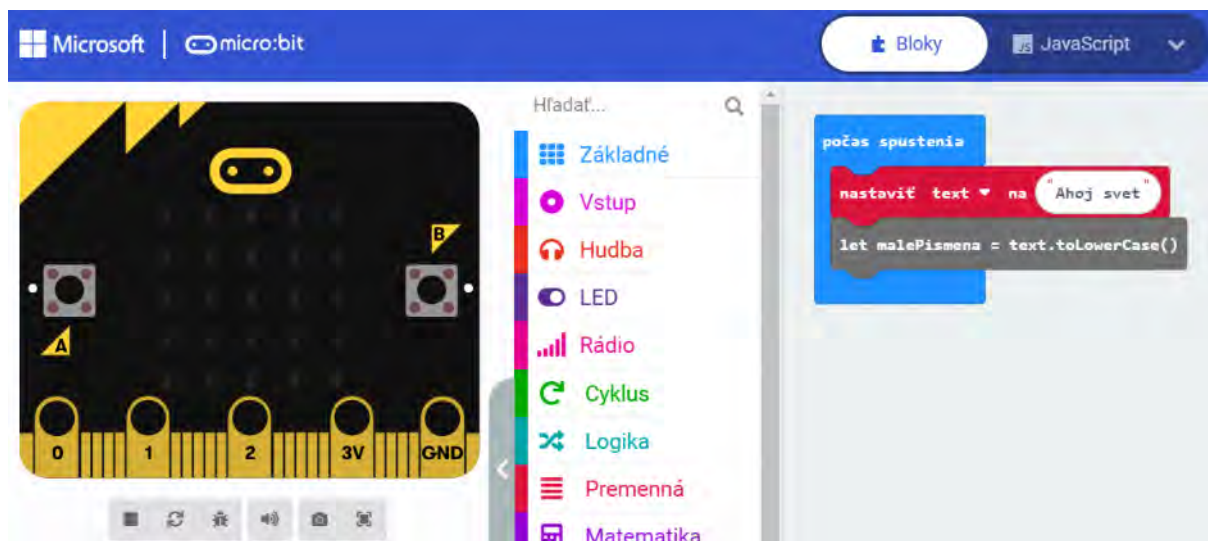
- Tento príkaz premení celý textový reťazec na veľké písmená.
- Použitie: Na zmenu textu na veľké písmená.
- Výsledok: "AHOJ SVET"

Príklad:

Obrázok 135 Príklad na zmenu textu na veľké písmená

Zmeniť na malé písmená:

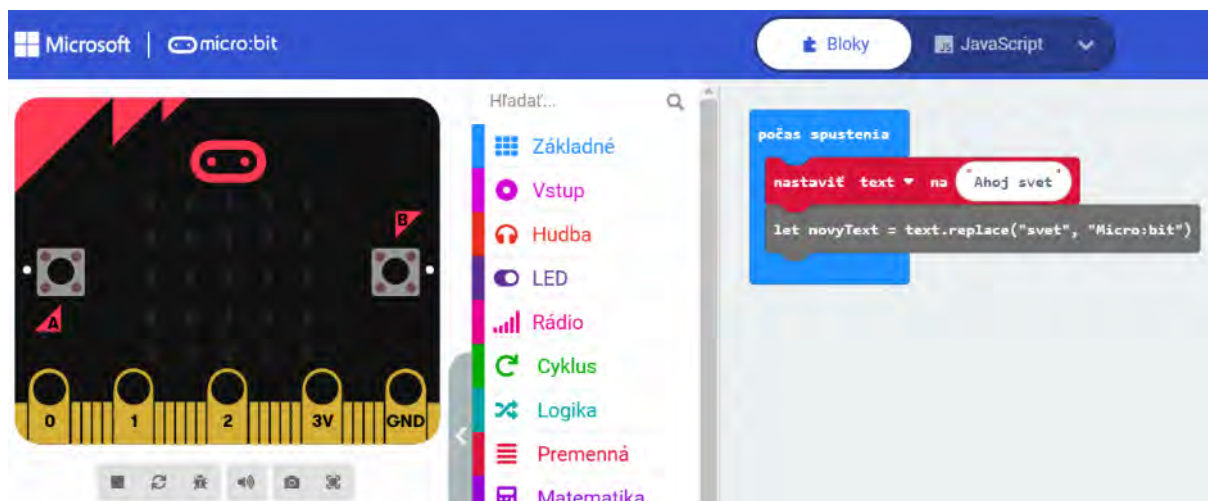
- Tento príkaz premení celý textový reťazec na malé písmená.
- Použitie: Na zmenu textu na malé písmená.
- Výsledok: "ahoj svet"

Príklad:

Obrázok 136 Príklad na zmenu textu na malé písmená

Nahradiť text:

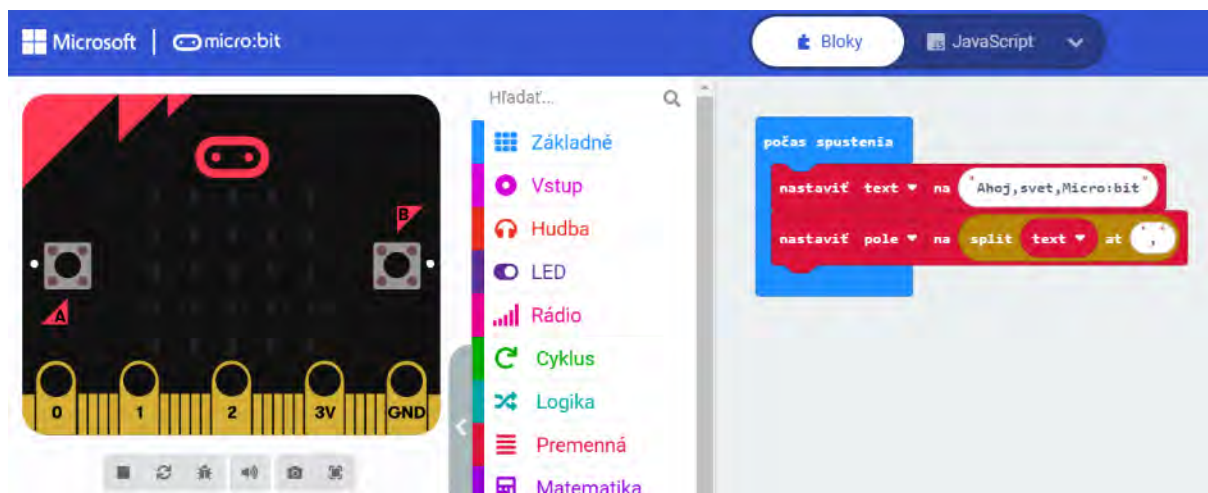
- Tento príkaz nahradí časť textového reťazca iným textom.
- Použitie: Na nahradenie určitého textu iným textom v reťazci.
- Výsledok: "Ahoj Micro:bit"

Príklad:

Obrázok 137 Príklad na nahradenie určitého textu iným textom v reťazci

Rozdeliť text:

- Tento príkaz rozdelí textový reťazec na pole (array) podľa určitého oddeľovača.
- Použitie: Na rozdelenie textu na jednotlivé časti.
- Výsledok: ["Ahoj", "svet", "Micro:bit"]

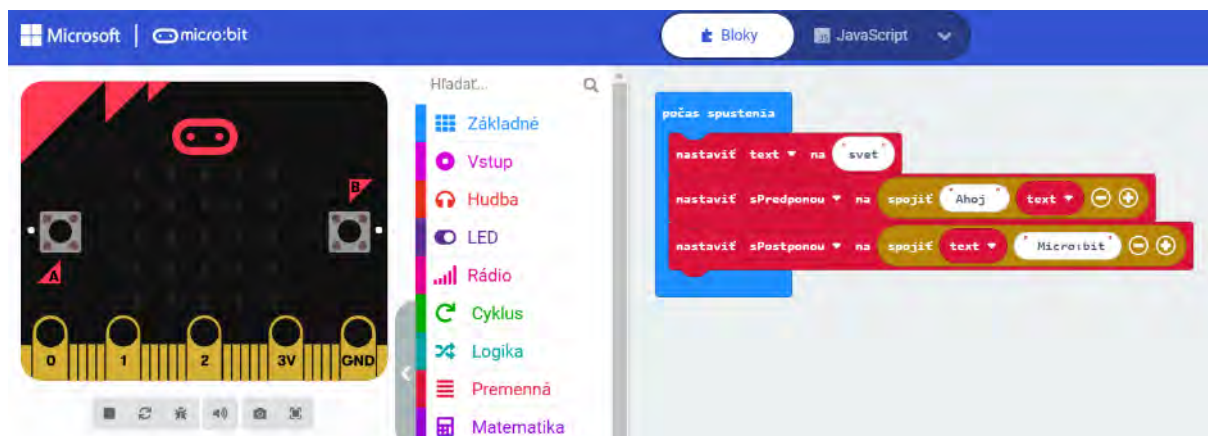
Príklad:

Obrázok 138 Príklad na rozdelenie textu na jednotlivé časti

Pridať predponu/postponu:

- Tento príkaz pripojí text na začiatok alebo koniec iného textového reťazca.
- Použitie: Na pripojenie textu pred alebo po inom texte.
- Výsledok: "Ahoj svet", "svet Micro:bit"

Príklad:

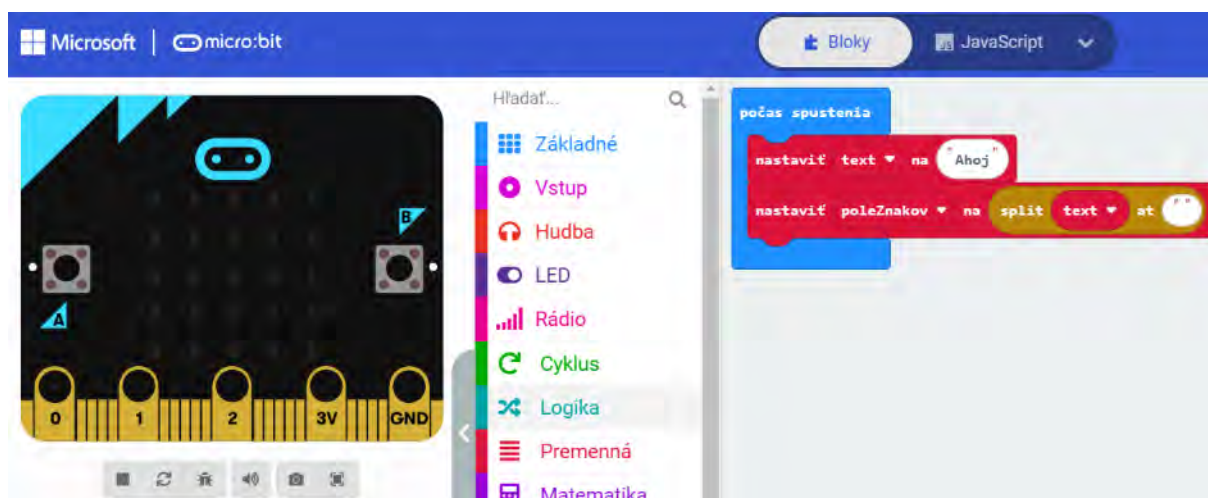


Obrázok 139 Príklad na pripojenie textu pred alebo po inom texte

Text ako pole znakov:

- Tento príkaz premení textový reťazec na pole jednotlivých znakov.
- Použitie: Na rozdelenie textu na jednotlivé znaky.
- Výsledok: ["A", "h", "o", "j"]

Príklad:



Obrázok 140 Príklad rozdelenie textu na jednotlivé znaky

14 Kolíky

Mikrokontrolér BBC micro:bit má na svojom okraji tzv. edge connector – súbor kontaktných plôch, ktoré slúžia ako piny (alebo „kolíky“) na pripojenie externých zariadení. Tieto piny sú rozdelené na digitálne, analógové, napájacie a komunikačné. Ich počet a funkcia závisí od verzie micro:bitu (v1 alebo v2), ale základné rozdelenie ostáva podobné.

Základné rozdelenie pinov micro:bitu

1. Digitálne piny - Slúžia na čítanie alebo zapisovanie digitálnych hodnôt (0 alebo 1, teda LOW alebo HIGH). V mikrokontroléri BBC micro:bit možno ako digitálne piny použiť napr.: P0, P1, P2 – hlavné dostupné piny. Ďalšie piny: P3–P20 (sú menšie, používajú sa pri rozšírených zapojeniach)

2. Analógové piny - Umožňujú čítanie alebo zapisovanie analógových hodnôt (napr. 0 až 1023 pri čítaní). Mnohé digitálne piny majú aj analógové funkcie (tzv. PWM – Pulse Width Modulation). Napríklad: P0, P1, P2 podporujú aj analog read a analog write. Pulse Width Modulation (skrátene PWM, po slovensky modulácia šírky pulzu) je technika, ktorou mikrokontrolér, ako napríklad BBC micro:bit, vytvára dojem analógového výstupu, hoci v skutočnosti pracuje iba s digitálnym signálom – teda stavmi zapnuté (HIGH) a vypnuté (LOW). Základom PWM je rýchle prepínanie výstupného pinu medzi stavom zapnuté a vypnuté v pravidelných cykloch. Každý takýto cyklus má určitú dĺžku (tzv. periódu) a v rámci nej je signál po určitý čas zapnutý. Práve pomer tohto času k celej dĺžke cyklu sa nazýva **strieda** (angl. duty cycle). Udáva sa v percentách a určuje, akú časť cyklu je signál v stave HIGH. Napríklad, ak je strieda 50 %, znamená to, že polovicu času je pin zapnutý a polovicu vypnutý. Výsledkom je, že zariadenie pripojené k pinu vníma priemerné napätie približne na úrovni polovice maximálneho napätia (napríklad pri 3,3 V to bude asi 1,65 V). PWM sa najčastejšie používa na riadenie jasů LED diód (čím vyššia strieda, tým jasnejšie svetlo), rýchlosti otáčania motorov alebo generovanie tónov v bzučiakoch. Na micro:bit-e je možné PWM využiť napríklad cez piny P0, P1 alebo P2, ktoré podporujú aj analógový výstup. Táto technika je veľmi užitočná v prípadoch, kde nie je k dispozícii skutočný analógový výstup, ale je potrebné simulovať plynulé riadenie zariadení pomocou digitálneho riadenia.

Kľúčové pojmy:

- **Perióda (cyklus):** čas jedného úplného „vlnového“ cyklu (zapnuté + vypnuté).
- **Frekvencia (Hz):** koľko cyklov sa uskutoční za sekundu.
- **Strieda (duty cycle):** percento času, počas ktorého je výstup v stave **HIGH** (napätie je zapnuté).

Príklad striedy:

Strieda	Význam	Výstupné napätie (priemer)
0 %	stále vypnuté (LOW)	0 V
50 %	polovicu času zapnuté	cca 1,65 V
100 %	stále zapnuté (HIGH)	3,3 V

3. Napájacie piny - 3V – v skutočnosti je výstupné napájanie 3.3 V pre externé zariadenia. GND (Ground) – zem.

4. Komunikačné piny - Na pokročilé pripojenie cez I2C, SPI alebo UART:

- o P5–P11: sú zdieľané s LED displejom, tlačidlami a inými funkciami.
- o Príklad: P6, P7, P8 – môžu slúžiť na SPI alebo I2C komunikáciu.
- o P13 (SCK), P14 (MISO), P15 (MOSI) – používané na SPI komunikáciu.
- o P19 (SCL), P20 (SDA) – I2C komunikácia.

Praktická poznámka

P0, P1, P2 – sú veľké kruhové otvory vhodné aj na pripojenie krokosvoriek, ideálne na prácu so žiakmi. Ostatné piny sú malé a vyžadujú špeciálny adaptér alebo konektor.

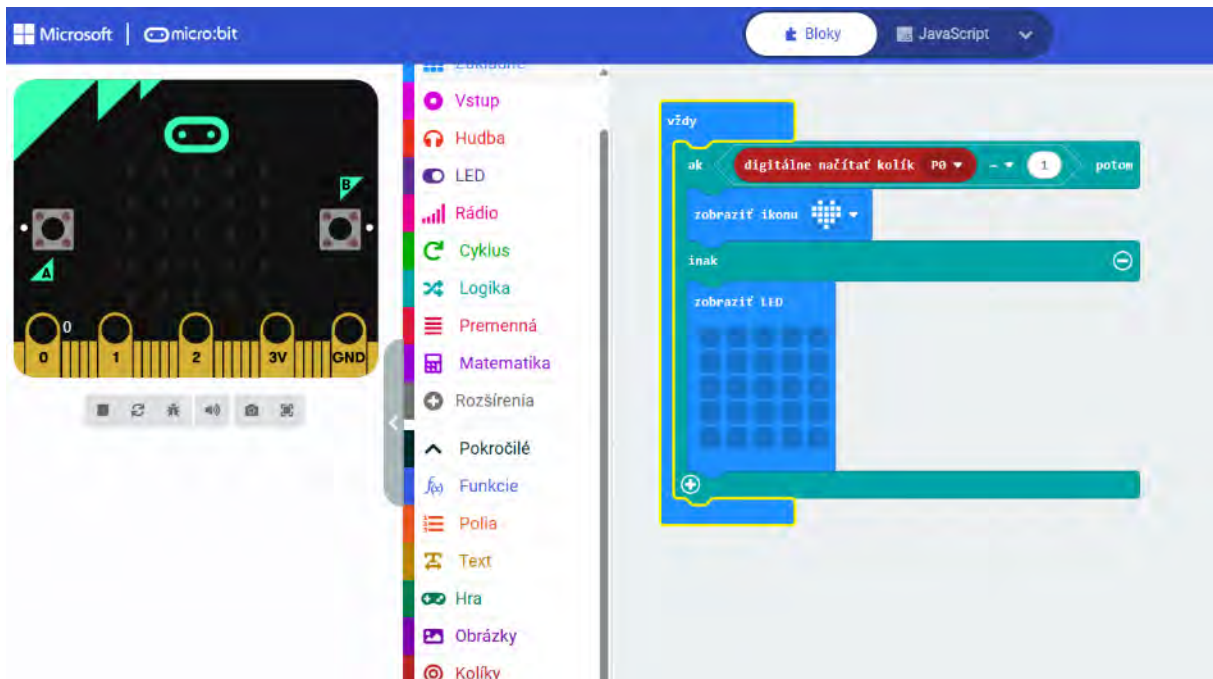
Príkaz „digitálne načítať kolík P0“

Príkaz „digitálne načítať kolík P0“ slúži na zistenie, aký digitálny signál (stav) je aktuálne prítomný na zvolenom vstupe – v tomto prípade na kolíku P0. Tento príkaz umožňuje zistiť, či je na pine elektrické napätie (HIGH – 1) alebo nie je (LOW – 0).

Ako to funguje?

1. Čítanie hodnoty: Príkaz načíta stav zvoleného pinu – výsledkom je číslo 0 alebo 1.
2. Reakcia na vstup: Na základe získanej hodnoty môžeme v programe rozhodnúť, čo micro:bit urobí (napr. rozsvieti LED, prehrá tón, zobrazí ikonu).

Príklad:



Obrázok 141 Príkaz „digitálne načítať kolík P0“

Ak chcete, aby micro:bit zobrazil srdiečko, keď je na kolíku P0 pripojený signál použite nasledujúci blok kódu:

Po spustení tohto kódu bude micro:bit neustále sledovať stav pinu P0. Ak zistí, že je zapnutý (digitálna hodnota 1), zobrazí srdce. Ak nie, obrazovka zostane prázdna.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke makecode.microbit.org.
2. Vložte do svojho programu blok „digitálne načítať kolík P0“ do podmienky „ak...inak“.
3. Pripojte napríklad tlačidlo medzi P0 a GND.
4. Spustíte program a stlačíte tlačidlo – micro:bit zobrazí obrázok podľa toho, či je pin aktívny.

Tento príkaz je veľmi užitočný na reakciu na externé zariadenia, ako sú tlačidlá, senzory alebo iné digitálne vstupy.

Príkaz „digitálne zapísať kolík“

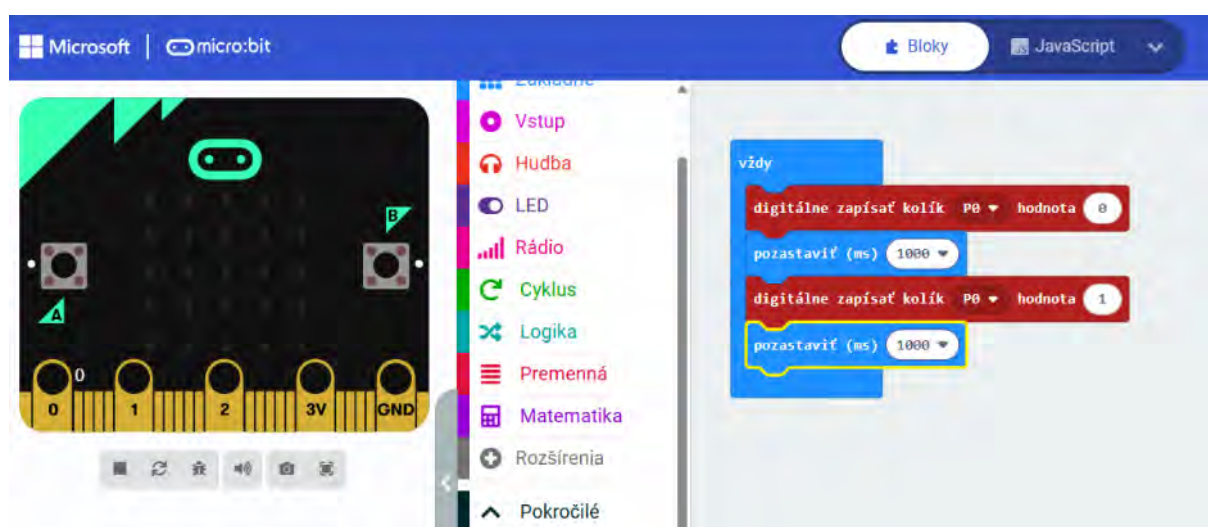
Príkaz „digitálne zapísať kolík“ slúži na nastavenie digitálneho výstupu na konkrétnom pine micro:bitu. Digitálne hodnoty sú len dve – 0 (vypnuté) a 1 (zapnuté), čo umožňuje jednoducho zapínať a vypínať napríklad LED diódu, bzučiak alebo iné zariadenia pripojené k micro:bitu.

Ako to funguje?

1. Výber kolíka: Vyberiete konkrétny pin (napr. P0, P1, P2), na ktorý je pripojené výstupné zariadenie.
2. Digitálny zápis: Nastavíte digitálnu hodnotu – 0 (nízka) alebo 1 (vysoká).
3. Ovládanie zariadenia: Podľa zvolenej hodnoty sa zariadenie zapne alebo vypne.

Príklad: Pravidelné blikanie LED diódy na pine P0

Ak chcete, aby LED dióda pravidelne blikala na kolíku P0, použijete nasledujúci blok kódu:



Obrázok 142 Príkaz „digitálne zapísať kolík“

Vyskúšajte si to!

1. Otvorte MakeCode editor.
2. Vložte príkaz „digitálne zapísať kolík“ do nekonečnej slučky (forever).
3. Pripojte LED diódu (s odporom) medzi pin P0 a GND na micro:bite.
4. Nahrajte program do micro:bitu.
5. Sledujte, ako dióda pravidelne bliká.

Tento príkaz je veľmi užitočný na jednoduché ovládanie hardvérových komponentov cez digitálne výstupy na micro:bite.

Príkaz „analogovo čítať kolík“

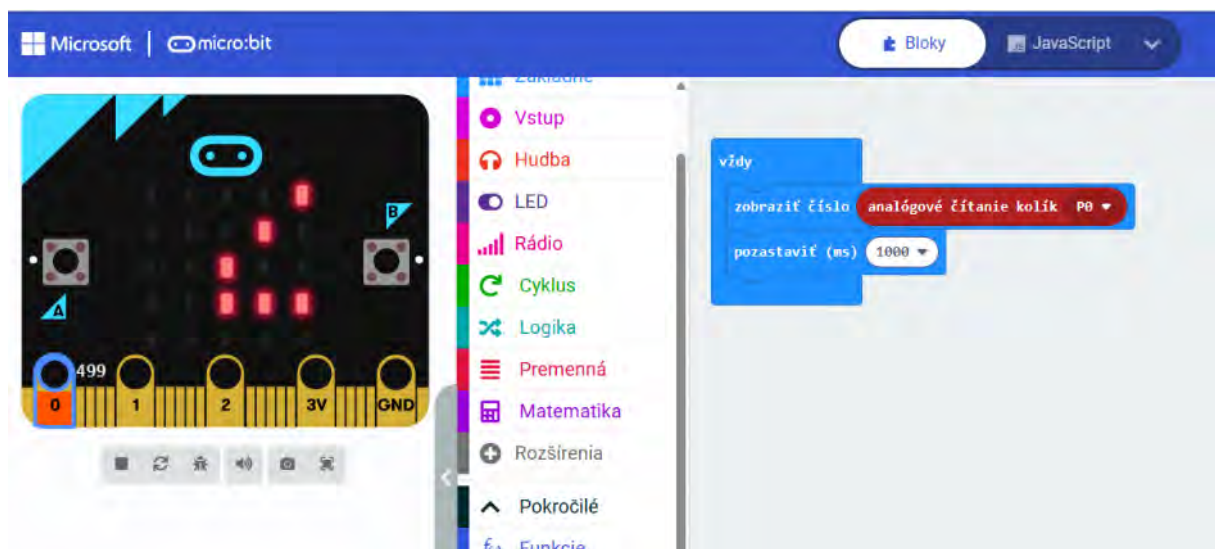
Príkaz „analogovo čítať kolík“ slúži na čítanie **analogovej hodnoty** z konkrétneho vstupného kolíka micro:bitu. Na rozdiel od digitálneho čítania (0 alebo 1), analogové čítanie vracia hodnoty v rozsahu **od 0 do 1023**, čo umožňuje merať napríklad úroveň svetla, polohu potenciometra alebo iné fyzikálne veličiny.

Ako to funguje?

1. **Výber kolíka:** Vyberiete kolík (napr. **P0, P1, P2**), na ktorom je pripojený analogový snímač (napr. svetelný senzor, potenciometer).
2. **Čítanie signálu:** micro:bit načíta napätie na tomto kolíku a prevedie ho na hodnotu od **0 (0 V) po 1023 (3,3 V)**.
3. **Zobrazenie alebo spracovanie:** Túto hodnotu môžete ďalej spracovať – zobraziť na displeji, porovnať s prahom alebo použiť na riadenie iných zariadení.

Príklad: Zobrazenie hodnoty z potenciometra pripojeného na kolík P1

Tento program každú pol sekundu zobrazí na LED displeji hodnotu, ktorú načítal z kolíka **P1**.



Obrázok 143 Príkaz „analogovo čítať kolík“

Vyskúšajte si to!

1. Otvorte MakeCode editor.
2. Pripojte analogový senzor (napr. potenciometer) medzi kolík **P1**, **GND** a **3V**.
3. V programe použite príkaz „analogovo čítať kolík“.
4. Nahrajte program do micro:bitu.
5. Sledujte hodnoty zobrazované na displeji podľa polohy potenciometra.

Tento príkaz je veľmi užitočný pri práci so snímačmi, ktoré poskytujú plynulo sa meniace údaje – napríklad pri meraní svetla, teploty alebo vlhkosti.

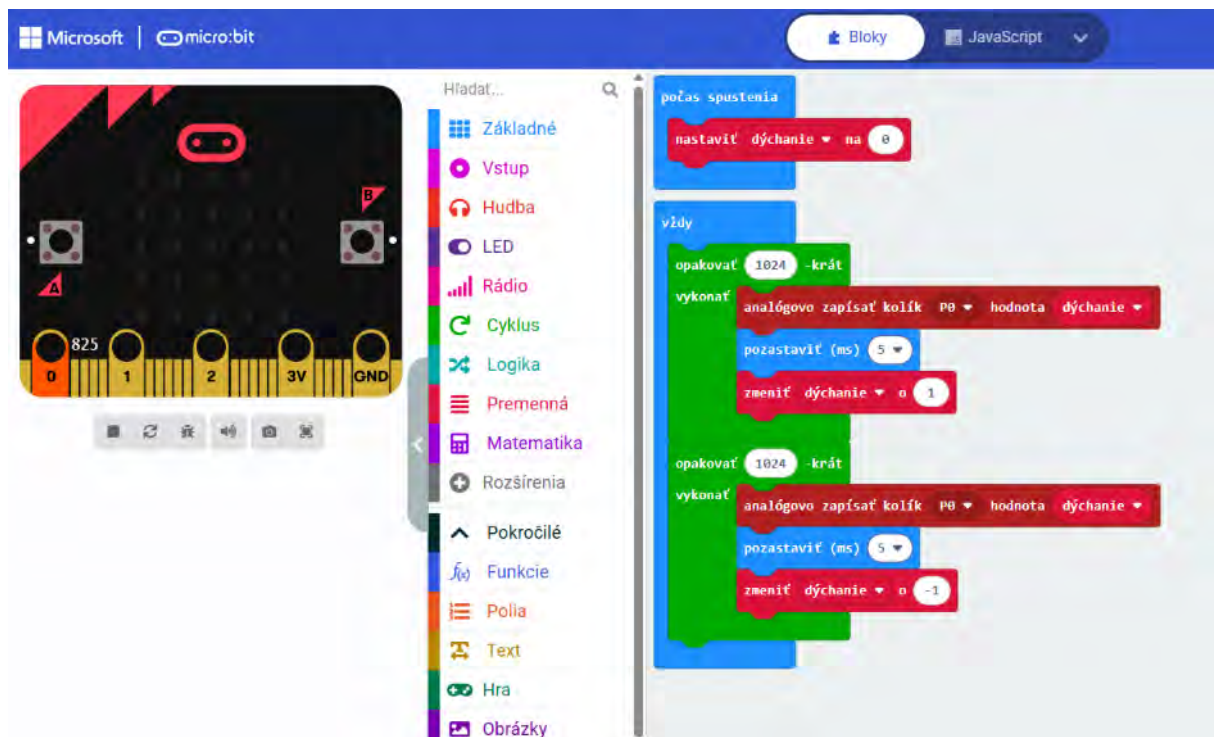
Príkaz „analogovo zapísať kolík“

Príkaz „analogovo zapísať kolík“ umožňuje poselať **analogovú (pulzne šírkovo modulovanú – PWM)** hodnotu na výstupný kolík micro:bitu. Na rozdiel od digitálneho zápisu (iba 0 alebo 1), analogový zápis umožňuje nastaviť výstup v rozsahu **od 0 do 1023**, čo umožňuje napríklad riadiť **jas LED diódy** alebo **rýchlosť motora**.

Ako to funguje?

1. **Výber kolíka:** Zvolíte výstupný pin (napr. **P0, P1, P2**).
2. **Nastavenie hodnoty:** Zvolíte hodnotu v rozsahu **0 – 1023**, kde:
 - **0** znamená žiadne napätie (LED zhasnutá),
 - **1023** znamená plné napätie (LED svieti naplno),
 - stredné hodnoty určujú stredný jas (napr. 512 = polovica jasu).
3. **PWM výstup:** micro:bit používa techniku PWM na vytvorenie efektu plynulého napätia.

Príklad: Plynulé zosvetľovanie a stmavovanie LED diódy na kolíku P0



Obrázok 144 Príkaz „analogovo zapísať kolík“

Tento program postupne zvyšuje a znižuje jas LED diódy pripojenej na **P0**, čím vytvára efekt "dýchania".

Vyskúšajte si to!

1. Otvorte MakeCode editor.
2. Pripojte LED diódu (s odporom 220 Ω) medzi kolík **P0** a **GND**.
3. Použite príkaz „analogovo zapísať kolík“ vo svojom programe.
4. Nahrajte program do BBC micro:bitu.
5. Sledujte, ako sa LED dióda plynulo zosvetľuje a stmavuje.

Tento príkaz je veľmi užitočný pre **jemné riadenie** výstupov – napríklad jas osvetlenia, rýchlosti ventilátora alebo intenzity zvuku.

15 Sériové rozhranie v blokovom programovaní micro:bit

Sériové rozhranie (UART – Universal Asynchronous Receiver/Transmitter) umožňuje micro:bitu odosielať a prijímať dáta prostredníctvom digitálnych signálov. V blokovom programovaní MakeCode sa využíva na komunikáciu medzi micro:bitom a počítačom alebo inými zariadeniami, napríklad senzormi, modulmi či inými mikrokontrolérmi.

Vlastnosti sériového rozhrania:

- Asynchrónna komunikácia – Dáta sa prenášajú postupne, bit po bite, bez potreby synchronizačného signálu.
- Prenosová rýchlosť (Baud Rate) – Udáva, koľko bitov sa preniesie za sekundu (napr. 9600, 115200 baud).
- Dátový formát – Prenášané dáta môžu byť vo forme textu, čísel alebo binárnych hodnôt.
- Jednosmerná alebo obojsmerná komunikácia – Micro:bit môže buď len odosielať, len prijímať, alebo vykonávať oboje.

Použitie sériového rozhrania:

- Debugging (ladenie programu) – Zobrazenie hodnôt premenných na počítači.
- Zber a vizualizácia údajov – Namerané dáta z micro:bitu (napr. teplota, svetlo) môžu byť odoslané na PC na spracovanie.
- Komunikácia s inými zariadeniami – Prepojenie micro:bitu s Arduino, Raspberry Pi alebo inými mikrokontrolérmi.

Sériová komunikácia je kľúčová pri práci s externými zariadeniami a umožňuje jednoduchý prenos údajov medzi rôznymi systémami.

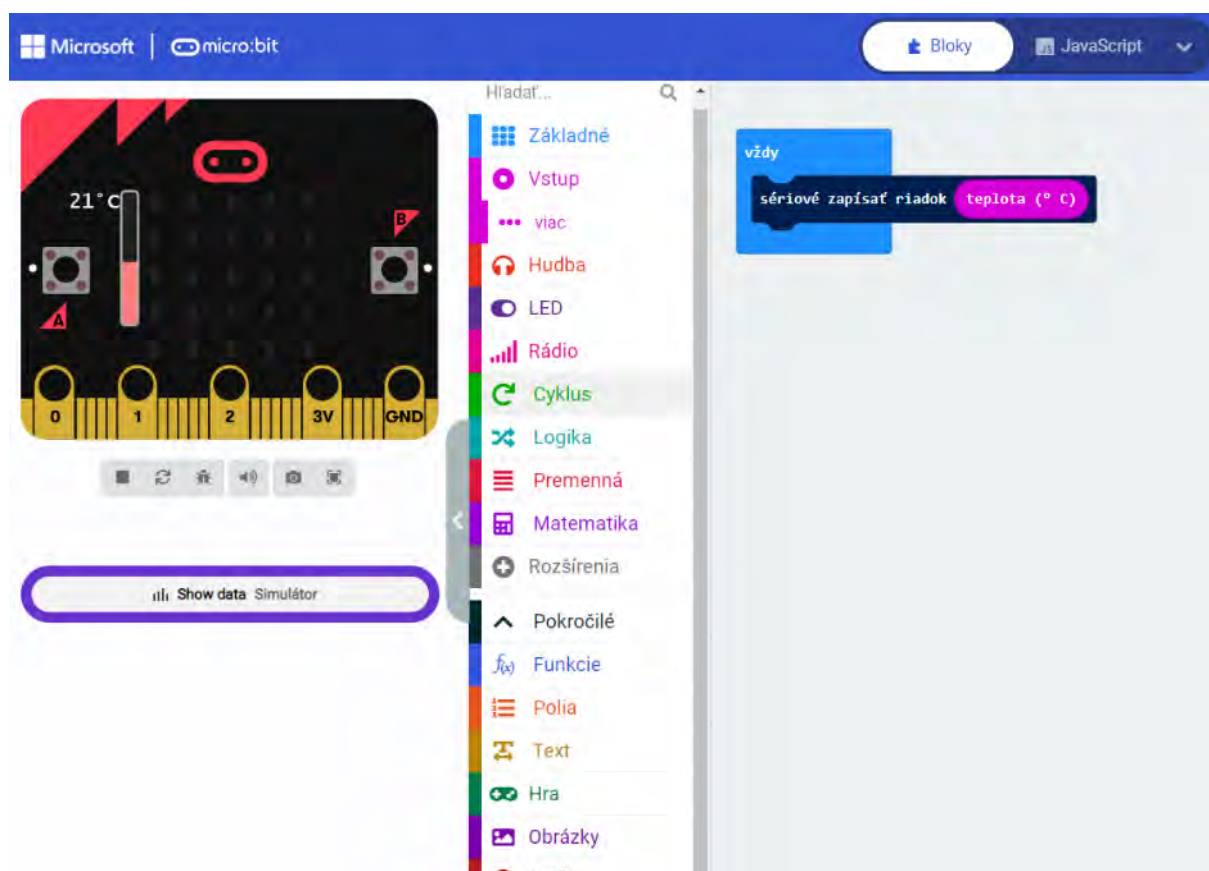
Príkaz "seriové zapísať riadok"

Príkaz "seriové zapísať riadok" textový reťazec, číslo alebo akúkoľvek hodnotu cez sériovú komunikáciu, pričom na koniec automaticky pridá nový riadok. Tento blok je užitočný pre komunikáciu medzi micro:bitom a počítačom alebo iným zariadením.

Ako to funguje?

Tento príkaz sa používa napríklad na odoslanie informácií, ako sú senzorové údaje, správy alebo výsledky programových výpočtov. Pomáha pri monitorovaní a debugovaní programu.

Príklad: Ako použiť príkaz "seriové zapísať riadok" v blokovom programe. V MakeCode editori si môžete otvoriť sekciu Sériové (Serial). Nájdete tam blok "seriové zapísať riadok". Do tohto bloku môžete vložiť text, čísla alebo iné hodnoty. **NaPríklad:**



Obrázok 145 Príkaz "seriové zapísať riadok"

Vyskúšajte si to!

1. Pripojte BBC micro:bit k počítaču cez USB.
2. Spustite program a v sériovom monitore MakeCode môžete sledovať, ako sa na obrazovke zobrazuje nameraná teplota.
3. Tento blok môžete kombinovať s inými blokmi, napríklad na odosielanie senzorových hodnôt v reálnom čase.

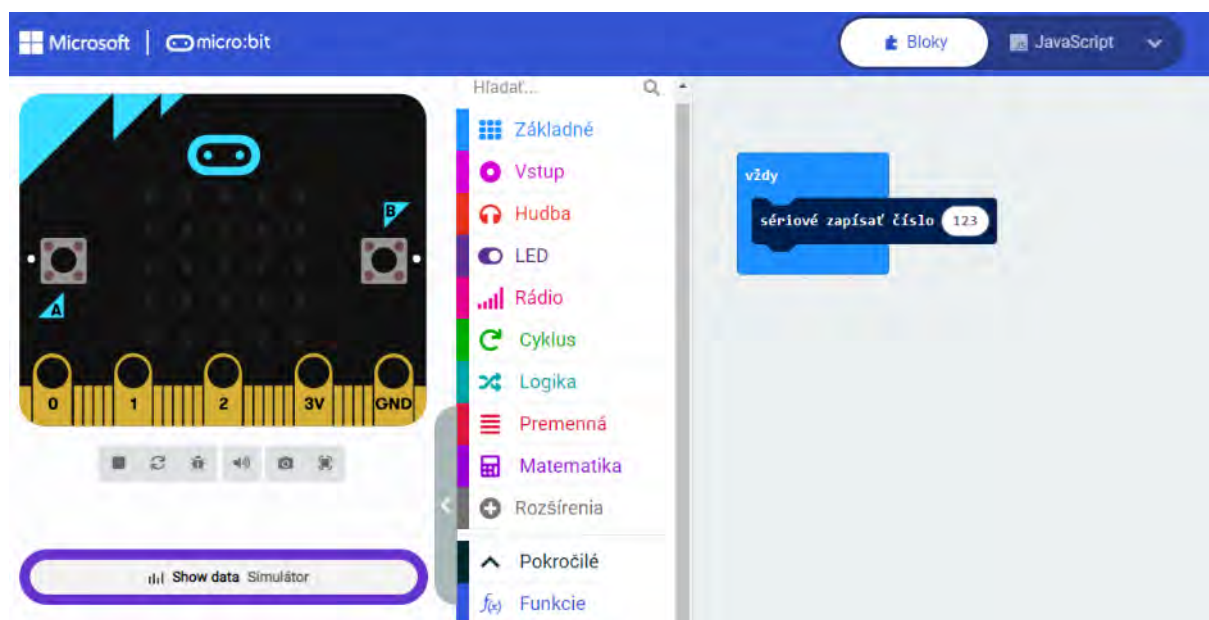
Príkaz "seriové zapísať číslo"

Príkaz "**seriové zapísať číslo**" sa používa na odoslanie čísla cez sériovú linku do pripojeného zariadenia, napríklad počítača. Tento príkaz umožňuje odosielať číselné hodnoty z micro:bitu, čo je užitočné na sledovanie hodnôt počas behu programu alebo pri prenose senzorových údajov.

Ako to funguje?

Odoslanie čísla: Príkaz) odošle číselnú hodnotu cez sériovú komunikáciu. Po odoslaní sa dáta zobrazia v sériovom monitore na pripojenom zariadení (počítači). **Príjem dát:** Ak je micro:bit pripojený k počítaču cez USB, prijaté číslo môžete sledovať v sériovom monitore alebo inom softvéri, ktorý umožňuje prijímať a spracovávať dáta zo sériovej linky.

Príklad: V blokovom programovaní vyberiete príkaz **sériovo zapísať číslo** a vložíte číslo **123**.



Obrázok 146 Príkaz "seriové zapísať číslo"

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Použite príkaz "**seriové zapísať číslo**"
3. Zadať číslo, napríklad **123**.
4. Pripojte micro:bit k počítaču cez USB a spustíte program.
5. Otvorte sériový monitor a sledujte, ako sa číslo **123** zobrazí na obrazovke.

Tento príkaz je veľmi užitočný, ak chcete sledovať hodnoty zo senzorov alebo výsledky výpočtov počas behu programu. Môžete ho použiť na zapisovanie číselných hodnôt v reálnom čase a jednoducho sledovať, ako sa váš program správa.

Príkaz "seriové zapísať hodnotu"

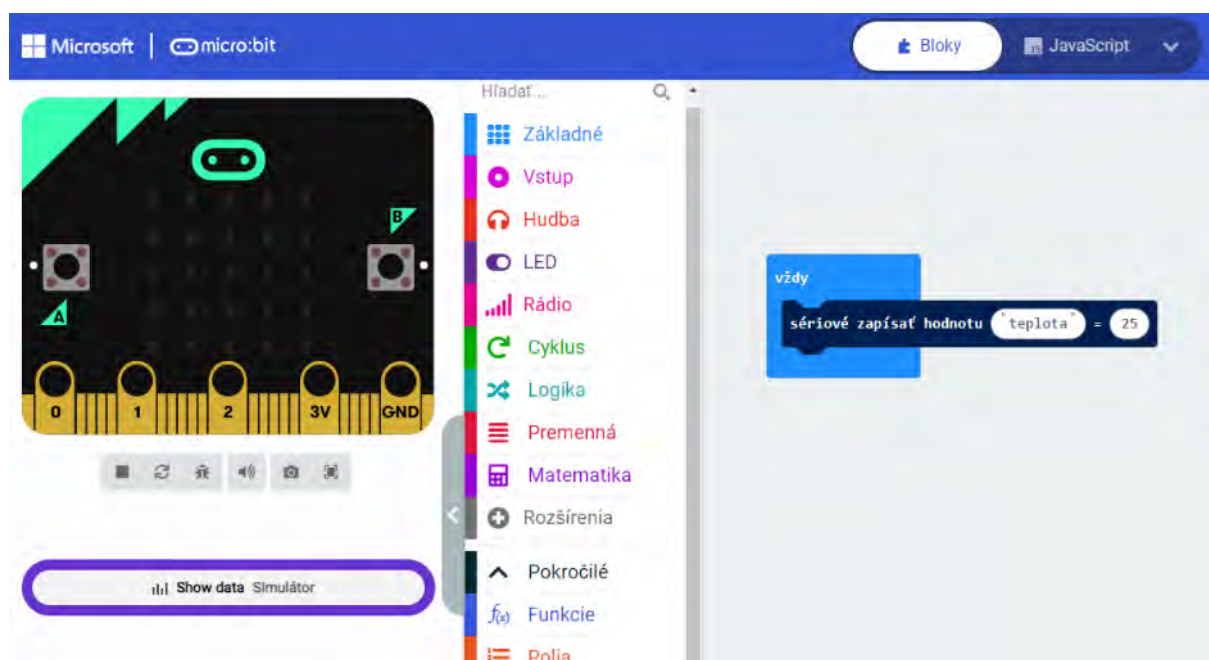
Príkaz "seriové zapísať hodnotu" slúži na odoslanie pomenovanej hodnoty cez sériovú linku. Tento príkaz je užitočný na odoslanie dvojíc dát, kde jedno je názov hodnoty (napríklad "teplota") a druhé je číselná hodnota (napríklad 25). Tento príkaz pomáha lepšie organizovať a interpretovať dáta prijaté zo sériovej komunikácie.

Ako to funguje?

Odoslanie pomenovanej hodnoty: Príkaz odošle dvojicu pozostávajúcu z názvu a príslušnej číselnej hodnoty. Napríklad, môžete odoslať názov "teplota" a číselnú hodnotu 25. Prijem dát: Na počítači alebo inom pripojenom zariadení môžete tieto údaje zobraziť v sériovom monitore, čo vám umožní vidieť nielen hodnotu, ale aj jej označenie.

Príklad:

V blokovom programovaní vyberiete príkaz serial write value, do prvého poľa napíšete text "teplota" a do druhého poľa číslo 25.



Obrázok 147 Príkaz "seriové zapísať hodnotu"

Vyskúšajte si to:

1. Otvorte MakeCode editor na stránke micro:bit.
2. Použite príkaz "seriové zapísať hodnotu" (v MakeCode je to blok "serial write value").
3. Vložte názov hodnoty, napríklad "teplota", a pridajte hodnotu, napríklad 25.
4. Pripojte micro:bit k počítaču cez USB a spustite program.
5. Otvorte sériový monitor a sledujte, ako sa zobrazí text "teplota: 25".

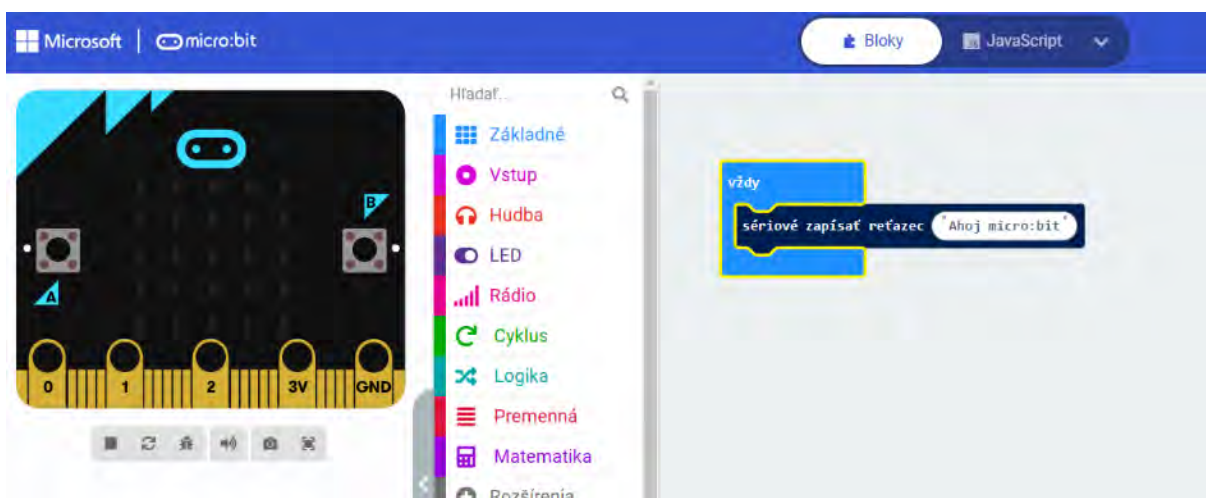
Tento príkaz je ideálny pre projekty, kde potrebujete odosielať viacero rôznych hodnôt s popismi, napríklad pri sledovaní senzorov (teplota, svetlo, vlhkosť atď.). Umožňuje jednoducho identifikovať, ktoré hodnoty sa práve odosielať.

Príkaz: "seriové zapísať reťazec"

Popis: Príkaz "seriové zapísať reťazec" slúži na odoslanie ľubovoľného textového reťazca cez sériovú linku. Tento príkaz je užitočný na prenos textových správ medzi mikrokontrolérom (napr. micro:bit, Arduino) a počítačom, alebo na zobrazovanie informácií v sériovom monitore

Ako to funguje?

1. **Odoslanie textového reťazca:** Príkaz odošle ľubovoľný textový reťazec, napríklad "Ahoj, micro:bit!".
2. **Príjem reťazca:** Na strane prijímača sa text zobrazí v sériovom monitore alebo sa použije na ďalšie spracovanie.
3. **Použitie v sériovom monitore:** Ak text odosielate z počítača, môžete ho vidieť na výstupe terminálu.



Obrázok 148 Príkaz: "seriové zapísať reťazec"

Vyskúšajte si to:

1. Otvorte MakeCode editor.
2. V sekcii **Rozšírenia** pridajte **serial** (ak nie je dostupné).
3. Vyberte "serial write line".
4. Do poľa napíšte text "Ahoj, micro:bit!".
5. Nahrajte program do Micro:bitu a otvorte sériový monitor.

Príklad: Ak do MakeCode zadáte "Dátová komunikácia OK", bude sa táto správa zobrazovať v sériovom monitore pri každom spustení programu.

Kedy použiť?

- Na odosielanie textových správ do počítača alebo iného zariadenia
- Na ladenie a testovanie sériovej komunikácie
- Na zobrazovanie diagnostických správ v sériovom monitore

Tento príkaz umožňuje jednoduchý prenos textu cez sériovú linku a je vhodný pre rôzne aplikácie, ako napríklad logovanie údajov, zobrazovanie správ alebo ladenie programov.

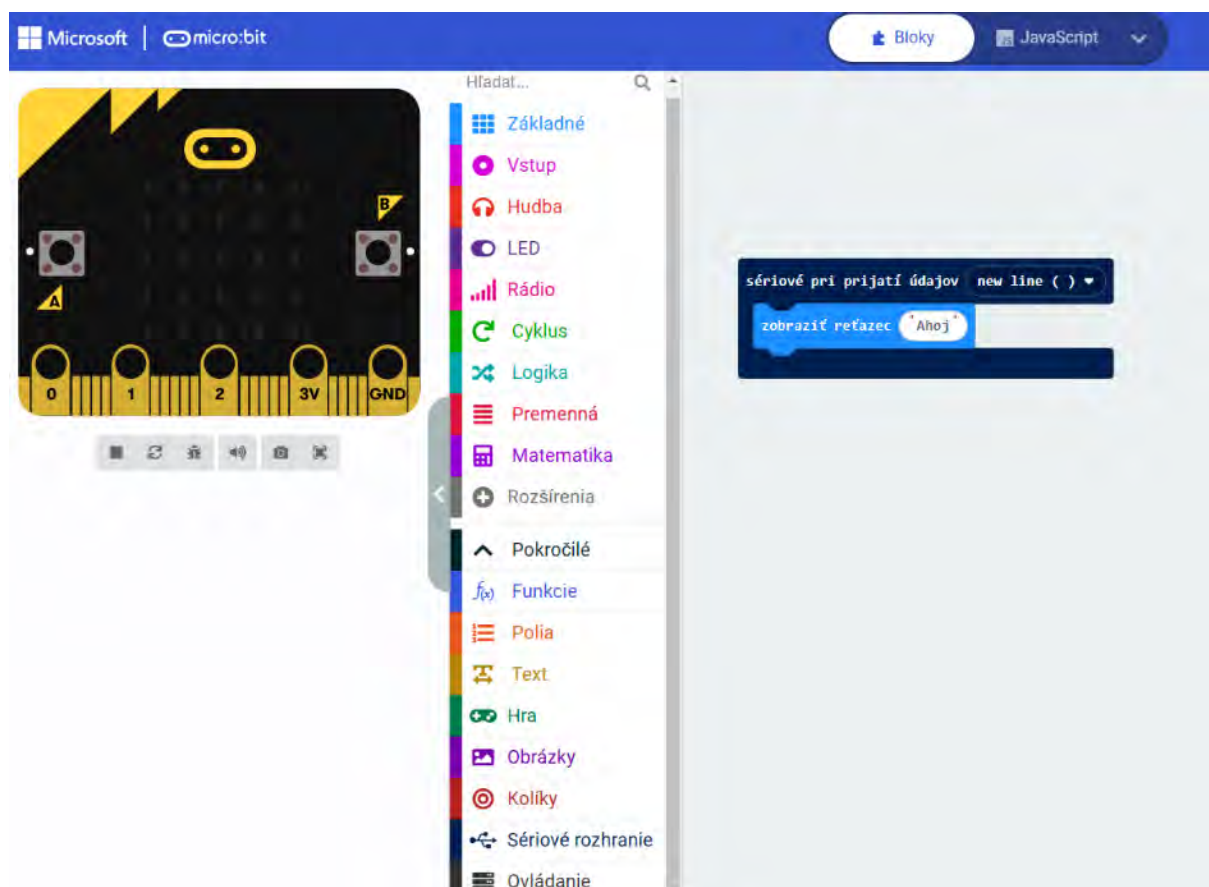
Príkaz: "seriové po prijatí údajov"

Popis:

Príkaz "**seriové po prijatí údajov**" slúži na spracovanie údajov, ktoré sú prijaté cez sériovú linku. Tento príkaz je užitočný na čítanie dát zo senzorov, prijímanie príkazov z počítača alebo komunikáciu medzi zariadeniami (napr. micro:bit a Arduino).

Ako to funguje?

1. **Príjem údajov:** Mikrokontrolér alebo počítač neustále čaká na údaje prichádzajúce cez sériový port.
2. **Spracovanie údajov:** Po prijatí údajov ich môžeme analyzovať a použiť na vykonanie akcie (napr. ak príde príkaz "LED_ON", zapne sa LED).
3. **Zobrazenie údajov:** Prijaté údaje môžu byť vypísané v sériovom monitore alebo použité na ďalšie spracovanie.



Obrázok 149 Príkaz: "seriové po prijatí údajov"

Použitie v micro:bit MakeCode

1. Otvorte MakeCode editor.
2. V sekcii **Rozšírenia** pridajte **serial** (ak nie je dostupné).
3. Použite blok "**on serial received**" (pri prijatí údajov).
4. Nastavte premennú na uloženie prijatých údajov.
5. Použite funkciu "**show string**", aby sa prijaté údaje zobrazili na LED displeji.

Príklad:

- Ak micro:bit prijme cez sériovú linku text "HELLO", zobrazí ho na LED displeji.
- Ak prijme "LED_ON", môže zapnúť LED na porte P0.

Kedy použiť?

Na čítanie údajov zo senzorov (teplota, vlhkosť, tlak)

Na prijímanie príkazov z počítača a ich vykonanie

Na zobrazovanie textových správ prijatých cez sériovú komunikáciu

Tento príkaz je ideálny pre projekty, kde je potrebné prijímať údaje a dynamicky reagovať na ich obsah.

Príkaz: sériové presmerovať na

Tento príkaz sa v blokovom programovaní micro:bit používa na presmerovanie sériovej komunikácie na konkrétne piny TX (transmit) a RX (receive) a nastavenie prenosovej rýchlosti (baud rate).

Vysvetlenie parametrov:

- TX pin – Odosiela údaje z micro:bitu do externého zariadenia.
- RX pin – Prijíma údaje z externého zariadenia do micro:bitu.

Prenosová rýchlosť – Najčastejšie hodnoty sú 9600, 115200, ale môžu byť aj iné podľa požiadaviek zariadenia.

Použitie:

- Prepojenie micro:bitu s iným mikrokontrolérom (napr. Arduino).
- Komunikácia s modulmi (napr. GPS, Bluetooth, Wi-Fi).
- Odosielanie a prijímanie dát z externých senzorov.

Tento príkaz umožňuje micro:bitu komunikovať s externými zariadeniami cez sériovú linku mimo USB pripojenia.

Príkaz: Sériové presmerovať na USB

Tento príkaz sa v blokovom programovaní micro:bit používa na presmerovanie sériovej komunikácie na USB port. Znamená to, že všetky dáta odosielané a prijímané cez sériové rozhranie budú komunikovať priamo s počítačom cez USB pripojenie.

Funkcia príkazu:

- Zabezpečí, že micro:bit bude posielat' a prijímať sériové dáta cez USB port.
- Dáta je možné zobrazit' v sériovom monitore
- Je užitočný pri debugingu (ladení programu) a zobrazovaní hodnôt premenných.

Použitie:

- Monitorovanie senzorov – zobrazovanie hodnôt (napr. teplota, svetlo, zrýchlenie).
- Debugging programu – kontrola správneho chodu programu.
- Odosielanie údajov z micro:bitu do počítača na ďalšie spracovanie.

Tento príkaz je štandardným spôsobom, ako sledovať výstupy micro:bitu v reálnom čase cez USB pripojenie.

Záver

Keď si sa pustil do tejto učebnice, možno si si kládol otázky: „Načo mi bude BBC micro:bit?“ alebo „Zvládnem vôbec programovanie?“ Dnes, keď si prešiel všetkými kapitolami, už vieš, že odpoveď na obe znie jednoznačne: „Áno.“

Počas svojej cesty si spoznal nielen základy mikropočítača BBC micro:bit, ale aj samotné princípy toho, ako sa programuje, ako sa dá pomocou jednoduchých blokov vytvoriť niečo zmysluplné, a ako môže technológia zjednodušiť alebo spestriť každodenný život. Ukázali sme si, ako sa ovláda displej z LED diód, ako reagovať na pohyb či zvuk, ako vytvoriť interaktívny projekt, prenášať údaje cez rádio či sériové rozhranie – a hlavne ako rozmýšľať ako **tvorca technológií**, nie iba jej používateľ.

Možno si vytvoril vlastnú hru, semafor, počítadlo stlačení, hudobný nástroj alebo si len s radosťou sledoval, ako micro:bit zareaguje na tvoje príkazy. Nech už si dosiahol čokoľvek, dôležité je, že si objavil silu programovania a logického myslenia – nie ako niečo zložité, ale ako **nástroj, ktorý máš vo vlastných rukách**.

Čo si sa naučil?

Zhrňme si hlavné oblasti, ktoré si v tejto učebnici zvládol:

- pochopiť **fungovanie mikropočítača** a jeho súčiastok (LED, senzory, tlačidlá, piny),
- používať **MakeCode** – blokové programovanie cez prehliadač,
- vytvárať **animácie, efekty a reakcie** na vstupy zo senzora,
- používať **premenné, podmienky, cykly a funkcie** v programoch,
- komunikovať medzi zariadeniami pomocou **rádia alebo sériového rozhrania**,
- zapojiť svoju **kreativitu** do navrhovania vlastných riešení,
- využiť micro:bit na pozorovanie prírody, meranie hodnôt či zábavné projekty.

Zároveň si si precvičil:

- čítanie s porozumením a dodržiavanie postupu,
- riešenie problémov,
- tímovú spoluprácu a komunikáciu,
- organizáciu údajov (napr. do premenných, polí, grafov).

Ako môžeš pokračovať ďalej?

- Aj keď sa táto učebnica končí, tvoja cesta rozhodne nekončí. Teraz máš pred sebou niekoľko ciest, ako pokračovať:
- **Vymýšľaj vlastné projekty.** Skús naprogramovať hru, budík, počítadlo krokov alebo automatický zavlažovač.
- **Získaj nový hardware.** Pripoj k micro:bitu servomotor, displej, senzor vlhkosti, teploty alebo svetla – a rozšír možnosti, ktoré si sa naučil.

- **Prejdi na textové programovanie.** Skús JavaScript alebo Python priamo v prostredí MakeCode.
- **Zdieľaj svoju prácu.** Nahraj svoje projekty na školský web, ukáž ich doma, alebo ich prezentuj pred spolužiakmi.
- **Zapoji sa do súťaží alebo klubov.** Mnohé školy alebo mestá ponúkajú krúžky robotiky, programovania alebo elektroniky.

Technológia ako nástroj zmeny

Možno si si počas práce uvedomil, že mikropočítače nie sú len hračka. Dajú sa použiť aj na vážne veci: merať kvalitu ovzdušia, pomáhať ľuďom so zdravotným postihnutím, zvyšovať bezpečnosť, šetriť energiu alebo zlepšovať prostredie v triede. Záleží len na tvojej predstavivosti.

Dnešný svet potrebuje ľudí, ktorí sa neboja hľadať nové cesty, ktorí vedia programovať a chápu technológiu. Táto učebnica ti ponúkla vstupenku do tohto sveta. Ako ju využiješ, je na tebe.

Slová na rozlúčku

Technológia a programovanie nie sú výsadou dospelých. Aj ty môžeš byť **tvorcom** – nie len používateľom. Môžeš meniť veci okolo seba k lepšiemu. Môžeš vytvárať zariadenia, ktoré pomáhajú ľuďom. Môžeš byť súčasťou tímu, ktorý jedného dňa vytvorí niečo, čo zmení svet.

- Nezabúdaj:
- Každý veľký projekt sa začína malým krokom.
- Každý programátor raz prvýkrát napísal svoj prvý riadok kódu.
- Každý nápad stojí za to preskúmať.

Literatúra

Barrett, A. (2017). Collaborative learning with micro:bit: Enhancing teamwork skills in the classroom. *Educational Technology Research Journal*, 29(3), 67-79.

Brown, J. (2019). Developing creative thinking through micro:bit programming. *Computing in Schools*, 21(4), 112-118.

Nominet Trust. (2017). Micro:bit Educational Foundation Report. Dostupné z <https://microbit.org/media/docs/Nominet-Trust-Report.pdf>

Smith, T. (2018). Introduction to block-based programming with BBC micro:bit. *Journal of Educational Programming*, 15(1), 32-45.

Thompson, R. (2020). Using BBC micro:bit in STEM education: A practical guide. *STEM Education Quarterly*, 12(2), 45-53.

Názov: Mikrokontroléry v interdisciplinárnej edukácii 1

Autor: Mgr. Miroslav ŠEBO, PhD.
PaedDr. Erik Krajínčák, PhD.

Recenzent: prof. PaedDr. Jarmila Honzíková, Ph.D.
prof. PhDr.PaedDr. Jiří DOSTÁL, Ph.D.

Rok vydania: 2025

Vydanie: prvé

Rozsah: 165 strán

Cover design: Mgr. Miroslav ŠEBO, PhD.

Vydavateľ: UKF v Nitre

Financované z projektu KEGA 019UKF-4/2023 s názvom „**Mikrokontroléry v interdisciplinárnej edukácii**“

Všetky práva vyhradené. Toto dielo ani žiadnu jeho časť nemožno reprodukovat' bez súhlasu majiteľov práv.

ISBN 978-80-558-2331-7