

Martin Malý

Micro:bit krok za krokem

Praktický úvod do programování
a elektroniky

Edice CZ.NIC



MICRO:BIT KROK ZA KROKEM

Praktický úvod do programování a elektroniky

Martin Malý

Vydavatel:

CZ.NIC, z. s. p. o.

Milešovská 5, 130 00 Praha 3

Edice CZ.NIC

www.nic.cz

1. vydání, Praha 2023

Kniha vyšla jako 29. publikace v Edici CZ.NIC.

ISBN 978-80-88168-70-6

© 2023 Martin Malý

Toto autorské dílo podléhá licenci Creative Commons BY-ND 4.0

(<http://creativecommons.org/licenses/by-nd/4.0/>). Dílo však může být překládáno a následně šířeno v písemné či elektronické formě, na území kteréhokoliv státu, za předpokladu, že nedojde ke změně díla a i nadále zůstane zachováno označení autora a prvního vydavatele díla, sdružení CZ.NIC, z. s. p. o. Překlad může být šířen pod licencí CC BY-ND 4.0.

— Martin Malý

Micro:bit krok za krokem

Praktický úvod do programování a elektroniky

— Edice CZ.NIC

Poděkování

Poděkování

Před téměř dvěma lety mi napsal Oldřich Horáček, provozovatel „bastlířského“ e-shopu HWKITCHEN, že by možná nebylo od věci napsat knihu o microbitu. Že si to hodně kupují učitelé do škol, a ptají se po materiálech v češtině. V té době jsem dopsal poslední knihu své „elektronické trilogie“ a myslel jsem si, že tím moje osvětová činnost končí. Ale ten microbit mi stále vrtal v hlavě... Největší dík proto patří právě Oldřichu Horáčkovi, který mě na toto téma přivedl, stále mi ho jemně předhazoval a hodně mi pomohl se sběrem materiálu, s testováním a sháněním modulů, stavebnic a všeho okolo, a nakonec knihu laskavě přečetl a opřipomínkoval.

Děkuju všem lidem, kteří mi při psaní fandili. Všem čtenářům předchozích knih, co mi psali, že s nimi je začala elektronika zase bavit. Všem lidem, kteří mi psali, že se těší na pokračování. Všem těm, kteří četli rukopis a přispěli radami a názory. Všem těm, kteří mi dali ve chvílích pochybností podporou najevo, že to není úplně zbytečná práce, že to někoho zajímá a že nepíšu do zdi.

Děkuju Barboře Havířové za souhlas s využitím některých jejích materiálů, které připravila pro svou výuku a zveřejnila na webu microbiti.cz, a za cenné poznámky k rukopisu.

Velké poděkování patří vydavateli a všem z Edice CZ.NIC, kteří z mého strojopisu udělali už počtvrté knihu.

Díky!

Předmluva vydavatele

Předmluva vydavatele

Vážení čtenáři,

právě držíte v ruce čtvrtou knihu Martina Malého, tentokrát na téma programování micro:bitu. Pokud s programováním teprve začínáte, může vám pomoci s prvními krůčky a zvládnete s ní svou první hru či animaci. S micro:bitem si ale vyhraje, i pokud už jste ostřílenější programátoři.

Když jsem na střední škole, tedy před více jak 20 lety, začínal s programováním hardwaru já, měli jsme k dispozici pouze nepájivé pole a logické obvody. Největší vymožeností tehdy byly mikroprocesory řady 8051 a k nim připojený programátor běžící na obstarožní 486 s DOSem. A studenti, ukažte, co umíte. Pro naučení základů a proniknutí do tajů digitální techniky to stačilo. Ovšem jen hrstce z nás to připadalo úžasné a skvělé. Assembler, ať je jakkoli užitečný a vlastně jednoduchý, totiž prostě není sexy.

Později na vysoké škole jsme se setkali s již sofistikovanějším přístupem programování mikroprocesorů přes programovací jazyk C, který byl uživatelsky mnohem příjemnější. Tehdy jsem si říkal, že tím už bych se snad i živit mohl, ale vzpomínky na těžké začátky mě od toho stejně nakonec odradily.

Při pročítání knihy Micro:bit krok za krokem mě ihned napadlo, že mít podobnou k dispozici v dobách středoškolského studia, možná bych měl k programování přívětivější vztah. Jednoduché připojení různých periférií, práce s interními bloky a další funkce jsou totiž přesně to, co (nejen) mladého člověka může nadchnout a vzbudit v něm potřebný zápal pro věc. To samé přeji i všem čtenářům této knihy. A ať je pro vás tato kniha průvodcem, který ve vás vzbudí další zvědavost pro budoucí práci s hardwarem.

Petr Bílek, CZ.NIC Praha, 9. srpna 2023

Předmluva

Předmluva

Vždy, když začnu psát novou knihu, představím si její čtenáře. Tedy vás. Kdo jste, proč jste po této knize sáhli, co v ní hledáte, co od ní čekáte? A mohu vám to splnit?

U této knihy se musím přiznat: představuju si několik různých skupin čtenářů, které se od sebe velmi liší. Jediné, co je spojuje, je zájem o elektroniku a programování.

Jedna skupina to o sobě ví, nebo alespoň tuší. To jsou lidé, kteří už přičichli třeba k programování, možná se tím i živí, ale rádi by vedle abstraktních programů zkusili programovat taky něco, na co si mohou sáhnout. Ovšem různých Arduin se bojí, protože to je příliš „elektronické“. Microbit je pro ně jako stvořený – je jednoduchý, z určitého úhlu na použití jednodušší než třeba to Arduino, ale zase výkonnější a méně omezený. Křivka učení pro ně bude velmi příznivá. Stejně tak ti, co rozumí elektronice, nebo třeba mechanice, rádi by stavěli třeba roboty, ale nevěří si u programování.

Druhá skupina to o sobě třeba ještě ani neví. Jsou to mladí lidé, žáci a studenti, kteří jsou sice školou a jejími postupy už dostatečně otráveni, ale zároveň v nich ještě nezemřela radost z poznávání. Ironicky řečeno: *Kluci a holky, kteří mají radost, když se něco naučí, i přesto, že chodí do školy*. Věřím, že tihle mladí lidé ocení, že s microbitem mohou udělat spoustu zajímavých věcí, od legráček a hříček po zajímavé aplikace, hry, nebo třeba říditelné roboty. Pro starší může být microbit vstupenkou třeba do světa umělé inteligence, strojového učení nebo internetu věcí (IoT).

A třetí skupina souvisí s tou druhou. Jsou to nadšení učitelé, kteří mají chuť opravdu něco naučit, baví je to, co učí, chtějí, aby to bavilo i děti, ale hodně často naráží na limity školství a systému. Oni o microbitu vědí, jako vědí o Arduinech, Raspberry, o 3D tiskárnách a dalších věcech. Často takové sady nakoupí do školy, někdy i ze svého platu, prostě proto, že věří, že to má smysl a že dokážou svým žákům a studentům předat něco ze svého nadšení.

Ale pak bohužel zjistí, že nejsou vhodné materiály, podle nichž by mohli postupovat.

Rád bych touto knihou oslovil všechny tři skupiny, jak nadšence, tak studenty i jejich učitele. Proto jsem do knihy zařadil nejen popis experimentů a konstrukcí s microbitem, ale k nim i pár pracovních a metodických listů. Kniha se kromě samotného microbitu věnuje i nejčastějším kitům, periferiím a stavebnicím, s nimiž se můžete na českém trhu a v českých školách potkat.

Věřím, že se mi podaří alespoň trochu zaplnit mezeru, která na českém trhu dlouho byla, a že kniha pomůže všem zájemcům o práci s microbitem v jejich prvních krocích.

Metodická poznámka

Metodická poznámka

Tuto knihu píšu na jaře roku 2023. Prostředí, v němž se microbity programují nejčastěji, je MakeCode. Najdete jej na adrese <https://makecode.microbit.org/>. Toto prostředí je z větší části přeloženo do češtiny. Někdy mám ke zvolenému překladu výhrady, ale většinou je překlad srozumitelný a pochopitelný. Proto se jej přidržím i v této knize. Co je přeloženo, budu zapisovat v přeložené podobě tak, jak je to přeloženo v době psaní knihy. Je možné, dokonce pravděpodobné, že po nějakém čase bude překlad vylepšený, změněný, nebo že části, které jsou zatím nepřeložené, budou dopřeloženy. Pro tyto případy poznamenávám i originální význam, aby byla kniha použitelná i do budoucna.

Výklad je sice zaměřen na začátečníky, ale předpokládám, že zájemce už ví, co je *počítač*, *programovací jazyk* nebo *program*, popřípadě že zná způsoby algoritmizace problému a podobně. Nevysvětluju, co je číslo a co je znak; u základních programových struktur vysvětluju jejich funkci jen obecně. Mým cílem není udělat z této publikace učebnici programování, k tomu sloužit nemá.

Při výkladu se soustředím především na prostředí blokového editoru MakeCode. Jsem si vědom toho, že pro pokročilejšího čtenáře není skládání bloků ideální způsob programování, přesto doporučuju, aby si i čtenář, znalý např. Pythonu nebo JavaScriptu / TypeScriptu, jednotlivé úvodní kapitoly prošel. Seznámí se tak s mnoha specifiky programování pro microbit. Pro snazší orientaci zařazuju do knihy výpisy některých příkladů v jazycích Python i JavaScript. Předpokládám, že zkušenější čtenář ví, co je např. podmínka nebo cyklus, a tak tyto věci osvětluju pouze pomocí bloků.

Knihu jsem rozdělil na několik částí. V první části probírám naprosté základy, včetně toho, co je proměnná a cyklus. Druhá část probírá pokročilejší techniky – rádio, kompas, magnet, práce s řetězcí, seznamy atd. Ve třetí části se věnuju propojení microbitu a vnějšího světa. Čtvrtá část

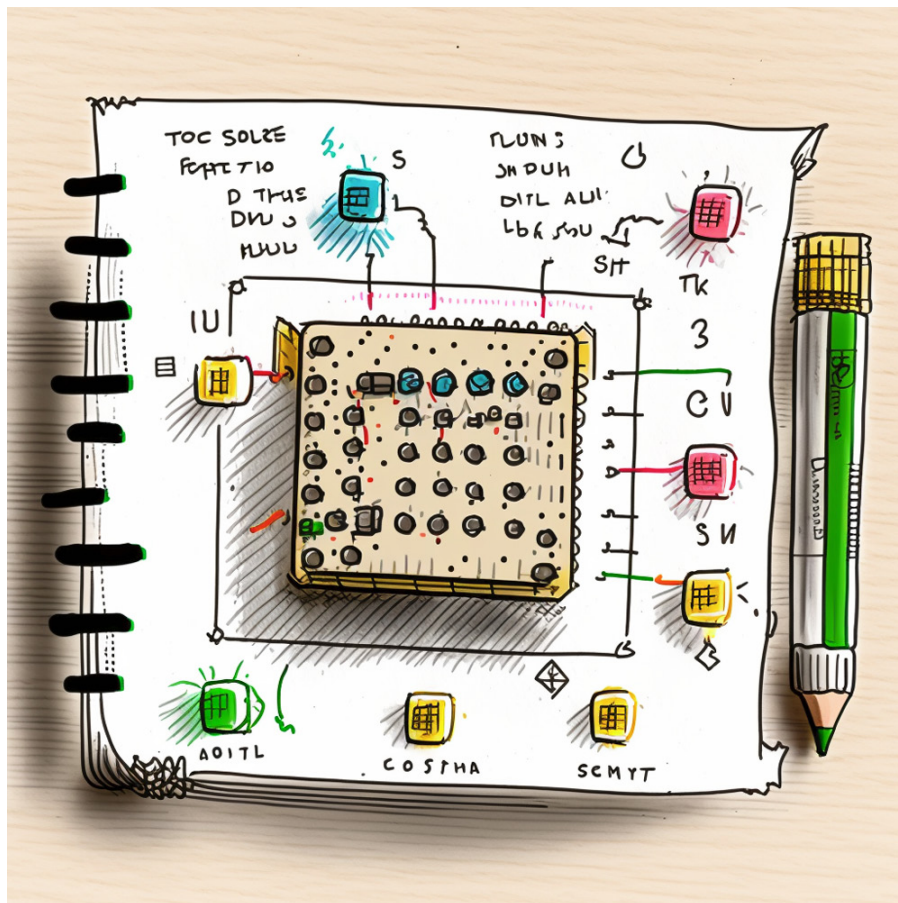
probírá pokročilá témata: mechanismus událostí a zpráv, vytváření vlastních rozšíření nebo některé alternativy k microbitu. Pátá část je věnována pokročilejším programátorům, kteří ovládají JavaScript a hodlají v něm programovat microbity. V dodatcích pak najdete schéma, praktické rady, jak například aktualizovat firmware, a v neposlední řadě pak odkazy na online materiály.

Jazyková poznámka

Jazyková poznámka

Originální název je „BBC micro:bit“ – se zjevným odkazem na legendární počítač BBC Micro z 80. let. V češtině se používá hned několik tvarů: s dvojtečkou i bez dvojtečky, s velkým M i s malým, někde i ve variantě „mikro...“ V této knize se budu držet varianty „microbit“ – tedy bez dvojtečky a s malým „m“. Varianta s dvojtečkou je ideální v titulcích nebo nadpisech, ale v běžném textu není její použití moc praktické, proto se budu držet tvarů microbit, microbitu, microbitem, microbity, ...

U názvů funkcí, bloků a dalších prvků se řídím tou podobou, jakou mají v době psaní knihy. Některé jsou přeložené, pak použiju český aktuální překlad, i když není ideální a bude pravděpodobně změněn; na tuto možnost upozorňuju. Pokud nějaká část přeložena není, použiju anglický název a možný český překlad.



Obsah

Poděkování 7

Předmluva vydavatele 11

Předmluva 15

Metodická poznámka 19

Jazyková poznámka 23

1 Jak číst tuto knihu? 31

2 Microbit - úplně první kroky 35

[2.1 První program 35](#)

[2.2 Než začneme doopravdy 41](#)

[2.3 Začínáme 47](#)

[2.4 Kreslení 54](#)

[2.5 Vstup 57](#)

[2.6 Zvuk 66](#)

[2.7 Kreslení 2 68](#)

[2.8 Odpočet 76](#)

[2.9 Hod mincí 85](#)

[2.10 Vodováha 89](#)

[2.11 Ovládání bodu náklonem 110](#)

[2.12 Hra Chyt' mě! 117](#)

3 S microbitem na další úroveň 133

[3.1 Rádiové spojení 133](#)

[3.2 Práce s textem 143](#)

[3.3 Pokročilejší funkce rádia 150](#)

[3.4 Kompas a magnet 162](#)

[3.5 Seznamy 166](#)

4 Jak ovládnout svět microbitem 179

[4.1 Konektor microbitu 180](#)

[4.2 Sensor:bit 197](#)

[4.3 Basic kit 199](#)

[4.4 Starter kit 201](#)

[4.5 Robotické vozítko Ring:bit 203](#)

[4.6 IoT kit 216](#)

5 Microbity pro pokročilé 229

[5.1 Soubor s programem 229](#)

[5.2 Události 230](#)

[5.3 Rozšíření 235](#)

6 Co po microbitu? 261

[6.1 Retro Arcade for Education 261](#)

[6.2 Pico:ed 262](#)

[6.3 Mbits 263](#)

[7 Doslov 267](#)

[8 CircuitPython 271](#)

[8.1 Rozdíly oproti MicroPythonu 271](#)

[8.2 První kroky s CircuitPythonem 272](#)

[8.3 Vývojové prostředí Thonny 273](#)

[9 JavaScript a Microbit 279](#)

[9.1 Robot Karel 279](#)

[9.2 Rychlý úvod do JavaScriptu / TypeScriptu pro microbit 282](#)

[10 JavaScriptová knihovna microbitu - referenční příručka 309](#)

[11 Dodatky 441](#)

[11.1 Schéma microbitu 441](#)

[11.2 Aktualizace firmware 441](#)

[11.3 Credits 443](#)

[Odkazy 447](#)

1 Jak číst tuto knihu?

1 Jak číst tuto knihu?

Nemyslím si, že čtenáři nevědí, jak číst knihu. *Stačí přeci číst slova a na konci pravé stránky vpravo dole obrátit list.*

Přesto dovolte jedno doporučení:

Pokud začínáte s microbitem a s programováním, začněte hned následující kapitolou. Čtěte postupně, zkoušejte si popsané příklady, zkoušejte si jednotlivá cvičení a postupujte tak, jak jde výklad.

Pokud umíte programovat, ale s microbitem nemáte zkušenosti, tak následující kapitoly berte spíš jako rozcvičku a ukázkou toho, co se s microbitem dá dělat. Nemusíte se zabývat „obrázkovým programováním“, můžete si přepnout na Python nebo JavaScript dle své libosti a zkusit rovnou programovat „psáním kódu“. Pravděpodobně nebudete vědět, jakou knihovnou a kde použít – editor MakeCode naštěstí vhodně napovídá a doplňuje.

Pokud si troufáte na microbit, ale rádi byste zkusili něco s hardware, tak můžete začít kapitolami o zabudovaných senzorech (akcelerometr, rádio) a pak plynule přejít na část o „ovládání světa pomocí microbitu“.

Pokud jste už s microbitem něco dělali, tak oceníte možná kapitoly „pro pokročilé“ – najdete v nich popis pokročilejších technik, popis mechanismu „událostí“ (Message Bus) a obsáhlejší kapitolu o tvorbě rozšíření pro MakeCode a jejich lokalizaci. Pokud je vaším jazykem JavaScript, tak na konci knihy najdete obsáhlou kapitolu s výpisem dostupných zabudovaných funkcí, jejich popisem a stručným úvodem, kde je vysvětlen rozdíl mezi standardním JavaScriptem a TypeScriptem, použitým v MakeCode.

Pokud jste učitel a hodláte zapojit microbit do výuky, tak si přečtěte celou knihu, včetně příkladů, metodických listů a návodů, všechno si na vlastní kůži vyzkoušejte, všechny úkoly vypracujte a na všechny řečnické otázky odpovězte, *máte na to 30 minut, pak nekompromisně vyberu testy! Kdo bude*

opisovat, má automaticky nedostatečnou! Ne, promiňte, nebudu vás zkoušet, ale ten zbytek myslím vážně. Opravdu si zkuste co nejvíc věcí a buďte připraveni na co nejvíc situací, které mohou při výuce nastat. Od problémů s připojeným hardware po „záhadné“ chyby. Trochu jsem se snažil napomoci „metodickými listy“ na koncích kapitol. Vyšel jsem hlavně ze skvělých materiálů Barbory Havířové (<https://microbiti.cz>), která laskavě souhlasila s použitím několika materiálů v této knize, za což jí ještě jednou děkuju. Navštivte i web <https://m-bit.cz>, kde je v Odkazech speciální sekce s metodickými a pracovními listy. *Držím vám palce!*

2 Microbit - úplně první kroky

2 Microbit - úplně první kroky

2.1 První program



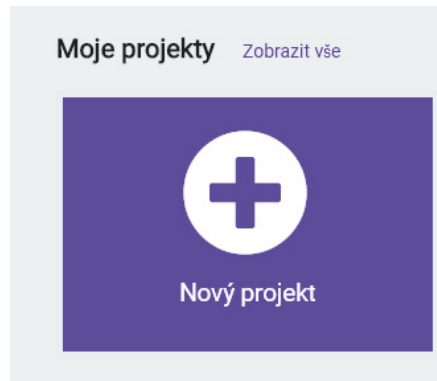
Vezměte microbit, připojte ho k USB kabelu (typ MicroUSB) a zapojte k počítači. Pokud je váš microbit úplně nový, spustí se ukázkový program, který předvede, co všechno microbit dokáže.

Zkuste jej chvíli pozorovat, zkuste si stisknout tlačítka A, B, zkuste dotykovou plošku a sledujte, co se děje. A až se vynadíváte, pustíme se do práce.

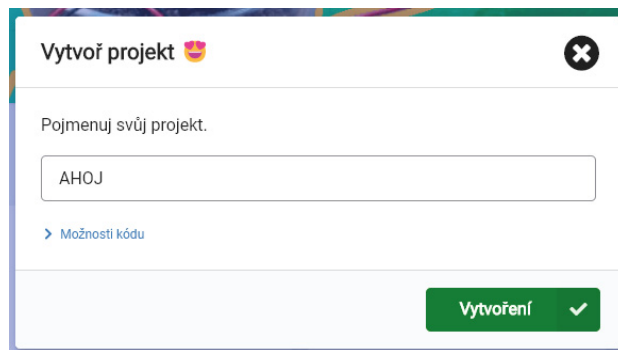
Každý programátor ví, že se začíná nápisem AHOJ SVĚTE. Nebo alespoň HELLO WORLD. U elektroniky je obdobou bliknutí LEDkou.

Na microbitu je ale celý displej, poskládaný z pětadvaceti LED v matici 5×5. Kterou rozsvítíme? No, uvidíme...

Otevřete prohlížeč a jděte na adresu <https://makecode.microbit.org/> Najdete zde spoustu materiálů, ukázek, kurzů a námětů. Zůstaneme ale v sekci *Moje projekty* – tam klikněte na **Nový projekt**.

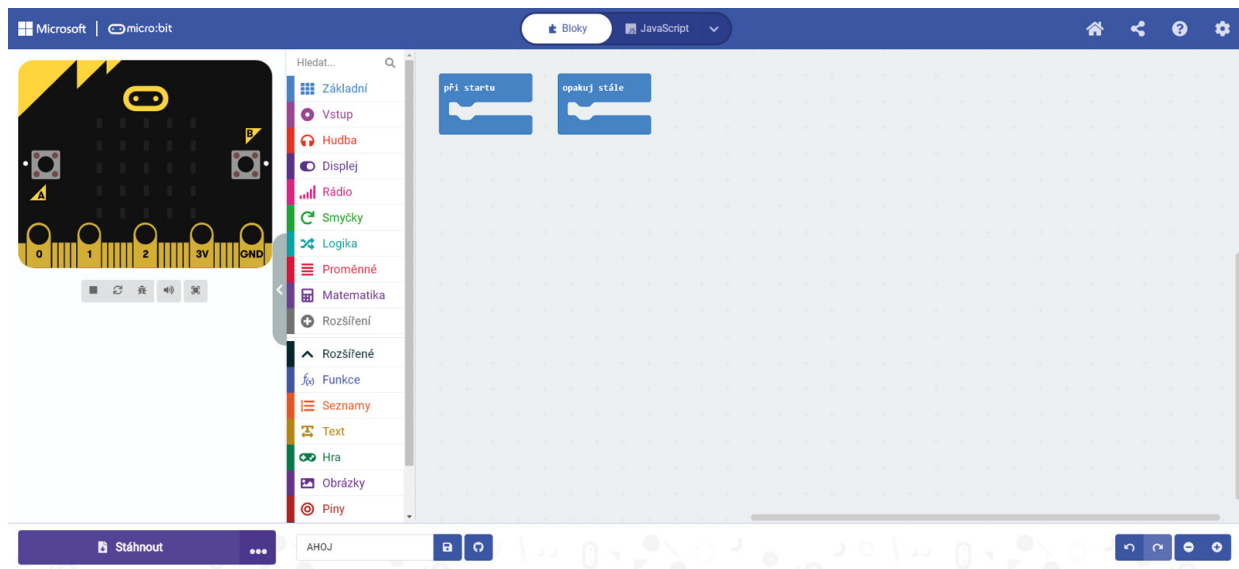


Náš nový projekt pojmenujeme třeba AHOJ a klikneme na tlačítko Vytvoření:



Po vytvoření nového projektu se ocitneme ve vývojovém prostředí microbitu, nazvaném „MakeCode“. Vlevo je simulátor microbitu, vedle něj sloupec se základními bloky, z nichž budeme program skládat, a vpravo prostor pro tvorbu programu. Nahoře je, jak je dobrým zvykem, menu, a dole pak několik ovládacích prvků, ke kterým se ještě dostaneme.

Pokud vám MakeCode připomíná jiný vizuální programovací nástroj, totiž Scratch, tak se nemyliťte. Blokovaný editor pro „skládání programů“ je totiž prostředím Scratch velmi silně inspirovan. Není naprosto shodný, ale je tak podobný, že člověk, který zná Scratch, bude v MakeCode intuitivně chápat, co a jak dělat.

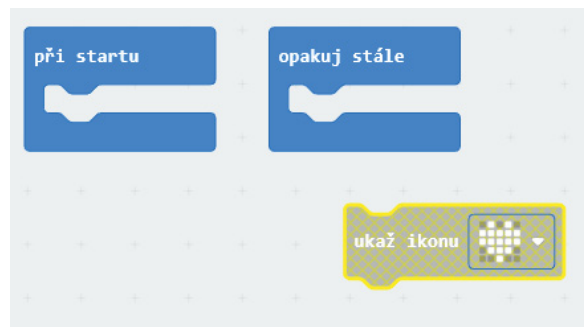


Na tomto místě musím uklidnit všechny odpůrce vizuálního programování: Nebojte, můžete použít i klasické programovací jazyky. Prostředí MakeCode přímo podporuje Python a JavaScript (respektive jejich „micro“ dialekty), a lze použít i C/C++. Ale začnu s vizuálním editorem.

Klikněte na seznam bloků s názvem Základní (v originále Basic). Rozevře se nabídka bloků jako na následujícím obrázku:

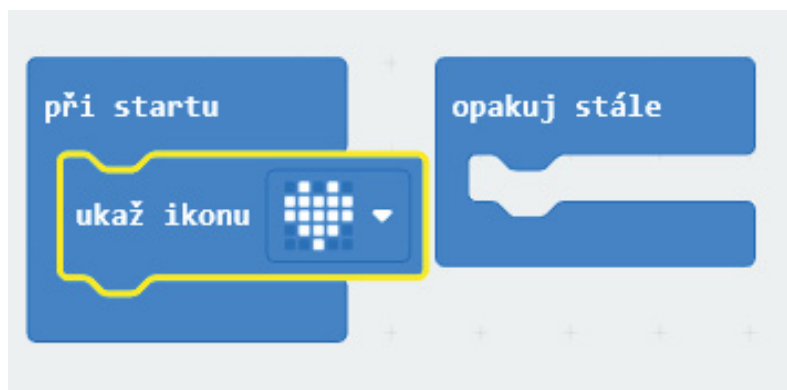


VeźmĚte blok „UkaŹ ikonu“ a pĚtĚhnĚte jej doprava na plochu editoru.



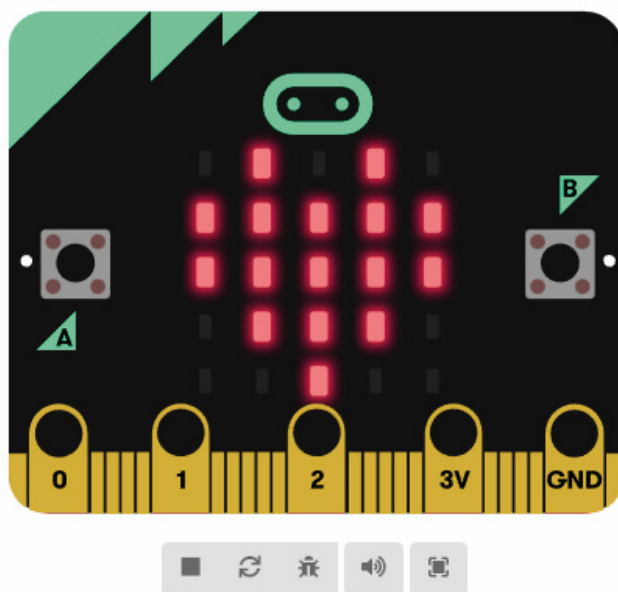
VšimnĚte si, Źe je zašrafovanĚ a vŮbec vypadĚ velmi neaktivnĚ. MyšĪ ji pĚtĚhnĚte do volnĚho prostoru v modrĚm bloku „při startu“. VšimnĚte si, Źe tento blok mĚ ve volnĚm prostoru takovĚ dva vŮřezy, do kterŮch blok „ukaŹ ikonu“ pĚsnĚ zapadne. KdyŹ blok pĚtĚhnĚte na vhodnĚ mĪsto, naznaĹĹ vĚm to editor tĪm, Źe ukĚŹe, kam se blok pĚpoĹĹ. UdĚlejte to.

VšimnĚte si, Źe jakmile na takovĚm mĪstĚ pustĪte blok, tak s jemnŮm zacvaknutĪm zapadne pĚsnĚ do vybranĚho mĪsta a zmodrĚ.



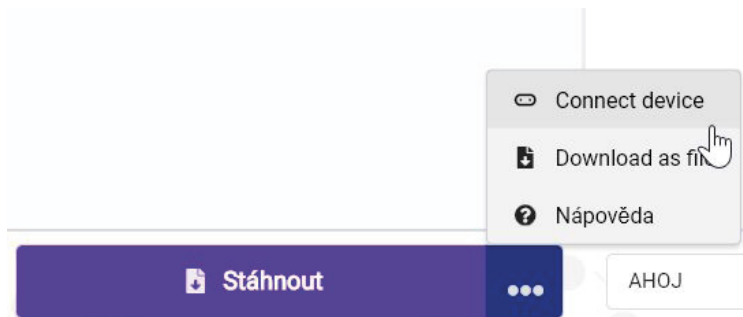
Tím jste vytvořili první program. Ano, je to tak. A tento program udělá přesně to, co je napsáno na obrazovce: **Při startu ukáže ikonu srdce.**

Po několika sekundách se program spustí i v simulátoru vlevo, a vy uvidíte výsledek:

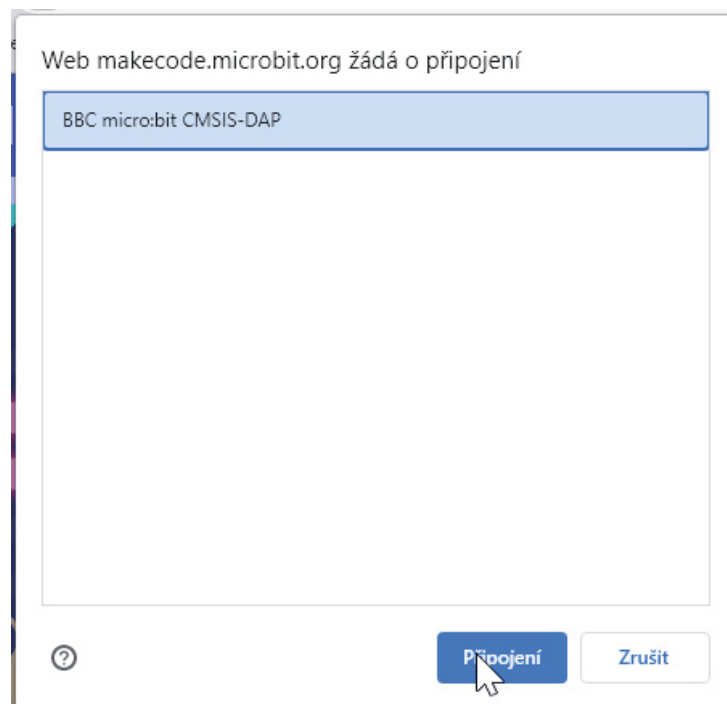


Takže program funguje, ukazuje to, co ukazovat má, ale pořád není v microbitu. Přišel čas, kdy jej tam nahrajeme.

Předpokládám, že máte microbit stále připojený kabelem k počítači. Klikněte na ikonu tří teček vedle tlačítka Stáhnout v levém dolním rohu a vyberte možnost „Connect device“ (v době psaní knihy ještě nebyla tato možnost přeložena do češtiny, pozn. aut.)



Průvodce připojením vám oznámí, že musíte v okně, které se objeví, vybrat možnost „BBC microbit CMSIS-DAP“ nebo „DAPLink CMSIS-DAP“ a potvrdit.



Všimněte si, že se změnila ikonka u tlačítka Stáhnout:



Z ikony „stažení dokumentu“ se stala ikonka microbitu. Teď klikněte na Stáhnout a nechte zázraky, ať se dějí. Po malé chvíli se rozsvítí nápis „Staženo“ (Downloaded) a na vašem microbitu se ukáže krásné červené srdce, úplně stejné jako v tom simulátoru.

Pokud se stane, že se microbit nepřipojí, např. máte starší prohlížeč, který nepodporuje standard WebUSB, tak můžete soubor stáhnout do počítače. Pak se například Průzkumníkem připojíte k microbitu, jako kdybyste se připojovali třeba k FLASH disku.

Uvidíte ho mezi ostatními úložišti. Stažený soubor (má příponu .hex) jednoduše zkopírujete do microbitu. Jakmile se dokopíruje, microbit jej spustí.


Pokud máte podporovaný prohlížeč (Chrome, Edge atd.) a přesto se nelze připojit, jako se to stalo mně, můžete zkusit:

- 1. Zkontrolovat, zda funguje ovladač winUSB, popřípadě jej přeinstalovat (utilitou Zadig)*
- 2. Ve Správci zařízení odinstalovat zařízení mbed (pokud jste s ním někdy experimentovali, jako já)*

Po restartu už by mělo vše fungovat k plné spokojenosti.

Gratuluji. Právě jste v seznamu úkolů splnili ten úplně první: **Vytvořili jste program pro microbit a spustili ho!**

A teď si všichni dáme zmrzlinu...

 *Podívejte se znovu na svůj program. Vedle ikony srdce vidíte malou šipku, která naznačuje, že je víc možností, stačí si jen vybrat. Zkuste to a místo srdce vyberte smajlík. Program nahrajte do microbitu.*

Metodický list

Co si připravit?

- Microbit, kabel USB – microUSB, propojení s PC
- Ověřit, zda PC s microbitem komunikuje
- Ověřit, zda je dostupné prostředí MakeCode

- Spárovat s microbitem přes WebUSB

Co jsme se naučili?

- Že programovat je fajn a zábava

Jak postupovat?

- Ideálně připravit microbity tak, aby byly připojené k PC a aby byly spárované s MakeCode
- Začít motivací – proč se učit programovat a že to může být i zábava
- Pomalu projít bod po bodu celý postup až do okamžiku, kdy se na displeji objeví ikona

- Podporovat je, ať se podívají, co tam jde vykreslit dál. Vysvětlit, že změnu v programu je potřeba „nahrát“ do microbitu.

Úkoly a výzvy

- Změnit srdíčko na jinou ikonu

- Zvědavé nechat, ať se podívají například na blok „vypiš text“, nebo ať zkusí použít dva bloky s výpisem ikony po sobě.

2.2 Než začneme doopravdy

Než se pustíme do všech těch skvělých a zajímavých věcí, co s microbitem lze dělat, položím vám jednu prostou otázku: **Máte microbit? A víte, co to vlastně je?**

Pokud ano, můžete klidně skočit na další kapitolu, protože v této kapitole se budeme věnovat právě tomu, kde microbit sehnat, jaké modely jsou k dostání, čím se od sebe liší a tak dál. Ale přesto bych vám doporučil tuto kapitolu nepřeskakovat, protože si řekneme i o doplňcích, které si můžete pořídit.

BBC microbit se představuje

Kdybych měl nějak popsat, co je microbit, tak použiju slova „miniaturní mikropočítač, zaměřený na výuku“. Co tam tedy dělá to BBC? Je to opravdu zkratka britské veřejnoprávní televize?

Pokud znáte historii počítačů, víte, že britská BBC měla už v jednom počítači prsty. Šlo o známý počítač BBC Micro na začátku 80. let. Tehdy BBC začala s výukovým projektem počítačové gramotnosti, čímž reagovala na předpovědi mnohých odborníků, že počítače čeká obrovský boom. Při přípravě tohoto vzdělávacího programu došlo jeho autorům, že potřebují, aby v Británii byl dostupný počítač, ke kterému se budou moci lidé dostat, buď ve škole, nebo doma, a který bude v projektu použit jako referenční. Proto ve veřejné soutěži oslovili několik britských výrobců s poptávkou po takovém počítači. Nakonec vyhrála společnost Acorn se svým počítačem Proton, který byl přejmenován na BBC Micro, a úspěšně jej mnoho let dodávala na trh. Výukový projekt nakonec velmi výrazně nastartoval a podpořil to, co dnes známe jako boom domácích počítačů.

V desátých letech tohoto století si BBC opět vzpomněla na to, že jejím posláním je i vzdělávání, a rozhodla se přijít s podobnou iniciativou „Make It Digital“. A během ní představila právě microbit – malý mikropočítač, vhodný pro výuku.

Proč raději nepředstavila obyčejné PC, nebo něco takového? Odborníci a učitelé z technických škol v té době upozorňovali na zajímavý jev. Zatímco v osmdesátých letech a na začátku devadesátých let nastupovali na technické školy studenti, kteří počítač uměli používat, programovat a chápali, jak pracuje, tak po přelomu století tahle

znalost pomalu vymizela. Na školy se hlásili lidé, kteří uměli počítač používat. Programovat uměla jen velmi malá část z nich, a ještě menší část chápala, jak počítač vůbec pracuje. A důvodem bylo mimo jiné právě to, že u domácího počítače v osmdesátých letech jste museli umět aspoň trošičku programovat, abyste jej mohli použít, a abyste jej mohli programovat lépe, museli jste chápat, jak pracuje. Na druhou stranu, když jste jej chtěli programovat, stačilo ho zapnout – a už na vás blikal nejčastěji nějaký druh BASICu. Vy jste napsali PRINT 1+1, a byli jste programátor. Jenže s moderními PC jste uživatel. Což je samozřejmě dobře, jen to úplně neprobouzí ten druh zvědavosti, který vyučující na *technikách* očekávali.

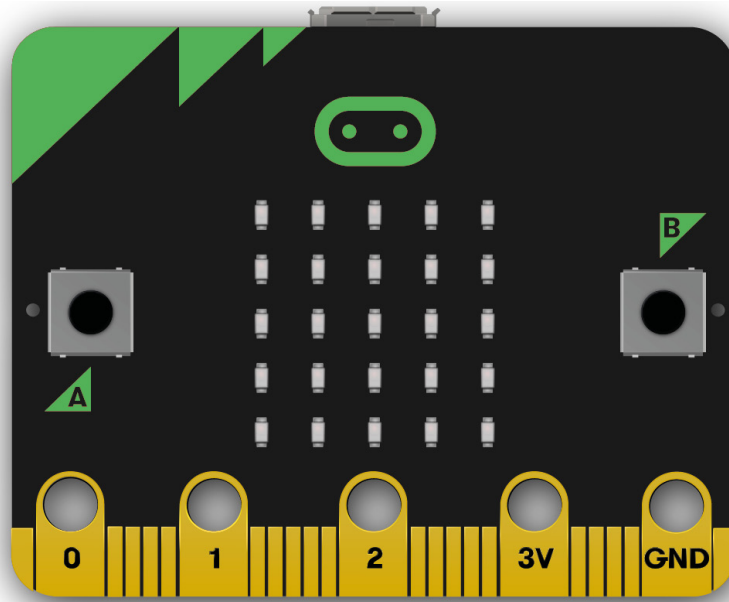
Proto v BBC chtěli počítač, který bude odolný, který bude mít periférie, který bude levný a z kterého budou, zjednodušeně řečeno, *viset dráty*. Počítač, který bude natolik jednoduchý, aby bylo možné ho pochopit. Počítač, kde bude programování základním způsobem, jak s ním komunikovat. A pokud možno bez složitých IDE a překladačů a knihoven.

A tak vznikl BBC microbit (též BBC Micro:Bit nebo Micro:Bit).

Verze 1? Verze 2?

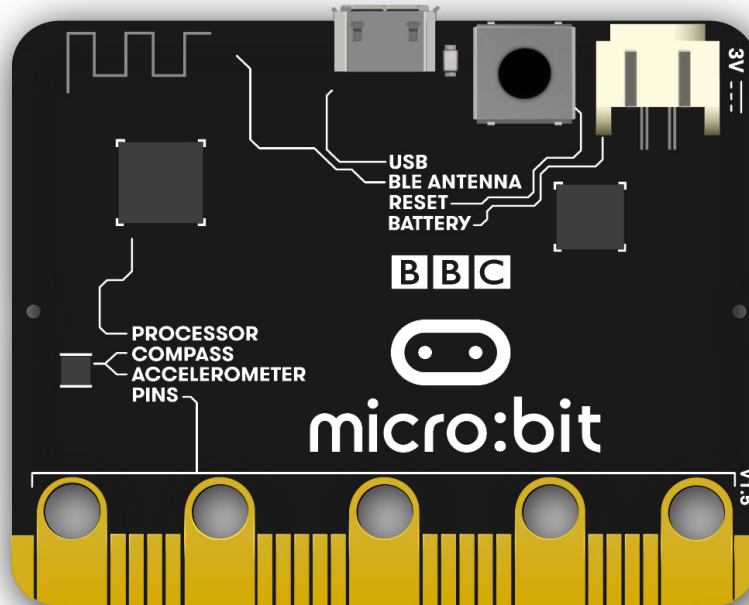
Zařízení je vybaveno procesorem ARM Cortex-M0, senzory akcelerometru a magnetometru, připojením Bluetooth a USB, displejem tvořeným 25 LED diodami, dvěma programovatelnými tlačítky a může být napájeno z USB nebo externí baterie. Vstupy a výstupy zařízení jsou prostřednictvím pěti kruhových konektorů, které jsou součástí většího 25pinového konektoru na hraně. V říjnu 2020 byla vydána fyzicky téměř identická deska v2, která je vybavena mikrokontrolérem Cortex-M4F, větší pamětí a dalšími novými funkcemi.

Verze 1



Fyzické rozměry desky jsou 43×52 mm a v první výrobní sérii zahrnovala:

- Nordic nRF51822 - 16 MHz 32bitový mikrokontrolér ARM Cortex-M0, 256 KB paměti flash, 16 KB statické paměti RAM, bezdrátovou síť Bluetooth low energy 2,4 GHz. Jádro ARM má možnost přepínat mezi frekvencí 16 MHz nebo 32,768 kHz.
- NXP/Freescale KL26Z - 48MHz mikrokontrolér s jádrem ARM Cortex-M0+, který obsahuje plnorychlostní řadič USB 2.0 On-The-Go (OTG), používaný jako komunikační rozhraní mezi USB a hlavním mikrokontrolérem Nordic. Toto zařízení také provádí regulaci napětí z napájení USB (4,5-5,25 V) na nominálních 3,3 V, které používá zbytek desky plošných spojů. Při provozu na baterie se tento regulátor nepoužívá.
- NXP/Freescale MMA8652 – tříosý akcelerometrický senzor, připojený na sběrnici I²C.
- NXP/Freescale MAG3110 – tříosý magnetometrický snímač, připojený na sběrnici I²C (funguje jako kompas a detektor kovů).
- Konektor MicroUSB.
- Konektor pro připojení baterie.
- 25pinový hranový konektor.
- Displej tvořený 25 LED diodami v matici 5×5.
- Tři tlačítka (dvě pro aplikace, jedno pro reset).



Hranový konektor obsahuje i pět kruhových konektorů (jeden napájecí 3 V, jeden zemní GND a tři datové vstupy/výstupy 0, 1, 2), do nichž lze zastrčit 4mm konektory „banánky“, nebo tzv. krokosvorky. Pro základní zapojení těchto pět pinů stačí.

Kompletní 25pinový hranový konektor nabízí podle konfigurace 2-3 výstupy PWM, 6 až 17 pinů GPIO, až šest analogových vstupů, sériové I/O, SPI a I²C.

Princip sběrnic SPI a I²C i s příklady využití najdete v knize Hradla, volty, jednočipy od stejného nakladatelství (dostupná i jako e-book zdarma).

Na rozdíl od prvních prototypů, které měly integrovanou baterii, lze k napájení zařízení jako samostatného nebo nositelného výrobku použít externí baterii (baterie AAA).

Dostupná dokumentace návrhu hardwaru se skládá pouze ze schématu a kusovníku šířeného pod licencí CC-BY, rozvržení desky plošných spojů není k dispozici. Existuje však plně dokumentovaný referenční návrh od Microbit Educational Foundation.

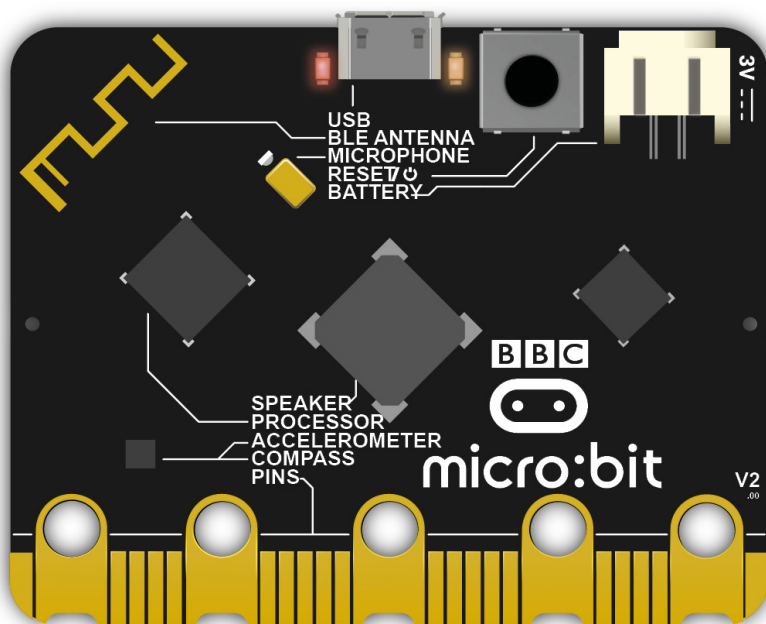
Verze 2

Druhá verze, vydaná 13. října 2020, obsahuje:

- Nordic nRF52833 - 64 MHz 32bitový mikrokontrolér ARM Cortex-M4, 512 KB paměti flash, 128 KB statické paměti RAM, bezdrátová síť Bluetooth low energy 2,4 GHz zajišťovaná zařízením Nordic S113 SoftDevice, integrovaný teplotní senzor.
- NXP/Freescale KL27Z - 48 MHz mikrokontrolér s jádrem ARM Cortex-M0+, předprogramovaný jako plnorychlostní řadič USB 2.0, používaný jako komunikační rozhraní mezi USB a CPU.

- ST LSM303 nebo NXP FXOS8700 – tříosý kombinovaný akcelerometr a magnetometr přes sběrnici I²C.
- Mikrofon MEMS Knowles s vestavěným indikátorem LED.
- Reprodukční Jiangu Huaneng MLT-8530.
- Konektor MicroUSB.
- Konektor pro připojení baterie JST PH.
- 25pinový hranový konektor.
- Displej tvořený 25 LED diodami v matici 5×5.
- Tři tlačítka (dvě pro aplikace, jedno pro reset) a dotykové tlačítko (logo).

U této verze microbit lze tlačítko reset použít k vypnutí desky jeho podržením po dobu 3 sekund.



Kde koupit?

Microbity nabízí mnoho obchodů, kamenných i online. Nemusí jít jen o specializované obchody s elektronikou, ale doporučuji začít právě tam, protože tam k Microbitu seženete většinou i nějaké základní příslušenství, nebo rovnou celé výukové sady se součástkami a moduly, které můžete připojit.

V České republice v době psaní knihy nabízely microbit například následující e-shopy:

- HWKITCHEN
- RPiShop
- Megarobot
- Rasel

- Farnell
- emo

Výčet není úplný ani autoritativní.

Co koupit dál?

Kromě samotného microbitu potřebujete už jen **kabel USB – MicroUSB**, kterým microbit připojíte k počítači a budete jej programovat. Většina z nás ho má doma, například od staršího telefonu nebo od jiného zařízení, ale doporučuju koupit nový, ideálně kratší, kolem jednoho metru. Díky tomu si ušetříte případné hledání problémů s „nefunkčním“ microbitem (vadný kabel způsobuje „záhadné“ výpadky, poruchy a problémy).

Pro první seznámení a vytvoření nějakého zajímavého zapojení toto bohatě stačí. Ale doporučuju po prvním seznámení pořídit i nějaký kit s dalšími součástkami. Kitům a stavebnicím se budu věnovat v příslušné části knihy, ale zde doporučím:

- Basic kit – jednoduchý kit s několika málo moduly, které rozšíří možnosti microbitu a demonstrují jeho propojení s vnějším světem
- Starter kit – vylepšený kit, který obsahuje kromě potřebných modulů i nepájivé kontaktní pole, sadu součástek a návod, takže nejste omezeni na předpřipravené moduly, ale můžete připojit *téměř cokoli*, co se vejde do nepájivého kontaktního pole.
- Ring:bit car – stavebnice robotického vozítka, které je řízené microbitem. Namísto „abstraktního“ zapojování součástek zde stavíte konkrétní věc, která funguje autonomně, a kterou lze dle libosti rozvíjet.
- Nezha Inventor Kit – kit, který jde v tvořivosti asi nejdál. Microbit a sada senzorů je zde doplněna sadou základních kostek, kompatibilních s LEGO Technic, takže fantazii se meze nekladou. Problémem však může být vyšší pořizovací cena, která je na druhou stranu vyvážena vysokou variabilitou a možnostmi. Při stavbě konstrukcí jste omezeni jen dostupností součástek a vlastní fantazií.

Kde hledat inspiraci?

Největším problémem při zkoušení podobných věcí, jako je microbit, není ani tak zjistit, jak věci udělat, ale vymyslet, co udělat. Buď vás napadá všechno, nebo nic. Dobré je začít tím, co vymyslel už někdo jiný.

V této knize popisuju několik různých projektů, nápadů a návodů, včetně námětů k dalšímu rozvoji, ale kniha je omezená svým rozsahem. Další místa, kde hledat inspiraci, najdete buď v odkazech na konci knihy, nebo na webu knihy <https://m-bit.cz/>.

Na tomto místě mi dovoluete uvést dva zdroje:

- Microbiti – stránky Barbory Havířové, která se specializuje na využití microbitů při výuce. Na webu má mnoho materiálů, které mohou být inspirací pro učitele, co s žáky či studenty probírat. <https://www.microbiti.cz/>
- Bastlárna HWKITCHEN – stránky Oldřicha Horáčka, kde najdete mnoho návodů a námětů k vlastní tvorbě. Nejsou zaměřeny didakticky, spíš techničtěji, ale jsou vhodné i pro začátečníky. <https://bastlirna.hwkitchen.cz/category/novinky/tutorialy/microbit/>

2.3 Začínáme

V této kapitole si projdeme úplné základy programování. Dozvíte se, co je proměnná, funkce, co je cyklus, co je podmínka, zkusíte si pracovat s blokovým editorem a naučíte se programovat microbit bez toho, abyste museli znát jediný příkaz v nějakém programovacím jazyce.

Předesílám to rovnou na začátku, abych předešel nedorozumění. Tato část je zaměřena na naprosté začátečníky, proto bude výklad velmi pomalý, plný příkladů, úloh a téměř výhradně orientovaný na blokový editor. Pokud jste si jisti, že toto všechno znáte, můžete tuto část přeskočit, nebo jen rychle prolistovat.

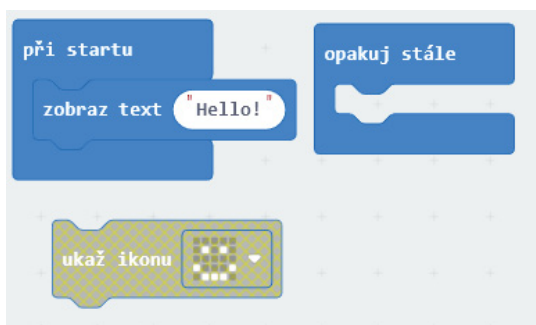
Základní nabídka

Text

Podívejte se do nabídky základních bloků, tam, odkud jste brali blok „ukaz ikonu“. Nad ním jsou bloky Zobraz číslo a Ukaž tvar, pod ním Zobraz text. Zkusme si teď vypsát to AHOJ.

Klikněte na „Zobraz text“. Blok se přesune do pracovního prostoru, ale nikam se nepřipojí. Nechte jej teď být – nejprve si musíme uklidit blok „Při startu“ – v něm zůstalo stále zobrazení ikony.

Veźměte blok „Ukaž ikonu“ v bloku „Při startu“ myšě a odtáhněte jej kousek stranou, mimo blok. Když jej pustěte, zůstane ležet na ploše. Teď místo něj do bloku „Při startu“ odtáhněte blok „Zobraz text“.

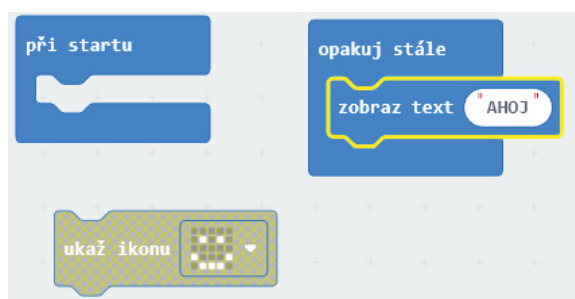


Jistě, můžeme být světoví a zobrazit Hello, ale řekli jsme si, že zobrazíme AHOJ, tak to taky dodržíme. Stačí kliknout na text Hello v bílé bublině a přepsat ho na AHOJ.

Jakmile se tak stane, uvidíte v simulátoru, že přes displej přejedou písmena A, H, O a J. A pak nic. Tma, černo. *To nebyla moc velká show, že?*

Problém je v tom, že microbit nemá moc možností, jak zobrazit delší text než ukázat písmeno po písmenu. A tak ukáže písmeno po písmenu a na konci – no, na konci je konec.

Udělejte takový trik. Vezměte blok „Zobraz text AHOJ“ a přesuňte jej z bloku „Při startu“ do bloku „Opakuj stále“.



A už teď v simulátoru uvidíte, že projede nápis AHOJ, a po něm projede nápis AHOJ, a po něm projede nápis AHOJ, a... No, bude se to dít stále a stále.

! A to je rozdíl mezi těmi dvěma bloky. Jakmile nahrajete program do microbitu, nebo zapnete napájení, nebo stisknete tlačítko RESET, tak microbit udělá to, co má napsáno v bloku „Při startu“. Udělá to přesně jednou. A pak udělá to, co je v bloku „Opakuj stále“, a to, co tam je, opakuje stále.

Zůstal nám na ploše jeden nepoužitý blok. Co s ním?

- Můžete jej odtáhnout myší doleva, do oblasti nabídky bloků. Zobrazí se ikona odpadkového koše. Když teď blok pustíte, smaže se.
- Můžete na něj kliknout a zmáčknout klávesu Delete. Smažete ho tím.
- Můžete kliknout pravým tlačítkem myši a vybrat „Odstranit blok“.
- Nebo ho můžete znovu použít...

☞ Zkuste nepoužitý blok s ikonou použít znovu v bloku **Při startu**. Co se stane? A zkuste jej použít v bloku **Opakuj stále**. Zkuste jej dát před blok zobrazení textu, zkuste jej dát za něj. Co se stane?

☞ Zkuste zadat v textu různé znaky, jako velká a malá písmena. Dokáže je microbit zobrazit? A co číslice? Dokáže zobrazit řadu číslic „0123456789“? A co třeba písmena

s diakritikou? Dokáže microbit zobrazit slovo „Příšerný“? Umí tečku, dvojtečku, vykřičník, otazník...?

Další základní funkce

Nabídka *Základní* obsahuje víc bloků pro práci s displejem.

Zobraz číslo zobrazí číslo

Ukaž tvar umí zobrazit jakoukoli kombinaci rozsvícených a zhasnutých bodů, kterou nakreslíte do mřížky 5×5 bodů

Zhasni displej udělá přesně to, co byste čekali: zhasne displej

Opakuj stále můžete použít, pokud chcete, aby probíhalo souběžně několik úloh, které se mají stále opakovat.

Při startu může být v programu jen jednou. Pokud si přihodíte na plochu další, tak se předchozí blok zneaktivní. Hodí se, pokud si jej omylem smažete.

Čekej se hodí k tomu, když potřebujete, aby nastala nějaká pauza. Hned si ukážeme jeho praktické využití.

Ukaž šipku funguje podobně jako Ukaž ikonu, ale místo nadefinovaných ikon pracuje s osmi světovými stranami (4 hlavní a 4 vedlejší). Hodit se bude při práci s kompasem, jak si ukážeme později.

☞ Použijte blok „ukaž tvar“ k zobrazení obrysu srdce, k zobrazení mřížky a k zobrazení vlnovky

☞ Ukažte na displeji šipku, která se bude točit stále dokola.

Metodický list

Co si připravit?

- Microbit, kabel micro-USB, propojení s PC
- Ověřit, zda PC s microbitem komunikuje

- Spárovat přes WebUSB

Co jsme se naučili?

- Procvičili jsme práci s editorem
- Využití displeje pro výpis textů

- Základní bloky „po startu“ a „opakuj stále“

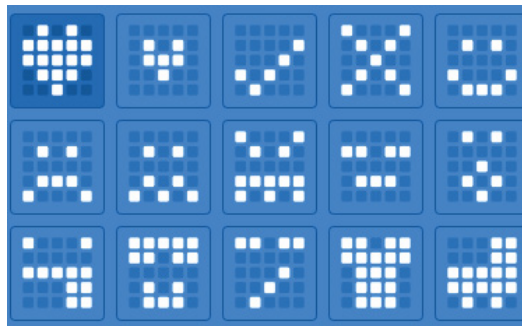
Úkoly a výzvy

- Kombinace dvou různých výpisů – text a ikona

- Vlastní výzkum na téma „co všechno půjde vypsát? Speciální znaky? Čeština? Emotikony?“

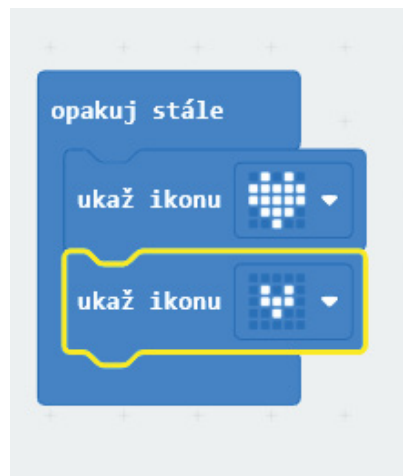
Blikající srdce

Pojďme si zkusit něco dalšího. Už víte, jak zobrazit srdíčko. Všimněte si, že v nabídce ikon jsou dvě srdíčka vedle sebe:



Pojďme si zkusit naše srdce rozpohybovat. Zobrazíme malé, a pak velké, a pak zase malé, a tak donekonečna.

Předpokládám, že vaše řešení vypadá nějak takto:



U microbitu zobrazení ikony chvíli počká, takže naše srdce hezky pravidelně tepe. Ale pojďme jej udělat tak, aby to menší bylo zobrazeno delší dobu než to větší.

Jestli nevíte, jak na to, zkusím napovědět: microbit by měl zobrazit velké srdce, pak menší srdce a pak chvíli **počkat**.

Už víte? Podívejte se do nabídky bloků v sekci Základní... Už vidíte?

(Jen dodám, že čas se zde udává v milisekundách, takže 500 ms je půl sekundy. A to je úplně ideální čas pro naše blikání.)

☞ *Zkuste přepracovat program „tepající srdce“ tak, že použijete funkci na ukázání šipky a budete šipkou točit kolem dokola...*

☞ *Vyzkoušejte funkci „Ukaž číslo“ a vypište postupně čísla 1, 2, 3, 4, 5. Co se stane, když má číslo víc číslic?*

! TIP: *Pokud si chcete program stáhnout, můžete k tomu použít ikonku diskety dole vedle názvu programu. Uloží se na disk jako soubor typu .hex – tedy ve stejné podobě, v jaké ho můžete nahrát do microbitu. Takový program ale můžete přetáhnout zpátky do prostředí MakeCode, a on se opět otevře v původní zdrojové podobě.*

Krátká odbočka pro pokročilé

Už jsem předesílal, že prostředí MakeCode nabízí kromě programování pomocí skládání funkčních bloků i dva „dospělé“ programovací jazyky, totiž Python a JavaScript (přesněji TypeScript). Pomocí přepínače v horní části můžete přepínat mezi blokovým editorem a řádkovými editory obou jazyků. Program, který vzniká, je stále tentýž. Když použijete blok „ukaž číslo 0“ a podíváte se do zdrojového kódu pro Python, uvidíte:

```
basic.show_number(0)
```

Jak správně tušíte, jde o použití metody `show_number` (ukaž číslo) z balíčku `basic` (základní). Zkuste přepsat číslo 0 na 123 a podívejte se zpátky do blokového editoru – hodnota se změnila i zde.

Po přepnutí na JavaScript uvidíte:

```
basic.showNumber(123)
```

Tedy podobně jako u Pythonu. Python se drží konvence s podtržítkem, JavaScript konvence `camelCase`.

Všechny zatím použité bloky (tedy ze sekce Základní) mají své odpovídající metody v obou jazycích. Příklad s blikajícím srdcem vypadá takto:

```
JS    basic.forever(function () {  
JS        basic.showIcon(IconNames.Heart)  
JS        basic.showIcon(IconNames.SmallHeart)  
JS        basic.pause(500)
```



```
JS    })
```

V Pythonu velmi podobně:

```
def on_forever():  
    basic.show_icon(IconNames.HEART)  
    basic.show_icon(IconNames.SMALL_HEART)  
    basic.pause(500)  
  
basic.forever(on_forever)
```

Můžete si svobodně vybrat, jestli použijete skládání bloků, Python nebo JavaScript.

Ve skutečnosti ale MakeCode Python není úplně *plnobodnotný Python*, je to „skript v jazyce pythonského typu“, tedy se stejnými syntaktickými pravidly, odsazením apod. Pro práci s microbitem ale můžete použít „skutečný (micro)Python“ – najdete jej na <https://python.microbit.org/>

Práce s MicroPythonem probíhá tak, že při prvním spárování a nahrání se v microbitu rozběhne interpret jazyka MicroPython, který můžete používat tak, jako se používá „velký“ Python, tj. včetně knihoven, souborů, importů, dokonce můžete využít i interaktivní REPL:

The screenshot displays the MakeCode Python editor interface for a Micro:bit project. The main code editor contains the following Python code:

```
1 # Imports go at the top  
2 from microbit import *  
3  
4  
5 # Code in a 'while True:' loop repeats forever  
6 while True:  
7     print("Hi")  
8     display.show(Image.HEART)  
9     sleep(1000)  
10    display.scroll('Hello')  
11  
12
```

Below the code editor, a terminal window shows the REPL output:

```
micro:bit flashed ✓  
Hi  
Hi  
Traceback (most recent call last):  
  File "main.py", line 10, in <module>  
KeyboardInterrupt:  
MicroPython v1.18 on 2022-11-10; micro:bit v2.1.1 with nRF52833  
Type "help()" for more information.  
>>> print(1+2)  
3  
>>> 
```

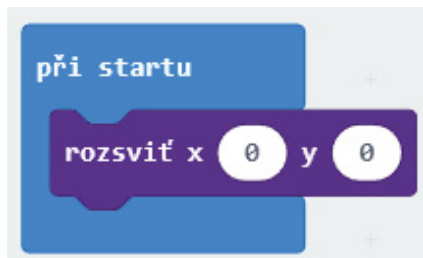
The right-hand side of the interface features a control panel with a play button, a 'Show serial' dropdown menu, and various sliders and buttons for controlling the device's behavior.

2.4 Kreslení

Naučili jste se, jak zobrazit na displeji microbitu tvary, znaky, písmena, číslice a ikony. Celý displej se přitom přepíše. Jak ale rozsvítit jenom jednu LEDku? A kterou?

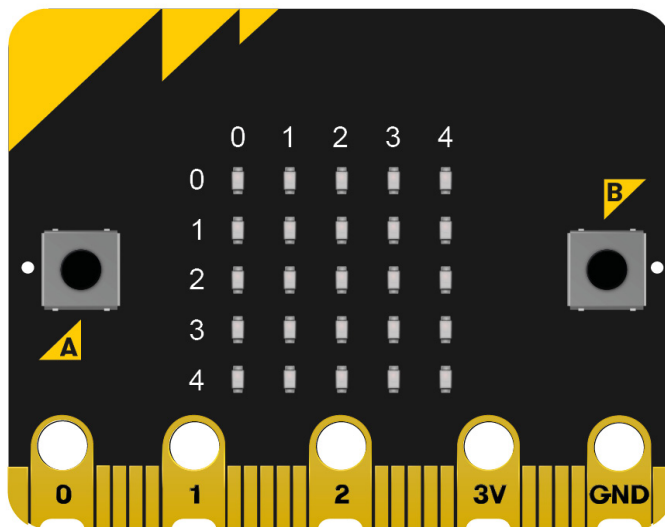
Vytvořte si nový projekt a použijte blok, který najdete v nabídce Displej a jmenuje se „Rozsvít“.

Tento blok má dva číselné parametry, X a Y.



Těmito parametry říkáte programu, která že LEDka má svítit. A jaká čísla tam mají být?

Podívejte se znovu na displej microbitu a všimněte si, že je uspořádán do mřížky 5 řádků × 5 sloupců LED. A teď si představte, že si sloupce i řádky očíslováme od 0 do 4 z levého horního rohu. Takto:



Každá jednotlivá LED může být určena kombinací dvou hodnot: čísla sloupce a čísla řádku. Číslo sloupce je X (stejně jako v matematice se u grafů vodorovná osa označuje X), číslo řádku je Y (opět stejně jako v matematice, ale tentokrát je nula nahoře a hodnota roste směrem dolů).

Pokud tedy řekneme microbitu, aby rozsvítil LED na pozici $X=0$ a $Y=0$ (zkráceně můžeme zapsat „na pozici $[X, Y]$ “, tedy $[0,0]$), měla by se rozsvítil LED v levém horním rohu.

Vyzkoušejte to.

☞ Zkuste si rozsvítil všechny LED ve všech rozích. Zkuste rozsvítil jen prostřední. Zkuste pomocí funkce „rozsvít“ nakreslit vodorovnou čáru uprostřed.

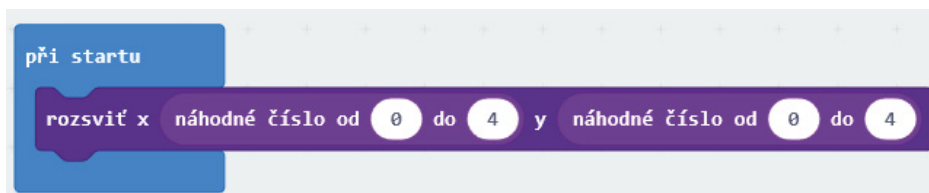
☞ Co se stane, když zadáte hodnotu větší než 4, nebo menší než 0?

Druhý blok, který si teď ukážeme, je blok „Zhasni“. Jak už název napovídá, měl by rozsvícenou LEDku opět zhasnout. Se znalostí těchto dvou bloků už dokážete rozblikat třeba prostřední LED. Zkuste si to.

A teď uděláme trochu divočejší blikání, ano? Místo toho, abychom říkali microbitu, kterou LED má rozsvítil, necháme to na něm, ať si rozsvítí nějakou náhodně.

Použijeme zase blok „rozsvít“ a jako hodnoty X a Y nezadáme číslo, ale přetáhneme tam blok, který najdeme v nabídce Matematika a který se jmenuje „Náhodné číslo od-do“.

Všimněte si, že tento blok má oválný tvar a zapadá tak do bílého prostoru pro číslo. Nemůžeme ho tedy použít jako funkční blok, a stejně tak nemůžeme například blok „čekej“ použít jako hodnotu.



Všimněte si, že jsem změnil hodnoty „od“ a „do“. Blok teď můžeme číst tak, že se má rozsvítil bod na pozici $[X, Y]$, kde X i Y jsou náhodná čísla od 0 do 4.

Když kód zkusíte spustit, rozsvítí se nějaký bod. Pokud jej spustíte znovu, rozsvítí se pravděpodobně jiný. Hezké, ale poněkud nudné. Zkuste proto celý blok „rozsvít“ přesunout do „Opakuj stále“.

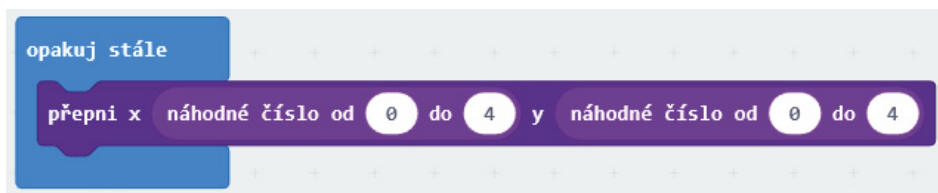
To už je jiná podívaná, že? Displej se během krátké chvíle zaplní rozsvícenými body, a pak...? A pak nic! Program sice rozsvěcí stále a stále nové LED na náhodných pozicích, ale ty už svítí, takže se změna nijak neprojevuje.

Pro přesnost je potřeba dodat, že čísla nejsou náhodná, ale pseudonáhodná. Je to proto, že jsou počítána matematicky z nějaké úvodní hodnoty, a nejsou tedy úplně náhodná. Získat v počítači opravdu náhodné číslo je problém, a tak se používají převážně tato

pseudonáhodná, která jsou pro naprostou většinu běžných aplikací dostatečně „nepředvídatelná“.

Zkuste funkci „rozsvít“ nahradit funkcí „přepni“ (opět v nabídce Displej). Jako X a Y dejte opět náhodná čísla a spusťte.

Funkce „přepni“ pracuje tak, že pokud LED na zadané pozici svítí, zhasne ji. Pokud je zhasnutá, rozsvítí ji. Proto se při každém zavolání vizuálně obrazec změní.



```
JS function led.plot(x,y)
JS function led.unplot(x,y)
JS function led.toggle(x,y)
JS function randint(min,max):number
⊞ def led.plot(x,y)
⊞ def led.unplot(x,y)
⊞ def led.toggle(x,y)
⊞ def randint(min,max):number
```

Metodický list

Co si připravit?

- Microbit, kabel micro-USB, propojení s PC
- Ověřit, zda PC s microbitem komunikuje

- Spárovat přes WebUSB

Co jsme se naučili?

- Ovládání displeje po jednotlivých bodech
- Adresace pozic na displeji pomocí dvojice souřadnic

- Náhodná čísla

Úkoly a výzvy

- V nabídce Displej – Více je blok pro rozsvícení LED s určitým jasem. Co jej použít u náhodného zaplňování displeje? Jak bude fungovat?

2.5 Vstup

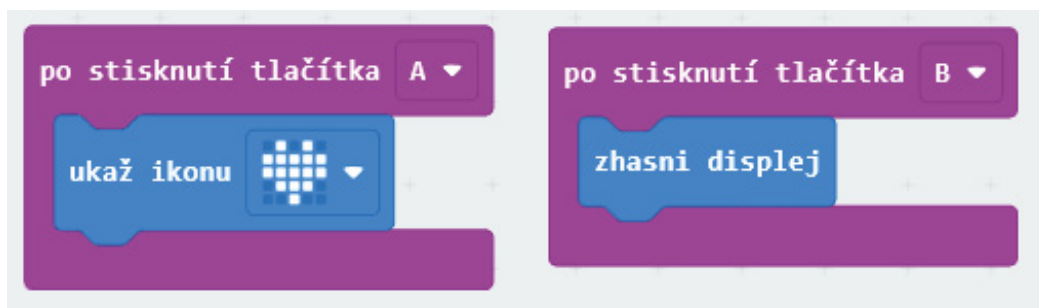
Určitě jste si všimli, že microbit má na přední straně dvě tlačítka, označená A a B. A teď nastala chvíle, kdy je zapojíme do našich experimentů.

A nejen tlačítka! Microbit obsahuje mnoho senzorů a vstupů, které si postupně představíme. Pokud chcete mít představu, co všechno naleznete na tak malé destičce, podívejte se do úvodní kapitoly.

Podívejte se do nabídky Vstup. Hned první funkční blok se jmenuje „Po stisknutí tlačítka...“ Přidejte ho na plochu.

Všimněte si, že se vzhledem ke svému tvaru nedá zapojit ani do „Po startu“, ani do „Opakuj stále“. Je samostatný a stojí mimo. Funguje vlastně nezávisle na vašem programu (programátoři jej nazvou třeba „obsluha asynchronní události“).

Dejme do něj naši oblíbenou funkci „ukaz ikonu srdce“. Přidejte ještě jeden tento blok, ale změňte tlačítko z A na B, a do něj dejte funkci „zhasni displej“.



Funkce tohoto programu je tak jednoduchá, že snad ani není potřeba popisovat, co se stane. Proto si tento program nahrajte do microbitu a vyzkoušejte, že dělá opravdu to, co má.

☞ *Psal jsem, že blok „Po stisknutí tlačítka“ funguje nezávisle na hlavním programu. Co se stane, když hlavní program bude stále dokola kreslit jinou ikonu, třeba smajlík, a přitom někdo zmáčkne tlačítko A?*

Pojďme si udělat náročnější program, který bude počítat, kolikrát jsme stisknuli tlačítko.

Počítadlo

Tak, jak na to?

Je jasné, že základ bude blok „Po stisknutí tlačítka“. A v něm bude blok „Zobraz číslo“. A logicky, že tam bude jednička... Nebo ne?

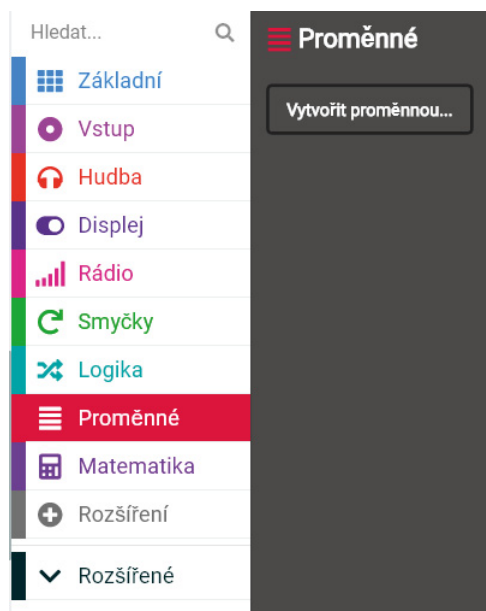
To neřeší náš problém. My potřebujeme, aby se někde počítalo každé stisknutí tlačítka, a pak se zobrazil celkový součet. Potřebujeme nějaké místo, kam si dáme na začátku nulu, a po každém stisknutí přičteme jedničku...

Přesně takové místo v programování existuje, a říká se mu „proměnná“. To proto, že hodnota se může měnit. Proměnná má nějaké své jméno, třeba „A“ nebo „xfh734“, nebo třeba „pocitadlo“.

Můžeme si proměnnou představit jako pojmenovaný kousek papíru, na kterém je napsáno obyčejnou tužkou číslo. Když je potřeba, můžeme ho vygumovat a napsat nové. Nebo jako tabuli, na kterou píšeme číslo křídou, smažeme a přepíšeme. Nebo jako pojmenovanou přihrádku, kam si vložíme číslo. Zkrátka jako nějaké pojmenované místo, kde může být nějaká hodnota (nejen číslo), která se může změnit.

Dobrym zvykem je pojmenovat proměnné tak, aby bylo jasné, co v nich je, takže například ne Bflm, ale „pocitadlo“. Taky je dobrým zvykem nepoužívat háčky a čárky. Některé jazyky to ani nedovolí. A důležité je nezapomenout na to, že většina jazyků má ještě nějaká další pravidla pro jména proměnných, takže ideální je držet se malých a velkých písmen, číslic a znaku podtržítka.

Takže si vytvoříme svoji první proměnnou. V hlavním menu je nabídka „Proměnné“. Klikněte a zadejte „Vytvořit proměnnou“.



Proměnnou pojmenujeme „pocitadlo“. V sekci „proměnné“ se objeví dva nové funkční bloky „nastav proměnnou“ a „změň proměnnou“ a jeden blok, který reprezentuje hodnotu proměnné (ano, to je ten oválný). Rovnou je zkusíme použít.

Do bloku „Při startu“ vložte dva bloky:

- Nastavení proměnné pocitadlo na hodnotu 0
- Zobrazení čísla

Jako hodnotu pro zobrazené číslo použijte hodnotu proměnné „pocitadlo“.

Výsledek by měl vypadat takto:



Po spuštění se na displeji ukáže nula. Jupí. Zvládli jsme vytvořit proměnnou a použít ji. A teď použijeme ještě blok „Změň proměnnou“. Přetáhněte ho do bloku „Po stisknutí tlačítka A“ a – máme vlastně hotovo. Po stisknutí tlačítka A se stav proměnné „pocitadlo“ zvedne o 1.

Když program přeložíte a nahrajete, zjistíte jednu zradu. Tlačítko sice mačkáte, ale na displeji je stále „0“. Důvod je jednoduchý: Sice jsme změnilí hodnotu, ale zapomněli jsme ji vypsat na displej!

Klikněte pravým tlačítkem na blok „Zobraz číslo ,pocitadlo“ a vyberte možnost Klonovat. Celý blok i s vloženou hodnotou proměnné „pocitadlo“ se zkopíruje na plochu. Vezměte jej a vložte za zvýšení stavu proměnné (tedy do bloku „Po stisknutí tlačítka A“).

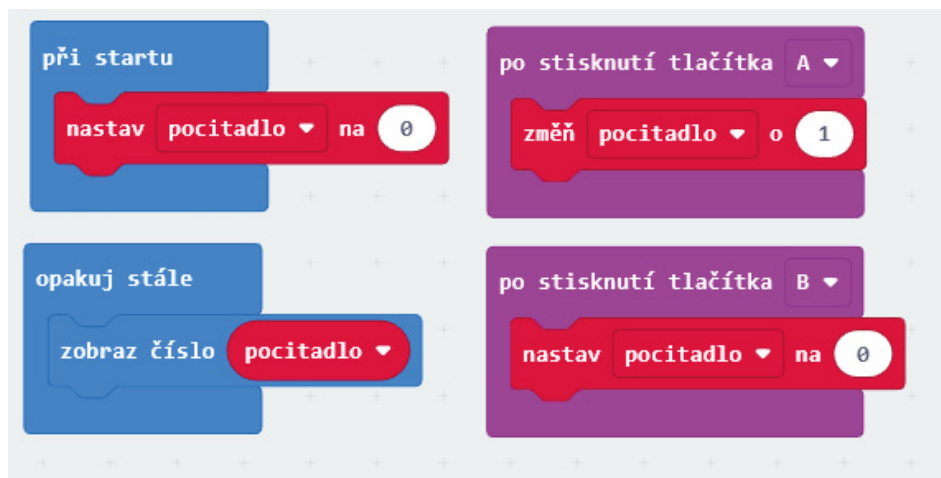
Teď už program krásně funguje. Počítá dál a dál...

☞ Přidejte funkci, která po stisknutí tlačítka B počítadlo opět vynuluje.

Předpokládám, že tento úkol byl pro vás hračkou. Vlastně stačilo jen zopakovat to, co se děje v bloku Při startu.

☞ Co se stane, když stisknete tlačítko A několikrát rychle za sebou?

A teď se podívejte na tento program. Co se změnilo? Jak se změní chování? A bude to vůbec fungovat?



Hlavní rozdíl je ten, že čísla jsou zobrazována neustále. Což u čísel 0 až 9 vidět není, ale od 10 výš už ano. Druhá změna je, že když stisknete tlačítko několikrát rychle po sobě, tak počítadlo skočí z 0 třeba rovnou na 4. Předchozí verze počítala 1... 2... 3... 4.

Důvod je ten, že microbit po zobrazení čísla čeká zhruba půl sekundy. Proto se může stát, že zobrazí hodnotu 0, půl sekundy čeká, vy během té doby dvakrát stisknete tlačítko, a pak, když se znovu zobrazuje hodnota proměnné, tam je už 2. Předchozí verze na dvojitý stisk zobrazila 1 a čekala půl sekundy. A po celou tu dobu čekala i informace o druhém stisku tlačítka...

Čekání se u funkce Zobraz číslo dá potlačit. Přepněte se např. do JavaScriptu, najdete řádek, kde je příkaz `basic.showNumber(pocitadlo)` a změňte jej na `basic.showNumber(pocitadlo,0)`. Stisknutí budou započítána a zobrazena ihned. Bohužel, nulové čekání způsobí, že vícemístná čísla budou po displeji létat tak rychle, že nebudou k přečtení...

Metodický list

Co si připravit?

- Microbit, kabel micro-USB, propojení s PC

Líbila se Vám ukázka
z knihy? Knihu
můžete zakoupit na
HWKITCHEN.cz