

PROGRAMOVANIE HARDVÉRU, ROBOTIKA A IoT

V PRAKTICKÝCH PRÍKLADOCH



Obsah

Úvod	5
Vývojové dosky pre výučbu, hobby riešenia a prototypovanie	6
Mikropočítačová doska BBC Micro:bit	6
<i>Konceptia mikropočítačovej dosky</i>	8
<i>Hardvér a senzory</i>	8
<i>Jednoduchá pripojiteľnosť</i>	9
<i>Zbernica a rozširujúce moduly</i>	9
<i>Programovanie pomocou blokov aj v Pythone</i>	11
<i>Micro:bit – digitálny výstup, blikanie LED diódami</i>	13
<i>Zavedenie kódu aplikácie do dosky Micro:bit</i>	15
<i>Micro:bit – digitálny vstup, obsluha tlačidla</i>	15
<i>Micro:bit – analógový vstup, obsluha tlačidla</i>	17
<i>Micro:bit – generovanie zvuku</i>	20
<i>Micro:bit – obsluha udalostí od hardvéru</i>	21
<i>Micro:bit – nastavenie farby svetla RGB diódy tlačidlami</i>	22
<i>Micro:bit – rozlíšenie zmeny úrovne napätia na vstupe</i>	23
<i>Micro:bit – spínanie a ovládanie zariadení</i>	24
<i>Micro:bit – ovládanie jednosmerných motorov</i>	27
<i>Micro:bit – ovládanie servomotorov</i>	28
<i>Micro:bit – stavebnica Grove na jednoduché zapojenia hardvérových komponentov</i>	30
<i>Micro:bit + Grove – snímanie miest v priestore</i>	32
<i>Micro:bit + Grove – meranie vzdialenosti ultrazvukom</i>	33
<i>Micro:bit + Grove – hudobný nástroj Theremin</i>	34
<i>Micro:bit + Grove – zobrazovanie na displeji</i>	35
<i>Micro:bit + Grove – meranie a zobrazovanie vzdialenosti</i>	36
<i>Micro:bit + Grove – pásik farebných RGB diód</i>	37
<i>Micro:bit + Grove – zabezpečujúce zariadenie indikujúce otvorenie skrinky, zásuvky alebo tašky</i>	38
<i>Micro:bit + Grove – zabezpečujúce zariadenie indikujúce priblíženie sa</i>	39
<i>Micro:bit Experiment Box na jednoduché pokusy s hardvérom</i>	40
<i>Micro:bit Smarthom kit na projekty simulujúce inteligentný dom</i>	41
Programovací jazyk Python	49
Aplikácie v Pythone pre Micro:bit	50
<i>Micro:bit Python - zobrazovanie na maticovom displeji</i>	51

<i>Micro:bit Python – definovanie vlastných obrázkov</i>	52
<i>Micro:bit Python – zisťovanie stavu tlačidiel</i>	54
<i>Micro:bit Python – digitálne vstupy a výstupy</i>	54
<i>Micro:bit Python – generovanie PWM signálu na emulovanie analógového výstupu</i>	55
<i>Micro:bit Python – ovládanie motorov a servomotorov pomocou PWM signálu</i>	57
<i>Micro:bit Python – generovanie zvuku</i>	57
<i>Micro:bit Python – Meranie teploty</i>	58
<i>Micro:bit Python – čítanie údajov z akcelerometra</i>	58
<i>Micro:bit Python – identifikácia polohy dosky a giest ruky s doskou</i>	59
<i>Micro:bit Python – kompas využívajúci magnetický senzor</i>	59
<i>Micro:bit Python – údaje v súboroch</i>	59
<i>Micro:bit Python – bezdrôtová komunikácia</i>	60
Roboticky a robotické podvozky s Micro:bitom	63
<i>Konštrukcia a programovanie robotov – ovládanie jednosmerných motorov</i>	64
<i>Diaľkové ovládanie pomocou Micro:bitu</i>	69
<i>Podvozok Ring:bit pre Micro:bit</i>	75
<i>Diaľkové ovládanie pomocou robotického podvozku Ring:bit druhým Micro:bitom</i>	78
<i>Robotický podvozok Ring:bit jazdiaci za zdrojom svetla</i>	80
<i>Doplnok k podvozku Ring:bit pre Micro:bit na sledovanie čiary</i>	81
<i>Ultrazvukový senzor na meranie vzdialenosti pre robot s Micro:bitom</i>	85
<i>Joystick na ovládanie zariadení s Micro:bitom</i>	88
<i>Ovládanie modelov Lego Technic mikropočítačom BBC Micro:bit</i>	95
<i>Príklad prototypu modelu LEGO Technic mikropočítačom BBC Micro:bit 2</i>	96
<i>Robotika – robot s kolesami Mecanum sa môže pohybovať všetkými smermi</i>	103
<i>Príklad s priamym ovládaním servomotorov</i>	106
<i>Príklady využívajúce knižnicu Wukong s podporou pre kolesá Mecanum</i>	108
<i>Micro:bit ako riadiaca jednotka populárneho robota mBot</i>	112
Pripojenie Micro:bitu na WiFi a komunikácia s cloudovou IoT službou	116
<i>Platforma ESP</i>	118
<i>Modul IoT: bit pre Micro:bit</i>	119
<i>Cloudová platforma ThingSpeak</i>	120
<i>Príklady - meranie a posielanie údajov do cloudu ThingSpeak</i>	122
<i>Zobrazovanie na OLED displeji (ak nemáte displej môžete vynechať)</i>	122
<i>Posielanie údajov do cloudu</i>	125
<i>Zobrazovanie údajov v službe ThingSpeak</i>	128
<i>Meranie, posielanie do cloudu a zobrazovanie údajov z akcelerometra</i>	129

Jednoduchá automatizácia s využitím IFTTT133

Úvod

Cieľom tejto publikácie je poskytnúť praktické návody na zaujímavé a pritom jednoduché riešenia využívajúce mikropočítačové dosky, ktoré sa využívajú pri výučbe informatiky na mnohých našich školách. Väčšina tém a príkladov je určená pre populárnu mikropočítačovú dosku BBC Micro:bit, ktorá má modernú koncepciu, výkonný hardvér a bohatú senzorovú výbavu. Niektoré témy sú z technických, prípadne praktických dôvodov demonštrované na iných mikropočítačových doskách.

Žiaci a študenti môžu pomocou názorných príkladov pochopiť, čo je to algoritmus a následne vytvárať a ladiť jednoduché programy na ovládanie hardvérových komponentov – snímanie údajov z rôznych typov senzorov, spracovávať tieto údaje a ovládať rôzne akčné členy ako sú LED diódy, motorčeky, servomotory a podobne. Poznatky potom môžu zúročiť v stati venovanej komplexnejším riešeniam ako sú roboty, či prototypy riešení pre inteligentný dom. Často práve pohľad na fungujúce riešenie je posledným kameňom do mozaiky umožňujúci pochopiť ako fungujú jednoduché algoritmy.

Veľa príkladov zasahuje do oblasti robotiky, keď mikropočítačová doska ovláda robotický podvozok vybavený potrebnými senzormi. Výber tohoto námetu zodpovedá aktuálnemu stavu „domácej“ robotiky. Ak nepočítame hračky, určite doma nemáte humanoidného robota, ani robotickú ruku na kuchynskej linke, ktorá pomáha pri varení a iných domácich prácach. Mnohí z vás však doma majú viac, alebo menej inteligentný robotický vysávač. Pohľad zospodu prezradí, že robotický vysávač sa pohybuje na rovnakom princípe, ako všetky robotické podvozky v príkladoch z tejto publikácie.

Pokročilejšie riešenia využívajú pripojenie k internetu, takže sú súčasťou internetu vecí. Prečo je praktické a v mnohých prípadoch dokonca nevyhnutné, aby ste údaje zo svojho hobby riešenia ukladali na internet do cloudových služieb? Predstavte si jednoduché riešenie, ktoré meria teplotu u vás v byte, prípadne v skleníku a podobne. A vy by ste si pomocou mobilného telefónu chceli tento údaj na diaľku skontrolovať. Pripojenie cez bluetooth umožňuje komunikáciu na krátke vzdialenosti – typicky do 10 metrov. Rovnako aj priama komunikácia cez WiFi si vyžaduje, aby obidva zariadenia, teda inteligentný teplomer, prípadne iné zariadenie a taktiež smartfón boli pripojené do rovnakej siete, napríklad k routeru u vás doma. Aby ste sa mohli pohodlne pripojiť zo smartfónu bez ohľadu na to, kde sa nachádzate, najjednoduchšie sa to dá vyriešiť tak, že senzor bude posielať údaje do cloudovej služby a vy sa v prípade potreby k tejto službe pripojíte a údaje si prečítate.

Internet vecí je jedným z najrýchlejšie sa rozvíjajúcim trendom nielen v IT, ale prakticky vo všetkých odvetviach. Informácie získané analýzou obrovského objemu údajov z prepojených zariadení má potenciál zmeniť smerovanie spotrebného priemyslu, výrobných odvetví, zdravotníctva, výskumu, bývania a podobne. Inteligentné prepojené zariadenia budú všade vrátane našich domácností, dokonca aj oblečenia. Dokážu reagovať na podnety, ktoré sa dosiaľ ignorovali. Internet vecí je obrovská príležitosť hlavne pre startupy, čiže začínajúce inovatívne firmy. Nielen z hľadiska nových výrobkov, ale aj vylepšenia existujúcich zariadení, ktoré po doplnení vhodného riadiaceho modulu a senzorov chytia druhý dych alebo, ešte výstižnejšie vyjadrené, získajú šiesty zmysel. Pridaná hodnota vznikne buď primárne doplnením zariadení o nové funkcie, prípadne znížením jeho energetickej náročnosti, alebo sekundárne analýzou údajov zozbieraných z prepojených zariadení.

Vývojové dosky pre výučbu, hobby riešenia a prototypovanie

Arduino, BBC Micro:bit, Raspberry Pi, Nucleo... To sú názvy najpopulárnejších vývojových dosiek s mikroprocesormi, primárne určené pre amatérske hobby riešenia, na výučbu programovania a robotiky, avšak tieto dosky využívajú aj v mnohých firmách pri vývoji a ladení prototypov. Dôvodom popularity týchto dosiek je sprístupnenie použitia miniatúrnych čipov určených pre plošnú montáž. Mikropočítačové dosky majú vyvedené signály zbernice na konektory s rozstupom pinov, alebo dutiniek 2.5 mm, na ktoré je možné pripojiť vodiče s dutinkami, alebo pinmi. Doska BBC Micro:bit má dokonca tri porty, napájacie napätie a zem vyvedené na dutinky, do ktorých môžete vložiť vodiče s banánikmi, alebo prichytiť vodiče s krokosvorkami.

Prakticky na všetky dosky je možné nasúvať rozširujúce moduly, buď špecializované, napríklad so senzormi, či výkonovými členmi pre krokové motory, alebo univerzálne s prepajovacím poľom, alebo plošným spojom s maticou pokovených otvorov, do ktorých je možné prispájkovať potrebné súčiastky a následne ich poprepájať. Pojem „nasunúť“ je relatívny. Na dosky typu Arduino, či Raspberry Pi sa moduly nasúvajú. Doska BBC Micro:bit sa do rozširujúcich modulov zasúva. Vytvoriť takýmto spôsobom riadiacu jednotku pre robotické vozidlo aj s výkonovou elektronikou na riadenie motorov je otázka jedného popoludnia.

Pri výbere typu mikropočítačovej dosky je potrebné prihliadať nielen na skúsenosti programátorov, ale aj na účel nasadenia napríklad v rámci študentského projektu. Na jednoduchý zber údajov je zbytočné používať výkonné jednodoskové minipočítače typu Raspberry Pi. Zbytočne na nich beží operačný systém, ktorý zvyšuje spotrebu zariadenia. Vhodnejšie sú vývojové dosky, ktoré obsahujú jednočipový mikrokontrolér s vyvedenými zbernicami a ďalšími podpornými perifériami. Pri výbere dosky potrebné zvážiť, či má vyvedené potrebné zbernice, ktoré budete potrebovať pre pripojenie senzorov. A samozrejme možnosť pripojenia k internetu. Hlavnou nevýhodou vývojových kitov nie je ich cena, ale objem, hmotnosť a hlavne spotreba. Typický rozmer čipu je menej ako 1x1 cm, naproti tomu vývojová doska má rozmery 5 x 10 cm, čiže minimálne 50 x viac. Približne rovnaký je aj pomer hmotnosti. Samozrejme k rozmerom a váhe čipu treba pripočítať aj plošný spoj a okolité súčiastky ale ten sa dá navrhnuť rozmerovo veľmi úsporne.

Zjednodušene povedané vývojová doska je perfektná do robotického vozidla a podobných väčších zariadení, ale ako riadiaca jednotka miniatúrneho zariadenia, či inteligentných hodiniek, kde záleží na rozmeroch, alebo drona, kde záleží na hmotnosti sa určite nehodí. Situácia sa však aj v tomto ohľade zlepšuje a na trh prichádzajú miniaturizované verzie známych dosiek, niektoré aj vo verzii na zabudovanie do nositeľných zariadení a odevu.

Mikropočítačová doska BBC Micro:bit

Video: <https://www.youtube.com/watch?v=60MYR0dnhoo>

Zdá sa vám názov BBC Micro povedomý? Máte pravdu. Tento mikropočítač vyrábala v 80-tych rokoch minulého storočia spoločnosť Acorn Computer pre vzdelávací projekt BBC Computer Literacy Project a využíval sa vo väčšine škôl v Spojenom kráľovstve. Napriek pomerne vysokej cene sa využíval aj ako domáci počítač.



BBC Microcomputer z 80-tych rokov minulého storočia

Vývoj ide po špirále a dopyt po názorných pomôckach na výučbu programovania pretrváva. Preto bol v roku 2016 vytvorený jednodoskový mikropočítač BBC micro:bit s rozmermi 50 x 40 mm, ktorý je 70 x menší a 20 x výkonnejší, než pôvodný BBC Micro. Rozdielne boli aj zámery tvorcov týchto mikropočítačov. Cieľom BBC Micro z 80-tych rokov bolo sprístupniť drahý mikropočítač žiakom, aby sa túto na svoju dobu revolučnú novinku naučili používať, prípadne aj programovať, čiže aby rozvíjali svoju kreativitu týmto perspektívnym smerom.

V súčasnosti sú počítače, tablety aj smartfóny bežnou súčasťou bežného života všetkých vekových kategórií. Pracovný trh vytvára veľký dopyt pre IT špecialistov a programátorov a tento dopyt sa bude stupňovať. Preto si tvorcovia mikropočítačovej dosky BBC Micro:bit vytýčili rovnaký cieľ ako ich predchodcovia – čiže aby žiaci a študenti rozvíjali svoju kreativitu perspektívnym smerom a začali programovať, prípadne vytvárať jednoduché konštrukcie a nenápadne hravým a tvorivým spôsobom prenikli do tajov odboru nazývaného mechatronika.



Doske BBC Micro:bit môžeme pokojne dať prívlastok školský mikropočítač. Je výsledkom spoločnej iniciatívy širokého spektra súkromných i verejných organizácií, ktorej cieľom bolo vytvorenie lacnej platformy na výučbu informatiky. Iniciatívu zastrešila britská verejnoprávna televízna spoločnosť BBC. Tá v roku 2016 rozdistribuovala jeden Micro:bit každému žiakovi siedmeho ročníka v Spojenom Kráľovstve (11 až 12 roční). Doska je dostupná a lacná, v e-shopoch ju kúpite za 15 eur. BBC Micro:bit

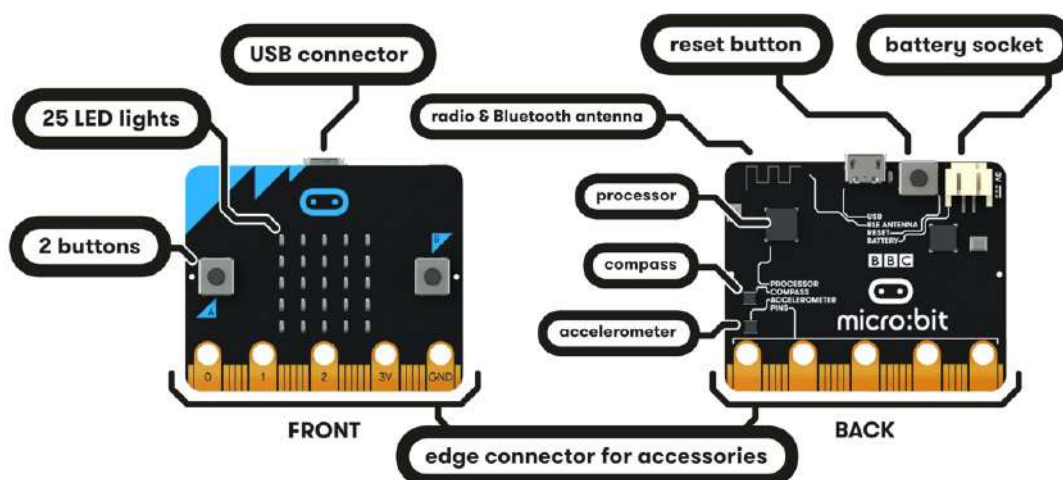
je rozšírený aj na mnohých našich školách, kde túto dosku s príslušenstvom distribuovali rôzne občianske združenia, niektoré z nich robia aj aktivity na edukáciu učiteľov.

Koncepcia mikropočítačovej dosky

Nech vás nemýli protiklad pojmu školský mikropočítač a konštrukčného formátu dosky BBC Micro:bit. Namiesto stolného počítača s klasickou klávesnicou má na výučbu algoritmizácie a programovania pre žiakov slúžiť malá doštička plošných spojov? Konštruktéri Micro:bitu však veľmi dobre vedeli, čo žiaci a študenti potrebujú. Doštička polovičnej veľkosti kreditnej karty sa pohodlne do vrečka košele a má zanedbateľnú hmotnosť, takže sa dá bez problémov nosiť do školy a domov. Doska má jednoduchý maticový displej z LED diód. Pôvodný Microbit využíval na zobrazovanie televízor, takže sa využíval len v školských učebniach. Ďalšou výhodou je univerzálnosť, doska sa dá využiť na výučbu programovania hravou formou, keď žiak môže vytvoriť jednoduchý grafický vzor, naprogramovať dosku ako hracu kocku, ktorá sa aktivuje potrasením indikovaným akcelerometrom a na maticovom displeji zobrazí náhodnú hodnotu. Možnosť bezdrôtovo prepojiť viac dosiek rozširuje množstvo pre žiakov atraktívnych a interaktívnych scenárov. K dispozícii je aj veľké množstvo príslušenstva, takže BBC micro:bit je možné použiť na vytvorenie riadiaceho systému pre akúkoľvek hobby konštrukciu, môžete vytvoriť robotický podvozok, riadiacu a spínaciu jednotku ovládajúcu svetlá a iné subsystemy modelu inteligentného domu a podobne. Bluetooth komunikácia umožňuje prepojiť micro:bit so smartfónom. K dispozícii je aj modul, ktorý umožňuje pripojiť BBC Micro:bit k Wifi a povýšiť túto školskú mikropočítačovú dosku na plnohodnotnú IoT platformu. Takže ak to zosumarizujeme, toto malé a lacné zariadenie poslúži ako pomôcka pre výučbu programovania, robotiky a pre starších žiakov a študentov aj na IoT projekty

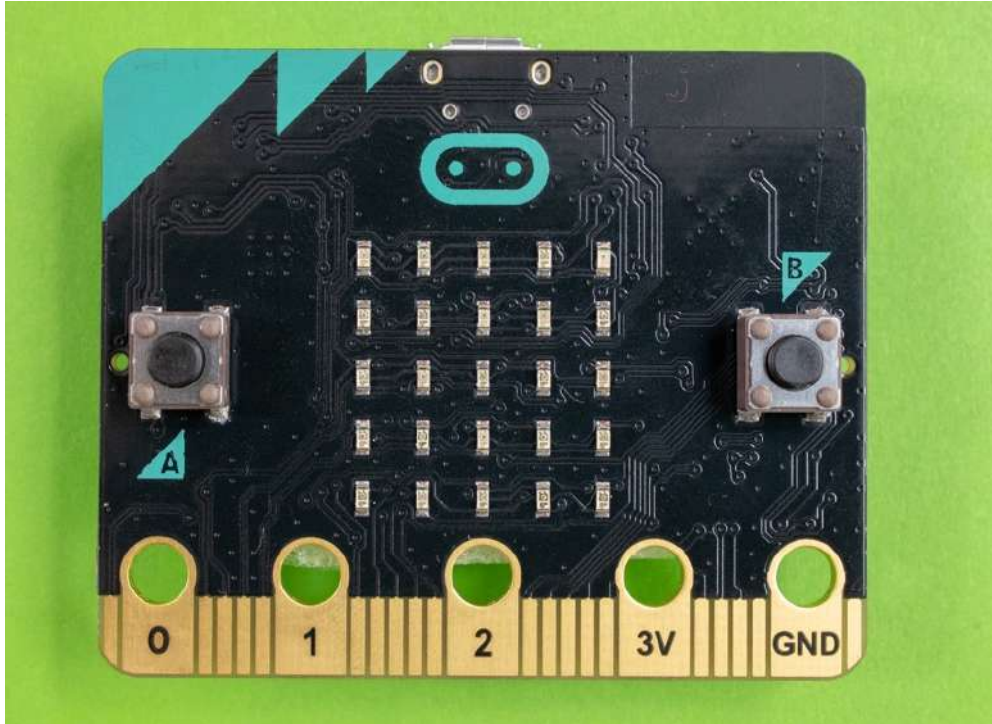
Hardvér a senzory

Hardvérová výbava BBC micro:bit je oveľa bohatšia, než by sa na prvý pohľad mohlo zdať. Na strane spojov sú dve tlačidlá označené ako A a B a maticový displej z 25 LED diód. Ten v reverznom móde funguje aj ako snímač intenzity okolitého svetla. Na strane súčastok je senzor pohybu, čiže akcelerometer, snímač naklonenia, kompas, čiže magnetometer a senzor teploty. Micro:bit podporuje dva druhy bezdrôtovej komunikácie - Bluetooth Low Energy (BLE) pre komunikáciu s mobilnými zariadeniami a rádiový prenos pre komunikáciu s inými zariadeniami BBC micro:bit. Taktiež je možná sériová komunikácia s počítačom cez USB kábel. Srdcom dosky je 32-bitový mikrokontrolér ARM Cortex-M0 taktovaný na 16MHz s 256KB Flash a 16kB RAM. Napájanie je realizovateľné tromi spôsobmi: cez USB kábel; dvoma batériami v špeciálnom držiaku bežne dodávanom s BBC micro:bit alebo pripojením batérie priamo na napájacie piny.



Jednoduchá pripojiteľnosť

Konektor na okraji plošného spoja má päť širokých kontaktov s pokovenými otvormi do ktorých je možné pripnúť vodiče s krokosvorkami, alebo zasunúť „banánikové“ konektory s priemerom 4 mm. Medzi nimi je 20 ďalších úzkych kontaktov. Tie je možné využiť keď vložíte BBC micro: bit do vhodného konektora.



Široké kontakty s pokovenými otvormi sú označené 0, 1, 2, 3V a GND, čiže elektrickú zem. Piny 0, 1 a 2 sú flexibilné GPIO (general-purpose input/output) porty, čiže vo voľnom preklade vstupy a výstupy na všeobecné použitie. Tieto tri piny taktiež umožňujú čítať analógové napätia pomocou analógovo-digitálneho prevodníka (ADC). Piny 3V a GND sú určené na napájanie dosky. Napätie 3V získate z dvoch do série zapojených tužkových batérií. Ak je doska napájaná z USB alebo z batérie cez konektor na druhej strane dosky, môžete použiť 3V pin na napájanie periférnych zariadení.

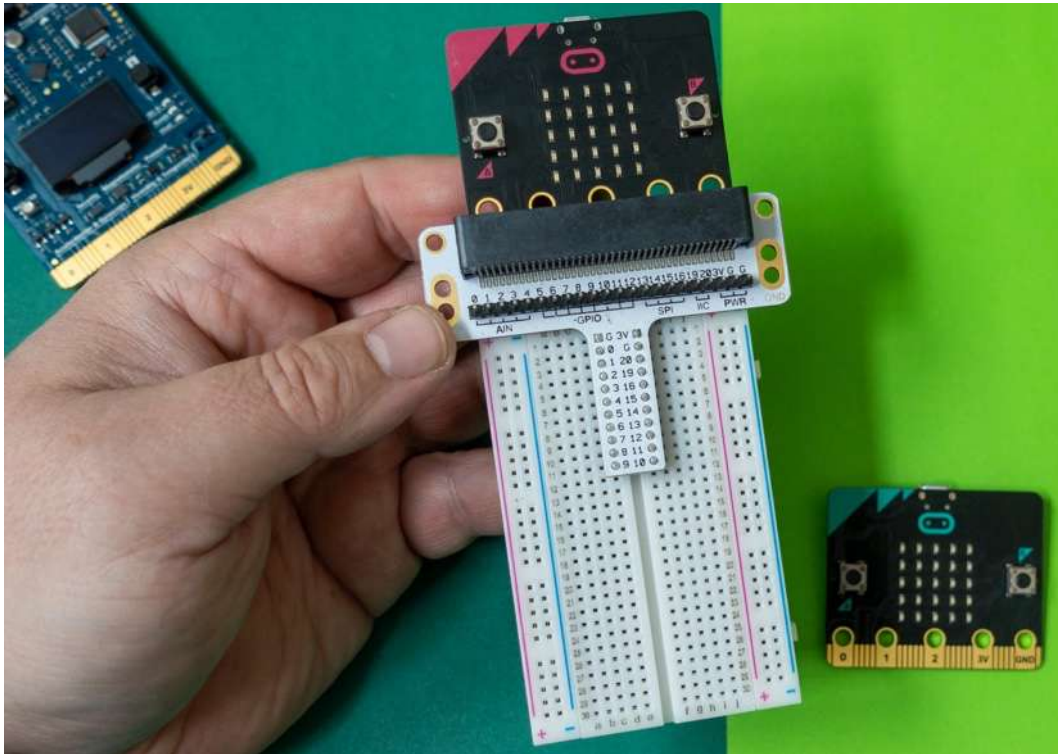
Cez široké kontakty môžete pomocou krokosvoriek, alebo banánikov pripojiť rôzne súčiastky a rozšíriť tak možnosti samotnej dosky. Tri piny, ktoré môžu fungovať buď ako vstupné výstupné digitálne porty, alebo ako analógové vstupy poskytujú oveľa viac možností, než by sa na prvý pohľad mohlo zdať. Bez toho, aby ste k nim čokoľvek pripájali ich môžete po príslušnom naprogramovaní využiť aj ako dotykové tlačidlá. Aby to fungovalo musíte sa druhou rukou dotýkať pinu GND. Cez rezistory môžete k týmto pinom pripojiť LED diódy, rôzne typy senzorov, z hliníkovej fólie vytvorené dotykové plochy, prípadne komponenty elektrotechnických stavebníc využívajúce banániky. Dokonca aj stavebnica robotického podvozku využíva na pripojenie riadiacej dosky len široké kontakty.

Zbernica a rozširujúce moduly

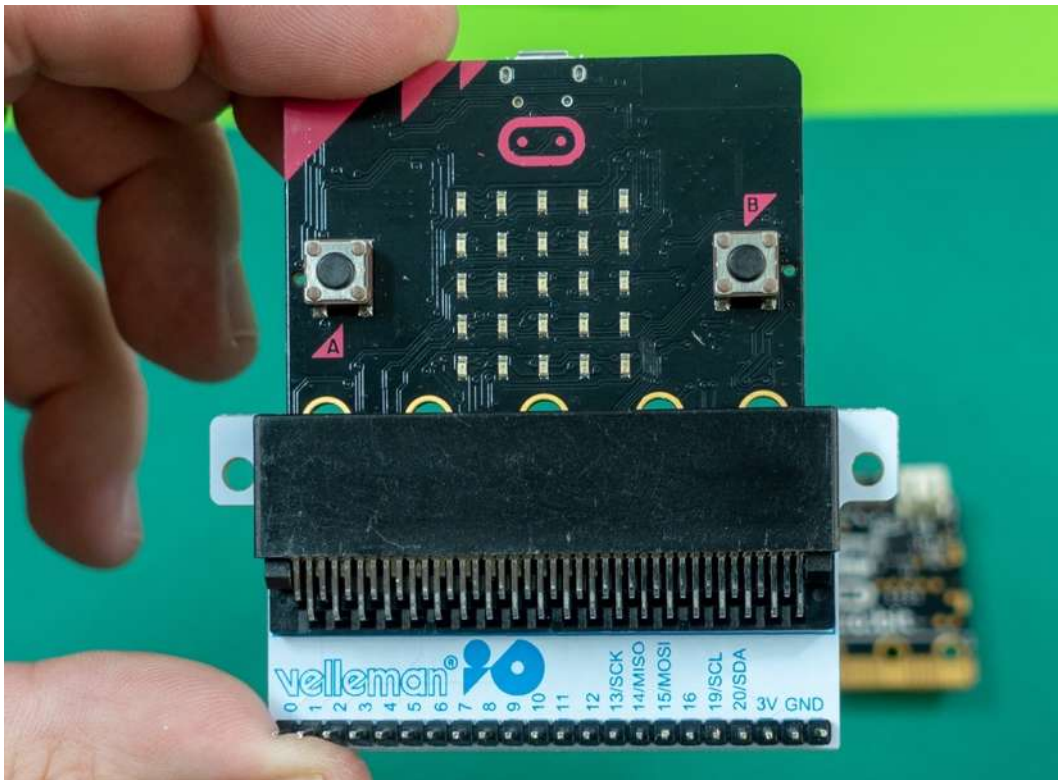
Video <https://www.youtube.com/watch?v=FFc0J6c7y8k>

Veľkou výhodou mikropočítačovej dosky BBC micro:bit je modulárnosť. Pozorný pohľad na zbernicu prezradí, že medzi širokými kontaktmi sú aj úzke neoznačené kontakty. Na rozdiel od prototypových dosiek typu Arduino, či Raspberry k týmto kontaktom nemôžete pripojiť prepojovacie vodiče, ale potrebujete konektor, do ktorého dosku micro:bit zasuniete. K dispozícii sú dva typy konektorov

v cenovej relácii 5 – 7 eur, ktoré umožnia pripojiť micro:bit k bezkontaktnému prepojovaciemu poľu, na ktorom potom môžete realizovať aj zložitejšie konštrukcie,

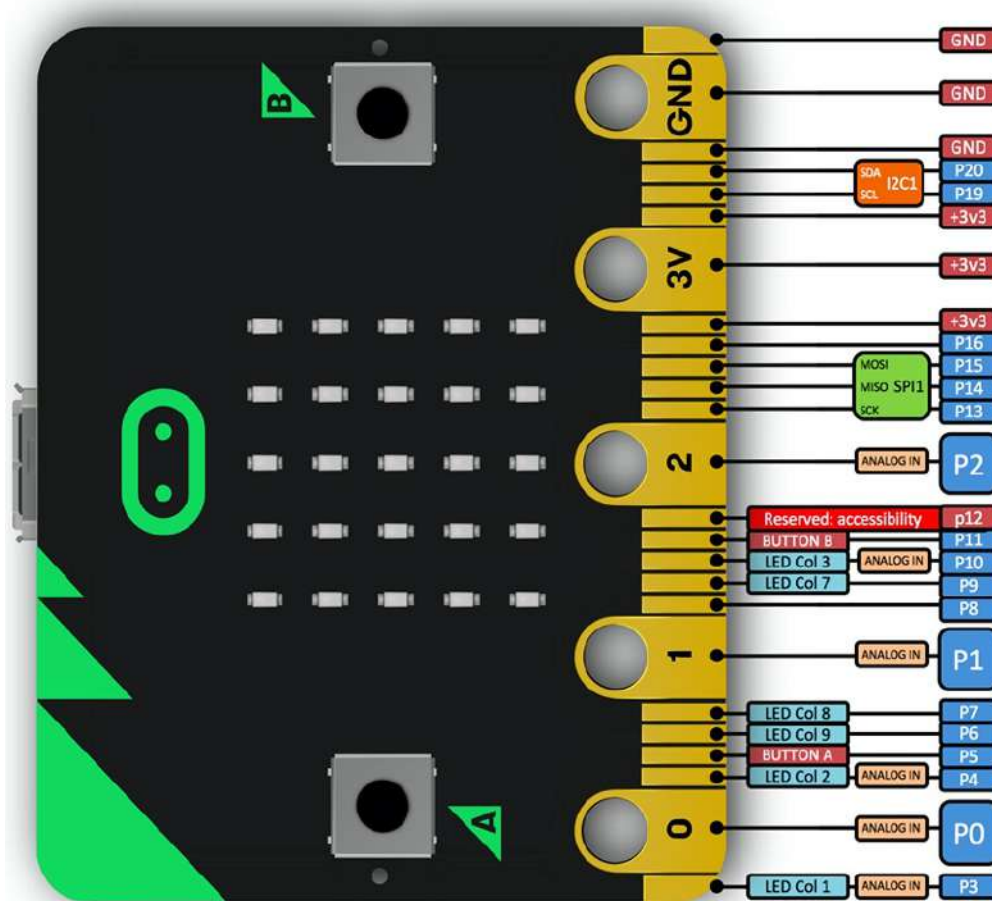


alebo druhý typ konektora má piny, na ktoré môžete nasunúť prepojovacie vodiče s dutinkami.



Úzke kontakty sú v dokumentácii číslované od 3 do 22. Niektoré z nich sú zdieľané s hardvérovými komponentmi. Napríklad piny 3, 4, 6, 7, 9, 10 sú zdieľané s niektorými LED diódami, takže ak využívate maticový displej, tieto piny použiť nemôžete. Pin 5 je zdieľaný s tlačidlom A a pin 11

s tlačidlom B. Pin 5 má pull-up rezistor, čo znamená, že na pin je cez rezistor privedené napätie 3V. Po pripojení externého tlačidla medzi pin 5 a GND je pri rozpojenom tlačidle na pine 3V, čiže logická jednotka. Po stlačení tlačidla je na pine 0 V, čiže logická nula. Nula na pine generuje udalosť stlačenia tlačidla.



Popis pinov konektora BBC micro:bit

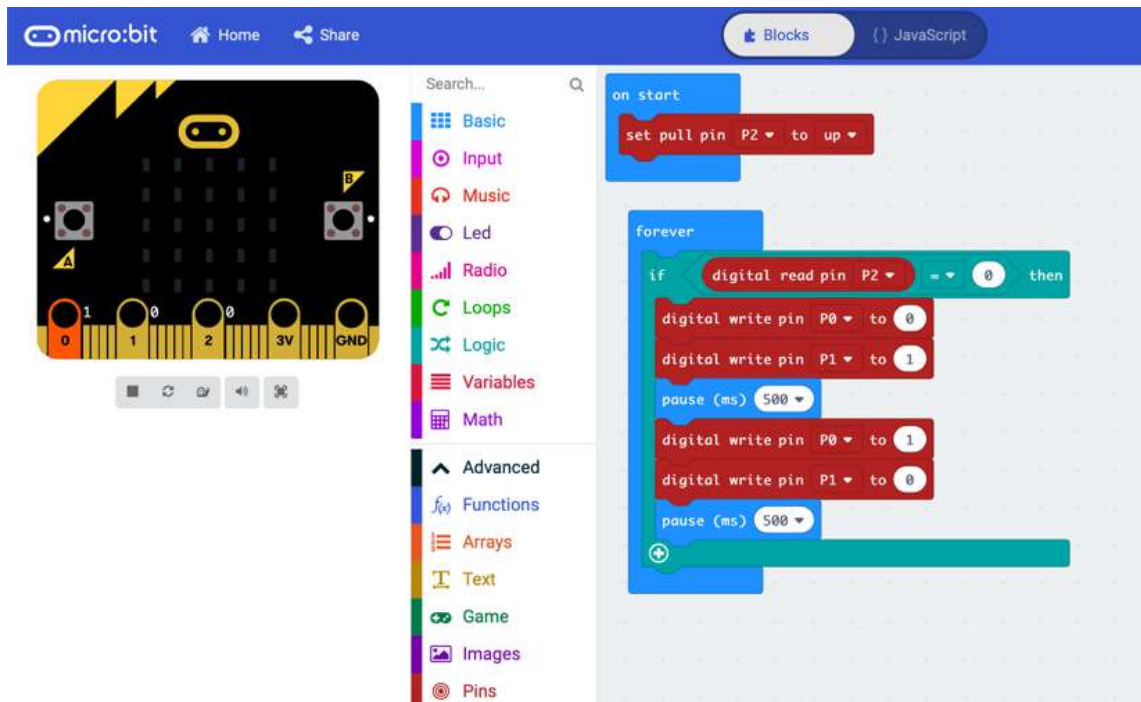
Pin 13 GPIO, ktorý sa využíva pre signál SCK pre rozhranie trojvodičovej zbernice Serial Peripheral Interface (SPI). SPI využíva pin 14 ako Master In Slave Out (MISO) a pin 15 pre signál Master Out Slave In (MOSI). Pin 16 sa využíva aj ako SPI Chip Select. Ďalšie komunikačné rozhranie I2C využíva piny 19 a 20 pre hodinový signál (SCL) a dátovú linku (SDA). Pomocou zbernice a protokolu I2C je možné pripojiť niekoľko zariadení súčasne. Na túto zbernicu je pripojený aj interný akcelerometer a magnetometer. Piny 21 a 22 sú pripojené na GND.

Na e-shopoch sú taktiež k dispozícii rozširujúce moduly umožňujúce ovládať jednosmerné motory, servomotory, spínať väčšiu napäťovú a prúdovú záťaž pomocou relé, stavebnice robotických podvozkov, moduly displejov, moduly s ovládacími prvkami usporiadanými ako na ovládačoch pre hernú konzolu, prípadne moduly umožňujúce pripojiť k doske micro:bit moduly pre Raspberry Pi. Všetko v cenovej relácii 10 – 25 eur.

Programovanie pomocou blokov aj v Pythone

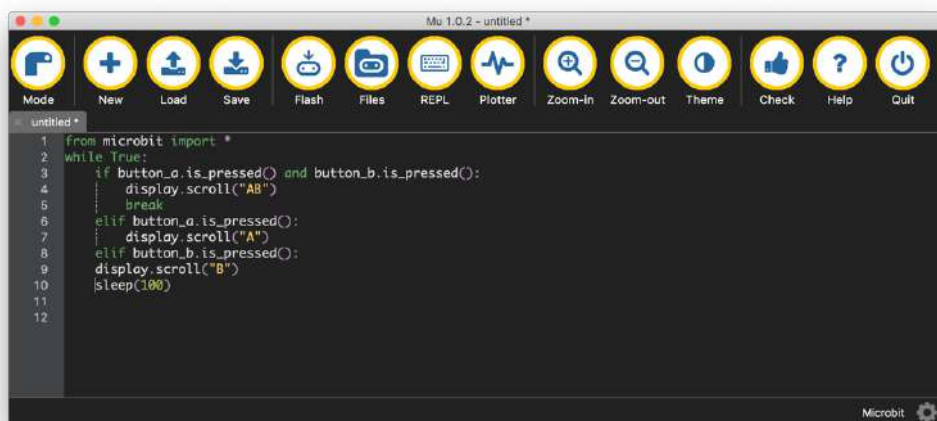
Na programovanie mikropočítačovej dosky Micro:bit sa používa buď online grafické prostredie MakeCode, v ktorom žiaci vytvárajú pomocou spájania blokov funkčné programy, alebo

programovací jazyk Python. Prostredie MakeCode na programovanie pomocou grafických blokov je vhodné najmä pre žiakov základných škôl.



Výhodou tohoto prostredia je nízka technická náročnosť prípravy pre učiteľa, nakoľko prostredie je online a nie je potrebná žiadna inštalácia. Vývojové prostredie má prepínač Bloky / JavaScript. Po prepnutí na JavaScript sa zobrazí kód v tomto programovacom jazyku, ktorý ste vytvorili pomocou blokov. Po vytvorení programu v prostredí si žiak program stiahne ako obyčajný súbor. BBC micro:bit, ktorý si žiak medzičasom pripojil k počítaču pomocou USB kábla, sa na počítači zobrazí ako klasické USB úložisko, na ktoré stiahnutý súbor s programom stačí skopírovať. Program sa následne automaticky spustí na zariadení.

Pre pokročilejších je k dispozícii aj MicroPython - špeciálna verzia jazyku Python na programovanie hardvéru, v ktorej syntax a veľká časť štandardných knižníc zostáva rovnaká. Žiaci sa tak nemusia učiť viacero nových jazykov, ten istý vedú využiť aj pri hardvéri a aj pri klasických programovacích úlohách. Podobne, ako pri blokovom programovaní, je možné využiť online MicroPython editor a stiahnuť vygenerovaný program na zariadenie,



no pre využitie rozšírenej funkcionality je vhodné používať prostredie Mu. Mu sprístupňuje interaktívnu konzolu REPL, ktorá slúži najmä na ladenie programu a hľadanie chýb, ako aj interné dátové úložisko, na ktoré je možné zapisovať nazbierané údaje zo senzorov.

Micro:bit – digitálny výstup, blikanie LED diódami

Video <https://www.youtube.com/watch?v=DwxFkJL0TWU>

Tento príklad, ktorého námetom je blikanie LED diódami je v každom učebnom texte programovania pre mikroprocesory. Na rozdiel od dosky Micro:bit, ktorá má maticový displej schopný zobraziť aj rolujúci viacznakový text, niektoré mikrokontrolérové vývojové dosky, napríklad Arduino nemajú žiadny displej. Preto najjednoduchším spôsobom ako sa presvedčiť o tom, že máte správna nainštalované a nakonfigurované vývojárske nástroje a správne pripojenú mikropočítačovú dosku na ktorej aplikáciu testujete je blikáť LED diódou pripojenou na niektorý z digitálnych výstupných portov. Tie má každá mikropočítačová doska. Cieľom aplikácie je blikáť LED diódou, čiže striedavo nastavovať na vybranom porte, ku ktorému je dióda pripojená úroveň „logická 0“ a „logická 1“.

V tejto aplikácii budeme blikáť dvoma LED diódami pripojenými k portom P0 a P1. LED dióda má malý vnútorný odpor, preto aby ste nezničili mikrokontrolér a LED diódu veľkým prúdom, LED diódy nie sú pripojené priamo na porty, ale cez rezistory 220 až 330 ohmov, ktoré obmedzujú prúd prechádzajúci diódami. Pri napätí 3.3 V a rezistore 330 ohmov bude prúd prechádzajúci cez LED diódu 10 miliampérov, čo je prípustná hodnota pre port aj LED diódu. Porty P0 a P1 sú vyvedené na široké kontakty, takže zapojenie môžete zrealizovať pomocou vodičov s banánikmi,

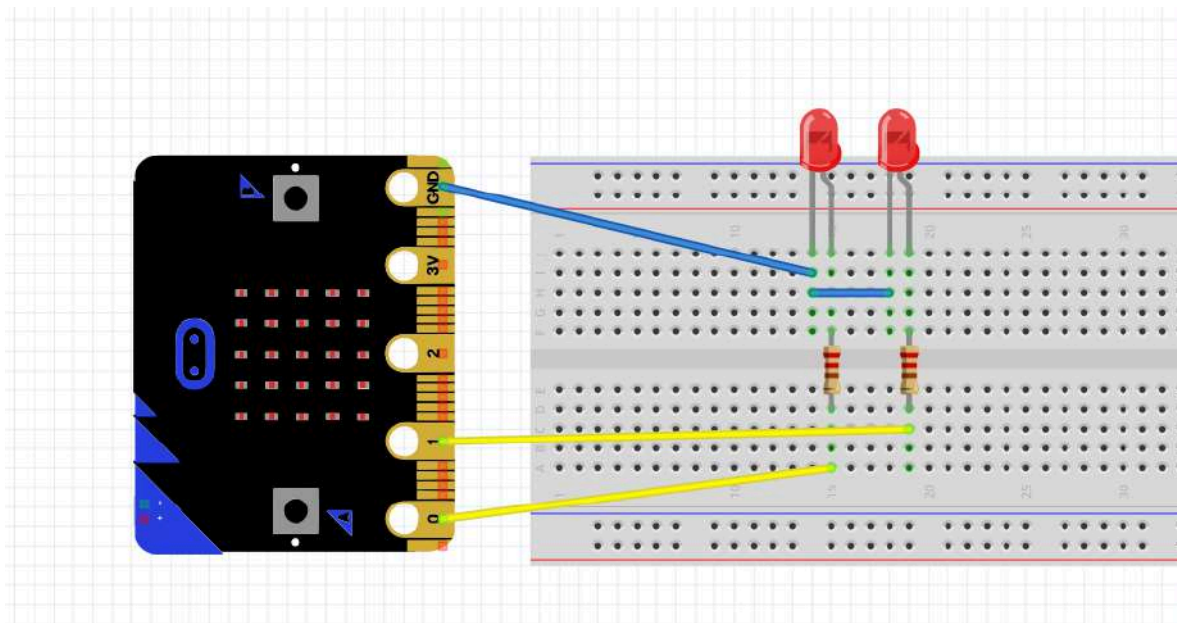
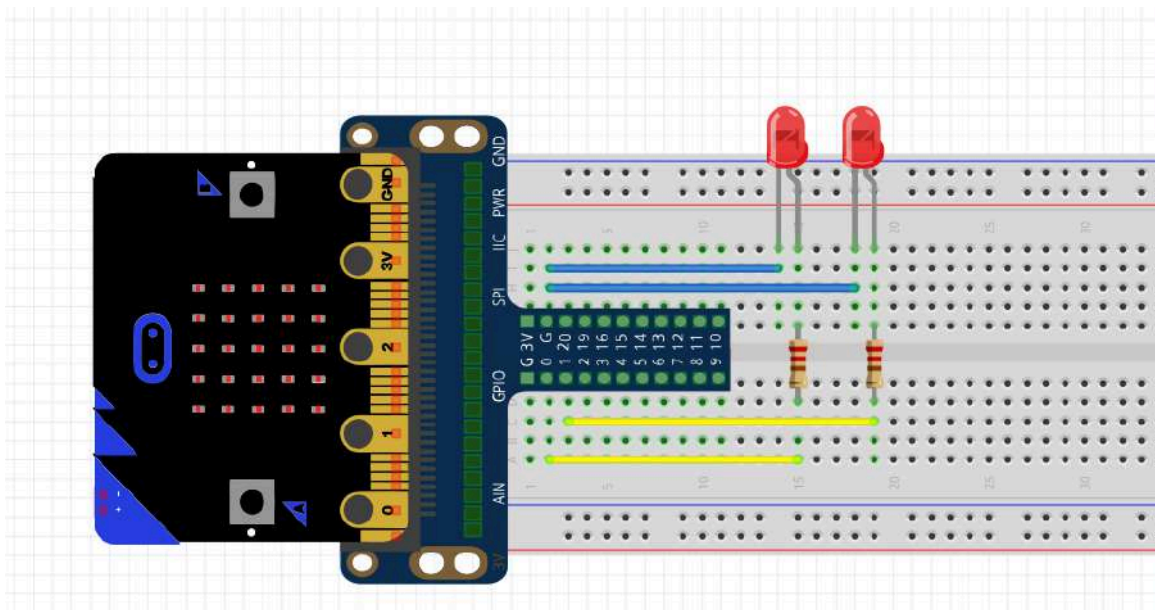
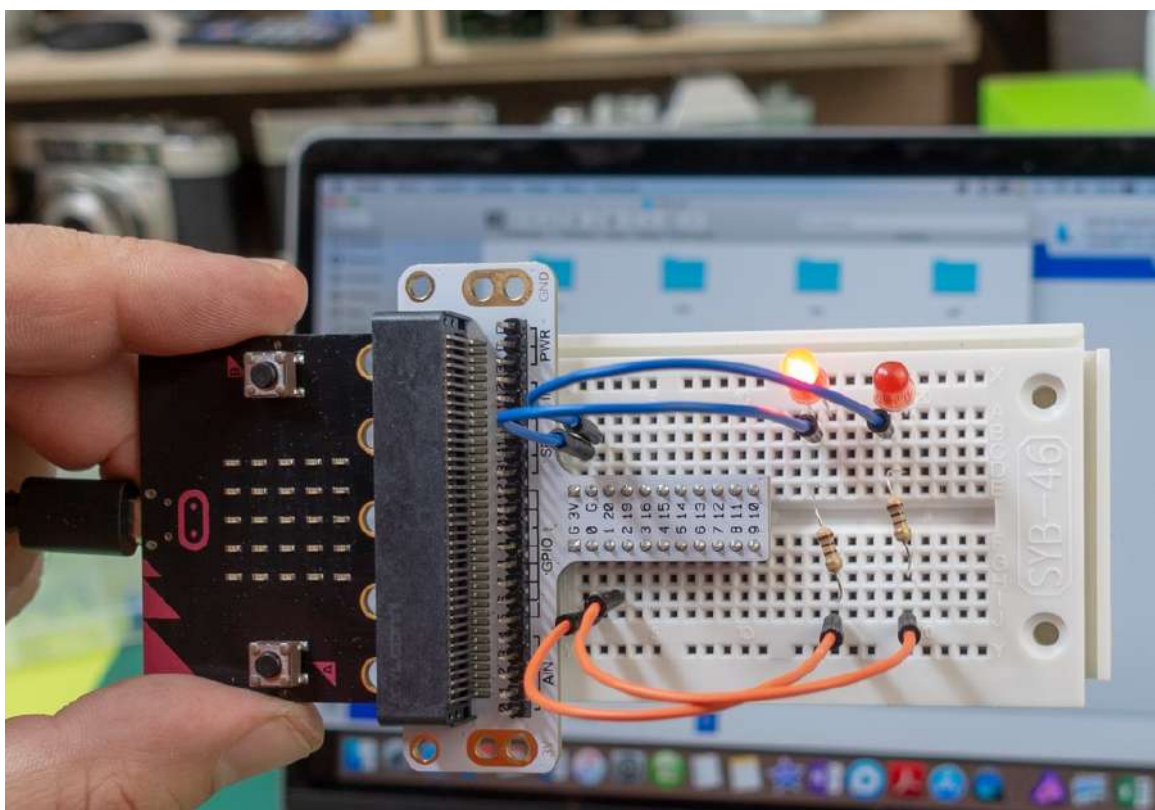


Schéma jednoduchého zapojenia, ktorá sa dá realizovať pomocou vodičov s krokosvorkami, avšak jednoduchšie a prehľadnejšie to bude na prepojovacom poli, pripojenom cez vhodnú redukciu

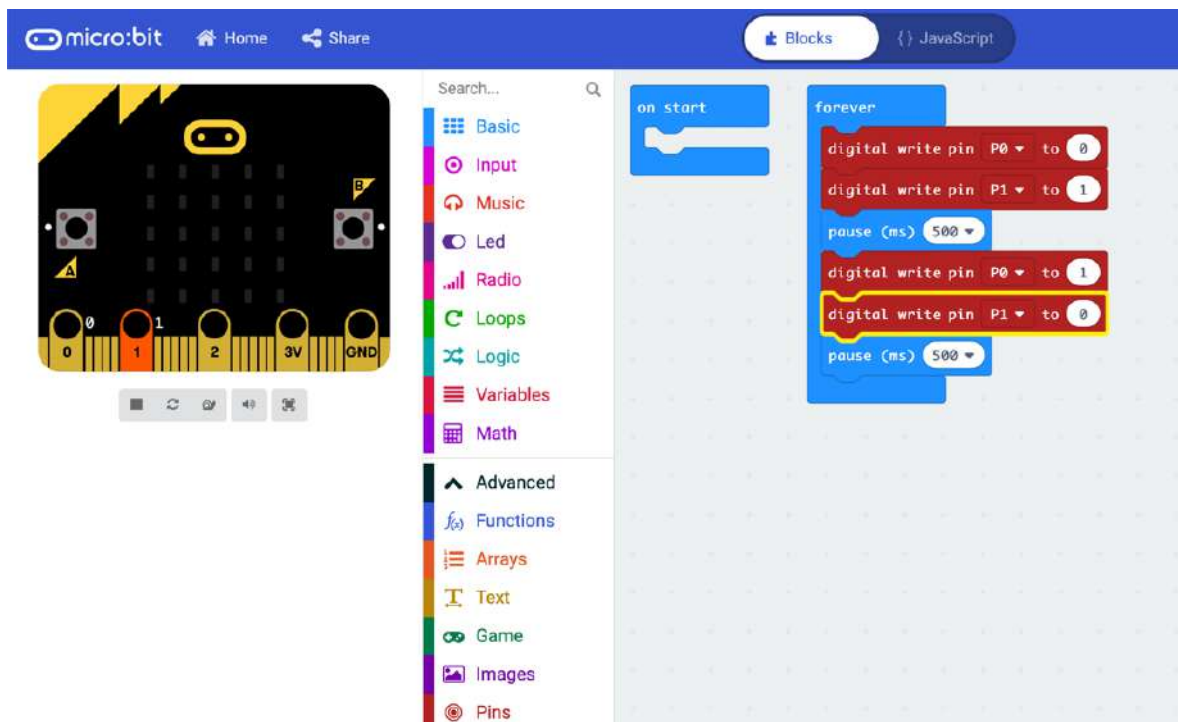


Reálne zapojenie na prepojovacom poli



Reálne zapojenie na prepojovacom poli

Ledky rozblikáte tak, že na príslušných portoch budete striedavo prepínať hodnoty logická 1 (3,3 V) a logická 0 (0 V).



JavaScript:

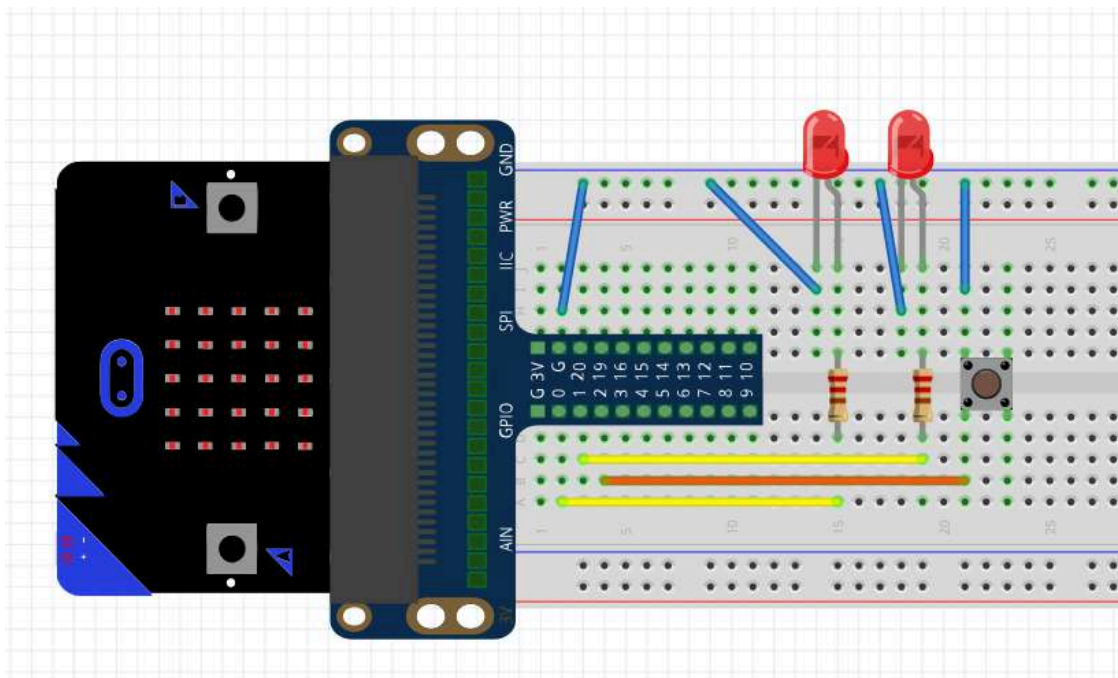
```
basic.forever(function () {
  pins.digitalWritePin(DigitalPin.P0, 0)
  pins.digitalWritePin(DigitalPin.P1, 1)
  basic.pause(500)
  pins.digitalWritePin(DigitalPin.P0, 1)
  pins.digitalWritePin(DigitalPin.P1, 0)
  basic.pause(500)
})
```

Zavedenie kódu aplikácie do dosky Micro:bit

Kód budete vytvárať priamo na stránke <https://makecode.microbit.org>. Pripojte vývojovú dosku do USB portu počítača. Počítač rozpozná zariadenie ako bežný USB kľúč - otvorí sa priečinok s názvom MICROBIT. Po vytvorení programu ho viete vo forme .hex súboru stiahnuť a nahráť do svojho zariadenia microbit kliknutím na tlačidlo "Stiahnuť". Hex súbor uložte do priečinka MICROBIT. Počas nahrávania na zariadení bliká LEDka pri USB konektore. Keď sa nový súbor s programom nahrá do zariadenia, LEDka prestane blikáť a program sa ihneď spustí. Zároveň sa súbor .hex z priečinka vymaže. Už nie je potrebný, pretože jeho obsah je už nahratý v pamäti mikrokontroléra, kde sa vykonáva. Všimnite si v hornej časti prepínač Bloky / JavaScript. Po prepnutí na JavaScript sa zobrazí kód v tomto programovacom jazyku, ktorý ste vytvorili pomocou blokov. Pri každom príklade ukážeme nielen grafické blokové znázornenie algoritmu, ale aj kód v JavaScripte.

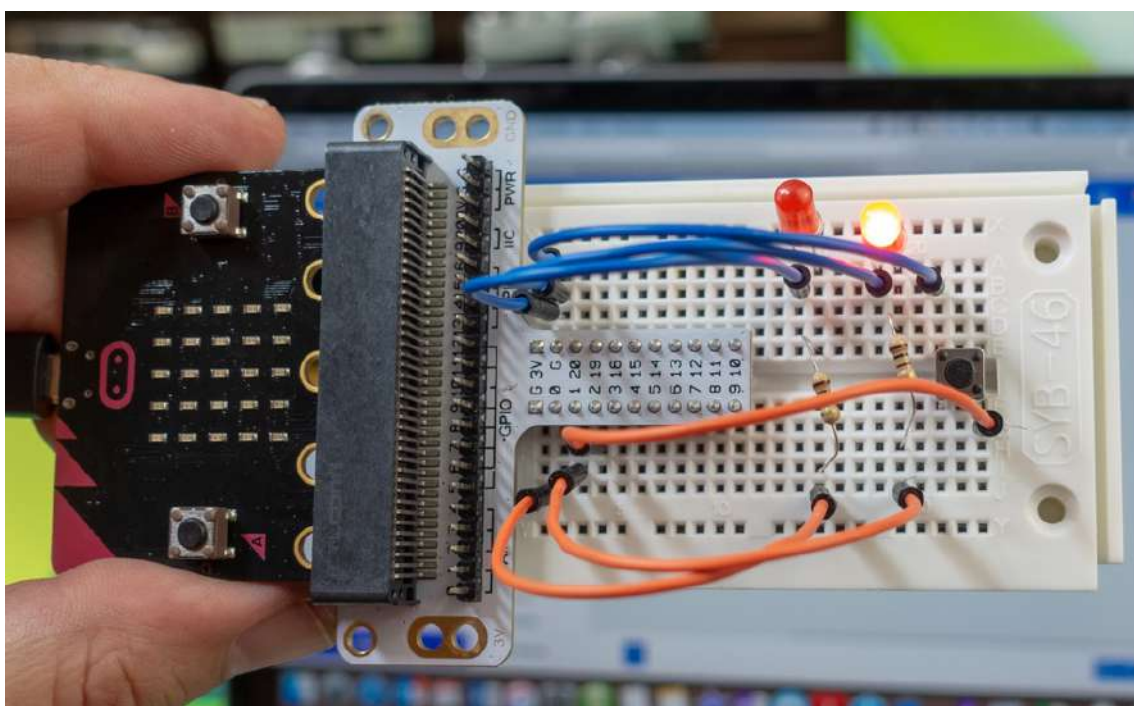
Micro:bit – digitálny vstup, obsluha tlačidla

Digitálne porty sa označujú aj skratkou GPIO – (General Purpose Input Output) čo v praxi znamená, že môžu fungovať ako výstupné, aj vstupné. Privedením napätovej úrovne 3.3 V na pin portu sa nastaví vstup na hodnotu logická 1. Zapojenie s dvoma LED diódami pripojenými na porty P0 a P1 rozšírime o tlačidlo pripojené k portu P2



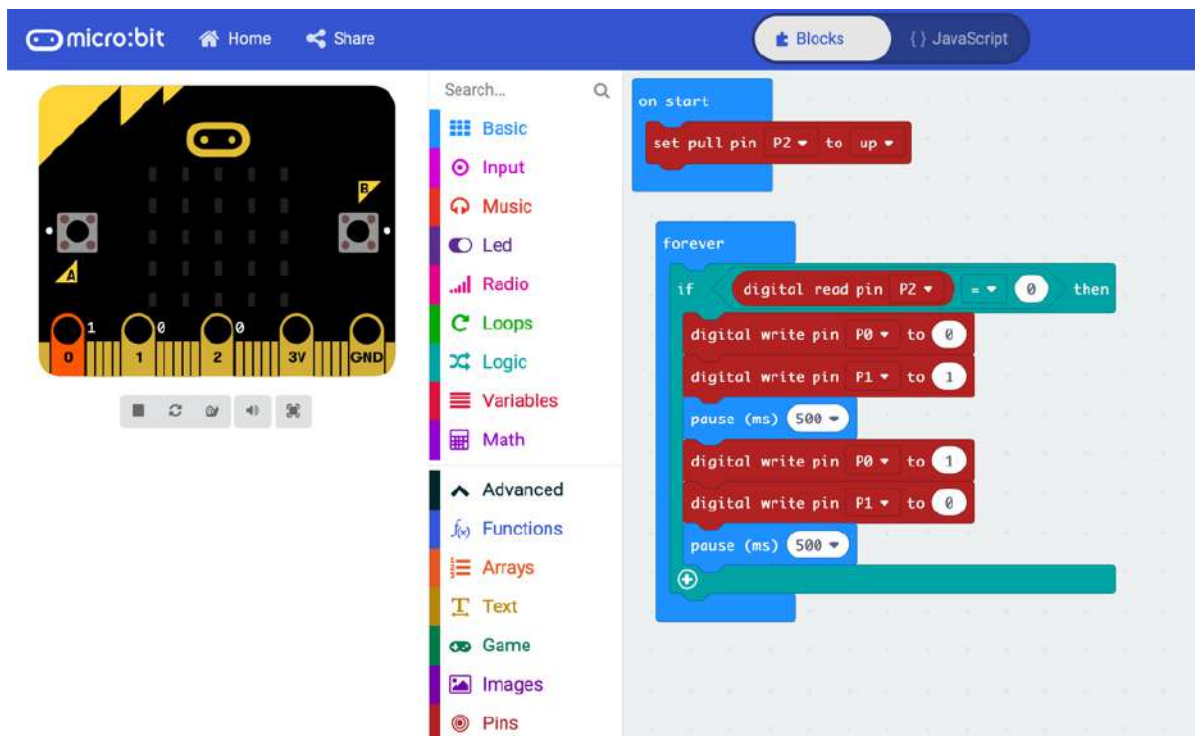
Rezistory majú hodnotu 220 Ohm.

a kód upravíme tak, aby LED diódy blikali iba vtedy, ak je stlačené tlačidlo.



Zapojenie s LED diódami a tlačidlom

V tomto príklade bude potrebné vstupný pin inicializovať. Bude fungovať tak, že pri nestlačení tlačidla bude na vstupnom porte logická 1 a po zatlačení logická nula. Aby ste nemuseli ku tlačidlu pridávať rezistor pripojený na jeden vývod tlačidla a napájacie napätie 3,3 V nakonfigurujeme vstupný pin do režimu pull up, kedy sa aktivuje vnútorný rezistor.



JavaScript

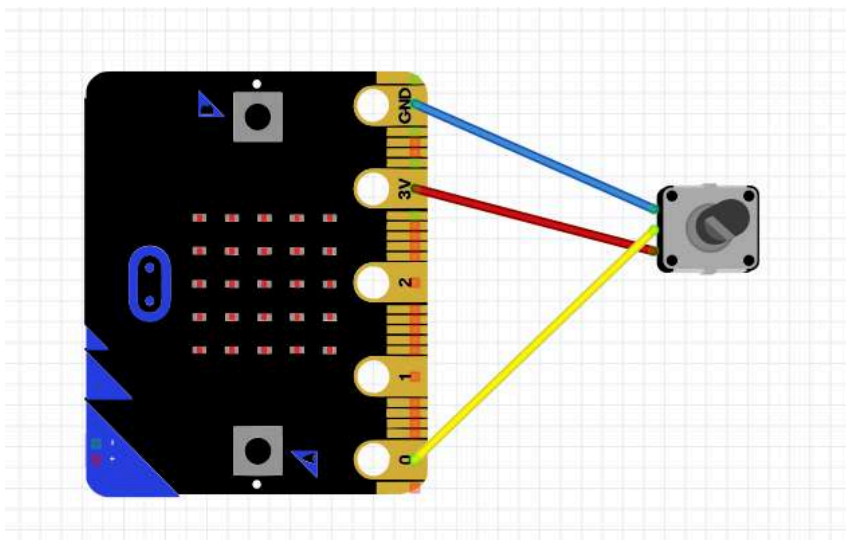
```

pins.setPull(DigitalPin.P2, PinPullMode.PullUp)
basic.forever(function () {
  if (pins.digitalReadPin(DigitalPin.P2) == 0) {
    pins.digitalWritePin(DigitalPin.P0, 0)
    pins.digitalWritePin(DigitalPin.P1, 1)
    basic.pause(500)
    pins.digitalWritePin(DigitalPin.P0, 1)
    pins.digitalWritePin(DigitalPin.P1, 0)
    basic.pause(500)
  }
})

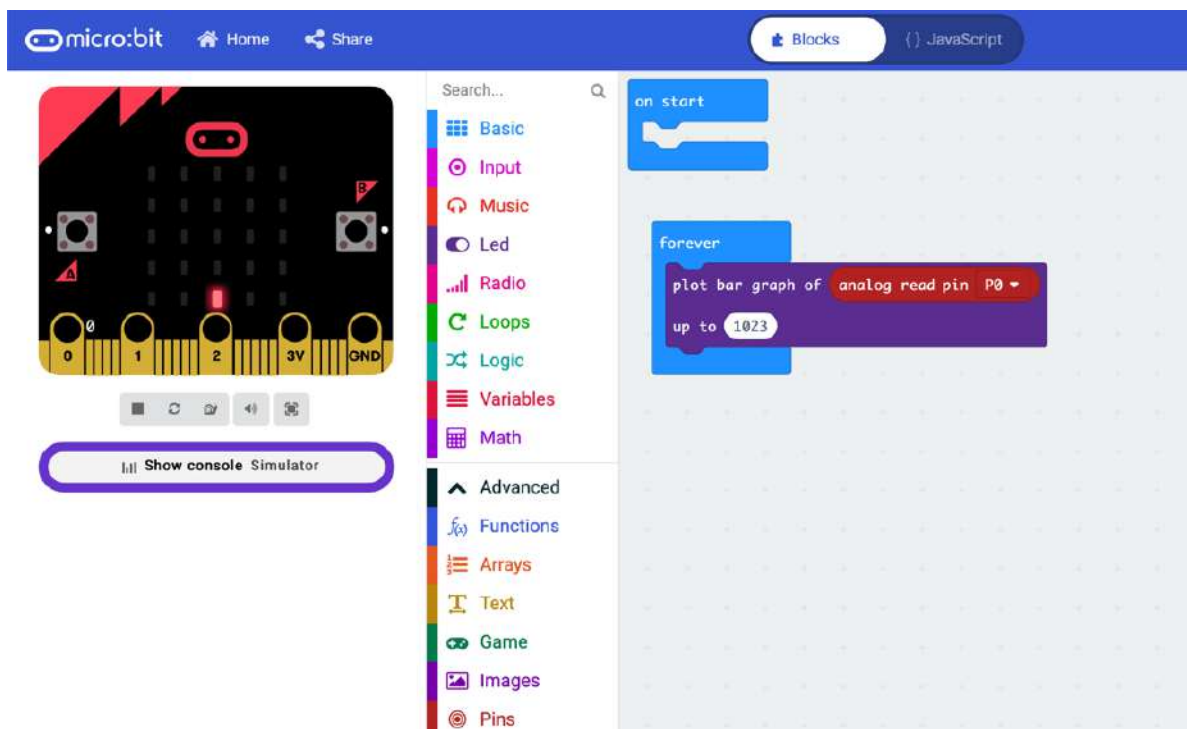
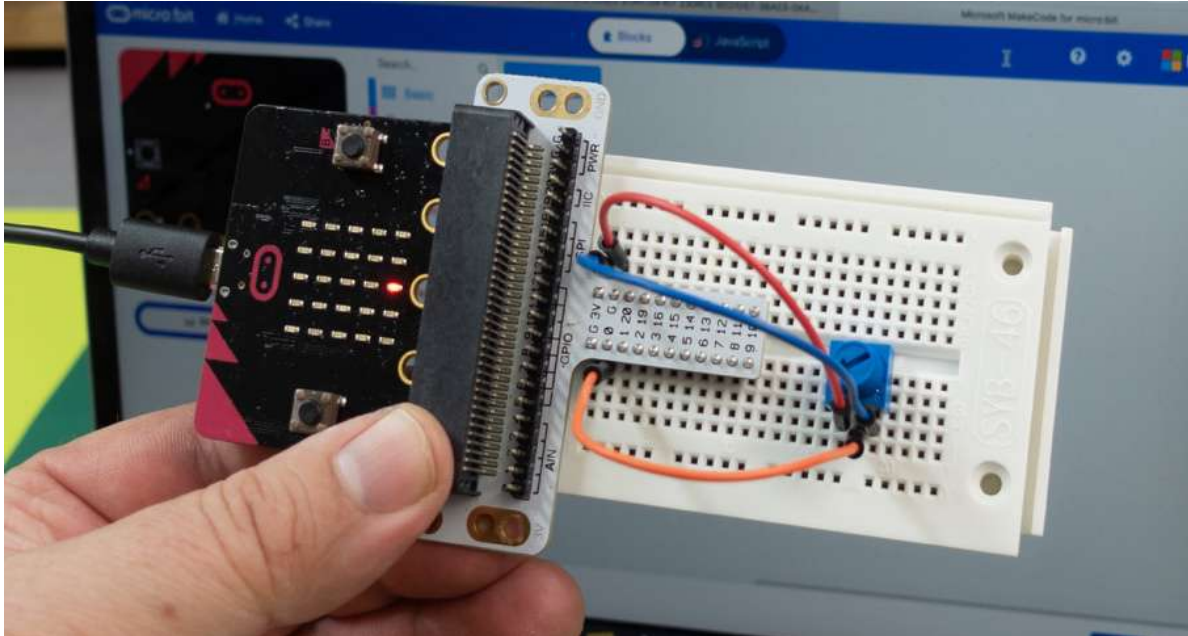
```

Micro:bit – analógový vstup, obsluha tlačidla

V ďalšom mini príklade sa presunieme do analógového sveta a ukážeme načítanie a jednoduché grafické znázornenie napätia na pine P0, ktorý v tomto prípade bude fungovať ako analógový vstup.



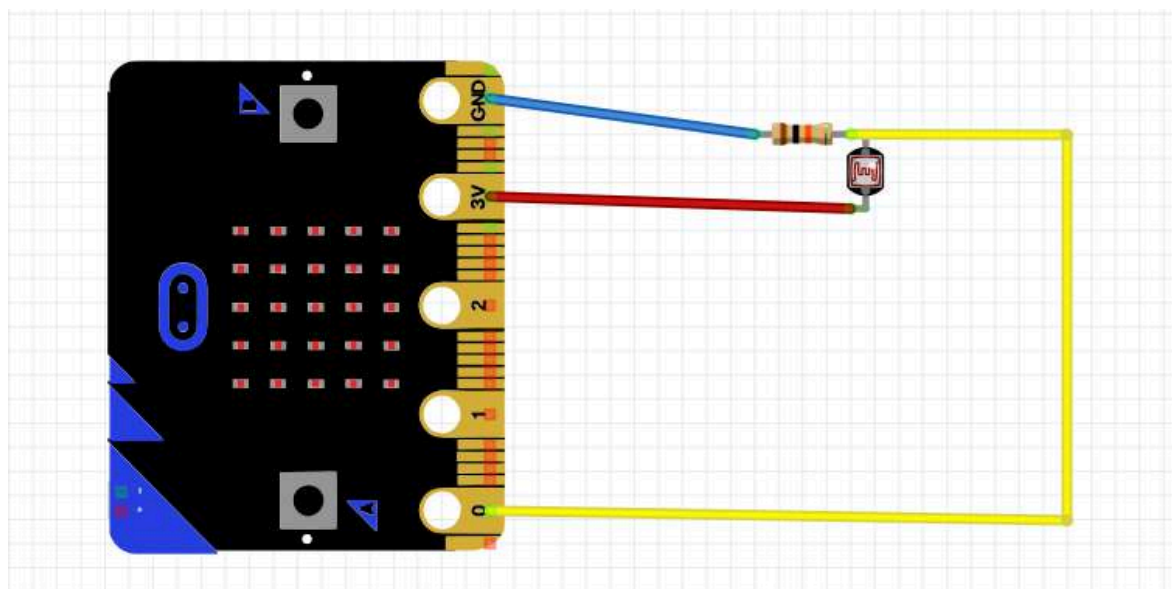
Takýmto spôsobom je možné napríklad zosnímať uhol natočenia potenciometra zapojeného ako delič napätia, hodnotu z termistoru, fotorezistoru a podobne. Vstavaný analógovo digitálny prevodník konvertuje analógovú hodnotu napätia na vstupe z intervalu napätí 0 – 3.3 V na interval číselných hodnôt 0 – 1024. Nameranú hodnotu zobrazíme ako plošný graf na displeji, presnejšie na matici diód na doske micro:bit. Napätie budeme regulovať pomocou potenciometra



JavaScript

```
basic.forever(function () {  
  led.plotBarGraph(  
    pins.analogReadPin(AnalogPin.P0),  
    1023  
  )  
})
```

Modifikujte predchádzajúci príklad tak, aby ste pomocou fotobunky mohli merať intenzitu okolitého osvetlenia. Softvérový kód zostane rovnaký, ako keď ste merali hodnotu napätia na potenciometri zapojenom ako napäťový delič. Fotobunka je zapojená podľa schémy. Rezistor má hodnotu 10 kOhm.



Rezistor má hodnotu 20 kOhm.

V poslednom príklade zostane rovnaké hardvérové zapojenie, no zmení sa softvér. Bude rozlišovať dve hodnoty osvetlenia. Po štarte sa aplikácia skalibruje, čiže do premennej „kalibrácia“ sa zapíše aktuálna hodnota osvetlenia. Ak bude intenzita okolitého svetla väčšia, alebo rovná tejto hodnote zmenšenej o 2 zobrazí sa na displeji jedna z preddefinovaných ikon. Ak bude hodnota úrovne osvetlenia menšia, zobrazí sa iná ikona.

```
on start
  set kalibracia to analog read pin P0

forever
  set svetlo to analog read pin P0
  if svetlo <= kalibracia - 2 then
    show icon [3x3 grid of dots]
  else
    show icon [2x2 grid of dots]
```

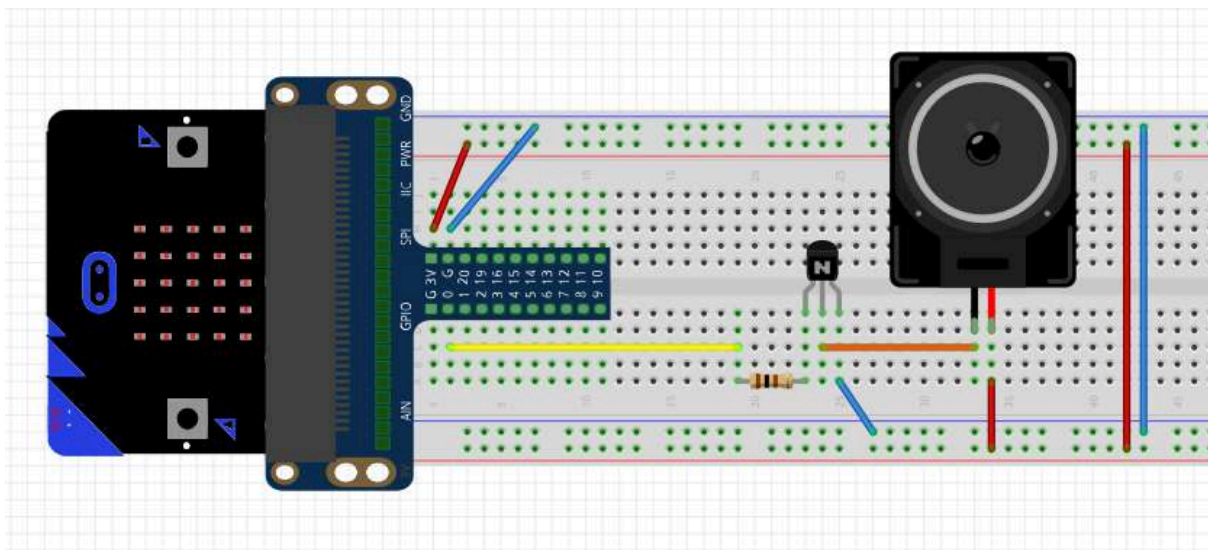
JavaScript

```
let svetlo = 0
let kalibracia = 0
kalibracia = pins.analogReadPin(AnalogPin.P0)

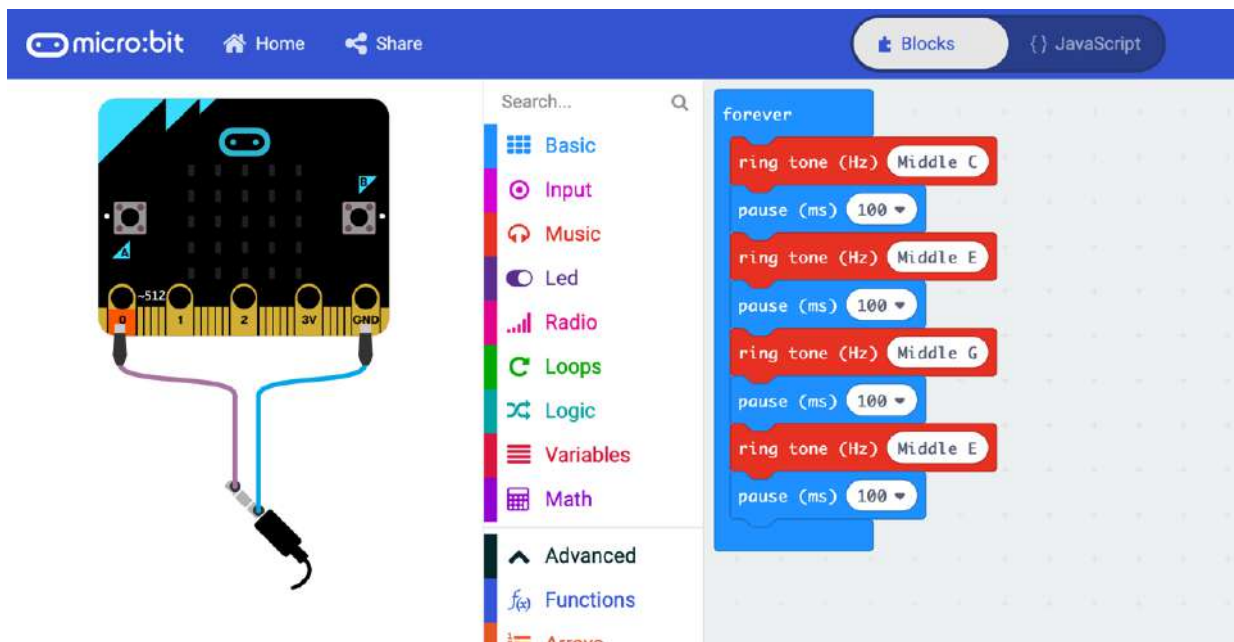
basic.forever(function () {
  svetlo = pins.analogReadPin(AnalogPin.P0)
  if (svetlo < kalibracia - 2) {
    basic.showIcon(IconNames.Heart) } else {
    basic.showIcon(IconNames.Square)
  }
})
```

Micro:bit – generovanie zvuku

Základná knižnica funkcií pre Micro:bit umožňuje aj generovanie tónov zvuku. Môžete nastaviť tóny buď podľa hudobnej stupnice, klaviatúry klavíra, alebo zadaním frekvencie. Nakoľko reproduktor keď hrá hlasnejšie má väčší prúdový odber, nepripojíme ho priamo na port P0, ale cez NPN tranzistor podľa schémy



Rezistor má hodnotu 100 Ohm.



JavaScript:

```
basic.forever(function () {
  music.ringTone(262)
  basic.pause(100)
  music.ringTone(330)
  basic.pause(100)
  music.ringTone(392)
  basic.pause(100)
  music.ringTone(330)
  basic.pause(100)
})
```

Micro:bit – obsluha udalostí od hardvéru

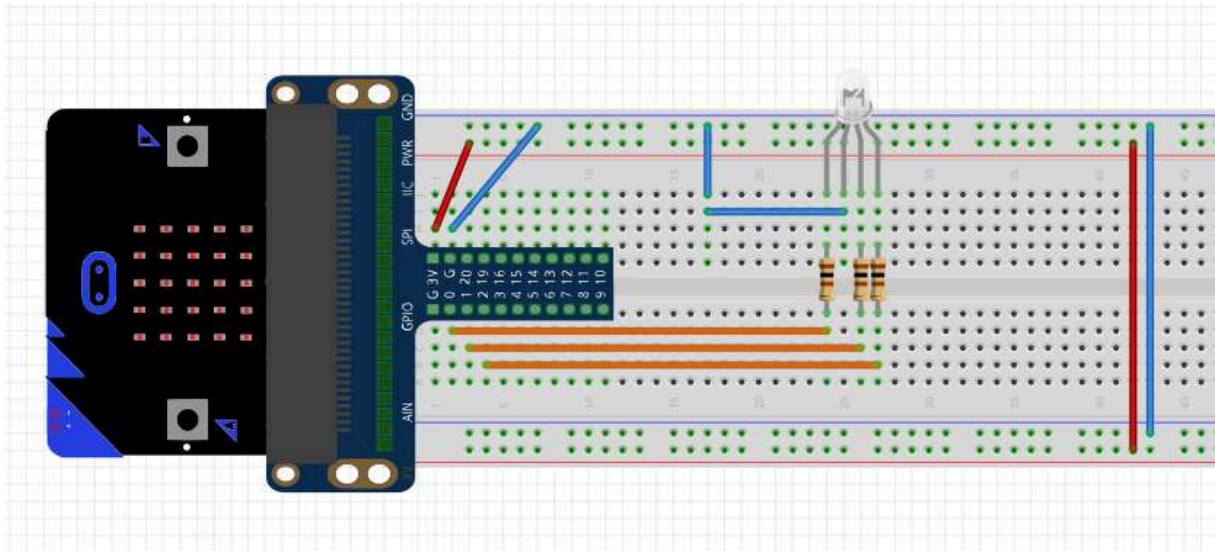
Video <https://www.youtube.com/watch?v=ptNAD9qTjml>

V predchádzajúcich príkladoch, kde ste testovali hodnoty na digitálnom, prípadne analógovom vstupe ste programovali mikropočítačovú dosku klasicky tak, že hlavný program ktorý niečo jednoduché robil periodicky bežal v nekonečnej slučke vytvorenej pomocou cyklu forever. Ak vytvárate aplikáciu, ktorá reaguje na podnety od používateľa, alebo od senzorov monitorujúcich veličiny okolia alebo procesu, bolo by monitorovanie týchto udalostí v slučke trochu komplikované na naprogramovanie podmienok. Kód by bol neprehľadný a hlavne mikroprocesor by musel stále bežať v normálnom režime a spotrebávať adekvátne množstvo energie a to aj v prípade, ak by sa pol dňa, prípadne dlhšie nedialo vôbec nič. Oveľa jednoduchšie a pri správnom naprogramovaní aj energeticky úspornejšie je vytvoriť udalosťami riadený program a naprogramovať obsluhu udalostí generovaných hardvérom.

Obsluha udalostí od používateľa rozlišuje interný hardvér, napríklad ak používateľ zatlačí niektoré z tlačidiel A a B integrovaných priamo na doske, prípadne ich zatlačí súčasne, prípadne udalosť od niektorého senzora, napríklad akcelerometra, snímača teploty, či osvetlenia a externý hardvér pripojený na zbernicu dosky micro:bit. V princípe v tom nie je žiadny rozdiel, akurát interný hardvér je možné vo vývojovom prostredí aj emulovať a graficky znázorniť. Nová verzia vývojového prostredia dokáže znázorniť aj fungovanie niektorých knižníc obsluhujúcich externý hardvér pripojený na zbernicu, napríklad servomotor.

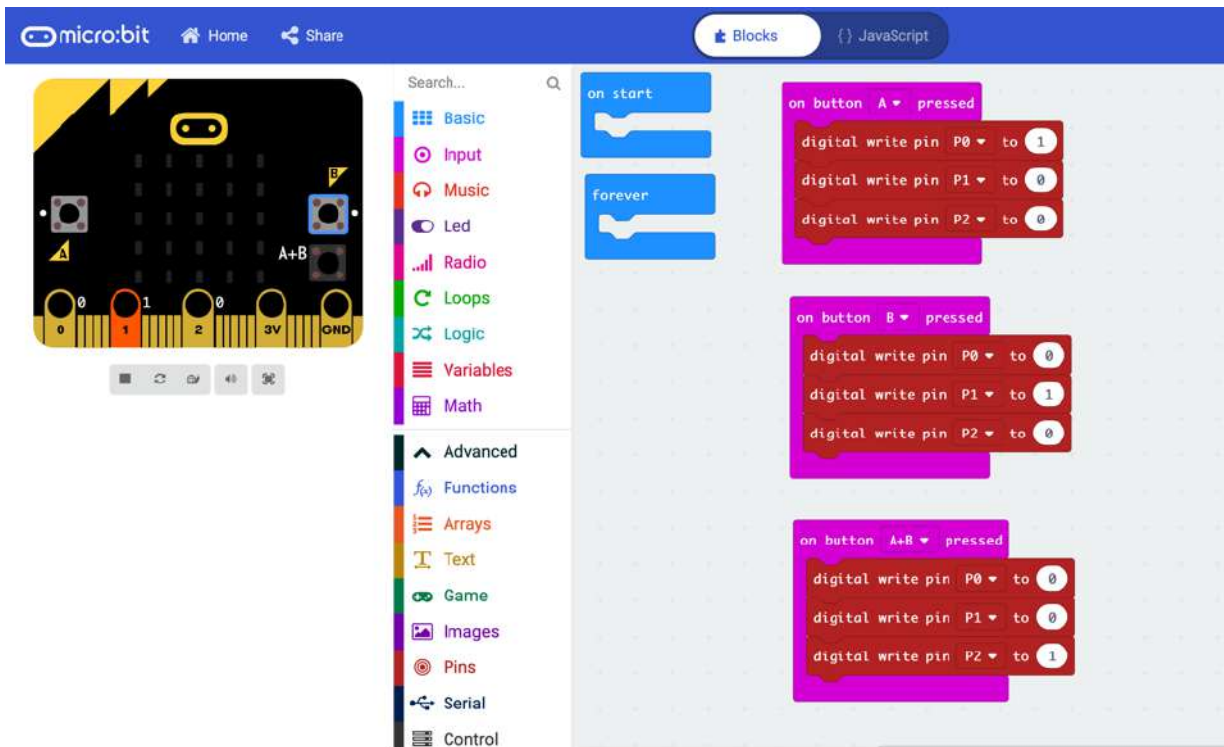
Micro:bit – nastavenie farby svetla RGB diódy tlačidlami

V príklade ukážeme rozsvietenie RGB LED, ktorej tri vývody sú pripojené k portom P0, P1 a P2 cez rezistory obmedzujúce prúd LED diódou. Hodnotu budeme nastavovať v móde digitálneho výstupu, čiže môže byť len 0, alebo 3.3 V. Ak na vývode R LED diódy bude logická 1 a na vývodoch G a B logická nula, LED dióda bude svietiť červenou farbou. Ak bude logická 1 na vývode G, dióda bude svietiť zelenou a ak na vývode B, tak dióda bude svietiť modrou farbou.



Rezistory majú hodnotu 100 Ohm.

Farba LED sa nastaví podľa toho, či používateľ stlačil tlačidlo A, tlačilo B, alebo obidva tlačidlá súčasne. Všimnite si, že hlavná slučka, čiže cyklus forever neobsahuje žiadny kód, všetko sa odohráva v obsluhu udalosti zatlačenia tlačidla `input.onButtonPressed`.

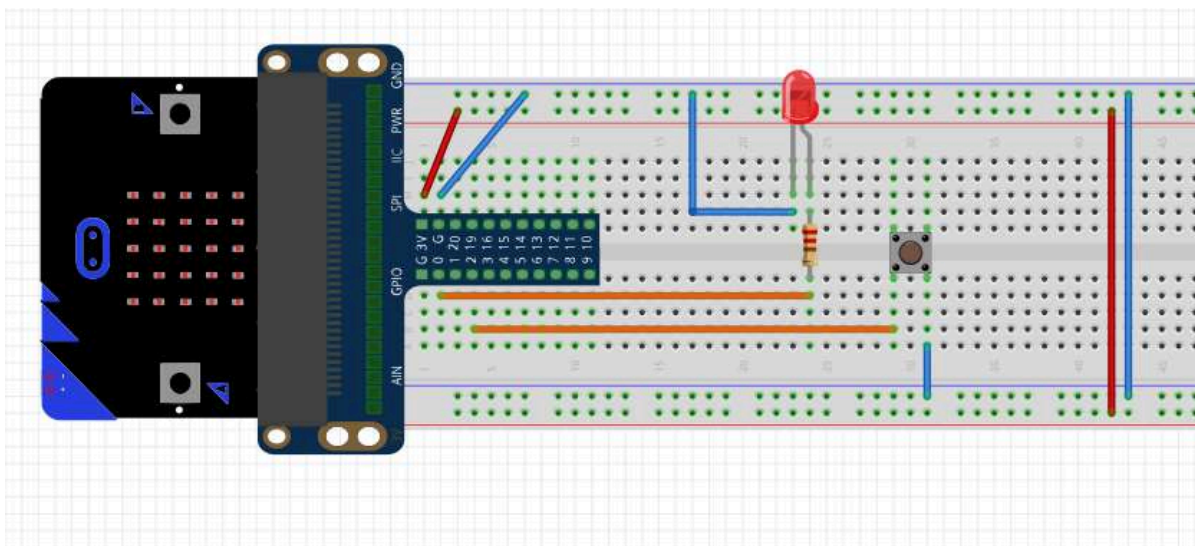


JavaScript:

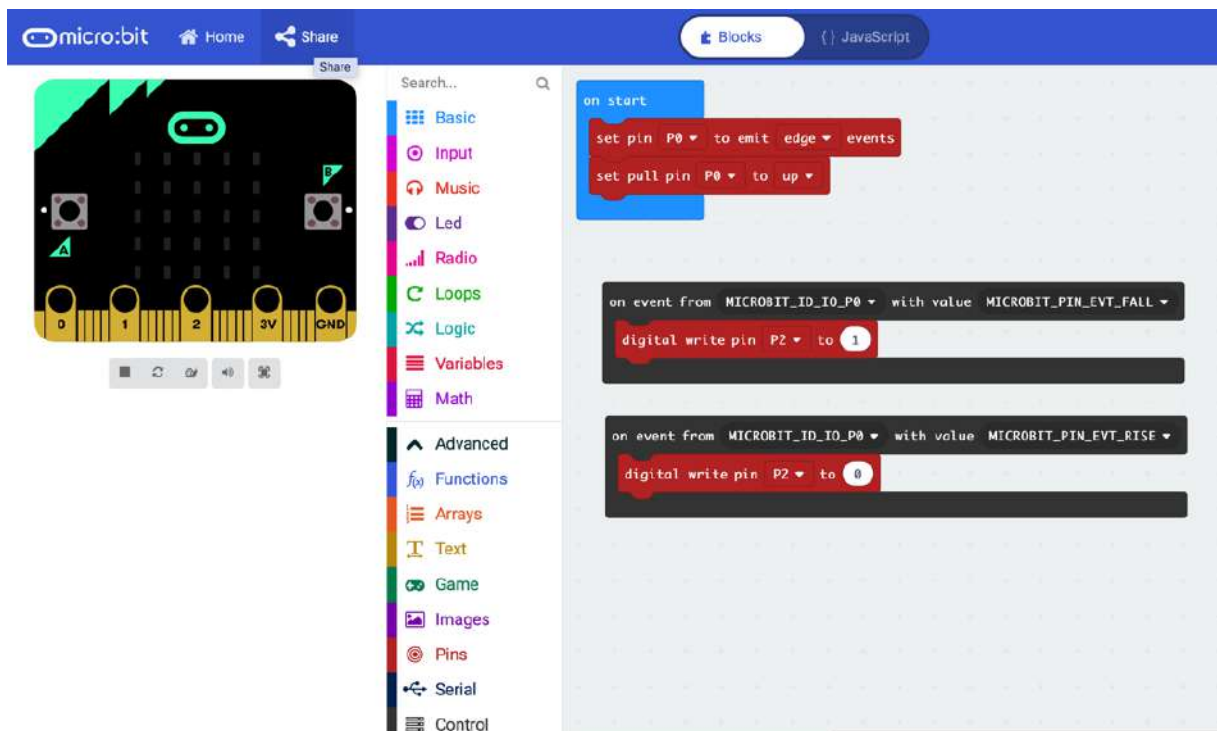
```
input.onButtonPressed(Button.A, function () {
  pins.digitalWritePin(DigitalPin.P0, 1)
  pins.digitalWritePin(DigitalPin.P1, 0)
  pins.digitalWritePin(DigitalPin.P2, 0)
})
input.onButtonPressed(Button.B, function () {
  pins.digitalWritePin(DigitalPin.P0, 0)
  pins.digitalWritePin(DigitalPin.P1, 1)
  pins.digitalWritePin(DigitalPin.P2, 0)
})
input.onButtonPressed(Button.AB, function () {
  pins.digitalWritePin(DigitalPin.P0, 0)
  pins.digitalWritePin(DigitalPin.P1, 0)
  pins.digitalWritePin(DigitalPin.P2, 1)
})
})
basic.forever(function () {
})
```

Micro:bit – rozlíšenie zmeny úrovne napätia na vstupe

Námetom príkladu je rozlíšiť pri spínaní, alebo impulzoch reakciu softvéru na nábežnú a zostupnú hranu signálu. Keď sa mení hodnota na vstupe z logickej 0 na logickú 1 jedná sa o nábežnú hranu, pretože napätie skokom nabehne z 0 na 3.3 V. Pri zmene z logickej 1 na logickú nulu pôjde o zostupnú hranu. To či sa jedná o nábežnú, alebo zostupnú hranu budeme rozlišovať pomocou LED diódy pripojenej na port P2.



Rezistor má hodnotu 220 Ohm



JavaScript:

```
pins.setEvents(DigitalPin.P0, PinEventType.Edge)
pins.setPull(DigitalPin.P0, PinPullMode.PullUp)

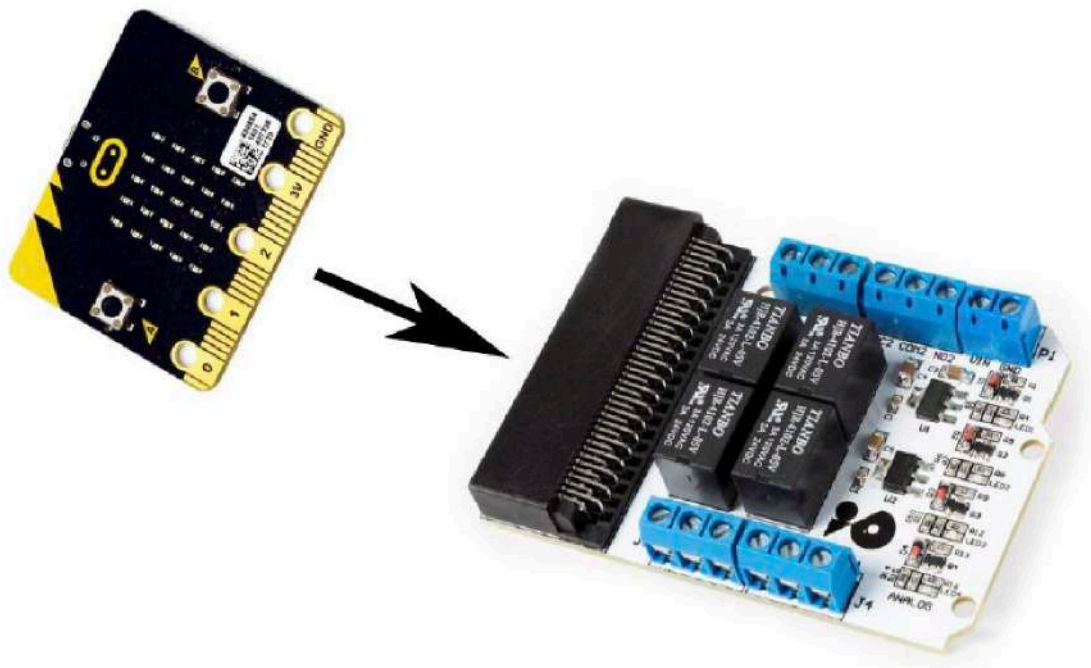
control.onEvent(EventBusSource.MICROBIT_ID_IO_P0,
EventBusValue.MICROBIT_PIN_EVT_FALL, function () {
    pins.digitalWritePin(DigitalPin.P2, 1)
})

control.onEvent(EventBusSource.MICROBIT_ID_IO_P0,
EventBusValue.MICROBIT_PIN_EVT_RISE, function () {
    pins.digitalWritePin(DigitalPin.P2, 0)
})
```

Micro:bit – spínanie a ovládanie zariadení

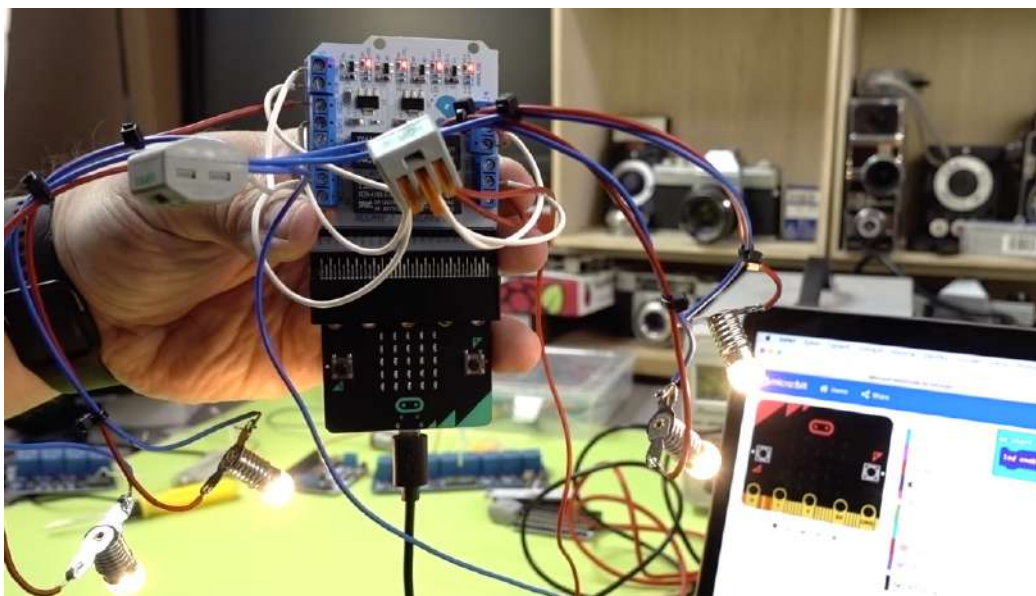
Video <https://www.youtube.com/watch?v=uwdCfGLY-A>

Pomocou portov dosky micro:bit môžete spínať pripojené zariadenia, avšak ich napájanie je limitované napätím 3.3 V a maximálnym prúdom niekoľko desiatok miliampérov. Pri napájaní cez USB prúdový regulátor zvládne záťaž 120 mA. Ak počítame 30 mA na napájanie obvodov na samotnej doske, na napájanie periférií zostáva celkový maximálny prúd 90 mA. Takže môžete napájať napríklad 4 LED diódy pripojené na portoch, pričom každá odoberá 20 mA. Aby ste mohli spínať aj zariadenia s vyšším prúdovým odberom, prípadne na vyššie napätie, potrebujete vhodný spínací prvok. Najjednoduchšie je použiť niektorý z modulov obsahujúcich relé, napríklad VMM400 od firmy Velleman. Cena modulu v e-shopoch je približne 15 eur.



Modul VMM400 so štyrmi relé

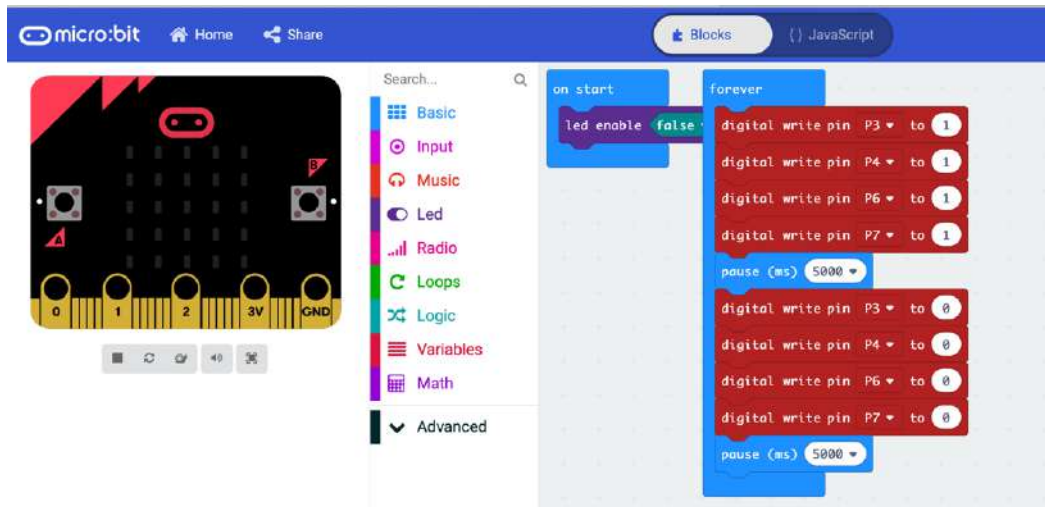
Tento modul má konektor, do ktorého sa dá zasunúť doska micro:bit. Na doske sú štyri relé, každé z nich môže spínať jednosmerný prúd s napätím maximálne 24 V a prúdom 3A. Je logické, že na modul k mikro počítačovej doske pre školy výrobca dal nízkonapäťové relé a nie relé na 230 V. Relé sú spínané podľa stavu logickej hodnoty na výstupoch P3, P4, P6 a P7. Doska potrebuje samostatné napájanie 5V a každý výstup mikro počítača micro:bit je zaťažený prúdom max. 5 mA. Stav zopnutia jednotlivých relé je indikovaný aj opticky pomocou LED diód na doske



Spínanie troch žiaroviek pomocou relé

V našich príkladoch sme k relé pripojili žiarovky do starších vreckových bateriek. Sú na napätie 4,5 V a pri tomto napätí nimi prechádza prúd 300 mA. Modul s relé je napájaný z externého 5V zdroja a napätie z tohoto zdroja sa cez kontakty relé privádza aj na žiarovky. V prvom príklade budeme

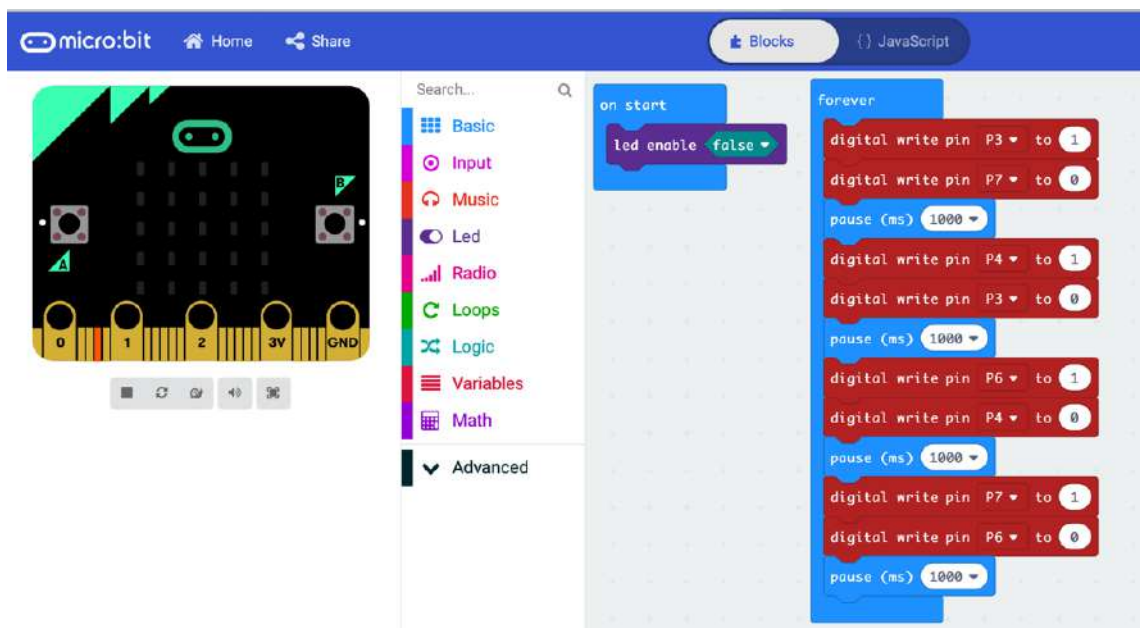
synchronne blikať všetkými žiarovkami súčasne. Technicky povedané na výstupy P3,P4,P6 a P7 budeme striedavo privádzať úroveň logická 0 a logická 1



JavaScript

```
led.enable(false)
basic.forever(() => {
  pins.digitalWritePin(DigitalPin.P3, 1)
  pins.digitalWritePin(DigitalPin.P4, 1)
  pins.digitalWritePin(DigitalPin.P6, 1)
  pins.digitalWritePin(DigitalPin.P7, 1)
  basic.pause(2000)
  pins.digitalWritePin(DigitalPin.P3, 0)
  pins.digitalWritePin(DigitalPin.P4, 0)
  pins.digitalWritePin(DigitalPin.P6, 0)
  pins.digitalWritePin(DigitalPin.P7, 0)
  basic.pause(5000)
})
```

Príklad môžete mierne zmodifikovať aby sa žiarovky rozsviecovali a zhasli postupne, tak, že sa rozsvieti prvá, v ďalšom takte prvá zhasne a zároveň sa rozsvieti druhá žiarovka a tak stále dookola



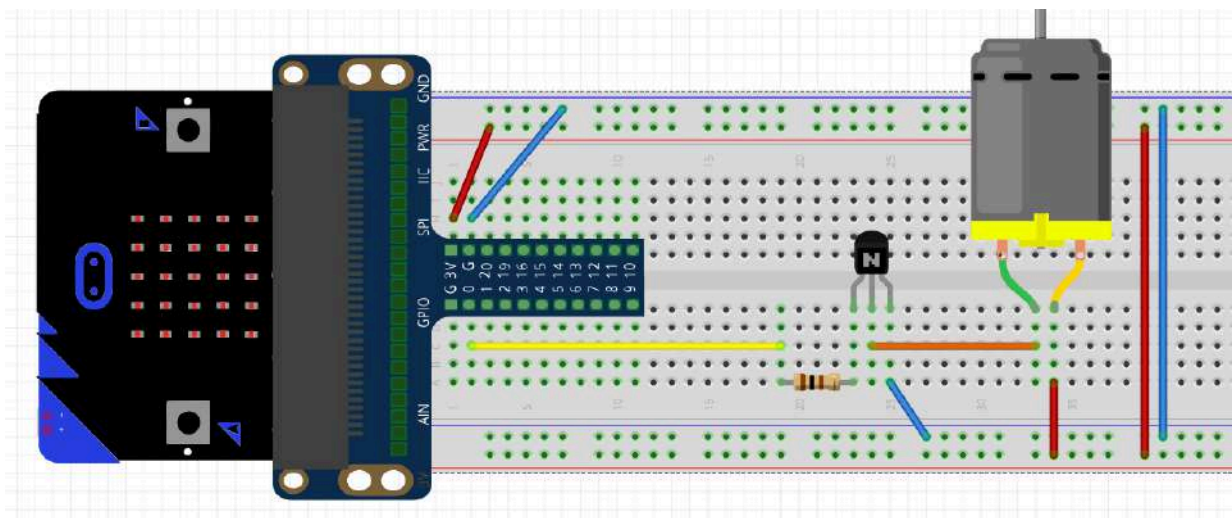
JavaScript

```
led.enable(false)
basic.forever(() => {
  pins.digitalWritePin(DigitalPin.P3, 1)
  pins.digitalWritePin(DigitalPin.P7, 0)
  basic.pause(1000)
  pins.digitalWritePin(DigitalPin.P4, 1)
  pins.digitalWritePin(DigitalPin.P3, 0)
  basic.pause(1000)
  pins.digitalWritePin(DigitalPin.P6, 1)
  pins.digitalWritePin(DigitalPin.P4, 0)
  basic.pause(1000)
  pins.digitalWritePin(DigitalPin.P7, 1)
  pins.digitalWritePin(DigitalPin.P6, 0)
  basic.pause(1000)
})
```

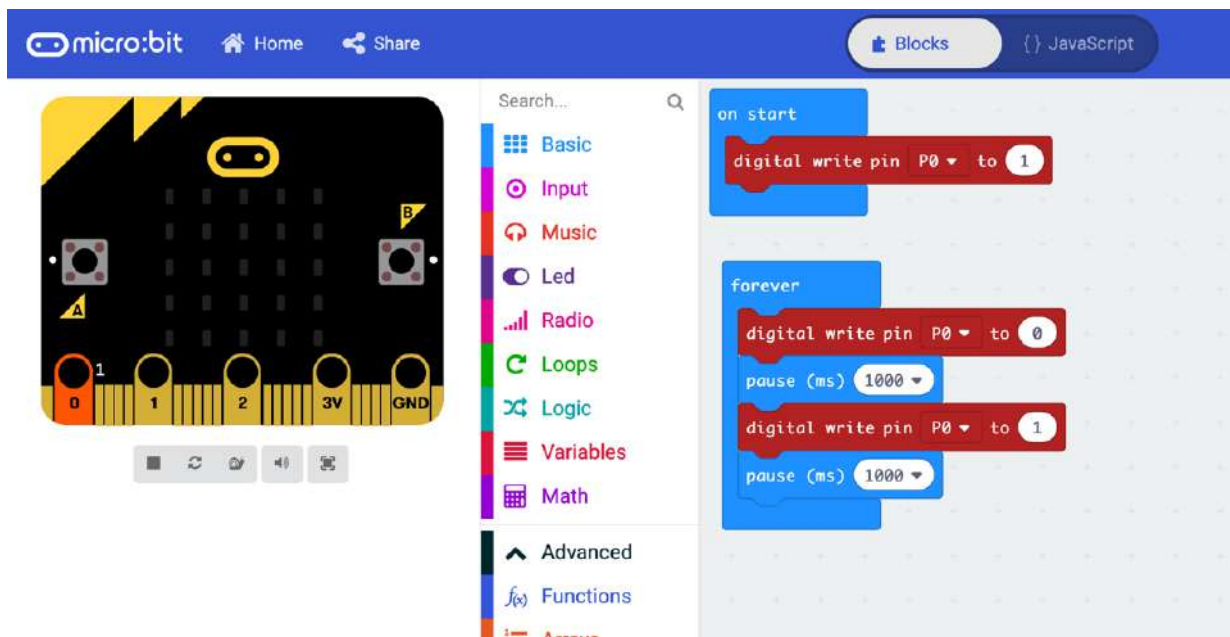
Micro:bit – ovládanie jednosmerných motorov

Video <https://www.youtube.com/watch?v=uwdCfGLY-A>

Rýchlosť otáčania jednosmerného motora závisí od úrovne privedeného napätia a smer otáčania závisí od polarítity privedeného napätia. V prvom príklade sa obmedzíme len na zapnutie a vypnutie motora zopnutím logickej úrovne na porte P0. Motor má väčší odber, než je schopná doska Micro:bit dodať pri napájaní cez USB, preto je potrebné napájať dosku z batérií a motor sa spína cez NPN tranzistor.



Jednosmerný motor, pripojený cez NPN tranzistor. Rezistor má hodnotu 100 Ohm



JavaScript:

```
pins.digitalWritePin(DigitalPin.P0, 1)
basic.forever(function () {
  pins.digitalWritePin(DigitalPin.P0, 0)
  basic.pause(1000)
  pins.digitalWritePin(DigitalPin.P0, 1)
  basic.pause(1000)
})
```

Micro:bit – ovládanie servomotorov

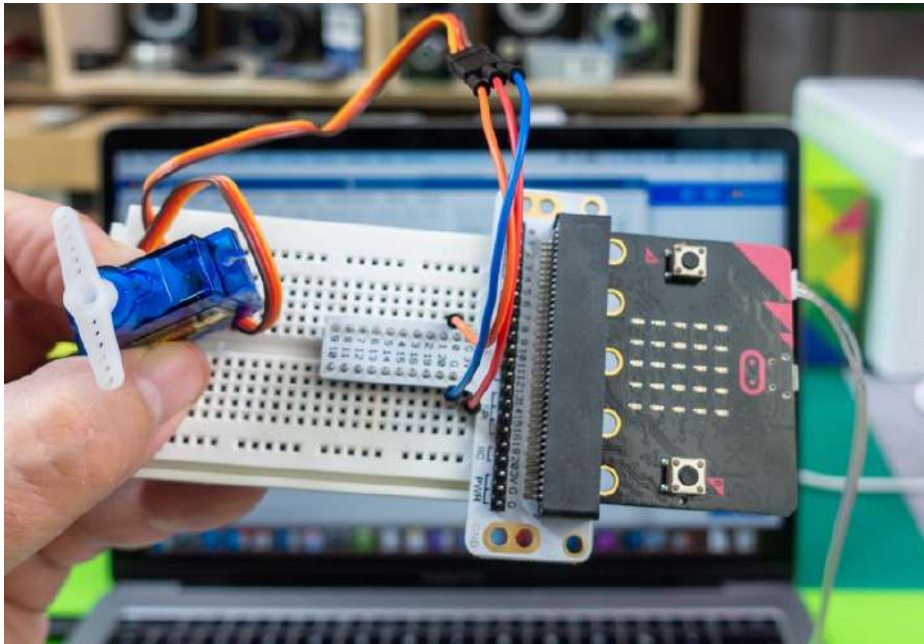
Video <https://www.youtube.com/watch?v=ptNAD9qTjml>

Servomotory, alebo skrátenie servá často využívajú modelári a konštruktéri robotov. Je to komponent umožňujúci pootočenie osky s ramenom o presne definovaný uhol, alebo v prípade kontinuálneho 360°serva sa dá regulovať rýchlosť a smer otáčania.



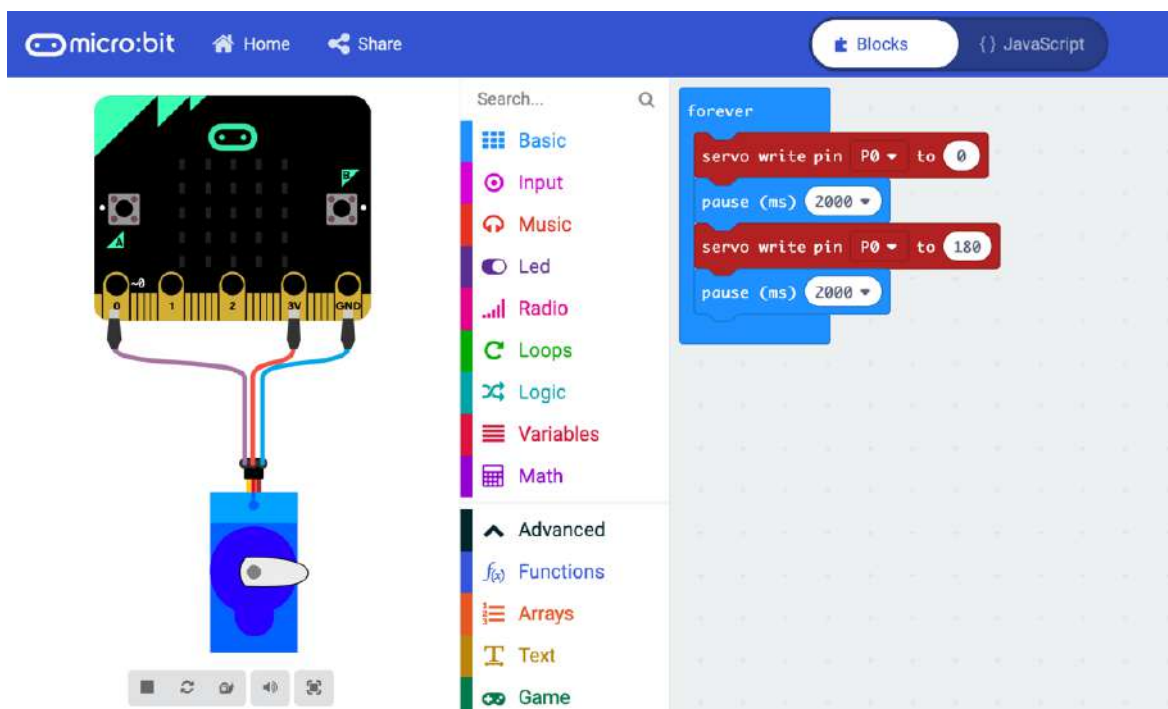
Môžete si ho predstaviť ako motor s prevodovkou, so spätnoväzobným potenciometrom, ktorým sa meria uhol natočenia a riadiacim obvodom. Pomocou PWM impulzov môžete regulovať uhol jeho natočenia. Servo je k riadiacej jednotke pripojené pomocou troch rôznofarebných vodičov. Na

červenom vodiči je kladné napájacie napätie, na čiernom (niekedy hnedom) vodiči je zem a na oranžový vodič sa privádzajú signály.



Pripojenie serva k Micro:bitu

Frekvencia impulzov privádzaných na oranžový vodič je 50 Hz. Uhol natočenia sa ovláda šírkou pozitívneho pinu. Napríklad u modelárskeho serva SG90 je šírka pulzu 500 až 2400 μ s a tejto hodnote zodpovedá uhol natočenia 0° až 180°. Na prvý pohľad to znie zložito, ale vďaka knižnici integrovanej do vývojového prostredia je ovládanie servomotorov veľmi jednoduché. Servomotor má väčší odber, než je schopná doska dodať pri napájaní cez USB, preto je potrebné napájať dosku z batérií. Oranžový vodič je vo vnútri serva pripojený na radiacu elektroniku, nemá veľký prúdový odber, takže ho môžete pripojiť priamo na port.



JavaScript:

```
input.onButtonPressed(Button.A, function () {
  pins.servoWritePin(AnalogPin.P0, 0)
})
input.onButtonPressed(Button.B, function () {
  pins.servoWritePin(AnalogPin.P0, 180)
})
input.onButtonPressed(Button.AB, function () {
  pins.servoWritePin(AnalogPin.P0, 90)
})
```

Micro:bit – stavebnica Grove na jednoduché zapojenia hardvérových komponentov

Video <https://www.youtube.com/watch?v=kLj8XR7o86A>

Ak ešte nemáte dostatok skúseností so zapojovaním elektronických obvodov a chcete si vyskúšať fungovanie čidiel a displejov a ďalších komponentov pripojených k vývojovým doskám, riešením sú modulové sú k dispozícii pre všetky populárne mikrokontrolérové vývojové dosky, čiže nielen pre Micro:bit, ale aj pre Arduino, či Raspberry. Veľa začiatočníkov, ktorí dosiaľ neexperimentovali so zapojením elektronických súčiastok má obavy aby obvody nezapojili zle a súčiastky, alebo Micro:bit nezničili. Pri pokusoch so súpravou Grove stačí moduly poprepájať káblíkmi, ktoré majú konektory.

Zapojovanie obvodov pomocou stavebnice Grove je navyše veľmi jednoduché, takže nemusíte špekulovať ako zapojiť obvody a môžete sa sústrediť na programovanie. Micro:bit má síce vyvedené tri porty na široké plôšky s otvormi, takže do nich môžete zasunúť vodiče s banánikmi, prípadne krokosvorky, ostatné porty sú však na úzkych plôškach plošného spoja, takže potrebujete redukciu, aby ste mohli realizovať zapojenie na prepojovacom poli.



Každý modul stavebnice plní jednu funkciu. Potrebne moduly stačí pripojiť základnej doske Base Shield a ste pripravení na testovanie svojich nápadov. Po overení fungovania si môžete skúsiť realizovať zapojenie na prepojovacom poli, prípadne pri finálnom zariadení na univerzálnej doske plošného spoja. Súprava Grove Inventor kit for Micro:bit obsahuje modul s LED diódou, snímačom

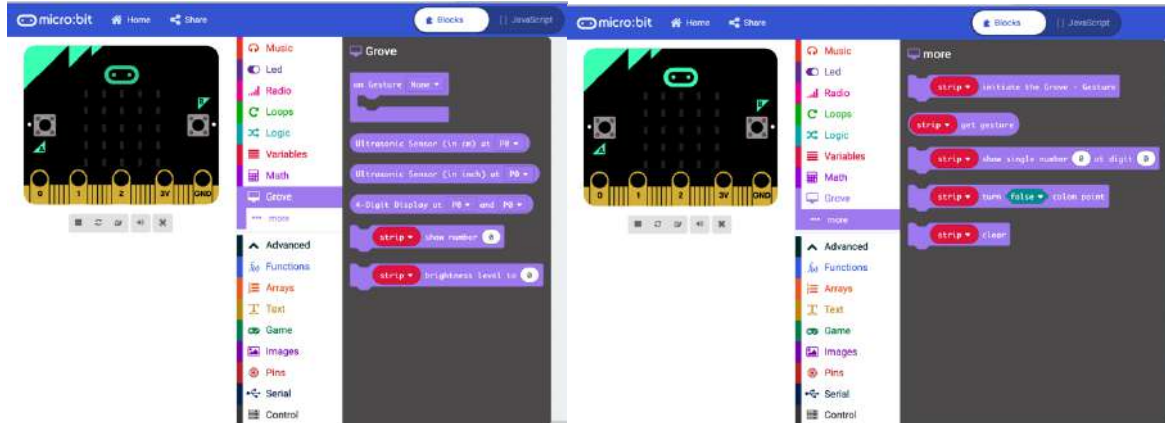
intenzity osvetlenia, potenciometrom, reproduktorom, ultrazvukovým meračom vzdialenosti, snímačom giest, sedemsegmentovým displejom so štyrmi číslicami a RGB LED pásik.



Na základnej doske Base Shield, do ktorej sa zasunie vývojová doska BBC Micro:bit sú štyri konektory. Jeden je označený I2C, k nemu sa pripájajú moduly využívajúce túto komunikačnú zbernicu. Súprava Grove Inventor kit obsahuje len jeden takýto modul - snímač giest. Ostatné moduly využívajú univerzálne analógové a digitálne vstupy a výstupy. Tie sú na ďalších troch konektoroch vždy po dvojiciach. P0/P14, P1/P15 a P2/P16. Na každom konektore je vyvedené aj napájacie napätie 3,3 V a zem GND.



Aby ste mohli jednoducho programovať aj zložitejšie komponenty ako je sedemsegmentový displej, ultrazvukový snímač vzdialeností, pásik RGB LED diód, či snímač pohybových giest potrebujete v online vývojom prostredí <https://makecode.microbit.org> pripojiť knižnice. Pripojíte ich pomocou položky Extensions. Potrebujete knižnicu github.com/seeed-studio/pxt-grove a aby ste mohli využiť pásik RGB LED diód aj Neopixel.

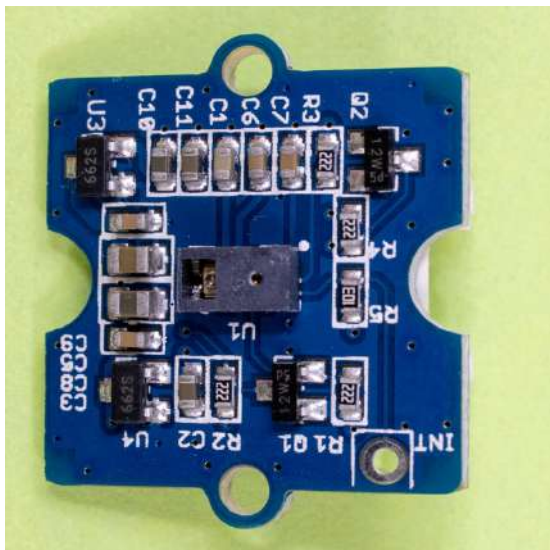


Funkcie v rozširujúcom module Grove

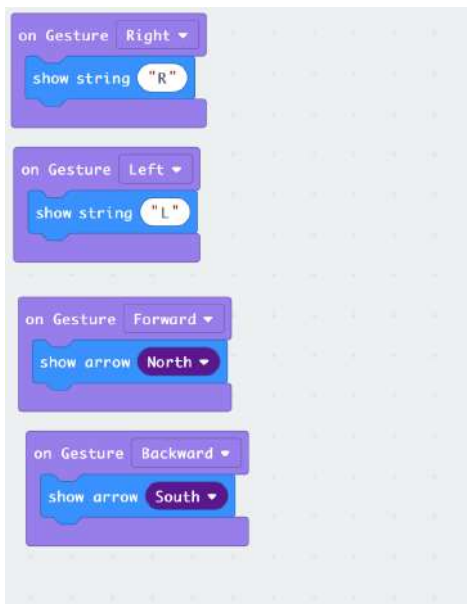
Nasleduje niekoľko príkladov možnosti stavby Grove. Ak stavbu nemáte, nevadí stačí zapojiť obvody obsahujúce príslušné hardvérové komponenty zapojiť na prepojovacom poli

Micro:bit + Grove – snimanie giest v priestore

Modul na rozpoznávanie giest je potrebné pripojiť na konektor označený I2C. Rozpoznávanie smeru a rýchlosti pohybu gesta funguje na podobnom princípe ako snímač optickej myši.



Podľa toho, ktorým smerom (doľava, doprava) budete pohybovať rukou pred snímačom zobrazí sa na maticovom displeji symbol L, alebo R



JavaScript:

```

grove.onGesture(GroveGesture.Right, function () {
  basic.showString("R")
})
grove.onGesture(GroveGesture.Left, function () {
  basic.showString("L")
})
grove.onGesture(GroveGesture.Forward, function () {
  basic.showArrow(ArrowNames.North)
})
grove.onGesture(GroveGesture.Backward, function () {
  basic.showArrow(ArrowNames.South)
})

```

Micro:bit + Grove – meranie vzdialenosti ultrazvukom

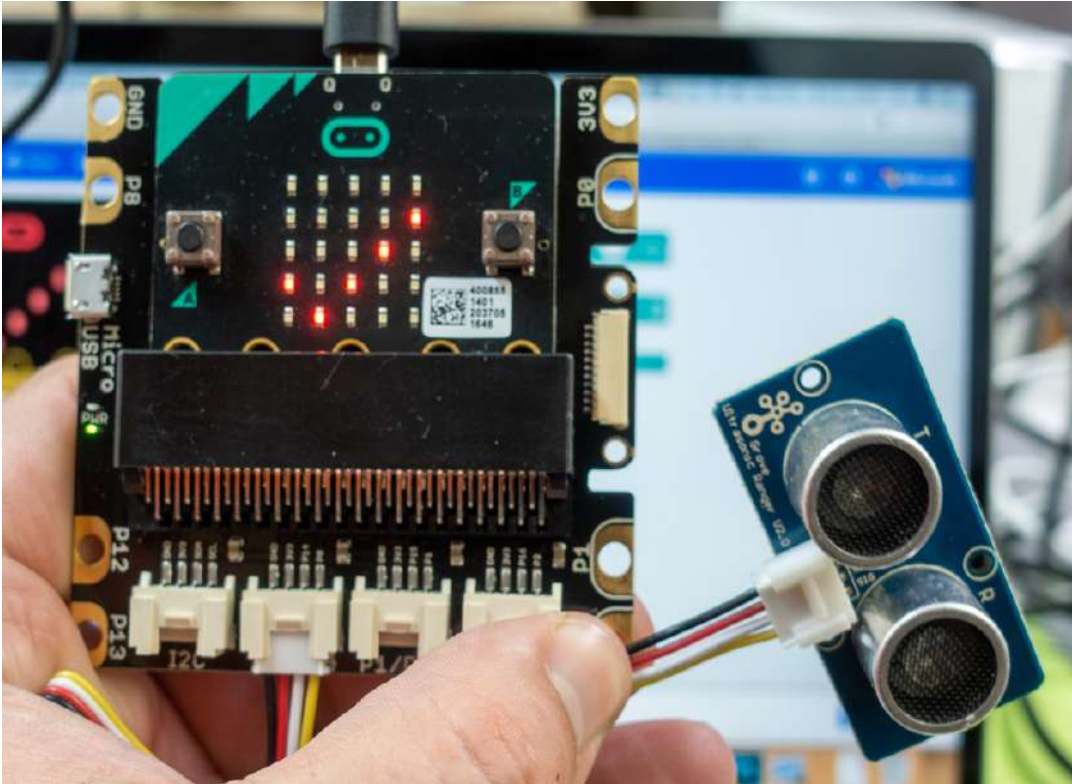
Ultrazvukový merač vzdialenosti indikuje vzdialenosť predmetu, napríklad ruky od snímača. Vzdialenosť za meria na základe oneskorenia ultrazvukového signálu, ktorý sa od predmetu odrazí. Možno vás prekvapí, že modul sa pripája len na jeden digitálny vstup P0/P14. Funguje tak, že jeho elektronika generuje obdĺžnikový signál a šírka signálu závisí od nameranej vzdialenosti. Pri experimentoch s ultrazvukovým snímačom použite externý batériový zdroj napájania.



V príklade je vzdialenosť predmetu nad 10 cm indikovaná ikonou OK a ak sa predmet priblíži k snímaču bližšie, zobrazí sa ikona „X“

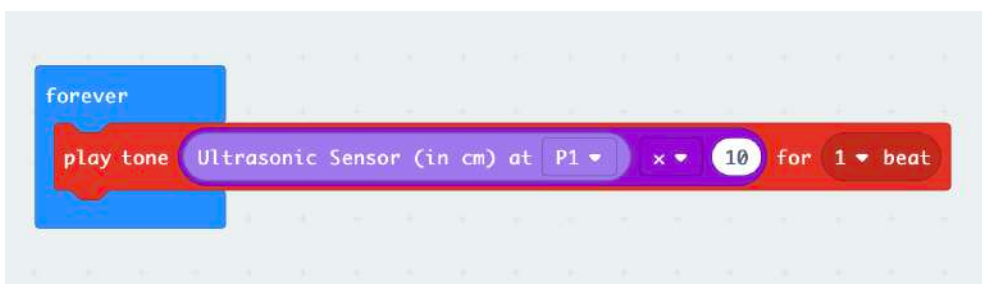
JavaScript:

```
basic.forever(function () {  
  if (grove.measureInCentimeters(DigitalPin.P0) >= 10) {  
    basic.showIcon(IconNames.Yes)  
  } else {  
    basic.showIcon(IconNames.No)  
  }  
  basic.pause(100)  
})
```



Micro:bit + Grove – hudobný nástroj Theremin

Námetom príkladu simulátor hudobného nástroja Theremin, ktorý vyludzuje tóny podľa pohybov rúk v priestore. V príklade sa výška tónu mení podľa vzdialenosti ruky od ultrazvukového snímača. Aby to spoľahlivo fungovalo pripojte napájanie z batérie.

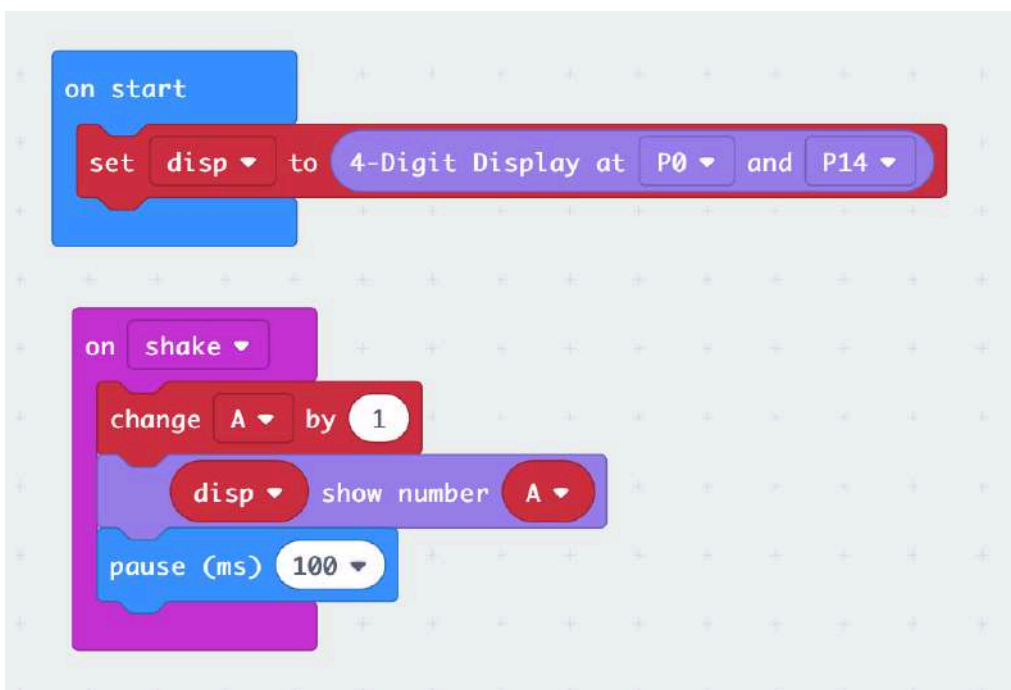


JavaScript:

```
basic.forever(function () {  
  music.playTone(grove.measureInCentimeters(DigitalPin.P1) * 10,  
    music.beat(BeatFraction.Whole))  
})
```

Micro:bit + Grove – zobrazovanie na displeji

V ďalšom príklade použijeme modul displeja, na ktorom budeme zobrazovať počet potrasení. Doska Micro:bit obsahuje akcelerometer, takže stačí obslúžiť udalosti generované týmto hardvérovým komponentom. Aj displej sa pripája na port P0/P14 a využíva obidva tieto digitálne výstupy.

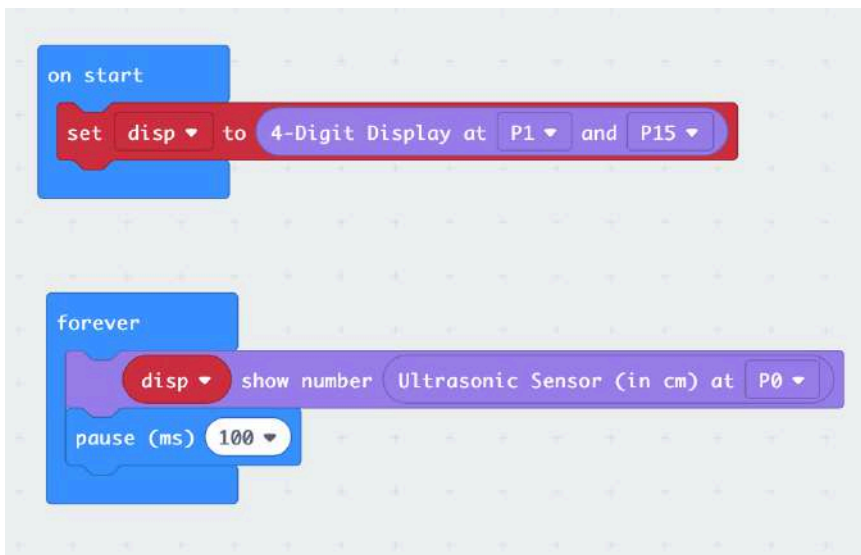
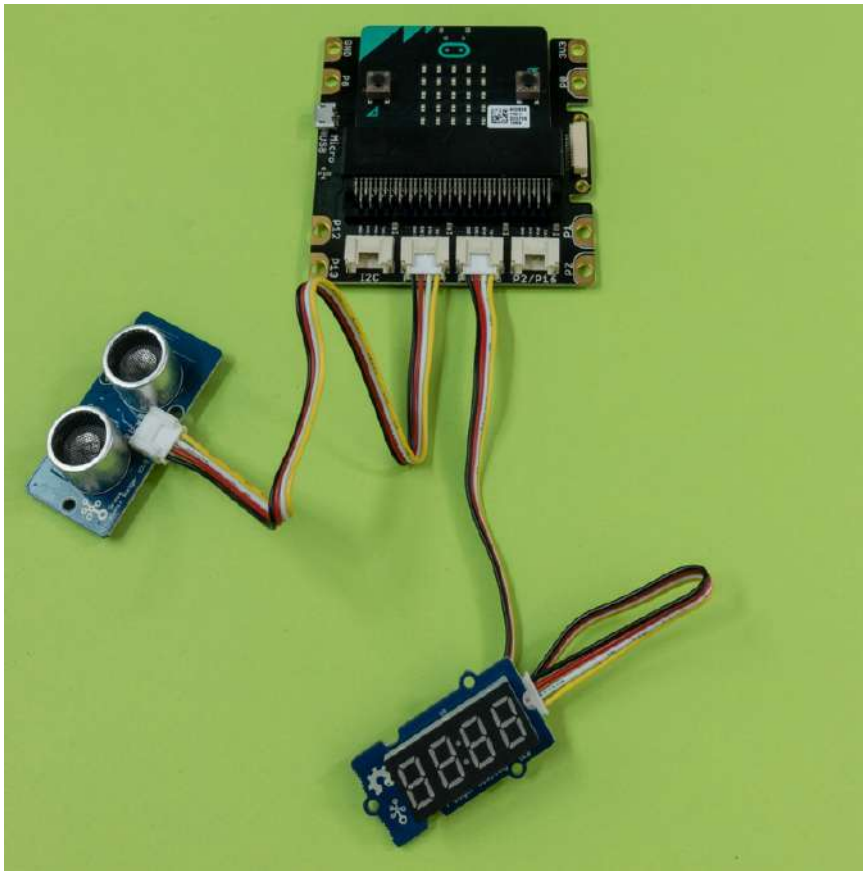


JavaScript:

```
let A = 0
let disp: grove.TM1637 = null
input.onGesture(Gesture.Shake, function () {
  A += 1
  disp.show(A)
  basic.pause(100)
})
disp = grove.createDisplay(DigitalPin.P0, DigitalPin.P14)
```

Micro:bit + Grove – meranie a zobrazovanie vzdialenosti

V nasledujúcom príklade je potrebné pripojiť dva moduly. Ultrazvukový merač vzdialenosti na konektor P0/P14 a displej na konektor P1/P15

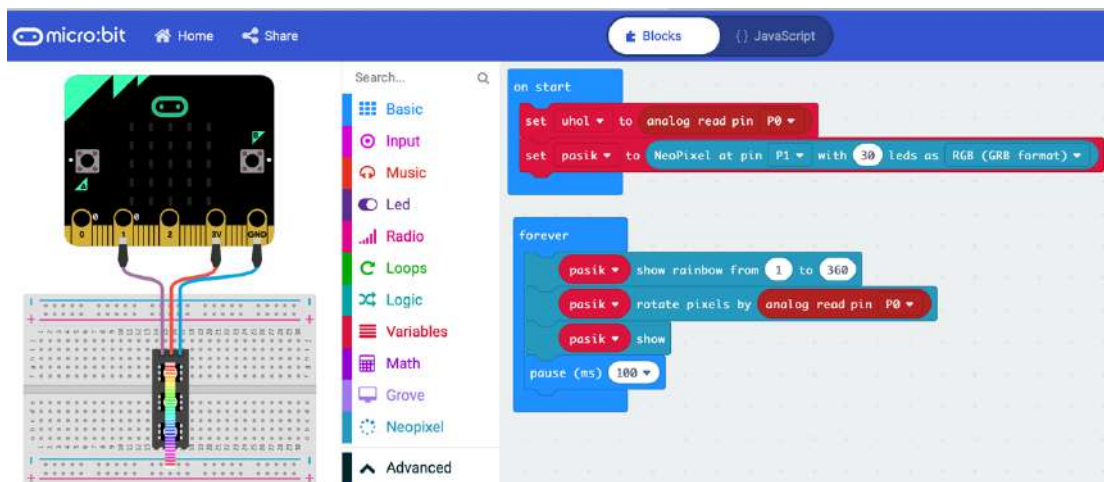
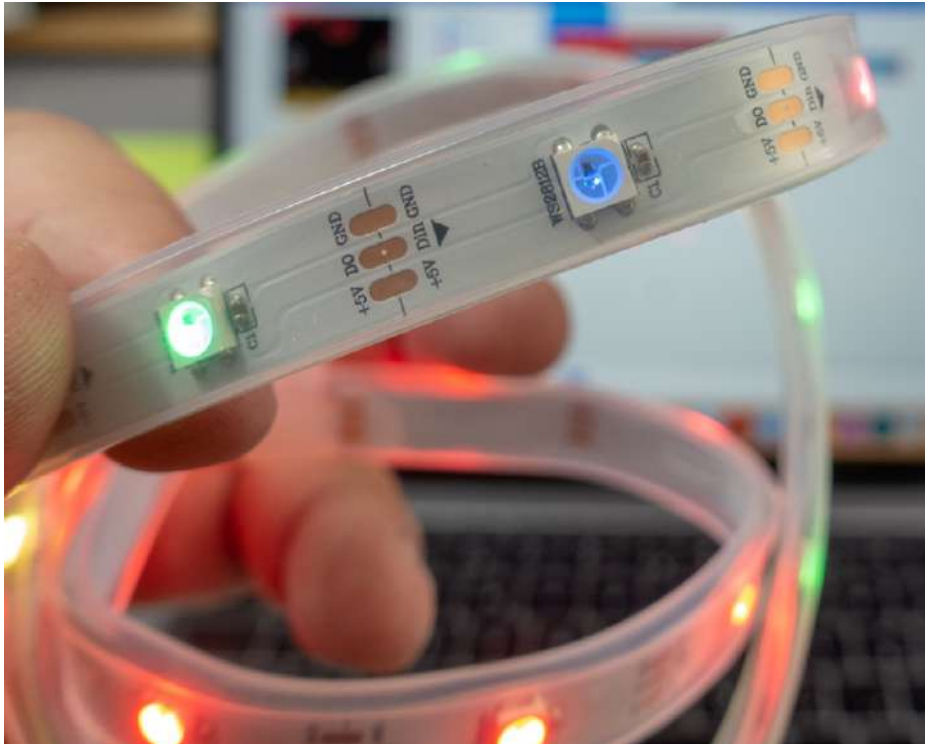


JavaScript:

```
let disp: grove.TM1637 = null
disp = grove.createDisplay(DigitalPin.P1, DigitalPin.P15)
basic.forever(function () {
  disp.show(grove.measureInCentimeters (DigitalPin.P0))
  basic.pause(100)
})
```

Micro:bit + Grove – pásik farebných RGB diód

V príklade je LED pásik pripojený ku konektoru P0/P14. Aby ste ho mohli programovať, potrebujete pridať knižnicu Neopixel. Blikanie a rotáciu farieb budeme ovládať potenciometrom pripojeným ku konektoru P0/P14

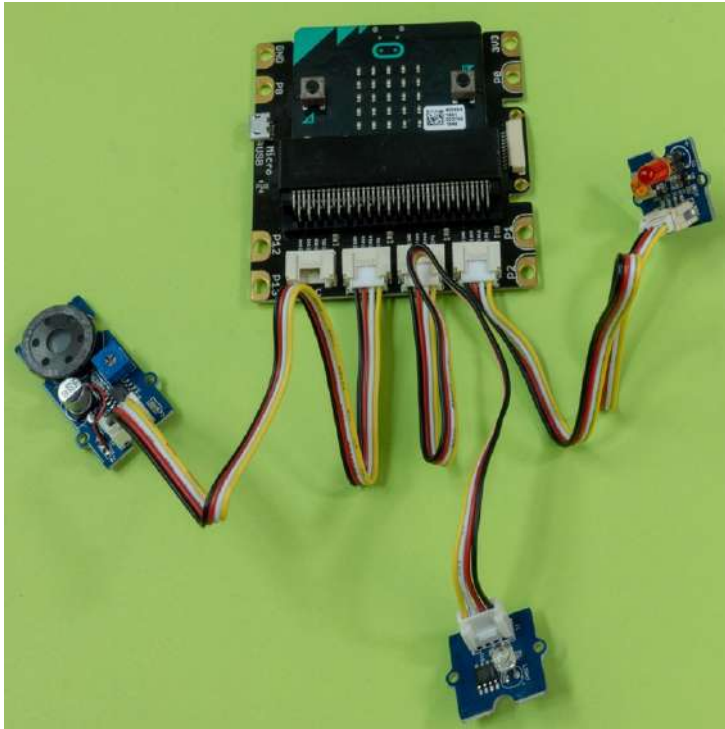


JavaScript:

```
let pasik: neopixel.Strip = null
let uhol = 0
uhol = pins.analogReadPin(AnalogPin.P0)
pasik = neopixel.create(DigitalPin.P1, 30, NeopixelMode.RGB)
basic.forever(function () {
  pasik.showRainbow(1, 360)
  pasik.rotate(pins.analogReadPin(AnalogPin.P0))
  pasik.show()
  basic.pause(100)
})
```

Micro:bit + Grove – zabezpečujúce zariadenie indikujúce otvorenie skrinky, zásuvky alebo tašky

Námetom príkladu je ochrana tajnej zásuvky, alebo tašky. Otvorenie zistí snímač svetla a indikuje sa výstražným tónom a blikaním LED diódy. Na konektor P0/P14 pripojte reproduktor, na P1/P15 svetelný senzor a na P2/P16 modul s červenou LED diódou



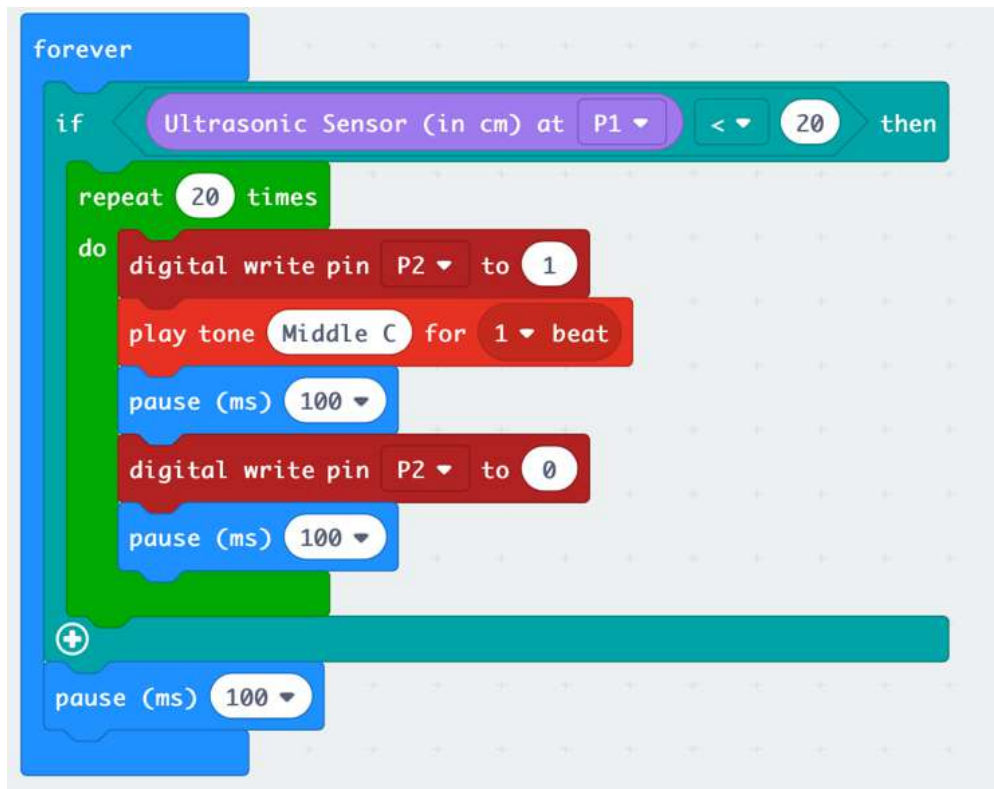
```
forever
  set svetlo to analog read pin P1
  if svetlo >= 100 then
    digital write pin P2 to 1
    play tone Middle C for 1 beat
    pause (ms) 100
    digital write pin P2 to 0
    pause (ms) 100
  else
    digital write pin P0 to 0
    digital write pin P2 to 0
  pause (ms) 100
```

JavaScript:

```
let svetlo = 0
basic.forever(function () {
  svetlo = pins.analogReadPin(AnalogPin.P1)
  if (svetlo >= 100) {
    pins.digitalWritePin(DigitalPin.P2, 1)
    music.playTone(262, music.beat(BeatFraction.Whole))
    basic.pause(100)
    pins.digitalWritePin(DigitalPin.P2, 0)
    basic.pause(100)
  } else {
    pins.digitalWritePin(DigitalPin.P0, 0)
    pins.digitalWritePin(DigitalPin.P2, 0)
  }
  basic.pause(100)
})
```

Micro:bit + Grove – zabezpečujúce zariadenie indikujúce prbliženie sa

V príklade je prototyp bezpečnostného zariadenia využívajúceho ultrazvukový merač vzdialenosti. Po prekročení nastavenej vzdialenosti sa ozve výstražný tón a blikanie. Na konektor P0/P14 pripojte reproduktor, na P1/P15 ultrazvukový senzor a na P2/P16 modul s červenou LED diódou



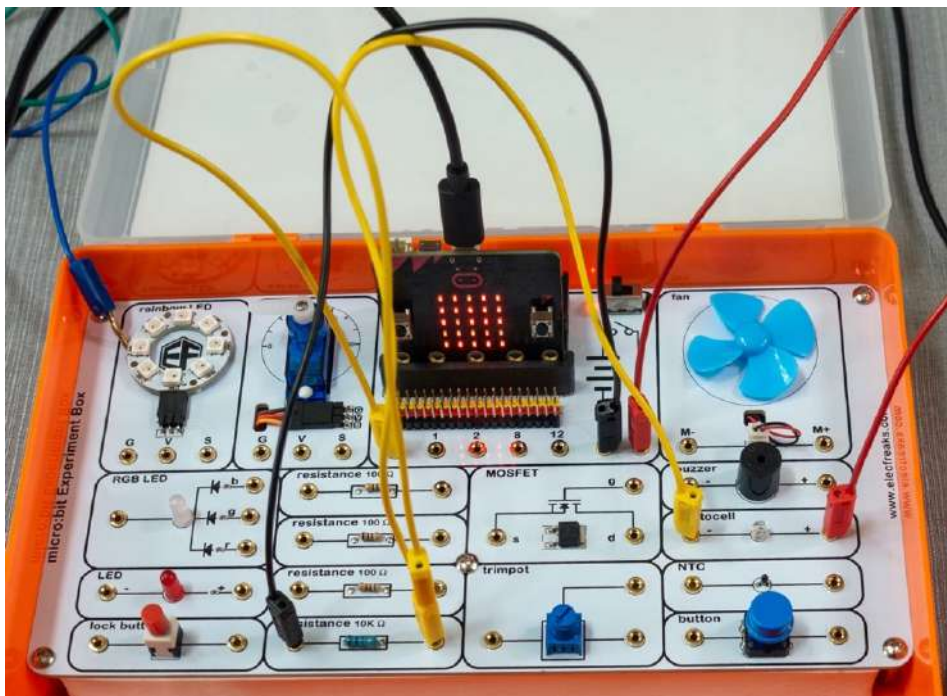
JavaScript:

```
basic.forever(function () {
  if (grove.measureInCentimeters(DigitalPin.P1) < 5) {
    for (let i = 0; i < 20; i++) {
      pins.digitalWritePin(DigitalPin.P2, 1)
      music.playTone(262, music.beat(BeatFraction.Whole))
      basic.pause(100)
      pins.digitalWritePin(DigitalPin.P2, 0)
      basic.pause(100)
    }
  }
  basic.pause(100)
})
```

Micro:bit Experiment Box na jednoduché pokusy s hardvérom

Video <https://www.youtube.com/watch?v=gHXlxcTStRQ>

Súprava Experiment Kit uľahčí žiakom prvé kroky pri poznávaní elektroniky a základov programovania. Využíva populárnu mikropočítačovú dosku BBC micro:bit. Obsahuje LED diódu, RGB LED, servomotor, jednosmerný motor, snímače teploty a osvetlenia, motor, tlačidlo, bzučiak, spínací MOSFET tranzistor, RGB LED prstenec, potenciometer a rezistory. Na základnej doske, na ktorej sú sústredené všetky súčiastky, je aj konektor typu GVS na pripojenie prípadných ďalších zariadení. Na zdielky, do ktorých sa zasunujú káble s banánikmi, sú vyvedené porty 0, 1, 2, 8 a 12. V súprave je 12 prepojujúcich káblikov s banánikmi dlhých 28 cm.



Plastový kufřík je konštruovaný tak, že sa dá zavrieť bez toho aby bolo treba zapojenie rozpojiť.



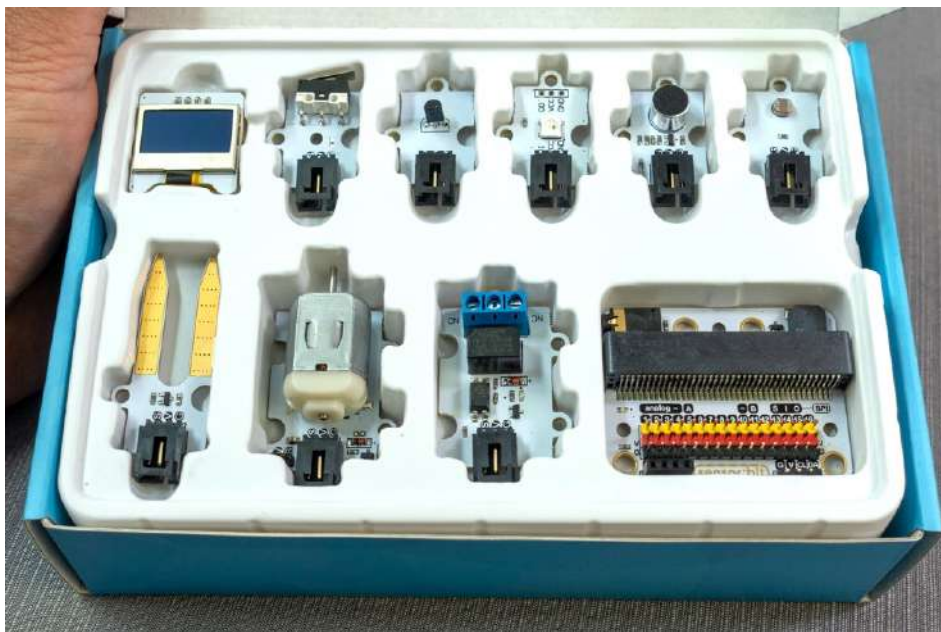
Micro:bit Smarthom kit na projekty simulujúce inteligentný dom

Video <https://www.youtube.com/watch?v=RVFhC-lsimg>

Systémy pre inteligentnú domácnosť sú vďačným námetom aj na školské, či hobby projekty s populárnou mikropočítačovou doskou BBC Micro:bit, ktorá sa používa v mnohých školách na výučbu programovania a robotiky. Všetky potrebné komponenty nájdete v súprave Smart home Kit.

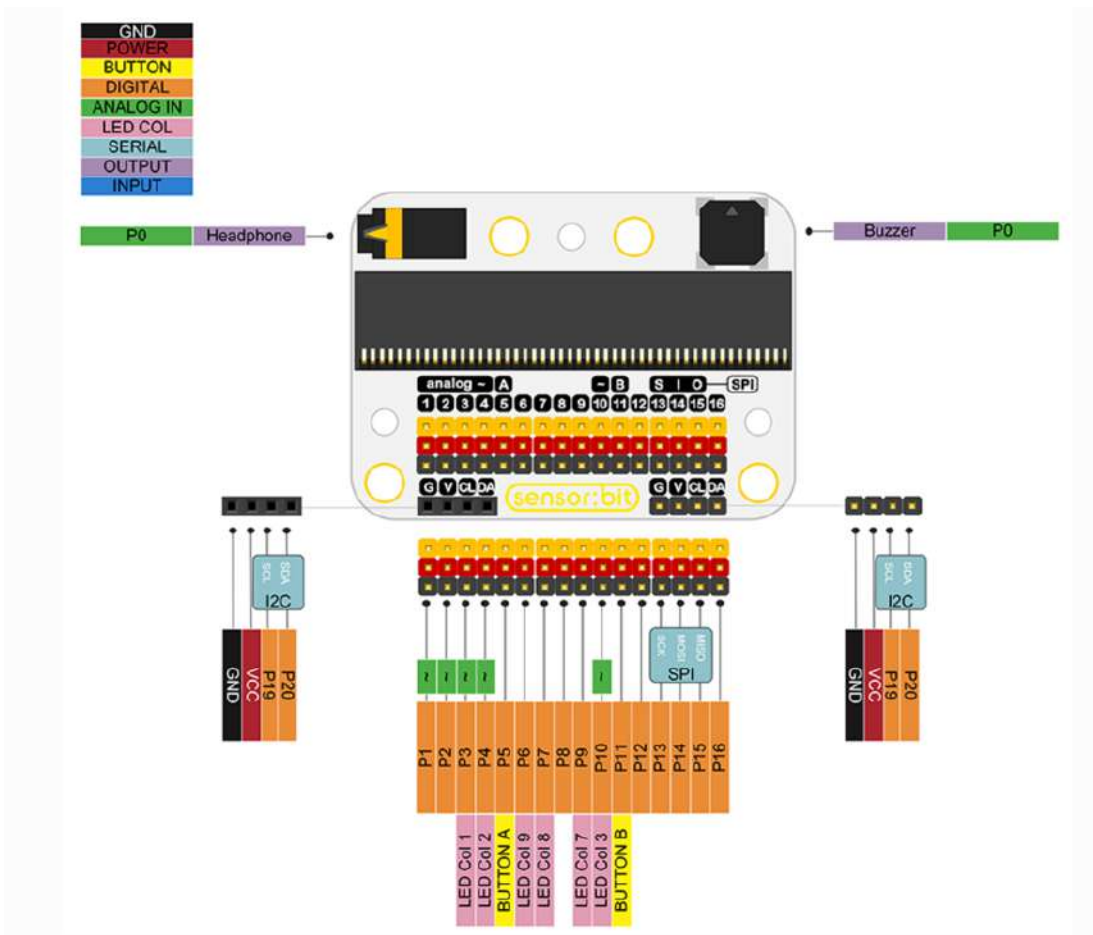
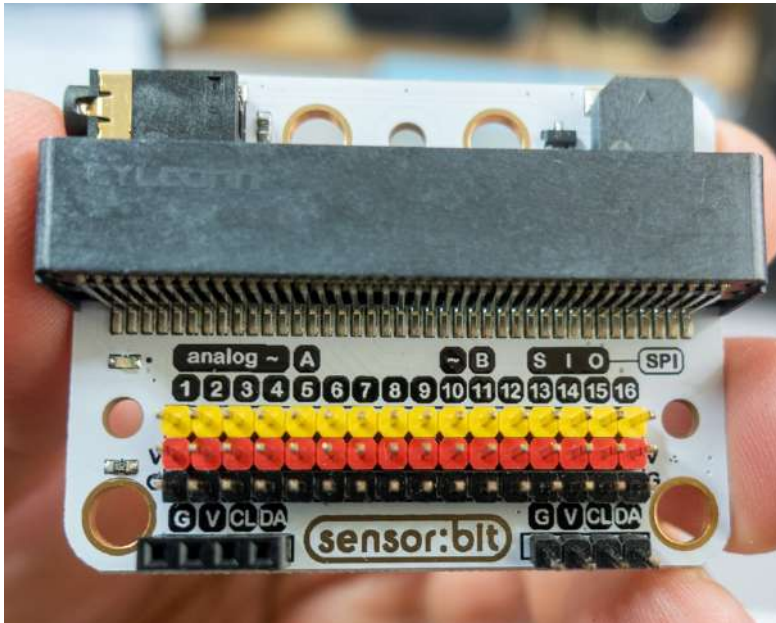


Súprava obsahuje najpoužívanejšie komponenty domácej automatizácie ako je teplotné čidlo, senzor intenzity zvuku, senzor na meranie intenzity svetla, snímač vlhkosti, jednosmerný motorček, servomotor, relé, OLED displej, modul s RGB diódou, čerpadlo na vodu a ďalšie súčiastky.



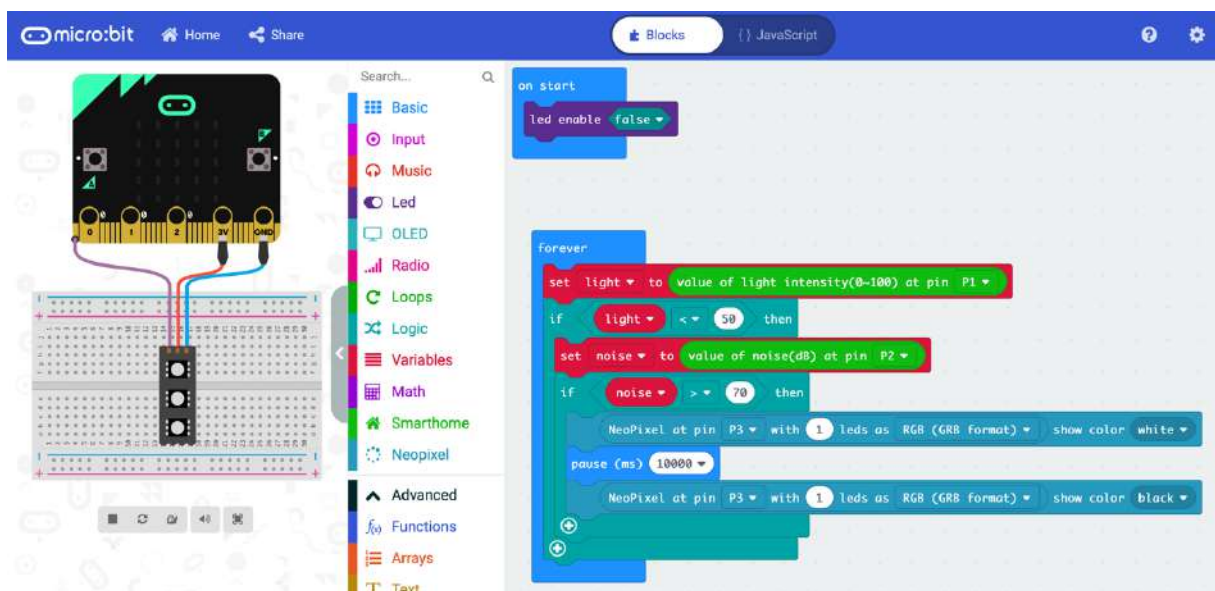
Samozrejme všetko je prispôbené tak aby fungovalo s bezpečným napätím. Na napájanie všetkých projektov vám stačia dve tužkové batérie.

Základom súpravy modul Sensor:bit, do ktorého sa zasunie mikropočítačová doska BBC Micro:bit. Modul má konektor, na ktorom je 16 GVS vstupov/výstupov a tiež konektor s rozhraním I2C do ktorého je možné priamo zasunúť OLED displej. Súčasťou modulu je aj mini reproduktor a konektor Jack 3.5 mm na pripojenie slúchadiel, alebo externého reproduktora.



Popis portov modulu sensor:bit

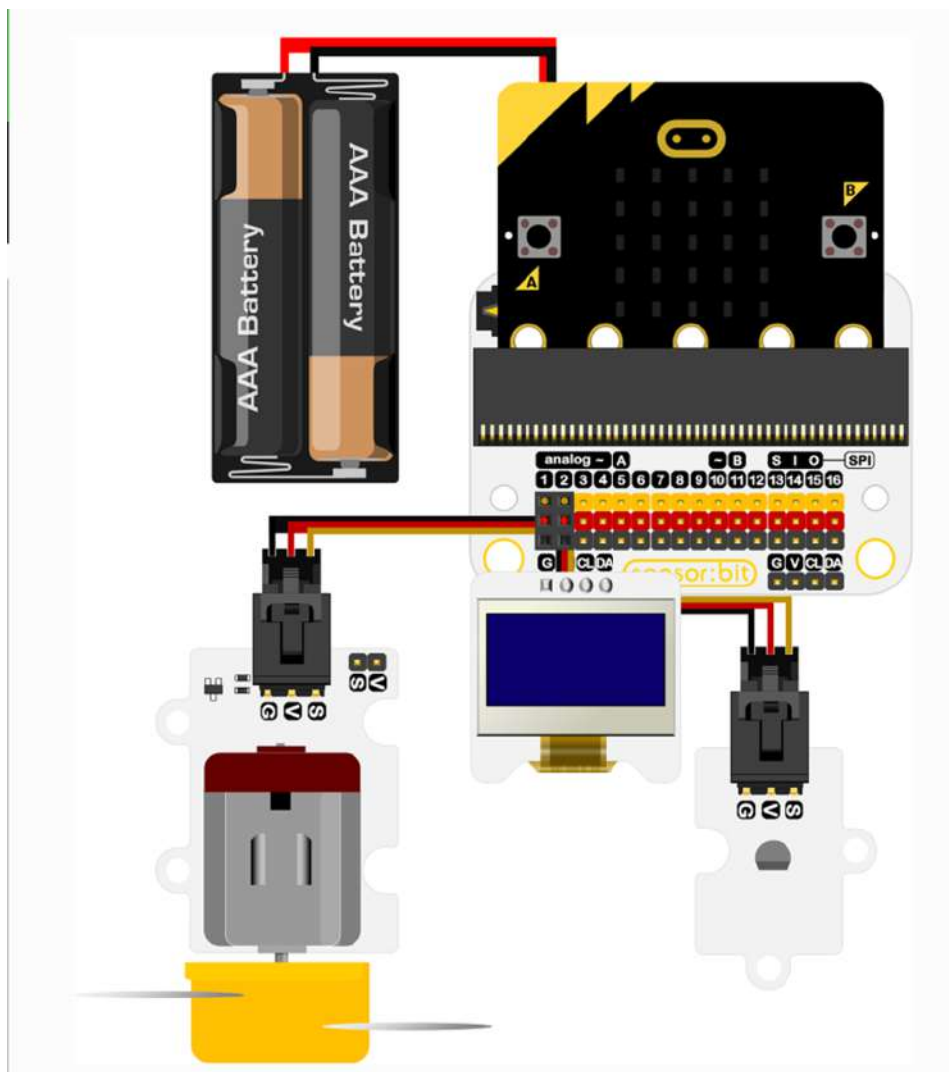
Začneme jednoduchým námetom svetla, ktoré sa rozsvieti zvukom. Po zaznamenaní zvuku určitej intenzity sa RGB LED dióda rozsvieti na 10 sekúnd. V originálnom návode vás výrobca motivuje aby ste urobili z kartónovej, alebo drevenek škatule model miestnosti, aby fungovanie modelu bolo čo najzrozumiteľnejšie.



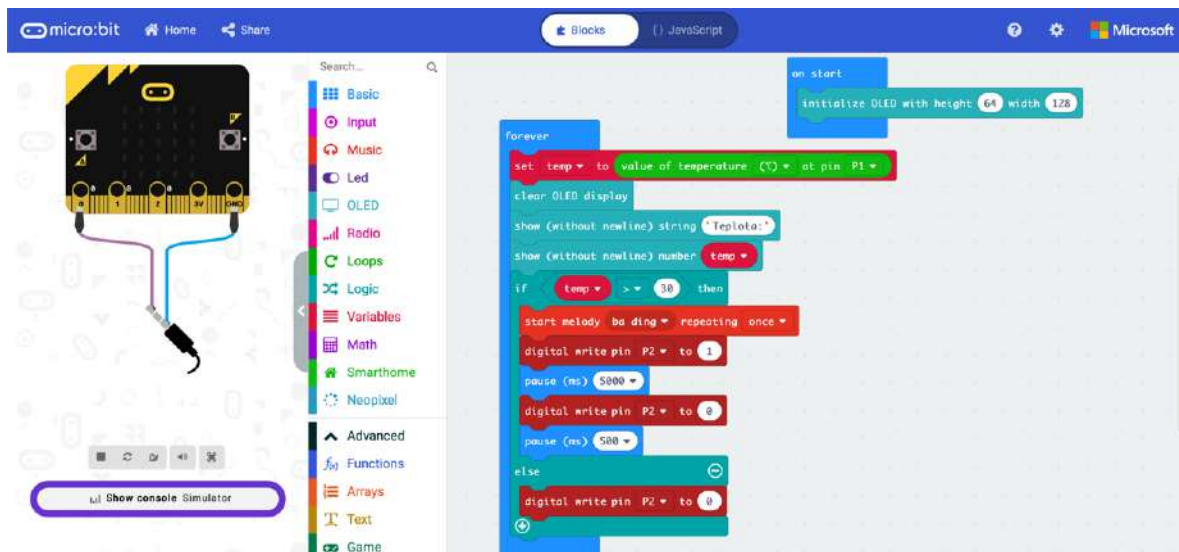
JavaScript:

```
let noise = 0
let light = 0
led.enable(false)
basic.forever(function () {
  light = smarthome.ReadLightIntensity(AnalogPin.P1)
  if (light < 50) {
    noise = smarthome.ReadNoise(AnalogPin.P2)
    if (noise > 70) {
      neopixel.create(DigitalPin.P3, 1,
NeoPixelMode.RGB).showColor(neopixel.colors(NeoPixelColors.White))
      basic.pause(10000)
      neopixel.create(DigitalPin.P3, 1,
NeoPixelMode.RGB).showColor(neopixel.colors(NeoPixelColors.Black))
    }
  }
})
```

Ďalší príklad simuluje reguláciu kúrenia, alebo klimatizácie. Akčný člen je simulovaný ventilátorom poháňaným DC motorčekom. Ak teplotné čidlo nameria vyššiu teplotu ako 30 °C, bude sa na 5 sekúnd opakovane zapínať ventilátor, kým sa teplota nezniží.



Ventilátorom sa samozrejme dá teplota znížiť iba vtedy ak ste čidlo ohriali napríklad držaním v ruke. Teplotu miestnosti ventilátorom neznížite, dosiahnete len prúdenie vzduchu a tým lepšie ochladzovanie tela.



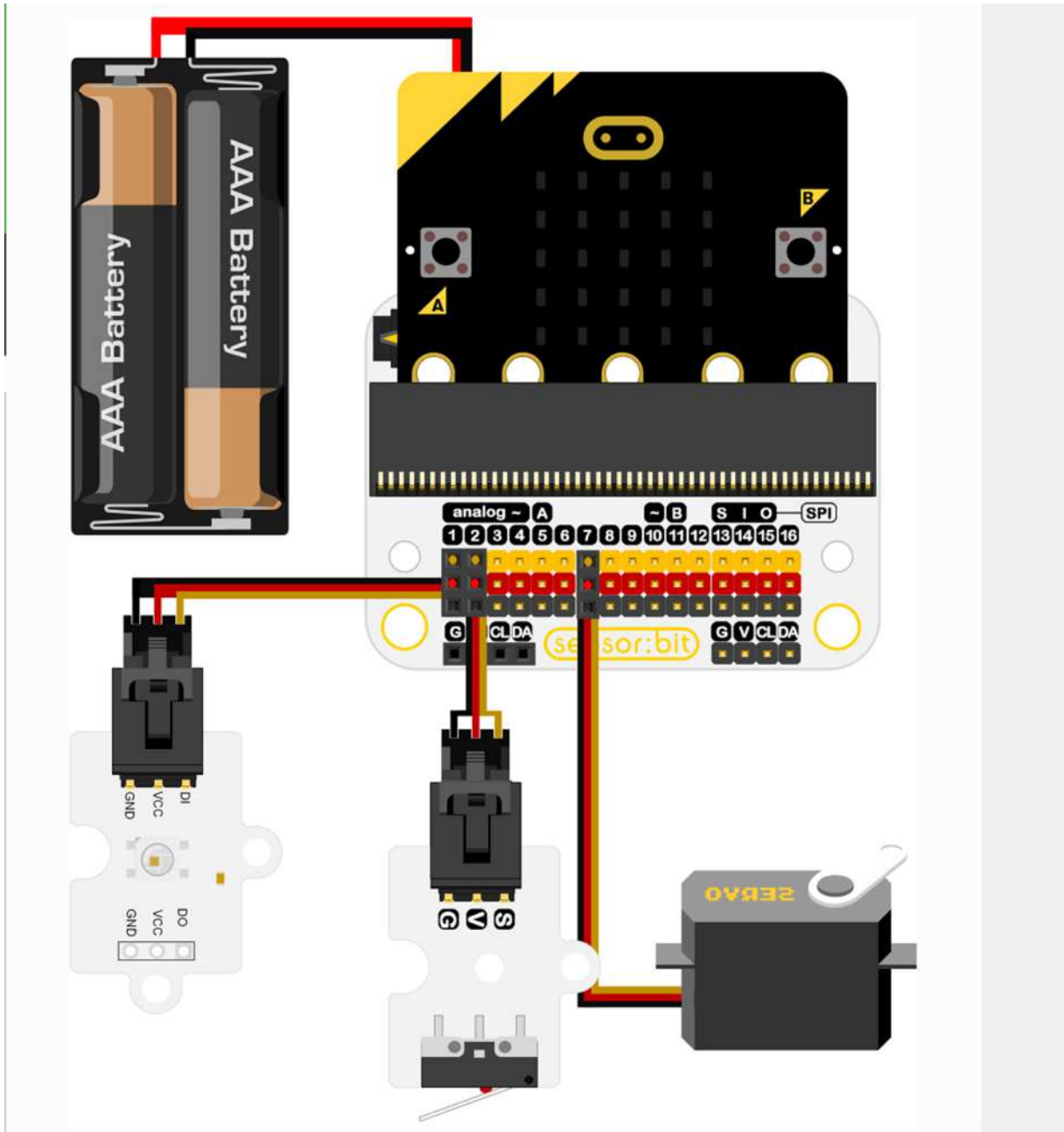
JavaScript:

```

let teplota = 0
OLED.init(64, 128)
basic.forever(function () {
    teplota = smartHome.ReadTemperature(TMP36Type.TMP36_temperature_C,
    AnalogPin.P1)
    OLED.clear()
    OLED.showStringNoNewLine("Teplota:")
    OLED.showNumberNoNewLine(teplota)
    if (teplota > 30) {
        music.beginMelody(music.builtInMelody(Melodies.BaDing), MelodyOptions.Once)
        pins.digitalWritePin(DigitalPin.P2, 1)
        basic.pause(5000)
        pins.digitalWritePin(DigitalPin.P2, 0)
        basic.pause(500)
    } else {
        pins.digitalWritePin(DigitalPin.P2, 0)
    }
})

```

V nasledujúcom príklade sa simuluje otvorenie okna, prípadne skrine pomocou servomotora



```

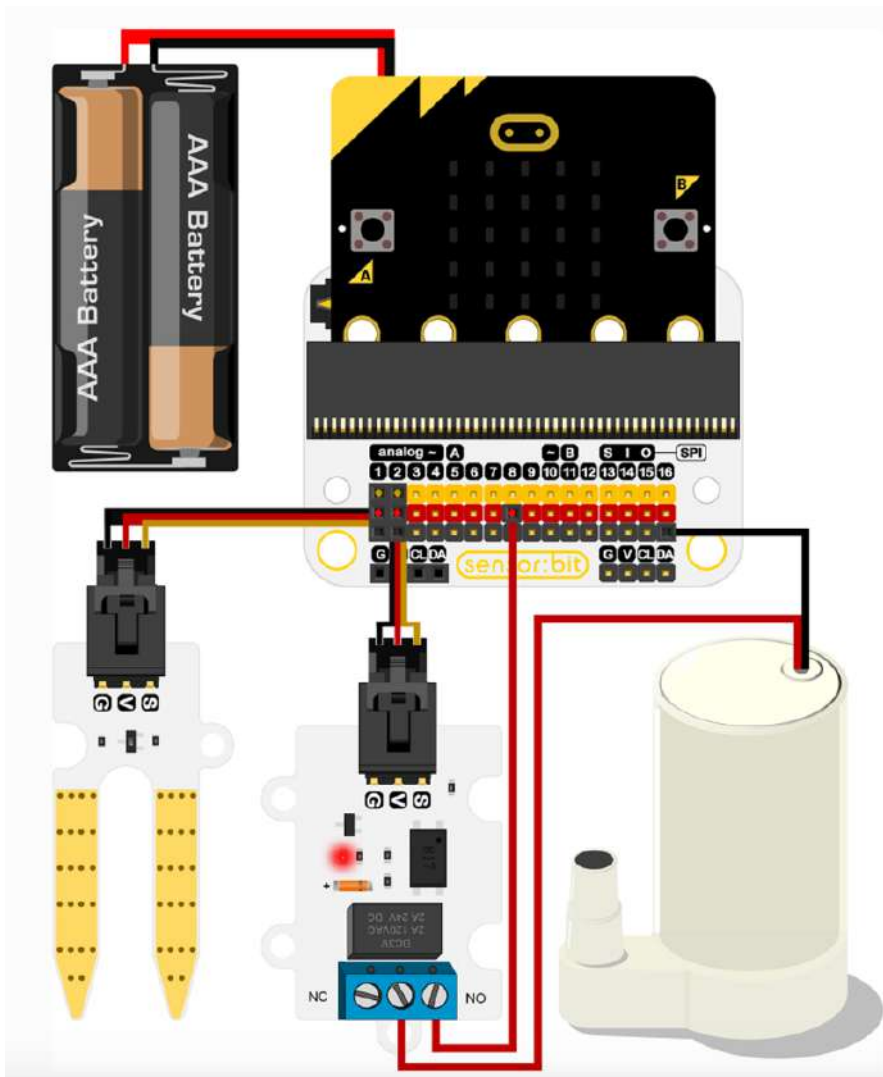
on start
  set pull pin P2 to up
  servo write pin P7 (write only) to 180
  set okno to -1

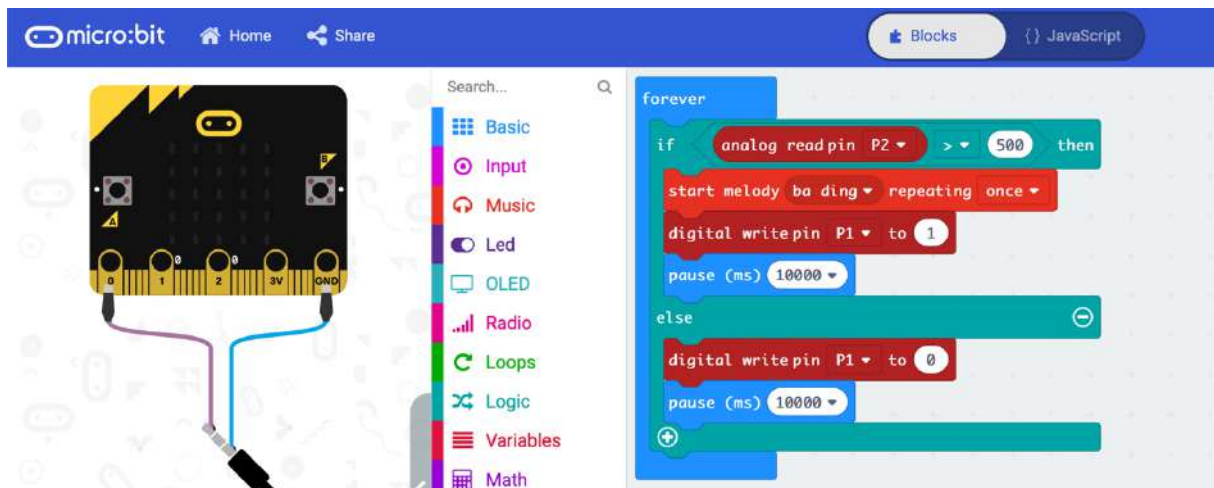
forever
  if digital read pin P2 == 0 then
    set okno to okno + x + -1
  if okno == 1 then
    NeoPixel at pin P1 with 1 leds as RGB (GRB format) range from 0 with 1 leds show color white
    servo write pin P7 (write only) to 0
    pause (ms) 2000
  else
    servo write pin P7 (write only) to 180
    pause (ms) 2000
    NeoPixel at pin P1 with 1 leds as RGB (GRB format) range from 0 with 1 leds show color black
  
```

JavaScript:

```
pins.setPull(DigitalPin.P2, PinPullMode.PullUp)
pins.servoWritePin(AnalogPin.P7, 180)
let okno = -1
basic.forever(function () {
  if (pins.digitalReadPin(DigitalPin.P2) == 0) {
    okno = okno * -1
    if (okno == 1) {
      neopixel.create(DigitalPin.P1, 1, NeoPixelMode.RGB).range(0,
1).showColor(neopixel.colors(NeoPixelColors.White))
      pins.servoWritePin(AnalogPin.P7, 0)
      basic.pause(2000)
    } else {
      pins.servoWritePin(AnalogPin.P7, 180)
      basic.pause(2000)
      neopixel.create(DigitalPin.P1, 1, NeoPixelMode.RGB).range(0,
1).showColor(neopixel.colors(NeoPixelColors.Black))
    }
  }
})
```

A napokon príklad, v ktorom ak senzor identifikuje zaplavenie vodou, zapne sa čerpadlo, ktoré vodu vyčerpá. Modifikáciou príkladu by ste mohli vytvoriť napríklad systém na automatické polievanie kvetov v kvetináči.





JavaScript:

```
basic.forever(function () {  
  if (pins.analogReadPin(AnalogPin.P1) > 500) {  
    music.beginMelody(music.builtInMelody(Melodies.BaDing), MelodyOptions.Once)  
    pins.digitalWritePin(DigitalPin.P2, 1)  
    basic.pause(10000)  
  } else {  
    pins.digitalWritePin(DigitalPin.P2, 0)  
    basic.pause(10000)  
  }  
})
```


Programovací jazyk Python

Podľa výsledkov posledného prieskumu portálu Stack Overflow je Python najrýchlejšie sa rozvíjajúci programovací jazyk v zmysle nárastu počtu vývojárov, ktorí ho používajú. V popularite už Python prekonal aj C#. Má všetky atribúty profesionálneho programovacieho jazyka, umožňuje rýchly vývoj aplikačného kódu, podporuje objektovo orientované programovanie a je vynikajúci pre začiatočníkov. Zároveň však Python búra zaužívaný mýtus, že jazyk vhodný na výučbu nie je vhodný pre prax a naopak. Podstatnou mierou k popularite Pythonu prispieva čistota a jednoduchosť syntaxe na jednej strane a možnosť nízkoúrovňových volaní a obrovské množstvo dostupných funkcií na strane druhej. Preto ho využívajú aj hackeri, odborníci na umelú inteligenciu a profesionálni programátori pracujúci vo veľkých globálnych firmách.

Python má už za sebou takmer 30 ročnú históriu. Vytvoril ho v roku 1989 Guido van Rossum, ktorý vtedy pracoval vo výskumnom ústave v Amsterdame. Nový programovací jazyk, ktorý mal byť jednoduchý a pritom by umožňoval aj nízkoúrovňové systémové volania vytvoril Van Rossum ako svoj hobby projekt cez dovolenku počas vianočných sviatkov. Jazyk nie je pomenovaný podľa hada, ako by sa mohli mnohí domnievať, ale podľa populárneho britského seriálu Monty Python's Flying Circus. Ako človek z praxe, snažil sa Van Rossum vytvoriť programovací jazyk na jednoduché a rýchle naprogramovanie bežných aj náročných úloh. Python je vyvíjaný ako open source projekt a na jeho vývoji participuje komunita nadšencov.

Napriek tomu, že Python je jednoduchý a prehľadný, rozhodne nezaostáva za ostatnými najpoužívanejšími programovacími jazykmi. Podporuje objektovo orientované, štruktúrované aj funkcionálne programovanie. Python bol navrhnutý tak, aby bol čo najjednoduchší a maximálne rozšíriteľný. Preto má malé jadro a množstvo knižníc na najrôznejšie úlohy, prípadne knižnice zohľadňujúce špecifiká platforiem na ktorých môže kód v Pythone bežať.

Pre migrantov z iných programovacích jazykov ohľadne zvláštností syntaxe Pythonu stačí informácia, že na definíciu štruktúry programových blokov ako sú telá procedúr, cyklov, alebo kódu, ktorý sa má vykonať v určitej verry podmienky sa nepoužívajú oddeľovače typu zložené zátvorky, ale len odsadenie riadkov kódu medzerami, alebo tabulátormi. Vďaka tomu je kód veľmi prehľadný.

Napríklad:

```
def fib(n):
    a, b = 0, 1
    while a < n:
        print(a, end=' ')
        a, b = b, a+b
    print()
fib(1000)
```

Tomuto krátkemu programovému kódu, ktorého úlohou je vypísať Fibonacciho čísla menšie ako zadaná hodnota, v príklade je volaná procedúra s parametrom 1000 porozumie aj začiatočník. Po preštudovaní niekoľkých príkladov a osvojení si syntaktického minima bude veľmi rýchlo vedieť realizovať v Pythone svoje nápady. Naproti tomu ak by ste ukázali začiatočníkovi kód v programovacom jazyku C, alebo C++, kde sa to len tak hemží smerníkmi, asi by ste ho dosť znechutili a nadobudol by dojem, že programovanie je veľmi zložitý.

Krátka ukážka kódu v predchádzajúcej stati prezradí aj ďalšiu zvláštnosť. Python nepoužíva deklarovanie typov premenných, ale typ premennej sa určí automaticky. Taktiež nie je problém v jednom riadku priradiť do viacerých premenných rôzne hodnoty, alebo výsledky výrazov. Na prvý pohľad by sa mohlo zdať, že skutočnosť, že Python nie je kompilovaný, ale interpretovaný jazyk bude

veľkým obmedzením. Je všeobecne známe, že kompilovaný kód beží podstatne rýchlejšie, než kód interpretovaný. To je síce pravda, ale interpreter zas umožňuje rýchlejšiu interakciu pri písaní a ladení kódu. Netreba kompilovať rozsiahle projekty, stačí spustiť interpreter. Jadro jazyka má malý počet základných dátových typov, konštrukcií a funkcií, avšak dajú sa rôzne kombinovať. Programové konštrukcie sa dajú štrukturovať, môžete vytvárať moduly, triedy, či bloky kódu. Všetky prvky v Pythone sú objekty. Programovací jazyk Python podporuje rôzne štýly programovania, takže si ho obľúbia tak priaznivci objektovo orientovaného programovania, prípadne programovania imperatívneho, či agilného.

Vďaka prehľadnosti a veľa ďalším výhodným vlastnostiam tvrdenie že Python programátor dokáže urobiť za mesiac to, čo nedokáže C++ programátor za rok nie je ďaleko od pravdy. Produktivita tvorby kódu sa prejaví tak pri najjednoduchších programoch, ako aj pri veľmi rozsiahlych aplikáciách. Pri jednoduchých, často jednoúčelových programoch určených napríklad na automatizáciu merania, prípadne programoch v študentských projektoch programátori ocenia predovšetkým stručnosť zápisu. U veľkých aplikácií sa vysoká produktivnosť dosahuje napríklad prirodzenou podporou menných priestorov, používaním výnimiek, či štandardne dodávanými nástrojmi na tvorbu testov. S vysokou produktivnosťou súvisí dostupnosť a jednoduchá použiteľnosť širokej škály knižničných modulov, umožňujúcich jednoduché riešenie úloh z prakticky akejkoľvek oblasti, vrátane sofistikovaných úloh z oblasti strojového učenia a umelej inteligencie.

Python je najlepšou voľbou nielen pre začiatočníkov, ale aj hackerov, testerov hardvéru a softvéru a taktiež na účely reverzného inžinierstva. Výhodou je, že ktokoľvek v ňom rýchlo a ľahko napíše potrebný kód, ktorý je prehľadný ľahko čitateľný. Vývojár sa pritom môže plne sústrediť na riešenie danej úlohy a nie na programové konštrukcie. Navyše majú vývojári k dispozícii veľké množstvo knižníc, počnúc nízkoúrovňovou podporou hardvéru, cez databázy, autentifikáciu, kontrolu správnosti používateľom zadaných hodnôt a podobne

Začiatočníci z radov žiakov vyšších ročníkov základných škôl spravidla prechádzajú na Python po tom, ako si osvojili základy algoritmizácie v grafických programovacích prostrediach typu Scratch, kde sa kód aplikačnej logiky tvorí skladaním do seba zapadajúcich blokov typu puzzle a dopĺňaním parametrov. Podľa skúsenosti z krúžkov zameraných na informatiku, programovanie a robotiku je hlavne u talentovaných žiakov dôležité, aby na vyšší, „textový“ programovací jazyk prešli čo najskôr, pretože grafické programovanie má svoje limity a žiaci po zvládnutí základov algoritmizácie na ne veľmi rýchlo narazia. Paradoxne, hlavným limitom grafického programovania je prehľadnosť, čiže atribút, ktorý sa spočiatku javil ako najväčšia výhoda Scratchu. Zložitejší stavový automat napríklad na komplexné ovládanie robota je oveľa prehľadnejší v Pythone, než v grafickom znázornení.

Aplikácie v Pythone pre Micro:bit

Na programovanie aplikácií pre mikropočítačovú dosku Micro:bit môžete použiť webový nástroj na adrese <https://python.microbit.org/v/1.1>, vtedy nebudete musieť na lokálny počítač nič inštalovať. Po vytvorení programu v prostredí si stiahnete z webu jeho binárny kód v súbore s príponou hex. Micro:bit, pripojený k počítaču pomocou USB kábla, sa na počítači zobrazí ako klasické USB úložisko, na ktoré stiahnutý súbor s programom stačí skopírovať. Program sa následne automaticky spustí na zariadení. V prípade aplikácie v jazyku Python, tento hex súbor obsahuje nielen kód aplikácie ale aj engine interpretera jazyka MicroPython

Je to trochu menej praktické, preto vývoj, zostavenie a zavedenie kódu do dosky Micro:bit môžete realizovať priamo na vývojárskom počítači. Stačí vám k tomu nástroj Mu-editor, ktorý je k dispozícii na <https://codewith.mu>. Program v jazyku Python pomocou tlačidla Flash zavediete priamo do

mikropočítačovej dosky Micro:bit pripojenej cez USB. Nemusíte tak ako u webových vývojových nástrojov riešiť stiahnutie súboru s príponou hex a jeho kopírovanie do micro:bitu-

Nebudeme primárne riešiť syntax jazyka Python, ale budeme sa venovať implementačným špecifikám jazyka Micro Python na mikropočítačovej doske BBC micro:bit, čiže obsluhou jeho hardvérových komponentov, čiže displeja, tlačidiel, zabudovaných senzorov a samozrejme aj možnosti práce so zbernicou, na ktorú je možné pripojiť rôzne zapojenia a externé moduly. BBC micro:bit má jednoduchý maticový displej z 25 LED diód, tlačidlá, senzor pohybu, naklonenia, teploty a intenzitu okolitého svetla. Podporuje dva druhy bezdrôtovej komunikácie - Bluetooth Low Energy (BLE) pre komunikáciu s mobilnými zariadeniami a rádiový prenos pre komunikáciu s inými zariadeniami BBC micro:bit. Taktiež je možná sériová komunikácia s počítačom cez USB kábel.

Príkazom `from microbit import *` sa importujú knižnice obsahujúce špecifické funkcie pre microbit. Hviezdička určuje, že sa importujú všetky knižnice

Micro:bit Python - zobrazovanie na maticovom displeji

Zobrazovacie možnosti dosky Micro:bit sú pomerne skromné, k dispozícii je maticový displej z LED diód v usporiadaní 5 x 5. Jednotlivé LED sú individuálne adresované od (0,0) v ľavom hornom rohu až po (4,4). Tretím parametrom funkcie `set_pixel()` je jas príslušnej LED diódy

```
from microbit import *
display.set_pixel(0,4,9)
```

v ďalšom príklade je ukázané rozsvetovanie LED diód maticového displeja v cykle

```
from microbit import *
display.clear()
for x in range(0, 5):
    for y in range(0, 5):
        display.set_pixel(x,y,9)
```

Na displeji možné staticky zobrazíť jeden znak, alebo iný grafický symbol. Viac sa tam nevojde. Napriek tomu je aj na takomto displeji možné zobrazíť aj niekoľko znakový textový reťazec. Trik je v tom že text vodorovne roluje.

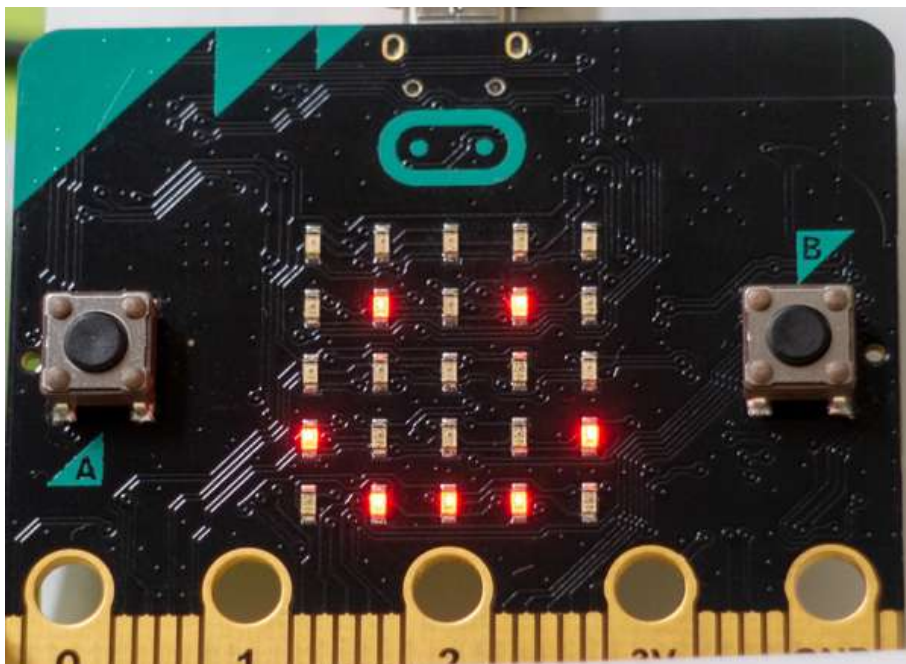
```
from microbit import *
display.scroll("Ahojte vsetci!")
```

Na zobrazenie statickej ikony preddefinovaného obrázku použijete príkaz v tvare

```
from microbit import *
display.show(Image.HAPPY)
```

K dispozícii je veľké množstvo preddefinovaných obrázkov

HEART, HEART_SMALL, HAPPY, SMILE, SAD, CONFUSED, ANGRY, ASLEEP, SMILE, SURPRISED, SILLY, FABULOUS, MEH, YES, NO, TRIANGLE, TRIANGLE_LEFT, CHESSBOARD, DIAMOND, DIAMOND_SMALL, SQUARE, SQUARE_SMALL, RABBIT, COW, MUSIC_CROTCHET, MUSIC_QUAVER, PITCHFORK, XMAS, PACMAN, TARGET, TSHIRT, ROLLERSKATE, DUCK, HOUSE, TORTOISE, BUTTERFLY, STICKFIGURE, GHOST, SWORD, GIRAFFE, SKULL, UMBRELLA, SNAKE



Zobraziť sa dajú ikony šípok: ARROW_N, ARROW_NE, ARROW_E, ARROW_SE, ARROW_S, ARROW_SW, ARROW_W, ARROW_NW kde smer je určený skratkami svetových strán na podčiarníkom a dokonca aj v analógovej forme vyjadriť čas s presnosťou na hodiny: CLOCK1 - CLOCK12

Hodiny aj šípky je možné zobraziť aj ako animovanú sekvenciu pomocou definovaných objektov polí ALL_ARROWS a ALL_CLOCKS . Takto môžete vytvoriť napríklad animovaný ukazovateľ hodín, ktorý používateľovi znázorňuje, že sa na niečo čaká:

```
from microbit import *
display.show(Image.ALL_CLOCKS, loop=True, delay=200)
```

Micro:bit Python - definovanie vlastných obrázkov

Ak vám žiadna ikona z vymenovanej pomerne bohatej ponuky nevyhovuje, môžete si v matici 5 x 5 LED diód, ktoré tvoria pixely, nadefinovať vlastný obrázok. V päťmiestnych textových reťazcoch čísla 0 znamená zhasnutú LED diódu a číslami od 1 do 9 definujete jas rozsvietených diód. Napríklad

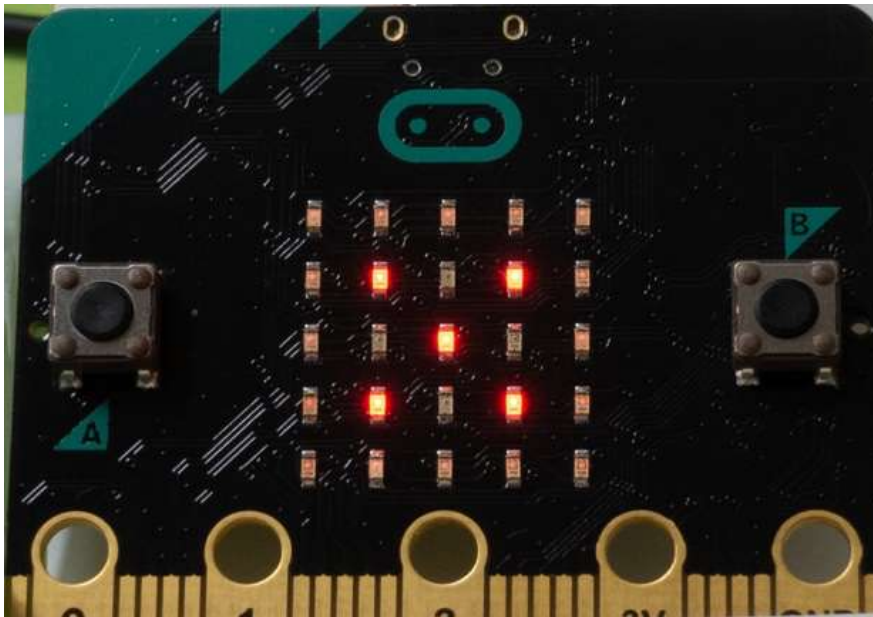
```
from microbit import *
lodka = Image("05050:05050:05050:99999:09990")
display.show(lodka)
```

Nie každý má takú predstavivosť, aby takto vedel navrhnuť obrázok, prípadne aby si z takéhoto zápisu vedel predstaviť, čo sa zobrazí. Oveľa prehľadnejší je takýto zápis, kedy číslice v reťazcoch tvoria maticu 5 x 5 bodov.

```
from microbit import *
obr = Image("33333:"
            "39093:"
            "30903:"
            "39093:"
            "33333")
```

```
display.show(obr)
```

Uhádli ste, je to hracia kocka, na ktorej je zobrazené číslo 5 a vidno aj obrysy kocky.



Aj z vlastných obrázkov je možné vytvoriť animáciu, stačí z nich vytvoriť pole. Napríklad hod kockou (pre jednoduchosť v jednoduchej postupnej sekvencii) by ste zobrazili pomocou animácie

```
from microbit import *
dice1 = Image("33333:"
              "30003:"
              "30903:"
              "30003:"
              "33333")

dice2 = Image("33333:"
              "30093:"
              "30003:"
              "39003:"
              "33333")

dice3 = Image("33333:"
              "39003:"
              "30903:"
              "30093:"
              "33333")

dice4 = Image("33333:"
              "39093:"
              "30003:"
              "39093:"
              "33333")

dice5 = Image("33333:"
              "39093:"
              "30903:"
              "39093:"
              "33333")

dice6 = Image("33333:"
              "39093:"
              "39093:"
              "39093:"
              "33333")

all_obr = [dice1, dice2, dice3, dice4, dice5, dice6]
display.show(all_obr, delay=500)
```

Micro:bit Python – zisťovanie stavu tlačidiel

Budeme rozlišovať prvky na interakciu dosky s používateľom a piny na zbernici určené na interakciu s pripojenými zariadeniami. Na interakciu s používateľom sú k dispozícii primárne tlačidlá A a B. V kóde sú reprezentované objektmi `button_a` a `button_b`. Migrantom z iných jazykov pripomíname, že telo cyklu `while True:` tvoria v jazyku Python všetky príkazy, ktoré sú pod týmto príkazom definujúcim cyklus odsadené od ľavého okraja začiatku príkazu o jeden tabulátor, prípadne alebo ako vývojári často používajú odsadenie o 4 medzery. V jednom súbore kódu však môžete použiť iba jeden druh odsadenia, buď tabulátormi, alebo medzerami. Rovnako sú odsadené aj príkazy v podmienke typu `if`.

```
from microbit import *
while True:
    if button_a.is_pressed() and button_b.is_pressed():
        display.scroll("AB")
        break
    elif button_a.is_pressed():
        display.scroll("A")
    elif button_b.is_pressed():
        display.scroll("B")
    sleep(100)
```

Funkcia `is_pressed()` kontroluje či bolo tlačidlo zatlačené v danom okamihu. V mnohých prípadoch je užitočná funkcia `was_pressed()`, ktorá vám prezradí, či tlačidlo bolo stlačené v čase, keď kód robil niečo iné.

```
from microbit import *
while True:
    if button_a.was_pressed():
        display.scroll("A")
    else:
        display.scroll(Image.ASLEEP)
    sleep(1000)
```

Ak potrebujete zistiť koľko krát bolo tlačidlo stlačené počas určitého času, kedy bola doska uvedená do režimu spánku, napriek tomu však evidovala udalosti zatlačenia tlačidiel, použijete funkciu `get_presses()`.

Tento príklad vypíše počet zatlačení tlačidla A počas intervalu 3s

```
from microbit import *
while True:
    sleep(3000)
    count = button_a.get_presses()
    display.scroll(str(count))
```

Micro:bit Python – digitálne vstupy a výstupy

Vstupno-výstupné piny zbernice mikropočítačovej dosky Micro:bit, z ktorých mnohé majú aj alternatívne funkcie sa adresujú ako `microbit.pin0` - `microbit.pin20`.

Piny 0,1 a 2 majú na doske zapojené 10 Mega ohmové rezistory na + 3.3V aby fungovali ako dotykový senzor. Funguje to tak, že jednou rukou chytíte pin GND a prstom druhej ruky pin 0,1, alebo 2. Na piny 5 a 11 sú pripojené rezistory 10 K, aby mohli fungovať tlačidlá A a B. Niektoré piny zbernice, konkrétne 3, 4, 6, 7, 9. a 10 sa využívajú aj pre displej. Ak ich chcete využiť na iný účel, je potrebné vypnúť displej príkazom `microbit.display.off()`. V tomto príklade využijeme dotyk kontaktných plôšok

prstom. Či sa človek dotýka, alebo nedotýka pinu sa dá zistiť príkazom **is_touched()**, ktorý vracia hodnoty True, alebo False.

```
from microbit import *
while True:
    if pin0.is_touched():
        display.show(Image.HAPPY)
    else:
        display.show(Image.SAD)
```

Ak je pin nastavený ako digitálny vstupný, úroveň napätia a tým aj logická hodnota sa číta pomocou príkazu **read_digital()**. Príkaz vracia hodnoty 0, alebo 1.

```
from microbit import *
while True:
    if pin0.read_digital():
        display.show(Image.HAPPY)
    else:
        display.show(Image.SAD)
```

Hodnotu na príslušnom pine nastavujete príkazom **write_digital(parameter)**, kde parameter môže nadobudnúť hodnoty 0, alebo 1.

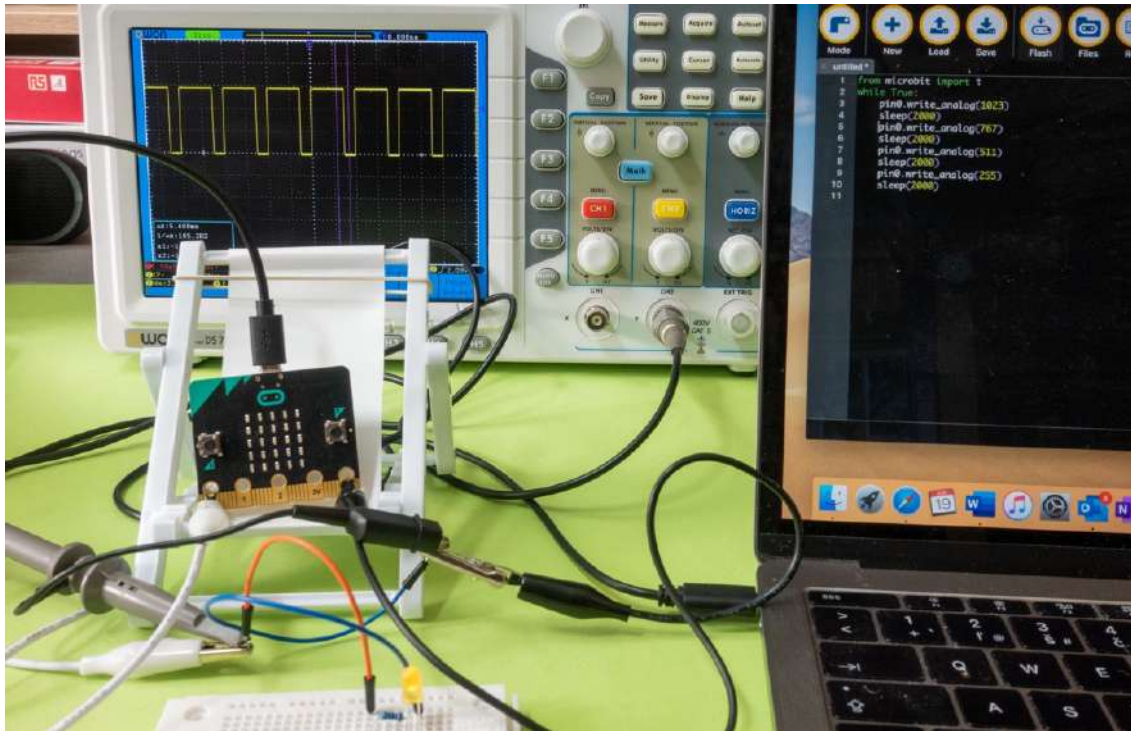
```
from microbit import *
while True:
    pin0.write_digital(1)
    sleep(500)
    pin0.write_digital(0)
    sleep(500)
```

Pokiaľ využívate piny v analógovom móde, tak na čítanie sa použije príkaz **read_analog()**, pričom napäťovému rozsahu na vstupe 0 – 3.3 V budú zodpovedať hodnoty od 0 do 1023.

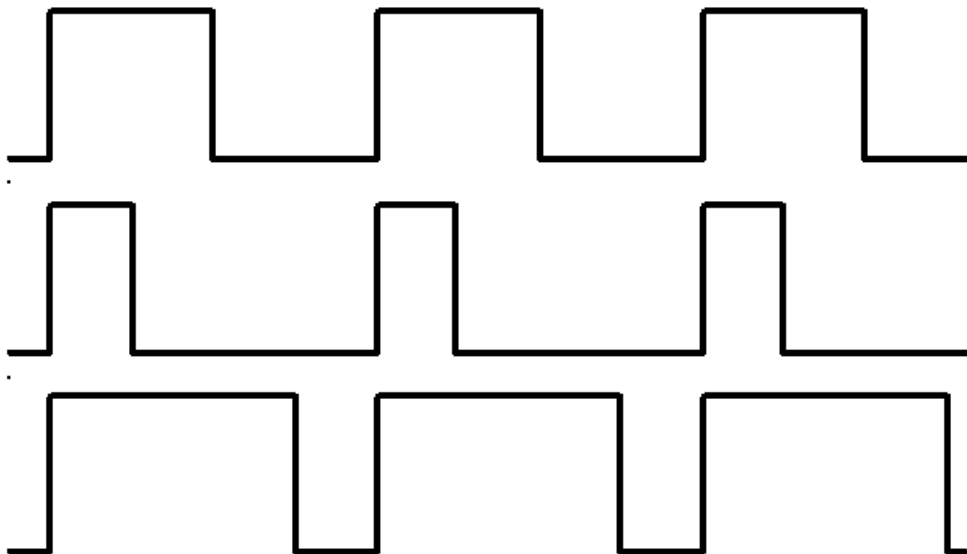
Micro:bit Python – generovanie PWM signálu na emulovanie analógového výstupu

Na pinoch dosky Micro:bit sice môžete nastaviť aj analógovú hodnotu **write_analog(parameter)**, avšak tá sa prejaví ako PWM signál, čiže priebeh signálu z pulzne šírkovou moduláciou. Hodnoty parametra sú od 0, čo je 0 % podiel signálu v úrovni logická 1 až po 1023, čo je 100 % podiel signálu v úrovni logická 1

```
from microbit import *
while True:
    pin0.write_analog(1023)
    sleep(2000)
    pin0.write_analog(767)
    sleep(2000)
    pin0.write_analog(511)
    sleep(2000)
    pin0.write_analog(255)
    sleep(2000)
```



Na obrázku sú tri priebehy signálu so šírkovou moduláciou:



- Prvý by bol generovaný `write_analog (511)`, pretože má presne 50% periódu - polovicu času každej periódy je na pine napätie 3.3 a polovicu času úroveň GND. Výsledkom toho je, že celková energia tohto signálu je rovnaká, ako keby bola nastavená na 1,65V namiesto 3,3V.
- Druhý signál má 25% pracovný cyklus a môže byť generovaný pomocou `write_analog (255)`. Má podobný účinok, ako keby bol na tomto pine trvale napätie 0,825V.
- Tretí signál má 75% pracovný cyklus a je generovaný pomocou `write_analog (767)`. Má trikrát toľko energie, ako druhý signál, a je ekvivalentný napätiu 2,475V.
- PWM dobre funguje dobre so zariadeniami, ako sú žiarovky alebo motory, ktoré majú zotrvačnosť, prípadne využívajú zotrvačnosť ľudského oka, typickým príkladom sú LED diódy.

Micro:bit Python – ovládanie motorov a servomotorov pomocou PWM signálu

PWM s presnou šírkou impulzu sa často využíva na ovládanie uhla natočenia servomotorov alebo na riadenie výkonu jednosmerných motorov, napríklad v dronoch. Preto je potrebné mať možnosť presného nastavenia periódy impulzov. Na nastavovanie sa používa príkaz

set_analog_period(pocet_milisekund), alebo

set_analog_period_microseconds(pocet_mikrosekund). U mikrosekund je minimálna hodnota 256µs. Pre servomotory sa využíva frekvencia PWM signálu 50 Hz, čo je 20 milisekund. Takže program, ktorý bude hýbať servom z 0 na 180° a späť bude:

```
from microbit import *
pin0.set_analog_period(20)
while True:
    pin0.write_analog(180)
    sleep(1000)
    pin0.write_analog(1)
    sleep(1000)
```

Micro:bit Python – generovanie zvuku

Ak pripojíte medzi piny 0 a GND malý reproduktor, môžete na ňom generovať tóny. Zdôrazňujem reproduktor a nie bzučiak, pretože bzučiak keď sa naň privedie napätie generuje len jeden tón.

Prípadne môžete priebežne meniť frekvenciu tónu

```
from microbit import *
import music
while True:
    for freq in range(880, 1760, 16):
        music.pitch(freq, 6)
    for freq in range(1760, 880, -16):
        music.pitch(freq, 6)
```

Môžete nechať zahrať niektorú z preddefinovaných melódií, napríklad:

```
from microbit import *
import music
music.play(music.BIRTHDAY)
```

K dispozícii sú melódie

DADADADUM, ENTERTAINER, PRELUDE, ODE, NYAN, RINGTONE, FUNK, BLUES, BIRTHDAY, WEDDING, FUNERAL, PUNCHLINE, PYTHON, BADDY, CHASE, BA_DING, WAWAWAWAA, JUMP_UP, JUMP_DOWN, POWER_UP a POWER_DOWN.

Alebo môžete pomocou tónov definovať vlastnú melódiu. Každá nota je definovaná textovým reťazcom v tvare `NOTA[oktáva][:trvanie]`

```
from microbit import *
import music
tune = ["C4:4", "D4:4", "E4:4", "C4:4", "C4:4", "D4:4", "E4:4", "C4:4",
        "E4:4", "F4:4", "G4:8", "E4:4", "F4:4", "G4:8"]
music.play(tune)
```

alebo zjednodušene

```
from microbit import *
import music
tune = ["C4:4", "D", "E", "C", "C", "D", "E", "C", "E", "F", "G:8",
        "E:4", "F", "G:8"]
music.play(tune)
```

Micro:bit Python – Meranie teploty

Doska micro:bit má integrovaný senzor teploty, takže urobiť z micro:bitu teplomer, ktorý bude ukazovať teplotu na displeji (pomocou vodorovného skrolovania) je zdanlivo hračka. Drobný problém je v tom, že teplotné čidlo je súčasťou mikrokontroléra, takže nemeria teplotu okolia, ale teplotu čipu, ktorá je podľa skúseností o 4 – 5 °C vyššia, než teplota okolia

```
from microbit import *
while True:
    teplota = temperature()
    display.scroll(str(teplota) + 'C')
    sleep(500)
```

Micro:bit Python – čítanie údajov z akcelerometra

Akcelerometer na doske BBC Micro:bit zachytáva pohyb v troch osiach:

- X – naklonenie doľava a doprava,
- Y – naklonenie dopredu a dozadu,
- Z – pohyb nahor a nadol.

Funkcia `accelerometer.get_x()` vracia hladnú, alebo zápornú hodnotu zrýchlenia v danom smere v jednotkách, ktoré by sa dali označiť ako mili-g. Predpona mili v tomto prípade kvôli jednoduchšej implementácii nie je 1/1000, ale 1/1024. Takže gravitačné zrýchlenie v osi Z má hodnotu 1024 mili-g. Na zobrazenie hodnôt zrýchlenia v jednotlivých osiach môžete použiť kód:

```
from microbit import *
while True:
    x = accelerometer.get_x()
    y = accelerometer.get_y()
    z = accelerometer.get_z()
    print("x, y, z:", x, y, z)
    sleep(500)
```

Ak držíte dosku micro:bit pokojne bez chvenia LED diódami nahor hodnoty zrýchlenia X a Y sú okolo nuly a zrýchlenie Z je približne -1024. To vám povie, že gravitácia pôsobí smerom nadol voči doske. Ak dosku otočíte LED diódami nadol mela by sa hodnota zmeniť na +1024 mili-g. Ak budete triasť s doskou dosť silne, uvidíte, že zrýchlenie dosiahne hodnoty až ± 2048 mili-G. To preto, lebo tento akcelerometer je nastavený na meranie maximálne ± 2048 mil-g. Skutočné zrýchlenie môže byť väčšie než namerané.

Príklad na určenie smeru

```
from microbit import *
while True:
    reading = accelerometer.get_x()
    if reading > 20:
        display.show("R")
    elif reading < -20:
        display.show("L")
    else:
        display.show("-")
```

Pretože akcelerometer je pomerne citlivý, je v príklade v podmienke tolerancia. Ak je hodnota zrýchlenia medzi -20 a 20 mili-g, nepovažuje sa to ešte za pohyb, ale toleranciu merania.

Micro:bit Python – identifikácia polohy dosky a giest ruky s doskou

Doska BBC micro:bit dokáže pomocou akcelerometra zachytávať a analyzovať aj komplexnejšie gestá. V kóde sú zadávané ako textové reťazce. Ich zoznam je: up, down, left, right, face up, face down, freefall, 3g, 6g, 8g a shake. Gestá 3g, 6g a 8g sa indikujú vtedy ak zrýchlenie dosky v ktoromkoľvek smere prekročí príslušnú hodnotu násobku gravitačného zrýchlenia. Nasledujúci príklad zobrazí ikonu šťastného smajlíka iba v prípade ak je zariadenie v polohe displejom nahor.

```
from microbit import *
while True:
    gesture = accelerometer.current_gesture()
    if gesture == "face up":
        display.show(Image.HAPPY)
    else:
        display.show(Image.ANGRY)
```

Micro:bit Python – kompas využívajúci magnetický senzor

Senzor magnetického poľa umožňuje zistiť polohu vzhľadom k svetovým stranám. Pred odčítaním musíte elektronický kompas kalibrovať, inak následné merania nebudú presné. Metóda **compass.calibrate()** spustí mini- hru pri ktorej budete podľa inštrukcií naklápať dosku micro:bit dokola, kým sa celý displej nezaplní svietiacimi pixelmi. Metóda **compass.heading()** určuje uhol v rozsahu od 0 do 360 v stupňoch, v smere hodinových ručičiek, pričom 0 je sever. Aby bolo možné zobrazíť smer graficky výpočet v nasledujúcom príklade určí index hodiny tak aby malá ručička zobrazených hodín na maticovom displeji zobrazovala sever.

```
from microbit import *
compass.calibrate()
while True:
    smer = ((15 - compass.heading()) // 30) % 12
    display.show(Image.ALL_CLOCKS[smer])
```

Micro:bit Python – údaje v súboroch

Aj pre malý mikroprocesorový systém je užitočné mať možnosť ukladať údaje do trvalého úložného priestoru, čiže do takého typu pamäti, kde zostanú aj po vypnutí napájania. MicroPython na platforme micro:bit podporuje jednoduchý súborový systém uložený v pamäti typu flash, teda tam, kde je aj programový kód. Na ukladanie je k dispozícii približne 30 kilobajtov. Na pripomenutie, Microbit má 256 kB flash pamäti a 16 kB pamäti RAM. Pri nahrávaní kódu v HEX súbore ste si určite všimli, že micro:bit pripojený cez USB sa voči počítaču tvári ako USB kľúč. Súbory ktoré vytvoríte a budete do nich zapisovať údaje však v tejto pripojiteľnej jednotke nebudú. Súborový systém na doske micro:bit je jednoduchý, takže nie je možné organizovať súbory do zložiek.

V prvom príklade uložíme do súboru nejaké údaje, pre jednoduchosť krátky text. Príkaz **open()** otvorí, alebo ak nie je zatiaľ vytvorený, tak vytvorí súbor súboru a priradí ho k objektu. Všimnite si, že parameter 'w' sa používa na nastavenie objektu subor do režimu zápisu. Na nastavenie režimu čítania sa používa parameter 'r', toto je zároveň implicitné nastavenie.

V príklade funkcia otvorí (vytvorí) súbor s názvom udaje.txt a priradí ho k objektu subor. Následne sa v bloku kódu odsadenom pod príkazom with použije objekt subor na zápis.

```
from microbit import *
with open('udaje.txt', 'w') as subor:
    subor.write("Lorem ipsum")
```

v ďalšom príklade tieto údaje prečítame a zobrazíme. Do objektu text sa priradí obsah načítaný zo súboru. Všimnite si, že riadok obsahujúci príkaz print() nie je odsadený. Blok kódu priradený k príkazu

with je len jeden riadok, ktorý číta súbor. Akonáhle je blok kódu zatvorený, Python automaticky zatvorí súbor.

```
from microbit import *
with open('udaje.txt') as subor:
    text = subor.read()
    display.scroll(text)
```

Podobne ako v klasických operačných systémoch aj Python na doske micro:bit dokáže so súbormi aj manipulovať, prípadne vypísať ich zoznam. Na to slúži modul os. Na zobrazenie zoznamu súborov sa používa funkcia **listdir()**

```
from microbit import *
import os
with open('udaje.txt', 'w') as subor:
    subor.write("Lorem ipsum")
with open('pokus.bin', 'w') as subor:
    subor.write("Lorem ipsum")
subory = os.listdir()
display.scroll(str(subory))
```

príkazom print () vypíšete zoznam súborov do konzoly vývojového prostredia.

```
from microbit import *
import os
with open('udaje.txt', 'w') as subor:
    subor.write("Lorem ipsum")
with open('pokus.bin', 'w') as subor:
    subor.write("Lorem ipsum")
subory = os.listdir()
print(str(subory))
```

na vymazanie súboru je určený príkaz remove()

```
from microbit import *
import os
os.remove('pokus.bin')
```

veľkosť súboru zistíte príkazom.

```
from microbit import *
import os
with open('udaje.txt', 'w') as subor:
    subor.write("Lorem ipsum")
velkost = os.size('udaje.txt')
print(str(velkost))
```

Micro:bit Python – bezdrôtová komunikácia

Integrovaný rádiový vysielač a prijímač na čipe dosky Micro:bit je primárne určený na bluetooth komunikáciu. Firmware dosky však tento modul využíva aj na rádiovú komunikáciu medzi doskami navzájom.

Micro:bit teda podporuje dva druhy bezdrôtovej komunikácie - Bluetooth Low Energy (BLE) pre komunikáciu s mobilnými zariadeniami a rádiový prenos pre komunikáciu s inými zariadeniami Micro:bit. Rádiovú komunikáciu je možné naladiť na rôzne kanály (očíslované 0-83). Všetci zainteresovaní sa naladia na ten istý kanál a každý počuje, čo všetci ostatní vysielaajú cez tento kanál. Môžete nastaviť aj vysielač výkon. Čím väčší výkon, tým bude väčší dosah komunikácie. Nakoľko sa

používa rovnaký čip a rovnaký výkon ako pri komunikácii cez bluetooth, maximálny dosah bude v otvorenom priestore približne 50-70 m.

Pri konfigurácii rádiovkej komunikácie môžete nastaviť dĺžku správy, štandardne je nastavená dĺžka 32 bajtov, maximálna dĺžka môže byť 251 bajtov.

Môžete posilať binárne správy s ľubovoľným obsahom, pri výučbe programovania bude najjednoduchšie posilať textové reťazce. Prijaté správy sa radia so fronty konfigurovateľnej veľkosti odkiaľ sa čítajú. Implicitná veľkosť fronty sú tri správy. Prečítaním sa správa odstráni z fronty. Ak sa fronta zaplní, čiže sa rýchlejšie prijímajú, než ich váš kód dokáže prečítať, nové správy sa ignorujú. Nakoľko príkazmi `radio.RATE_250KBIT`, `radio.RATE_1MBIT`, alebo `radio.RATE_2MBIT` sa dá nastaviť pomerne vysoká komunikačná rýchlosť od 250 kBit/s až po dva megabity za sekundu, môže sa pri zle napísanom kóde fronta prijatých správ zaplniť pomerne rýchlo. Implicitne je nastavená prenosová rýchlosť 1 Mbit/s.

Rádiový modul umožňuje zadať niekoľko parametrov a taktiež adresu a skupinu. Skupina môže byť 0 – 255, pričom implicitne je nastavená 0. Dôležité je, že rádio modem odfiltruje správy, ktoré prijíma, ktoré nezodpovedajú vašej adrese a skupine. V dôsledku toho je dôležité vopred nastaviť adresu a skupinu, ktorú bude vaša aplikácia používať. Micro:bit samozrejme stále prijíma správy o vysielaní pre iné kombinácie adresa / skupina.

Posielanie a príjem správy na vybranom kanáli môžete realizovať pomocou kódu:

```
from microbit import *
import radio

radio.on()
radio.config(channel=19)
radio.config(power=7)

my_message = "Voz ker minie ker voz nie krava o ker nos trie"

while True:
    radio.send(my_message)
    incoming = radio.receive()
    if incoming is not None:
        display.show(incoming)
        print(incoming)
    sleep(500)
```

Ak nenastavíte kanál, môžete na implicitne nastavenom 7 kanáli využiť hromadnú komunikáciu. Nasledujúci kód vyšle správu. Všetky dosky, ktoré ju prijmu zobrazia srdiečko. Každá doska však navyše s pravdepodobnosťou 1:3 pošle rovnakú správu, ktorej prijatie ostatné dosky potvrdia a niektoré z nich, teoreticky každá tretia správu znovu pošlú.

```
from microbit import *
import radio
import random

radio.on()

while True:
    if button_a.was_pressed():
        # vysielanie.
        radio.send('x')
        # príjem.
        incoming = radio.receive()
        if incoming == 'x':
            display.show(Image.HEART)
```

```

sleep(500)
display.clear()
# obcas odpovedz
if random.randint(0, 3) == 0:
    sleep(500)
    radio.send('x')

```

Ak to zosumarizujeme trochu odbornejšie, na vývojárskom počítači, alebo webe kde programujete online v MicroPythone sa vytvorí súbor s príponou HEX, ktorý obsahuje engine jazyka MicroPython a k nemu pribalený váš kód. Engine kód predkompiluje a spustí. Ak sa ponoríme ešte hlbšie ho hardvérovej architektúry, tak nad hardvérom je vrstva mini operačného systému mbed OS určeného (aj) pre systémy s mikrokontrolérmi Cortex M0. Na vyššej vrstve sú ovládače hardvérových komponentov a na najvyššej vrstve runtime či už TypeScriptu, alebo Micro Pythonu

A na záver ešte niekoľko trikov. Programovací jazyk Python je modulárny a objektovo orientovaný. Určite ste si všimli, že všetky príklady začínajú importom modulov pre hardvérovú implementáciu MicroPythonu na doske BBC:micro:bit

```
from microbit import *
```

Otvorte v editore Mu konzolu REPL a zadajte

```
>>> import microbit
>>> dir (microbit)
```

Zobrazí sa zoznam importovaných modulov

```
['_name_', 'Image', 'display', 'button_a', 'button_b', 'accelerometer',
'compass', 'i2c', 'uart', 'spi', 'reset', 'sleep', 'running_time', 'panic',
'temperature', 'pin0', 'pin1', 'pin2', 'pin3', 'pin4', 'pin5', 'pin6', 'pin7',
'pin8', 'pin9', 'pin10', 'pin11', 'pin12', 'pin13', 'pin14', 'pin15', 'pin16',
'pin19', 'pin20']
```

Môžete taktiež jednoducho zistiť, do akej triedy patrí príslušná funkcia, napríklad

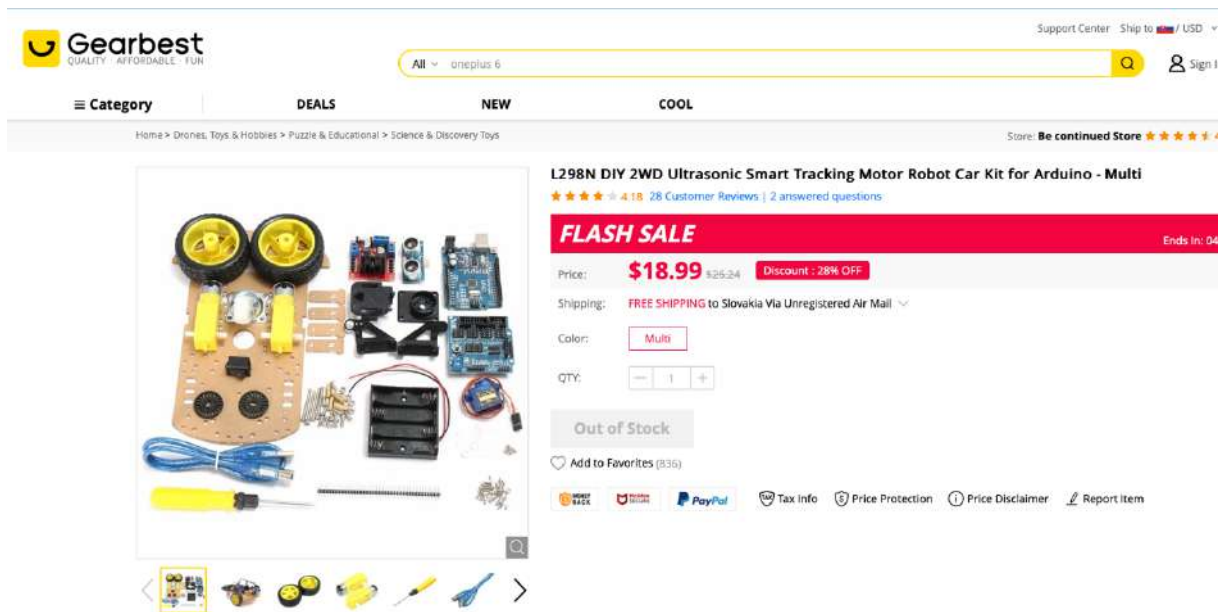
```
>>> type(display)
<class 'MicroBitDisplay'>
>>> type(pin0)
<class 'MicroBitTouchPin'>
>>> type(pin9)
<class 'MicroBitDigitalPin'>
```

A ešte jedna vec: Písať IoT aplikácie na vývojárskom počítači s operačným systémom Windows, či MacOS či už cez webové rozhranie <https://python.microbit.org/v/1.1>, alebo vývojové prostredie Mu <https://codewith.mu> je také trochu menej cool. Čo tak skúsiť pripojiť Micro:bit k doske Raspberry Pi s nainštalovaným operačným systémom Raspbian, pripojiť k Raspberry Micro:bit a skúšať písať Python kód v editore Mu na Raspberry Potom stiahnite hex súbor do Raspberry a prekopírujte ho do pripojenej dosky Micro:bit.

Roboticky a robotické podvozky s Micro:bitom

Video <https://www.youtube.com/watch?v=8I0I6Sqk3qU>

Dosiaľ preberané témy typu spínanie zariadení a ovládanie jednosmerných a krokových motorov a servomotorov bola vlastne prípravou na tému robotika, ktorá bude zameraná na robotické podvozky. Na e-shopoch nájdete stavebnice robotických podvozkov, či už samostatných s motorčekom, alebo stavebnice podvozkov s Arduino. Najlacnejší a pritom plnohodnotný podvozok získate v e-shopoch už za 20 eur a to vrátane dosky Arduino, dosky na riadenie motorov, prepájacej dosky, servomotora a ultrazvukového senzora vzdialenosti. Tento podvozok sa vám bude hodiť aj na stavbu robotického vozidla ovládaného doskou BBC Micro:bit



Gearbest
QUALITY. AFFORDABLE. FUN

Support Center Ship to / USD

All

Category DEALS NEW COOL

Home > Drones, Toys & Hobbies > Puzzle & Educational > Science & Discovery Toys

Score: **Be continued Store** ★★★★★

L298N DIY 2WD Ultrasonic Smart Tracking Motor Robot Car Kit for Arduino - Multi
★★★★★ 4.18 28 Customer Reviews | 2 answered questions

FLASH SALE Ends In: 04

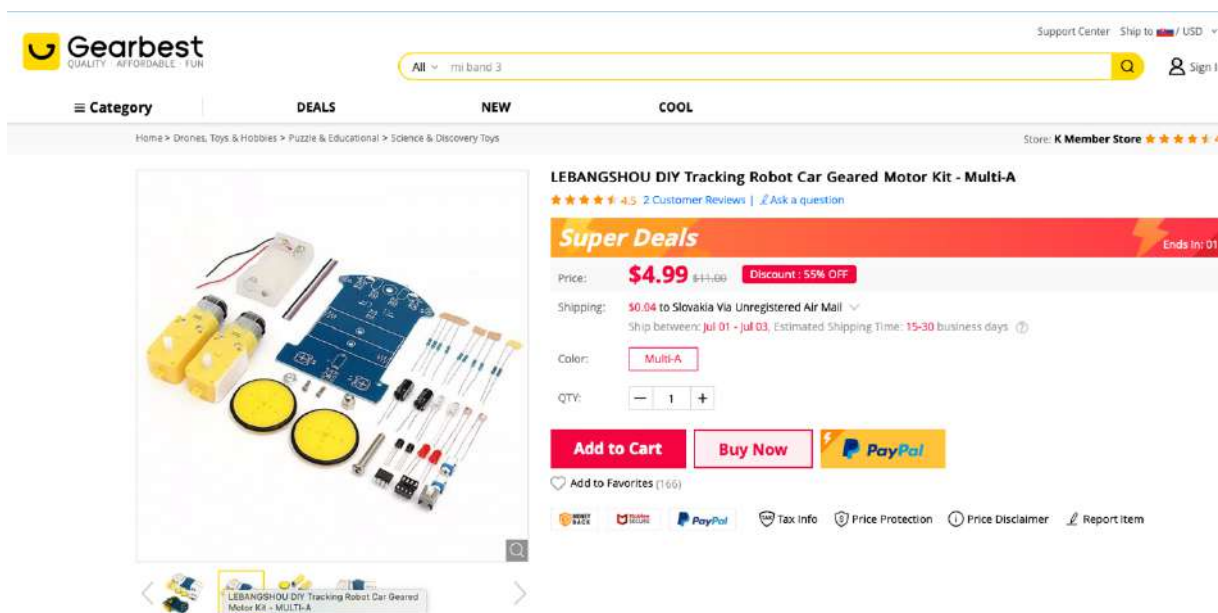
Price: **\$18.99** ~~\$26.24~~ Discount: 28% OFF

Shipping: **FREE SHIPPING to Slovakia Via Unregistered Air Mail**

Color:

QTY:

Alebo samotný kompletný podvozok s motorčekom za približne 7 eur.



Gearbest
QUALITY. AFFORDABLE. FUN

Support Center Ship to / USD

All

Category DEALS NEW COOL

Home > Drones, Toys & Hobbies > Puzzle & Educational > Science & Discovery Toys

Score: **K Member Store** ★★★★★

LEBANGSHOU DIY Tracking Robot Car Geared Motor Kit - Multi-A
★★★★★ 4.5 2 Customer Reviews | [Ask a question](#)

Super Deals Ends In: 01

Price: **\$4.99** ~~\$11.00~~ Discount: 55% OFF

Shipping: **\$0.04 to Slovakia Via Unregistered Air Mail**
Ship between: Jul 01 - Jul 03, Estimated Shipping Time: 15-30 business days

Color:

QTY:

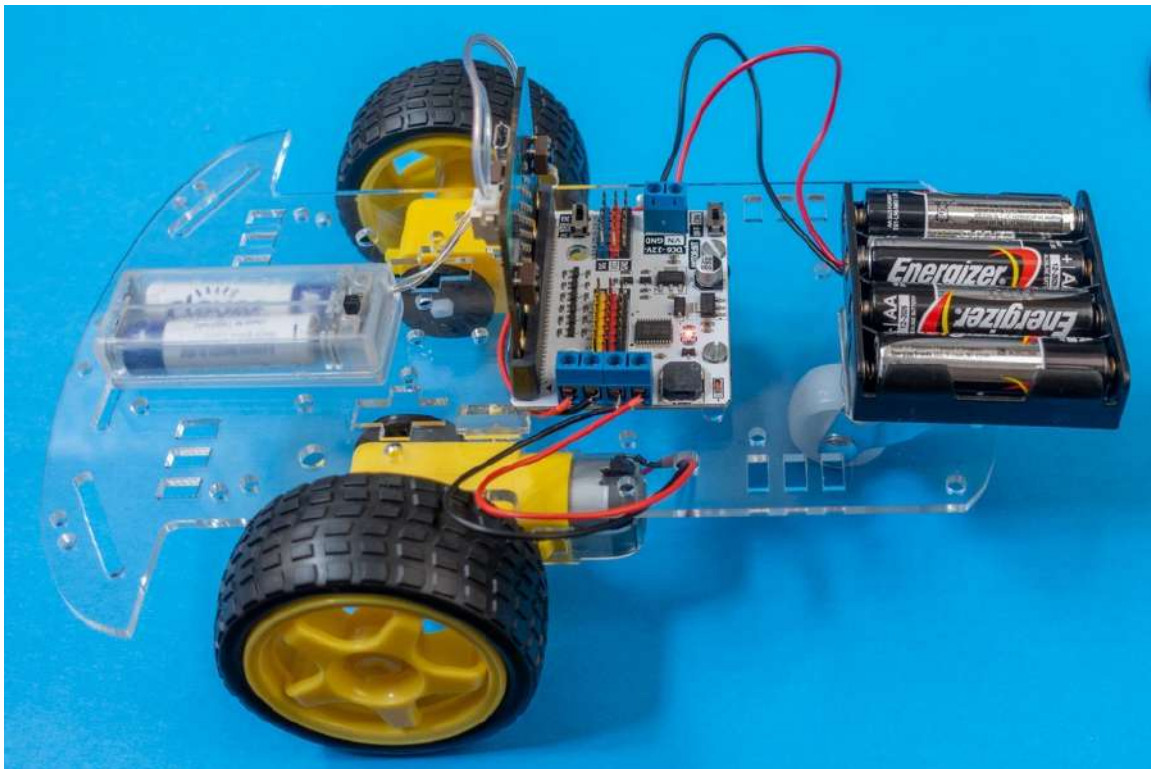
K nemu ešte potrebujete dosku micro:bit (14 eur) a dosku pre micro:bit na riadenie motorčekov (10 eur)

Konštrukcia a programovanie robotov – ovládanie jednosmerných motorov

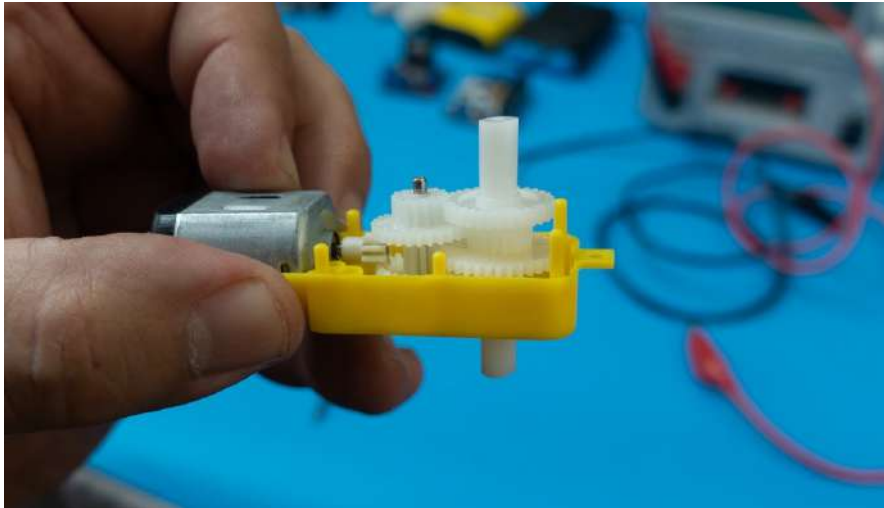
Video <https://www.youtube.com/watch?v=74fAhcdZfJQ>

Doska Micro:bit má v porovnaní s Arduinoom veľa výhod, predovšetkým možnosť vytvorenie rádiového prepojenia medzi dvomi, prípadne viacerými doskami, takže jedna doska môže riadiť hardvér robotického podvozku a pomocou druhej dosky môžete robotický podvozok ovládať. Bud' pomocou dvojice tlačidiel A a B, prípadne naklápaním, či natáčaním dosky Micro:bit, ktorá má zabudovaný akcelerometer a náklonomer.

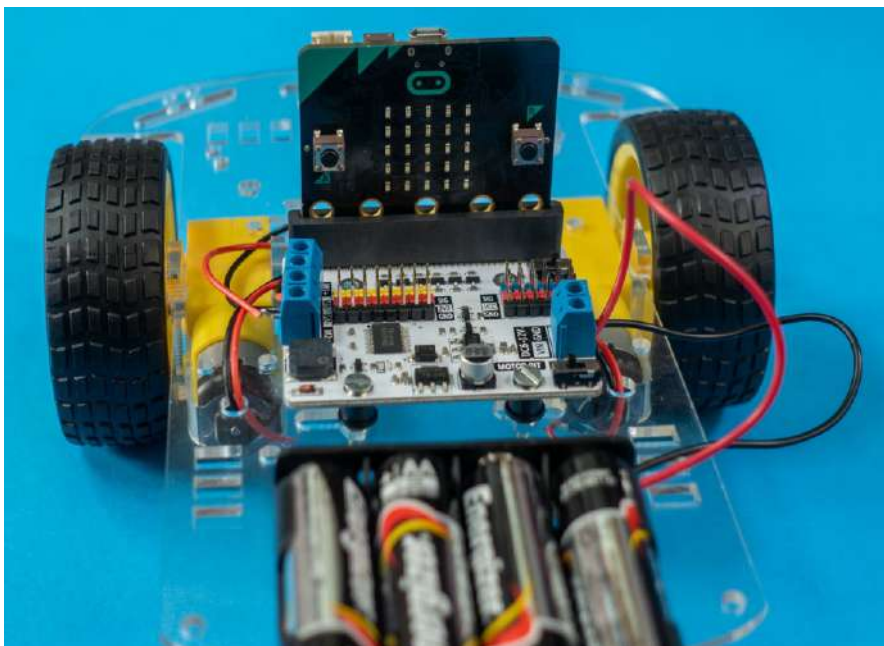
Základným komponentom robotického podvozku sú dva motorčeky, ktoré poháňajú kolesá. Môžu byť jednosmerné, krokové, alebo servomotory. Najjednoduchšie sa, aspoň zdanlivo ovládajú jednosmerné motorčeky. Otáčky sa menia zmenou napájacieho napätia a smer otáčania sa zmení tak, že medzi sebou vymeníte kladný a záporný pól napájania.



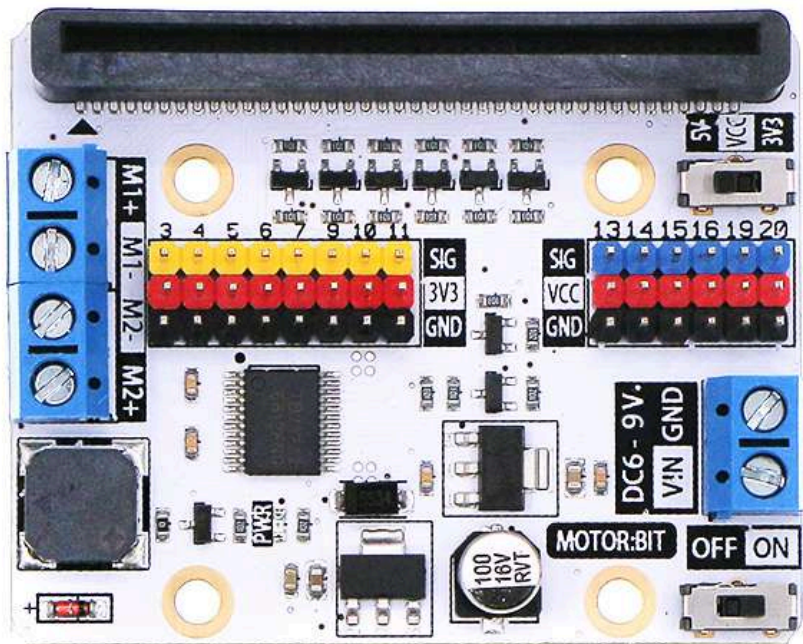
Väčšina hobby konštrukcií malých robotických vozidiel používa elektromotorček s planétovou prevodovkou s prevodovým pomermom 1:48 do pomala. Motorček má osky na oboch stranách, na jednu z nich môžete dať koleso podvozku a na druhú kotúč s otvormi, pomocou ktorého môžete snímať o koľko sa oska pootočila. Motorček je možné napájať napätím v rozsahu 3 - 6V. Má krútiaci moment 0,078Nm a odber prúdu naprázdno je 75 mA



Ovládanie spočíva v napájaní jednosmerným prúdom primeraného napätia na aké ja motorček konštruovaný. Čím je napätie vyššie, tým sa motorček otáča rýchlejšie. Napríklad, ak je moptorček konštruovaný na 6 V, môžete robiť reguláciu otáčok zmenou napájacieho napätie, v rozsahu približne 2 – 6 V. Pri nižšom napätí by sa motorček nerozbehol, vyššie napätie by ho mohlo poškodiť. Motorček odoberá vyšší prúd, než sú schopné dodať porty riadiacej dosky. Preto je potrebná doska na výkonové prispôsobenie. Dosky ako Micro:bit, či Arduino nemajú skutočné analógové výstupy, ale simulujú ich pomocou PWM, čiže pulzne šírkovej modulácie. Toto pri použití knižníc pre danú dosku nepotrebujete vedieť, pre hlbavejších čitateľov je princíp PWM vysvetlený v stati Generovanie PWM signálu.

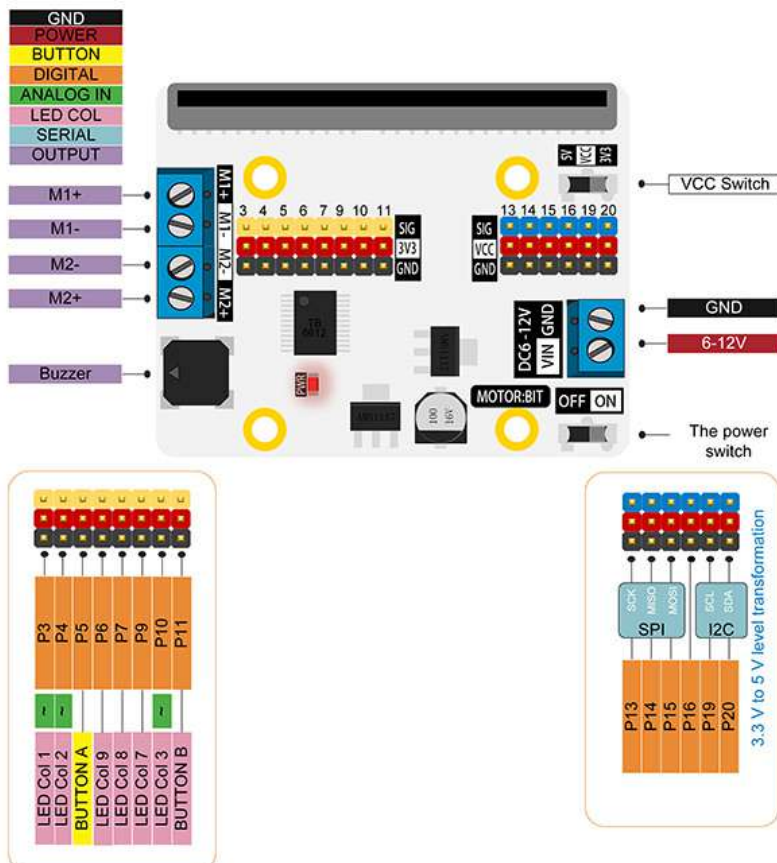


Na ovládanie motorčekov sme použili rozširujúci modul Motor:bit, ktorý dokáže nezávisle riadiť dva jednosmerné motory, alebo jeden krokový motor. Srdcom modulu je integrovaný obvod TB6612 s maximálnym prúdom do motoru 1,2 A. Na module sú vyvedené aj ďalšie porty, ktoré môžete používať na ovládanie servomotorov, prípadne na snímanie údajov zo senzorov.



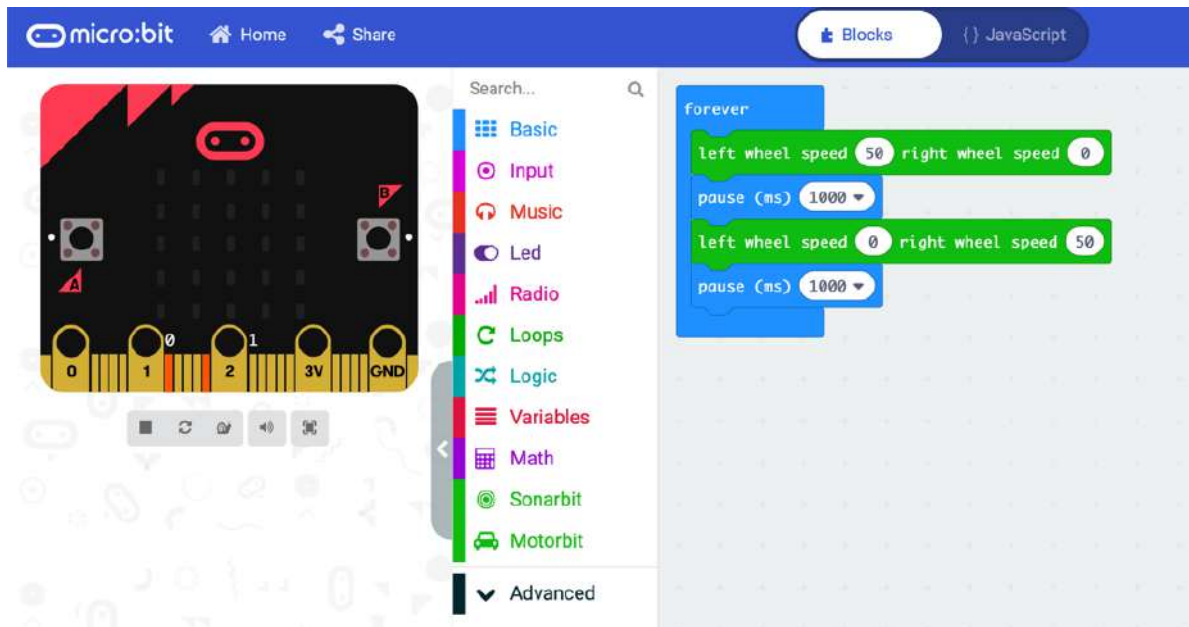
Porty P3 - P7 a P9 - P11 sú určené na pripojenie periférií využívajúcich 3.3 V logiku a porty P13 - P16 a P19- P20 podporujú logiku 3,3 V alebo 5 V. Na výber napäťovej úrovne je na doske prepínač. Na konektory sú vyvedené aj SPI a I2C zbernice. Na doske je aj malý reproduktor, takže robotické vozidlo môže vydávať aj zvuky. Napájacie napätie doky je 6 – 12 V.

ELECFREAKS MOTOR:BIT V1.6



Pomocou analógových, alebo v prípade Micro:bitu pomocou PWM portov P1 a P2 sa ovládajú otáčky motorom M1 a M2. Smer otáčania motora M1 sa mení logickou úrovňou na porte P8 a smer otáčania motora M2 portom P12. Do vývojového prostredia MakeCode potrebujete importovať rozširujúci modul Motorbit.

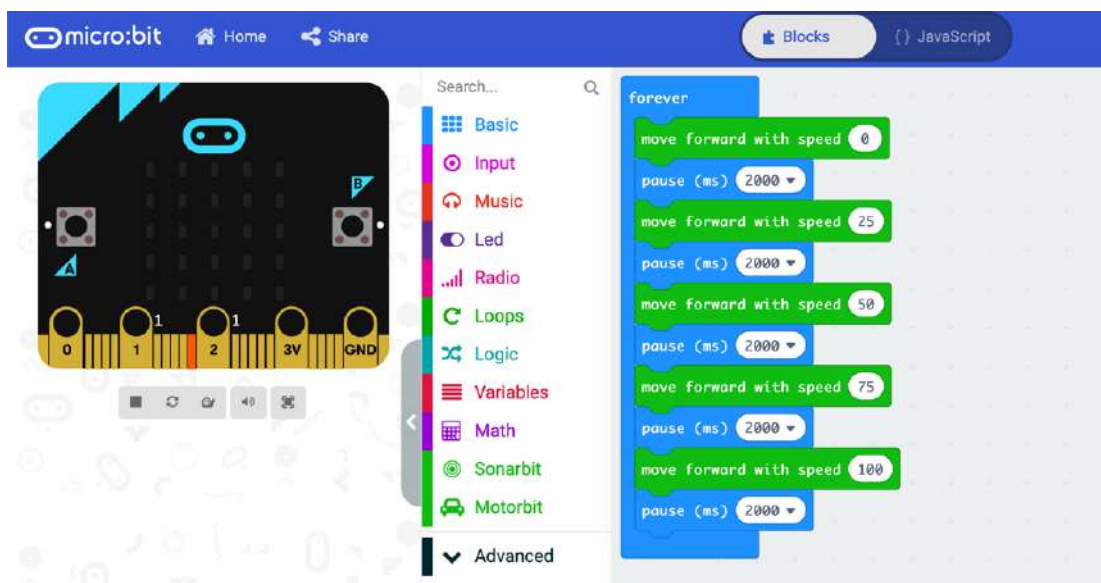
V prvom príklade ukážeme ovládanie motorčekov tak aby sa v sekundových intervaloch otáčali raz na jednu a potom na druhú stranu.



JavaScript

```
basic.forever(function () {
  motorbit.freestyle(50, 0)
  basic.pause(1000)
  motorbit.freestyle(0, 50)
  basic.pause(1000)
})
```

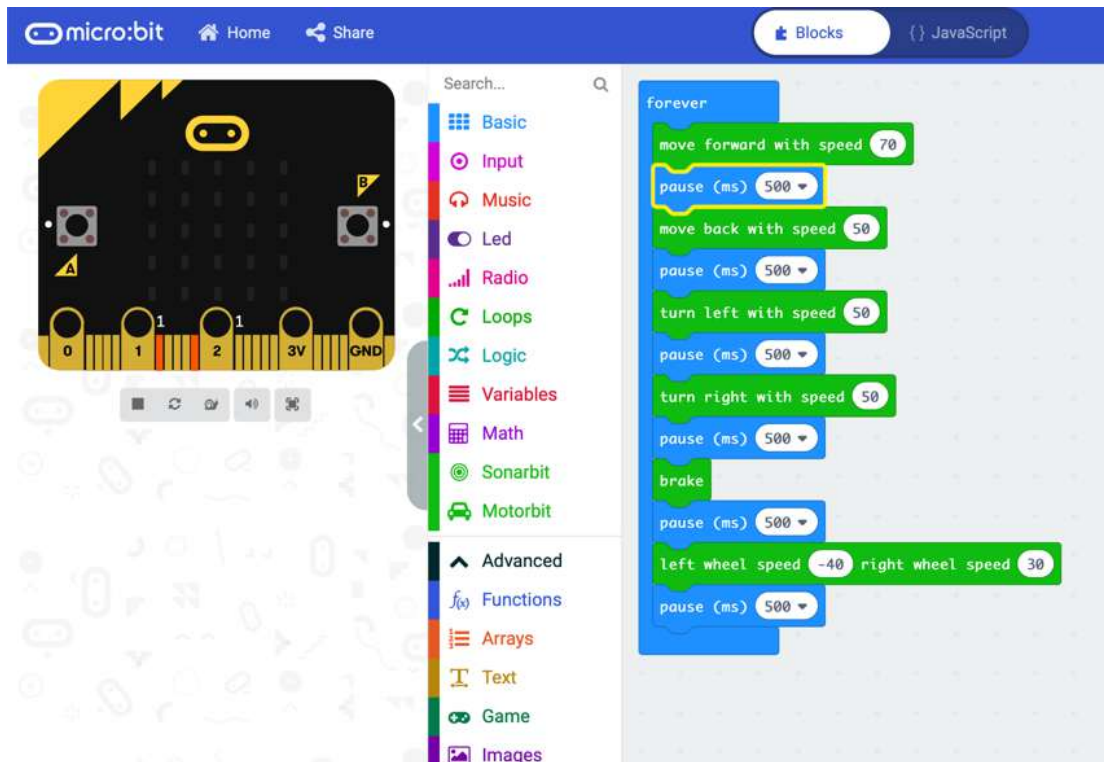
Ďalší príklad je zameraný na zmenu otáčok. Parameter v príkaze forward() udáva výkon v rozmedzí 0 – 100, čiže de facto v percentách.



JavaScript

```
basic.forever(function () {  
  motorbit.forward(0)  
  basic.pause(2000)  
  motorbit.forward(25)  
  basic.pause(2000)  
  motorbit.forward(50)  
  basic.pause(2000)  
  motorbit.forward(75)  
  basic.pause(2000)  
  motorbit.forward(100)  
  basic.pause(2000)  
})
```

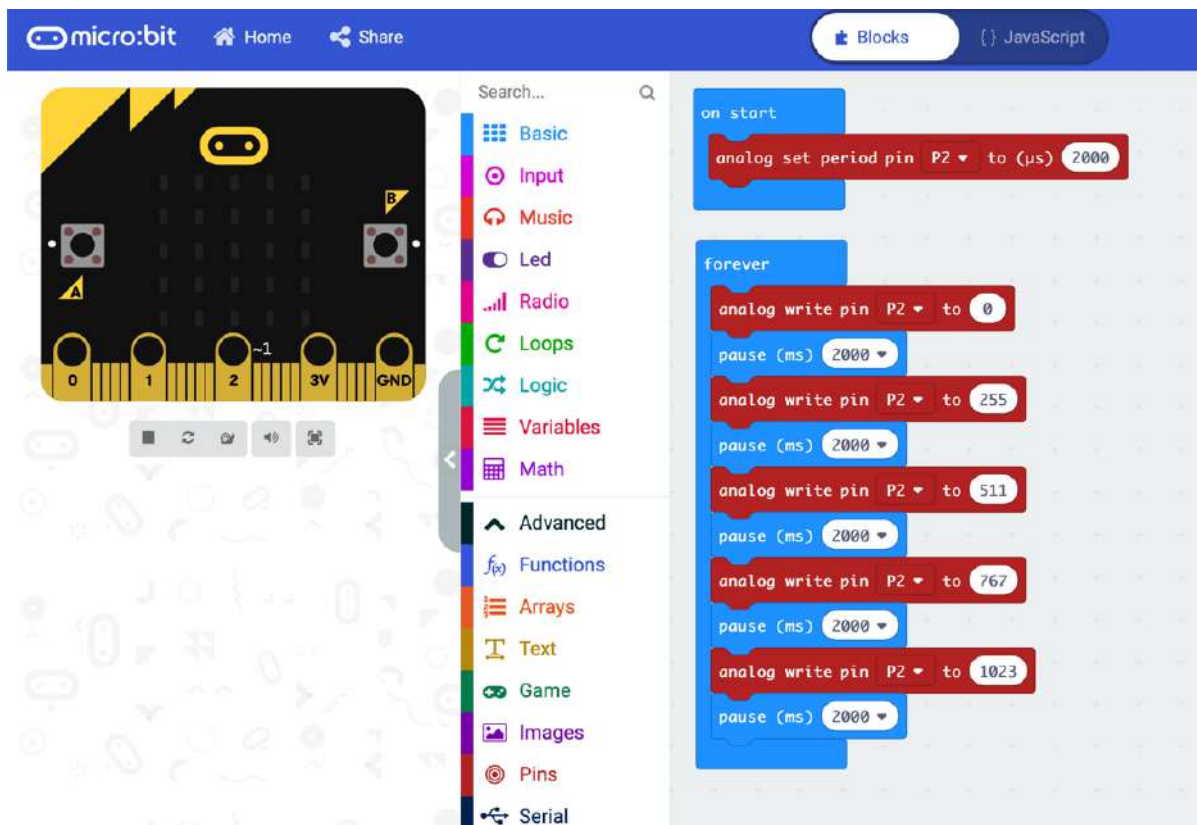
Malá ukážka možností príkazov a funkcií v rozširujúcom module Motorbit.



JavaScript

```
basic.forever(function () {  
  motorbit.forward(70)  
  basic.pause(500)  
  motorbit.back(50)  
  basic.pause(500)  
  motorbit.turnleft(50)  
  basic.pause(500)  
  motorbit.turnright(50)  
  basic.pause(500)  
  motorbit.brake()  
  basic.pause(500)  
  motorbit.freestyle(-40, 30)  
  basic.pause(500)  
})
```

Môžete využiť aj priame generovanie analógového PWM výstupu. V tomto prípade sa pomer striedy a teda aj výstupné napätie nastavuje v rozsahu hodnôt 0 – 1023



JavaScript

```
pins.analogSetPeriod(AnalogPin.P2, 2000)
basic.forever(function () {
  pins.analogWritePin(AnalogPin.P2, 0)
  basic.pause(2000)
  pins.analogWritePin(AnalogPin.P2, 255)
  basic.pause(2000)
  pins.analogWritePin(AnalogPin.P2, 511)
  basic.pause(2000)
  pins.analogWritePin(AnalogPin.P2, 767)
  basic.pause(2000)
  pins.analogWritePin(AnalogPin.P2, 1023)
  basic.pause(2000)
})
```

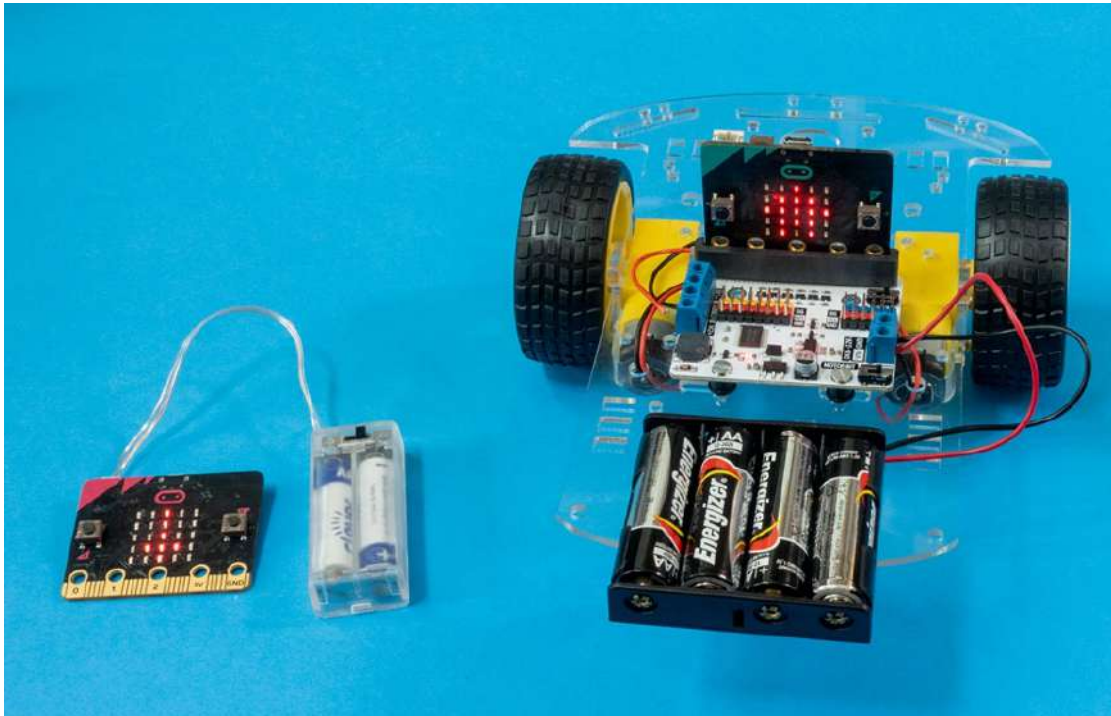
- Hodnota 255 definuje 25% pracovný cyklus a Má podobný účinok, ako keby bolo na tomto pine trvale napätie 0,825V.
- Hodnota 511 definuje 50% pracovný cyklus a Má podobný účinok, ako keby bolo na tomto pine trvale napätie 1,65V.
- Hodnota 767 definuje 50% pracovný cyklus a Má podobný účinok, ako keby bolo na tomto pine trvale napätie 2.475V.
- Hodnota 1023 privedie na výstup napätie 3.3 V

Diaľkové ovládanie pomocou Micro:bitu

Video <https://www.youtube.com/watch?v=7FEgJV6SyUs>

Micro:bit umožňuje vytvorenie rádiového prepojenia medzi dvomi, prípadne viacerými doskami, takže jedna doska môže riadiť hardvér robotického podvozku a pomocou druhej dosky môžete

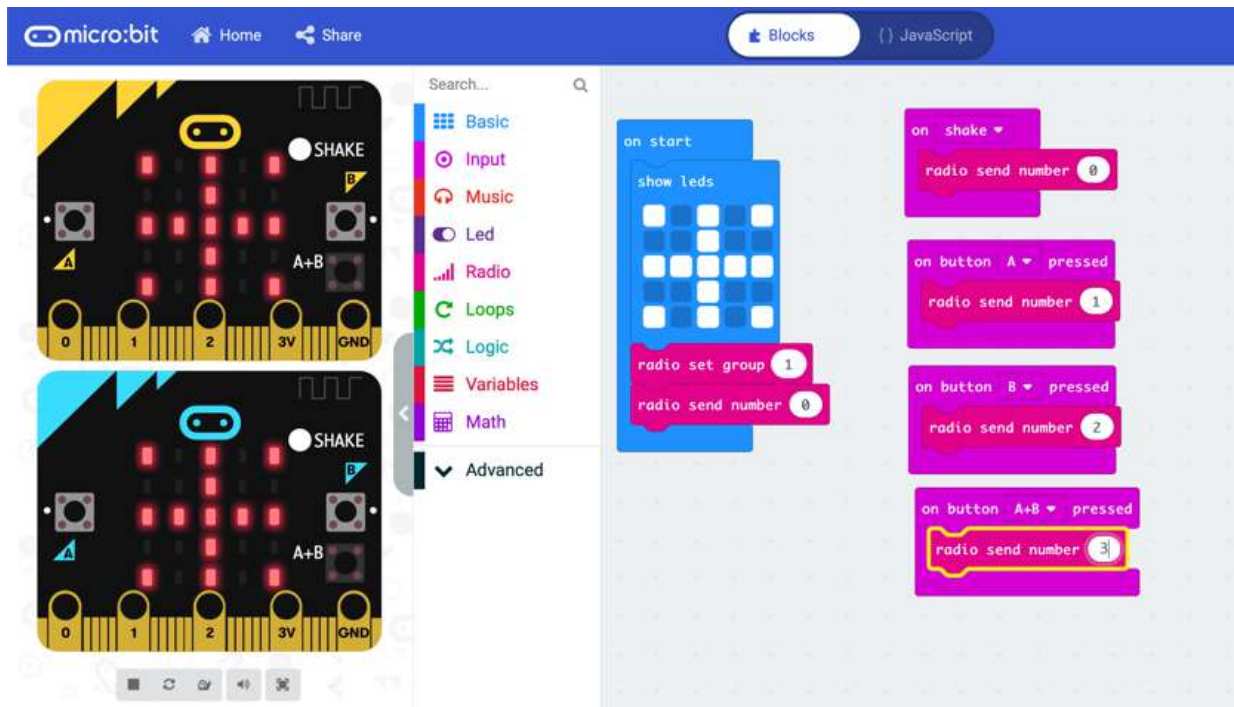
robotické vozidlo ovládať. Buď pomocou dvojice tlačidiel, prípadne potrasením, naklápaním, či natáčaním dosky, ktorá má zabudovaný akcelerometer a náklonomer.



Najskôr vysvetlíme princíp pohybu robotického podvozku s dvoma kolesami, pričom každé poháňa samostatný motor. Ak sa kolieska robotického podvozku otáčajú rovnakou rýchlosťou jedným, alebo druhým smerom, podvozok sa pohybuje priamo dopredu, alebo dozadu. Ak sa kolieska otáčajú rôznou rýchlosťou, robot zatáča do tej strany, na ktorej sa koliesko pohybuje pomalšie. Ak jedno z kolies úplne zastane a druhé sa otáča, tak robot sa začne otáčať na mieste, pričom stredom otáčania je koleso, ktoré sa nepohybuje a polomer otáčania je daný vzdialenosťou medzi kolesami. Ak potrebujete robotický podvozok otočiť na mieste ale na menšom priemere, tak kolieska sa musia otáčať opačným smerom. Vtedy bude polomer otáčania rovný polovičke vzdialenosti medzi kolesami a os otáčania bude v strede pomyslenej spojnice kolies.

Najjednoduchší, aj keď nie najintuitívnejší spôsob ovládania je pomocou tlačidiel A a B. Tlačidlá sú len dva a sú umiestnené tak, že sa pomocou nich bude dobre ovládať pohyb doprava, alebo doľava. Potrebovali by ste ešte ďalšie dva tlačidlá. Jedno na aktivovanie pohybu dopredu, prípadne zrýchľovanie a druhé na pohyb dozadu, prípadne zabrzdzenie. Dá sa to vyriešiť napríklad tak, že súčasným zatlačením tlačidiel A + B sa aktivuje pohyb dopredu, tlačidlami A a B sa ovláda smer pohybu a robota zastavíte potrasením ovládacej dosky. Aktivovaný smer pohybu môžete indikovať šípkou zobrazenou na displeji, v našom kóde je z dôvodu úspory miesta len úvodné zobrazenie smerového kríža.

Kód pre dosku Micro:bit, ktorá bude fungovať ako diaľkový ovládač využívajúci tlačidlá A a B a akcelerometer indikujúci potrasenie. Stavý sú indikované ikonami:



JavaScript

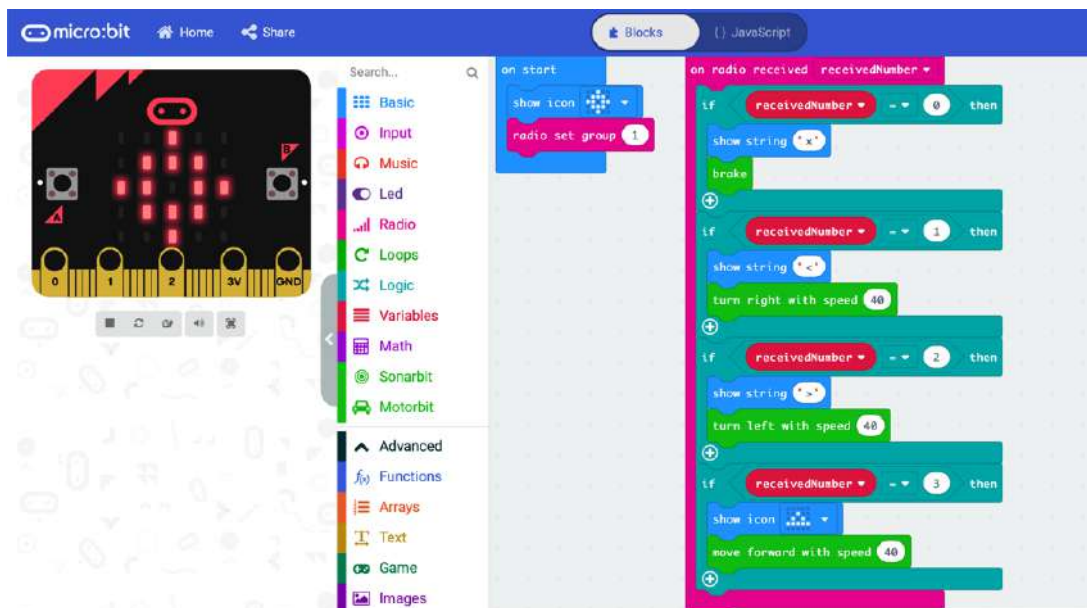
```

basic.showIcon(IconNames.Target)
radio.setGroup(1)
radio.sendNumber(0)

input.onGesture(Gesture.Shake, function () {
  basic.showString("x")
  radio.sendNumber(0)
})
input.onButtonPressed(Button.A, function () {
  basic.showString("<")
  radio.sendNumber(1)
})
input.onButtonPressed(Button.B, function () {
  basic.showString(">")
  radio.sendNumber(2)
})
input.onButtonPressed(Button.AB, function () {
  basic.showIcon(IconNames.Triangle)
  radio.sendNumber(3)
})

```

Kód pre dosku Micro:bit, ktorá bude prijímať povely od ovládača a riadiť motory robotického podvozku pri ovládaní pomocou tlačidiel A a B



JavaScript

```
basic.showIcon(IconNames.Target)
radio.setGroup(1)
```

```
radio.onReceivedNumberDeprecated(function (receivedNumber) {
    if (receivedNumber == 0) {
        basic.showString("x")
        motorbit.brake()
    }

    if (receivedNumber == 1) {
        basic.showString("<")
        motorbit.turnright(40)
    }

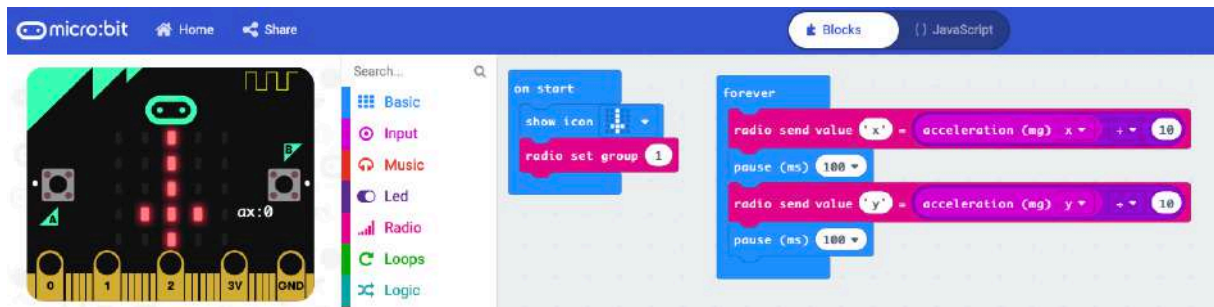
    if (receivedNumber == 2) {
        basic.showString(">")
        motorbit.turnleft(40)
    }

    if (receivedNumber == 3) {
        basic.showIcon(IconNames.Triangle)
        motorbit.forward(40)
    }
})
```

Druhý, pohodlnejší a presnejší spôsob ovládania je, že robot budete ovládať naklápaním dosky v ruke a do riadiacej dosky robota sa budú posilať údaje z akcelerometra v osiach X a Y. V inicializačnom bloku je potrebné nastaviť rádiový kanál, konkrétne číslo skupiny. Toto číslo musí byť rovnaké pre obidve dosky, ktoré majú spolu komunikovať. V príklade je pomocou príkazu setGroup() ako číslo skupiny nastavené 1. Aby ste vedeli, že inicializácia bola úspešná, odporúčame zobrazit' na maticovom displeji vhodnú ikonu.

Ovládanie, čiže načítanie údajov z akcelerometra a ich poslanie do druhej dosky sa odohráva v hlavnej slučke programu. Parameter X, ktorým sa ovláda smer pohybu robota sa nastavuje naklonením dosky doľava alebo doprava. Zrýchlenie, alebo spomalenie sa nastavuje naklonením dosky dopredu, alebo dozadu. V kóde je potrebný prepočet rozsahu, pretože akcelerometer poskytuje údaje z intervalu – 1 023 až 1023, avšak knižničné funkcie na ovládanie motorov pracujú

s parametrami v rozsahu 0 – 100, prípadne od -50 do 50. Cyklus posielania hodnôt sa opakuje každých 100 ms.

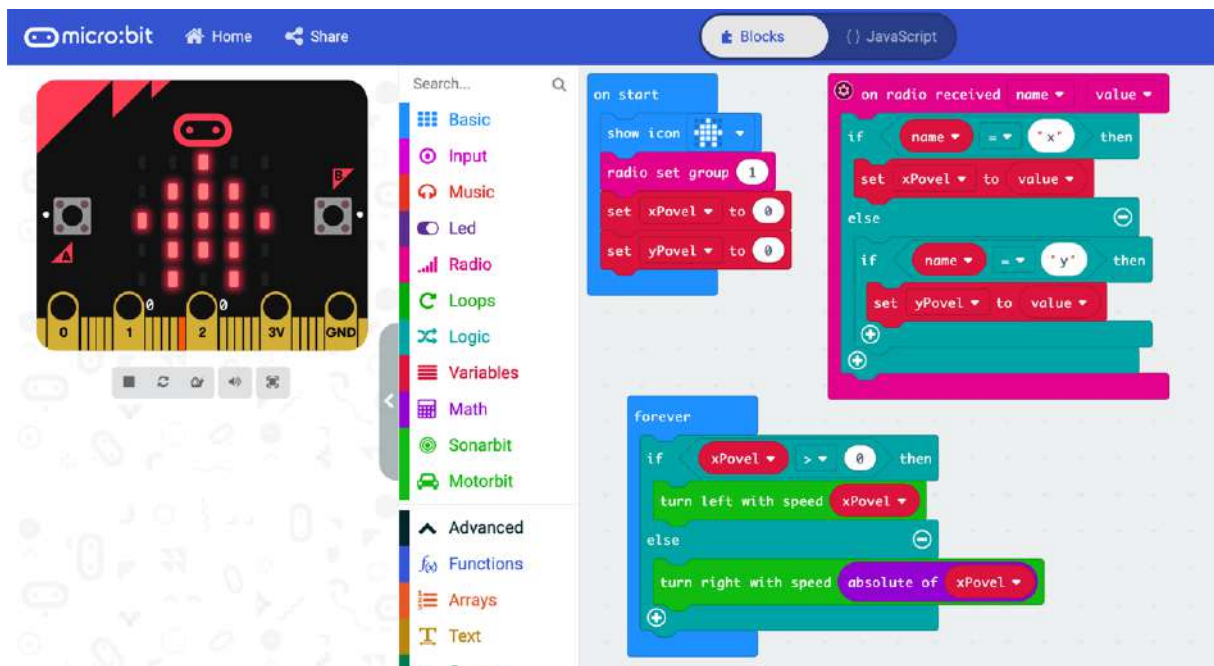


JavaScript

```
basic.showIcon(IconNames.Sword)
radio.setGroup(1)
basic.forever(() => {
  radio.sendValue("x", input.acceleration(Dimension.X) / 10)
  basic.pause(100)
  radio.sendValue("y", input.acceleration(Dimension.Y) / 10)
  basic.pause(100)
})
```

V hlavnej slučke sa spracovávajú povely, ktoré obsahujú informácie o naklonení druhého micro:bitu, Pohyb dopredu a dozadu je odvodený od hodnoty akcelerometra v osi Y. Predpokladajme zatiaľ, že naklonenie v druhej osi X je nulové. V takom prípade, ak je doska ovládača naklonená dopredu potrebujete aby sa robot pohyboval dopredu a naopak. Na ovládanie pohybu robota do strán sa použije hodnota naklonenia ovládača v osi X.

Kód pre dosku Micro:bit, s rozširujúcim modulom motorbit, ktorá bude prijímať povely od ovládača a riadiť motory robotického podvozku. Možno vás prekvapí, že v tomto cvičnom kóde sa využíva len pohyb doľava a doprava, čiže reakcia na naklonenie ovládacej dosky v osi x.



JavaScript

```
let xPovel = 0
let yPovel = 0

radio.onDataPacketReceived(({ receivedString: name, receivedNumber: value }) => {
  if (name == "x") {
    xPovel = value
  } else {
    if (name == "y") {
      yPovel = value
    }
  }
})

basic.showIcon(IconNames.House)
radio.setGroup(1)
xPovel = 0
yPovel = 0

basic.forever(() => {
  if (xPovel > 0) {
    motorbit.turnleft(xPovel)
  } else {
    motorbit.turnright(Math.abs(xPovel))
  }
})
```

Ak by totiž kód obsahoval aj logiku pre pohyb dopredu a dozadu .

```
...
basic.forever(() => {
  if (xPovel > 0) {
    motorbit.turnleft(xPovel)
  } else {
    motorbit.turnright(Math.abs(xPovel))
  }

  if (yPovel > 0) {
    motorbit.forward(yPovel)
  } else {
    motorbit.back(Math.abs(yPovel))
  }
})
```

nefungovalo by to, pretože povely reagujúce na naklonenie v osi Y by zrušili povely na zatáčanie do strán. Preto použijeme funkciu freestyle, ktorá umožňuje zadať parametre pre obidva kolesá

JavaScript

```
let praveKoleso = 0
let laveKoleso = 0
let yPovel = 0
let xPovel = 0
basic.showIcon(IconNames.House)
radio.setGroup(1)
xPovel = 0
yPovel = 0

radio.onDataPacketReceived(function ({ receivedString: name, receivedNumber: value
}) {
  if (name == "x") {
    xPovel = value
  } else {
    if (name == "y") {
      yPovel = value
    }
  }
})
```

```

    }
  })

  basic.forever(function () {
    laveKoleso = yPovel + xPovel
    praveKoleso = yPovel - xPovel
    motorbit.freestyle(praveKoleso, laveKoleso)
  })

```

Podvozok Ring:bit pre Micro:bit

Video <https://www.youtube.com/watch?v=zGyO9VICoSg>

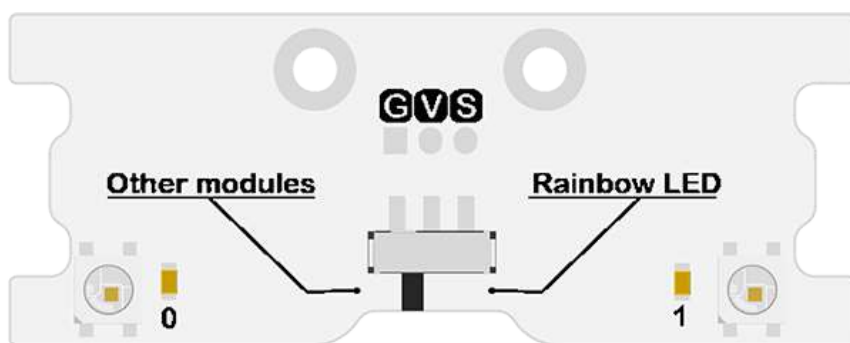
Stavebnica Ring: bit obsahuje mechanické diely, servomotory a dosku elektroniky pomocou ktorých môžete za niekoľko minút poskladať robotický podvozok, ktorý bude riadiť populárny jednodoskový mikropočítač BBC Micro:bit.



Micro:bit sa k doske elektroniky pripevní pomocou piatich skrutiek, ktoré zároveň tvoria elektrické prepojenie pre tri porty P0, P1 a P2, napájacie napätie 3.3 V a elektrickú zem GND. Dva GPIO porty P1 a P2 sú určené na ovládanie servomotorov poháňajúcich kolesá a tretí port P0 sa dá použiť na iný účel, napríklad ultrazvukový senzor a podobne.



Na pomocnú dosku elektroniky na ktorej sú aj dve RGB diódy sa dajú umiestniť prídavné senzory, ktoré nie sú súčasťou štandardného balenia stavebnice.

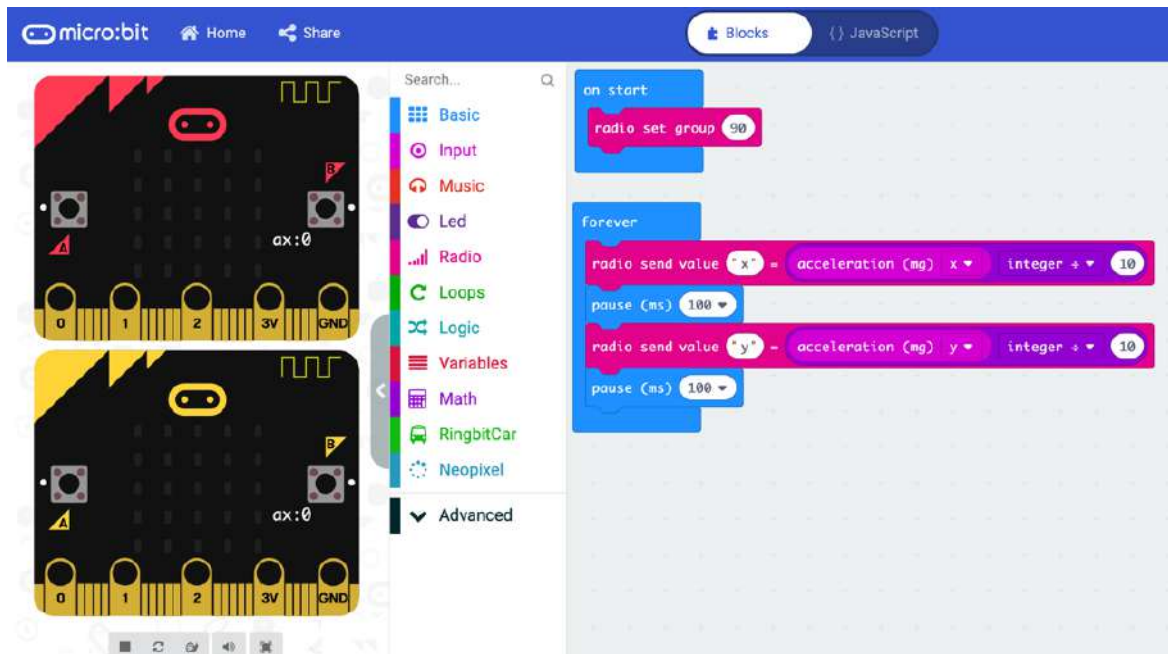


Servomotor, alebo servo je jednosmerný motorček s prevodovkou snímačom uhla natočenia a riadiacou elektronikou. K riadiacej elektronike sa pripája pomocou troch vodičov, pričom dva vodiče sú napájacie a jeden vodič slúži na prenos riadiaceho signálu. Servo sa riadi impulzmi PWM signálu s periódou 20ms a šírkou pulzu v intervale 1 až 2ms.

Servomotory sa buď otáčajú o definovaný uhol, ktorý zodpovedá aktuálnej šírke impulzu, alebo servá označené ako 360° sa otáčajú kontinuálne dopredu, alebo dozadu. Rýchlosť otáčania závisí od šírky impulzu. Pri 1.5 ms motor stojí, v intervale 1 – 1.5 ms sa otáča na jednu stranu a v intervale 1.5 – 2 ms sa ootáča na druhú stranu.

Aby ste nemuseli programovať generovanie PWM signálu na jednotlivých portoch, môžete využiť pripravené funkcie. V prostredí <https://makecode.microbit.org> je potrebné do projektu importovať rozšírenie “ringbitcar”.

V prvom najjednoduchšom príklade bude robotické autíčko ovládané tlačidlami na svojej vlastnej riadiacej doske. Tlačidlom A sa bude aktivovať pohyb dopredu a tlačidlom B dozadu. Súčasným zatlačením oboch tlačidiel vozidlo zastane



JavaScript

```
RingbitCar.init_wheel(AnalogPin.P1, AnalogPin.P2)
```

```
input.onButtonPressed(Button.A, function () {
  RingbitCar.forward()
})
input.onButtonPressed(Button.B, function () {
  RingbitCar.back()
})
input.onButtonPressed(Button.AB, function () {
  RingbitCar.brake()
}).
```

V ďalšom príklade skúsime nechať robotické autíčko kresliť. Má totiž v zadnej časti držiak, do ktorého sa dá upevniť fixka. Pozrite si vo videu aký obrazec nakreslí tento kód. Predpokladali by ste, že mnohouholník. Je to však skôr niečo pripomínajúce ozubený kotúč. Je to dané kombináciou dvoch faktorov. Jednak tým, ako je implementovaná funkcia RingbitCar.turnright(), kedy robot nezatočí plynule, ale na mieste a tým, že držiak na fixku je umiestnený mimo stredu spojnice kolies.



JavaScript

```
RingbitCar.init_wheel(AnalogPin.P1, AnalogPin.P2)
basic.forever(function () {
  RingbitCar.forward()
  basic.pause(400)
  RingbitCar.turnright()
  basic.pause(100)
})
```

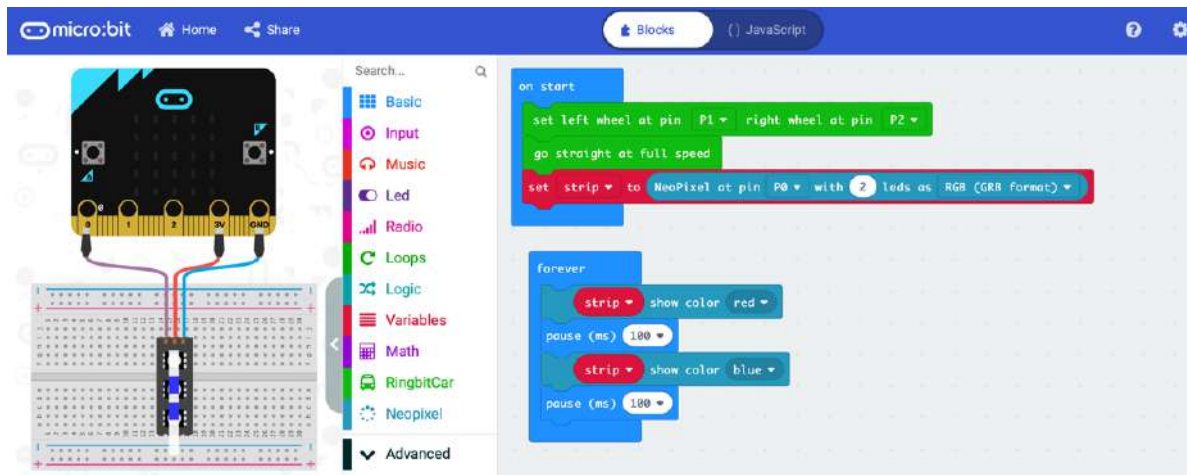
Naproti tomu kružnicu vykreslí takéto robotické autíčko takmer dokonalú. Stačí zariadiť, aby sa kolesá otáčali rovnakým smerom, avšak rôznou rýchlosťou. Čím je rozdiel medzi rýchlosťou otáčania kolies väčší, tým bude polomer vykreslenej kružnice menší

JavaScript

```
RingbitCar.init_wheel(AnalogPin.P1, AnalogPin.P2)

input.onButtonPressed(Button.A, function () {
  RingbitCar.freestyle(100, 50)
})
input.onButtonPressed(Button.B, function () {
  RingbitCar.brake()
})
```

Využiť môžete aj dve RGB LED diódy na pomocnej dosičke, ktoré sú nasmerované tak, aby svietili na podložku. Kombináciou modrého a červeného svetla môžete napodobniť maják. Prípadne môžete rozsvietením týchto LED signalizovať nejakú udalosť.



JavaScript

```
RingbitCar.init_wheel(AnalogPin.P1, AnalogPin.P2)
RingbitCar.forward()
let strip = neopixel.create(DigitalPin.P0, 2, NeoPixelMode.RGB)
basic.forever(function () {
  strip.showColor(neopixel.colors(NeoPixelColors.Red))
  basic.pause(100)
  strip.showColor(neopixel.colors(NeoPixelColors.Blue))
  basic.pause(100)
})
```

Diaľkové ovládanie pomocou robotického podvozku Ring:bit druhým Micro:bitom
Najintuitívnejší spôsob ovládania je nakláňaním dosky v ruke, pričom sa do riadiacej dosky robota budú posilať údaje z akcelerometra v osiach X a Y. V inicializačnom bloku je potrebné nastaviť

rádiový kanál, konkrétne číslo skupiny. Toto číslo musí byť rovnaké pre obidve dosky, ktoré majú spolu komunikovať. V príklade je pomocou príkazu `setGroup()` ako číslo skupiny nastavené 90.

Ovládanie, čiže načítanie údajov z akcelerometra a ich poslanie do druhej dosky sa odohráva v hlavnej slučke programu. Parameter X, ktorým sa ovláda smer pohybu robota sa nastavuje naklonením dosky doľava alebo doprava. Zrýchlenie, alebo spomalenie sa nastavuje naklonením dosky dopredu, alebo dozadu. V kóde je potrebný prepočet rozsahu, pretože akcelerometer poskytuje údaje z intervalu – 1 023 až 1023, avšak knižničné funkcie na ovládanie motorov pracujú s parametrami v rozsahu 0 – 100. Preto je potrebné hodnotu z akcelerometra vydeliť desiatimi. Cyklus posielania hodnôt sa opakuje každých 100 ms.

JavaScript

```
radio.setGroup(90)
basic.forever(function () {
  radio.sendValue("x", Math.idiv(input.acceleration(Dimension.X), 10))
  basic.pause(100)
  radio.sendValue("y", Math.idiv(input.acceleration(Dimension.Y), 10))
  basic.pause(100)
})
```

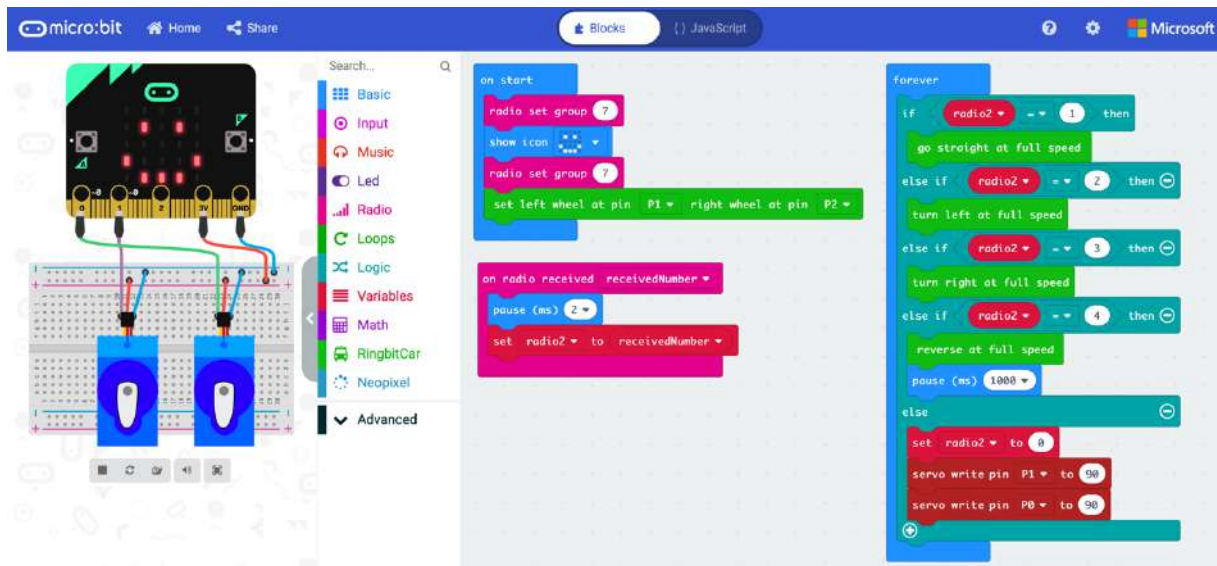
V hlavnej slučke sa spracovávajú povely, ktoré obsahujú informácie o naklonení druhého micro:bitu.

Kód pre dosku Micro:bit, s rozširujúcim modulom motorbit, ktorá bude prijímať povely od ovládača a riadiť motory robotického podvozku. Budeme samostatne nastavovať smer a rýchlosť otáčania pre každý motor. V hlavnej slučke sa na základe prijatých povelov, ktoré obsahujú informácie o naklonení druhého micro:bitu, slúžiaceho ako ovládač vypočítava rýchlosť a smer otáčania obidvoch motorčekov. Pohyb dopredu a dozadu je odvodený od hodnoty akcelerometra v osi Y. Predpokladajme zatiaľ, že naklonenie v druhej osi X je nulové. V takom prípade, ak je doska ovládača naklonená dopredu potrebujete aby sa robot pohyboval dopredu a naopak. Preto ak je hodnota Y kladná budú sa kolesá otáčať takým smerom, aby sa robotický podvozok pohyboval dopredu a pokiaľ je hodnota Y záporná, tak dozadu. V obidvoch prípadoch sa budú kolesá otáčať rovnakou rýchlosťou. Na ovládanie pohybu robota do strán sa použije hodnota naklonenia ovládača v osi X. Vtedy sa budú kolesá otáčať každé iným smerom. Pre ľavé koleso sa použije hodnota nameraná v osi X priamo a pre pravé hodnota s prevráteným znamienkom.

Z toho vyplývajú vzorce pre otáčky kolies:

```
laveKoleso = yPovel + xPovel
praveKoleso = yPovel - xPovel
```

Na základe znamienok hodnôt nameraných akcelerometrom ovládača v jednotlivých osiach sa nastaví smer otáčania motorčekov poháňajúcich kolesá.



JavaScript

```
radio.setGroup(90)
```

```
radio.onDataPacketReceived(function ({ receivedString: name, receivedNumber: value
}) {
  if (name == "x") {
    xValue = value
  } else {
    if (name == "y") {
      yValue = value
    }
  }
})
```

```
let rightwheel = 0
let leftwheel = 0
let yValue = 0
let xValue = 0
RingbitCar.init_wheel(AnalogPin.P2, AnalogPin.P1)
basic.showIcon(IconNames.Triangle)
xValue = 0
yValue = 0
basic.forever(function () {
  leftwheel = yValue + xValue
  rightwheel = yValue - xValue
  RingbitCar.freestyle(leftwheel, rightwheel)
})
```

Robotický podvozok Ring:bit jazdiaci za zdrojom svetla

A ešte jedno zaujímavé využitie dosky Micro:bit. LED diódy v maticovom displeji sa dajú využiť aj na meranie intenzity okolitého osvetlenia. Hľbavejší čitatelia nájdu viac podrobností o tom ako to funguje na <https://lancaster-university.github.io/microbit-docs/extras/light-sensing/>

V najjednoduchšej aplikácii sa bude robotický podvozok pohybovať za svetlom. Po spustení programového kódu sa bude otáčať na mieste a ak nameria intenzitu svetla vyššiu než referenčná hodnota v podmienke, začne sa pohybovať. Takže sa v konečnom dôsledku začne. Pohybovať tým smerom z ktorého zachytil svetlo



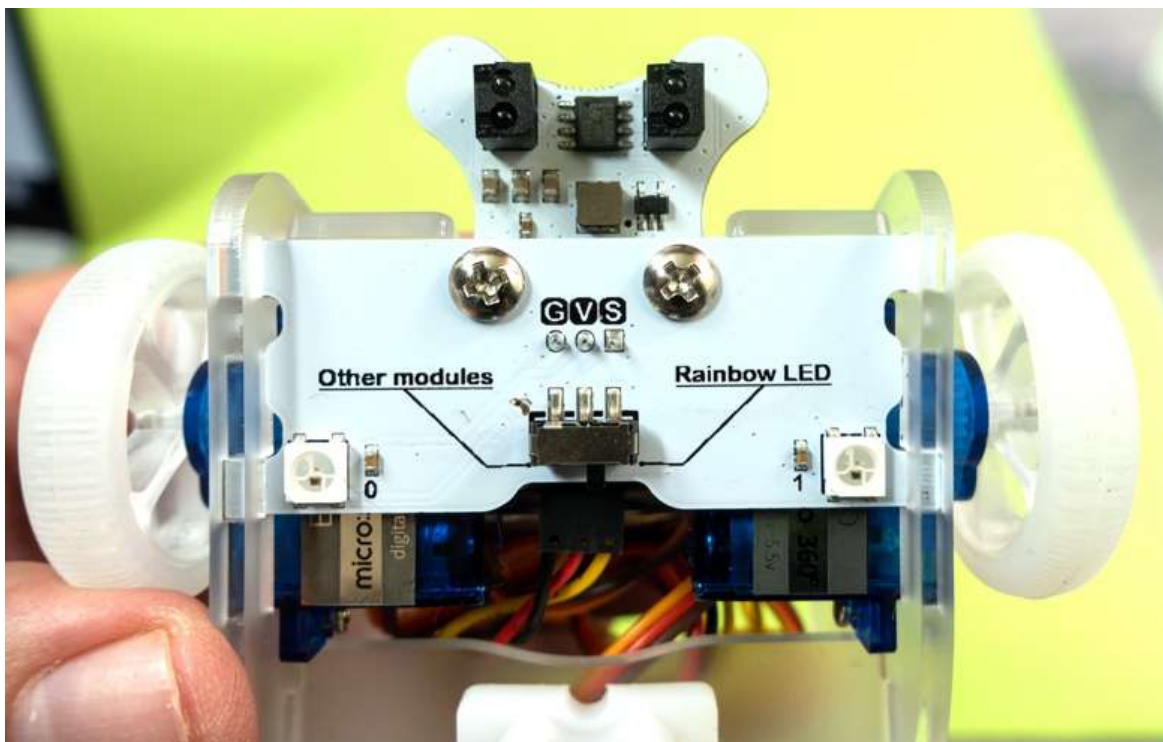
JavaScript

```
RingbitCar.init_wheel(AnalogPin.P1, AnalogPin.P2)
basic.forever(function () {
  if (input.lightLevel() > 40) {
    RingbitCar.forward()
  } else {
    RingbitCar.turnleft()
  }
})
```

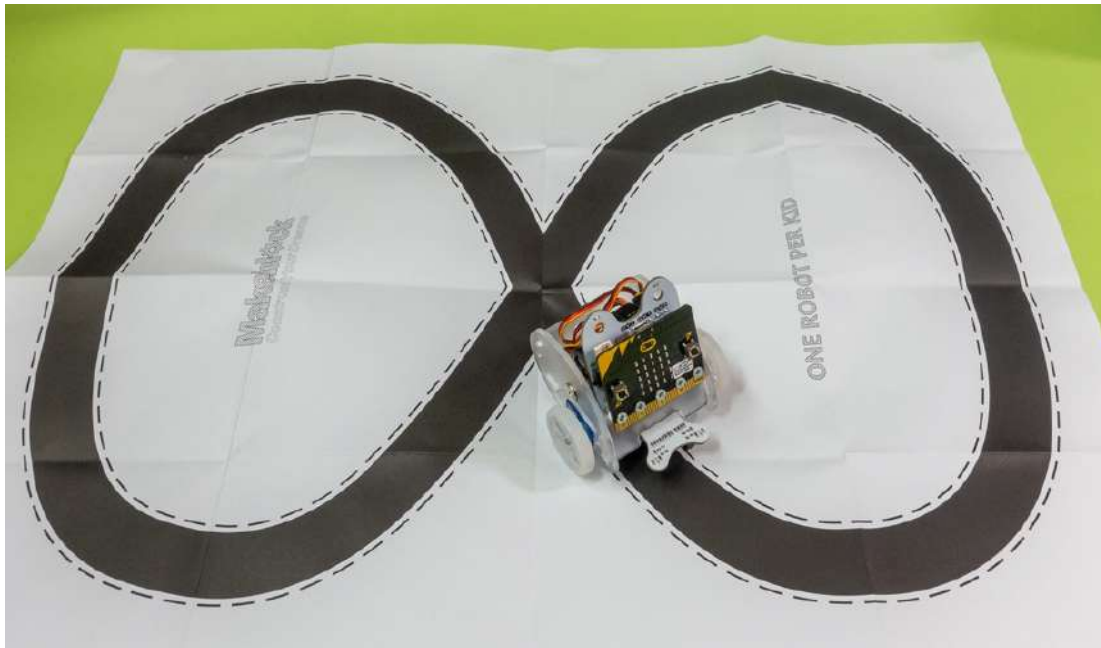
Tento kód sa dá jednoducho vylepšiť, aby sa robot pohyboval za svetlom kontinuálne

Doplnok k podvozku Ring:bit pre Micro:bit na sledovanie čiary

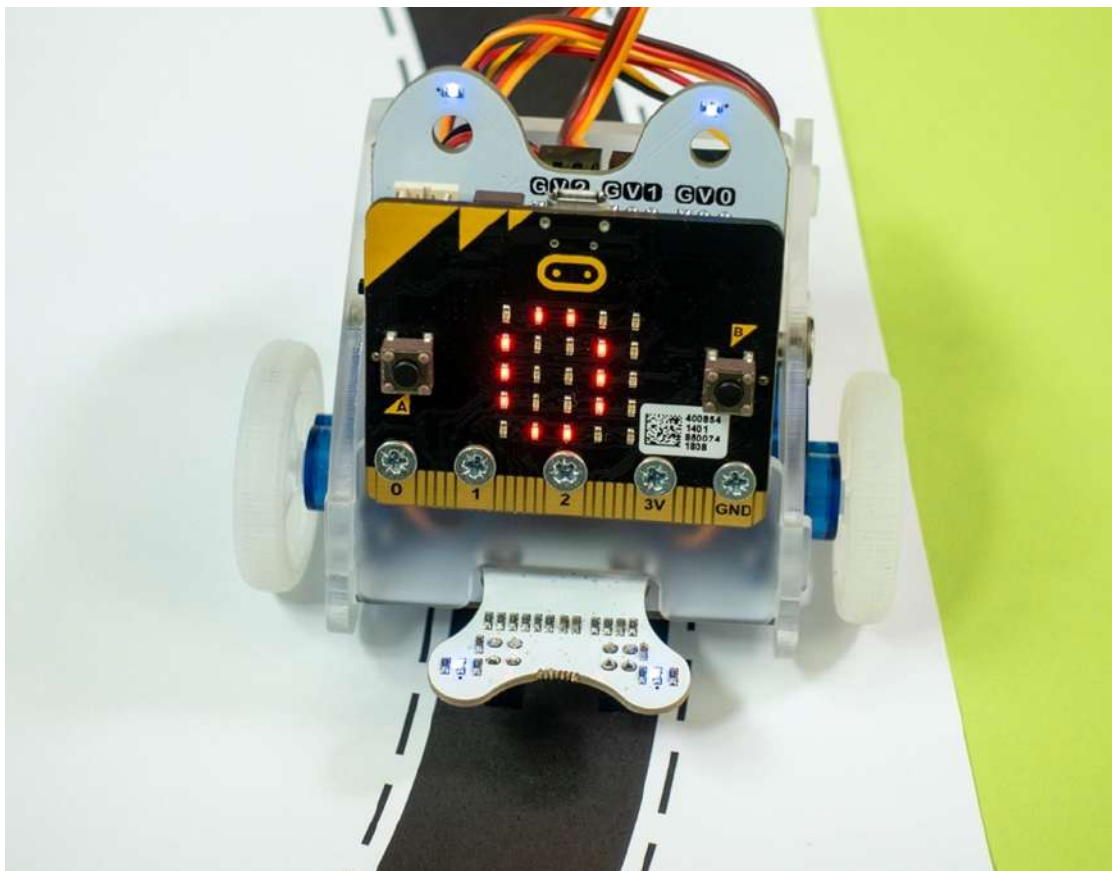
Dva GPIO porty P1 a P2 na hlavnej doske Ring:bitu sú určené na ovládanie servomotorov poháňajúcich kolesá a tretí port P0 sa dá použiť na iný účel, napríklad na optický senzor na sledovanie čiary, ktorý sa dá upevniť zospodu na pomocnú dosku elektroniky.



V príklade budeme realizovať jednu z najobľúbenejších funkcií robotických podvozkov – sledovanie čiary.



Snímač čiary pre robota Ring: bit V2, ktorý sa dodáva ako príslušenstvo dokáže sledovať čiernu čiaru na svetlej podložke, alebo detegovať hranu stola, po ktorom sa pohybuje. Snímač čiary je vybavený dvoma infra senzormi, ktoré dokážu rozpoznať čiernu čiaru alebo hranu povrchu na vzdialenosť $2 \div 12$ mm. Princíp je jednoduchý, od svetlého povrchu sa infračervený signál odrazí a čierny povrch ho pohltí, takže sa neodrazí. Podobne to funguje napríklad po dosiahnutí hrany svetlého stola.



Najskôr si podmienky v aplikácii odladíme tak, že budeme identifikovať kedy sa program nachádza v ktorej vetve podmienky. V prípade ak sú obidva snímače senzora, ľavý aj pravý na čiernej čiare bude

na displeji zobrazená číslica 0. Ak bude robot v polohe, že bude ľavý senzor (pri pohľade spredu) mimo čiernej čiary zobrazí sa 1 a ak bude mimo čiary pravý senzor, zobrazí sa 2. Prepínač na spodnej doske musí byť v polohe Other Modules.

JavaScript

```
RingbitCar.init_wheel(AnalogPin.P1, AnalogPin.P2)
basic.showNumber(0)
basic.forever(function () {
  if (RingbitCar.tracking(RingbitCar.TrackingStateType.Tracking_State_2)) {
    basic.showNumber(2)
    basic.pause(200)
  }
  if (RingbitCar.tracking(RingbitCar.TrackingStateType.Tracking_State_1)) {
    basic.showNumber(1)
    basic.pause(200)
  }
  basic.showNumber(0)
})
```

Teraz po odladení podmienok, budeme namiesto zobrazovania číslic aktivovať pohon pravého, alebo ľavého kolesa, aby sa robotický podvozok vrátil na čiernu čiaru. Pripomíname, že ľavé koleso robotického podvozku je pripojené na porte P1 a pravé koleso na porte P2. Program funguje tak, že ak sa ľavá časť snímača čiary odkloní od čiernej čiary, potom sa pravé koleso zastaví a ľavé koleso zaistí návrat späť na čiaru rýchlosťou 50. Pravá časť snímača funguje rovnakým spôsobom akurát obrátene. Ak obe časti snímača detegujú čiernu čiaru otáčajú sa obidva kolesá rovnakým smerom rýchlosťou 100, takže robotický podvozok ide po čiare dopredu.



JavaScript

```
RingbitCar.init_wheel(AnalogPin.P1, AnalogPin.P2)
RingbitCar.forward()
basic.forever(function () {
  if (RingbitCar.tracking(RingbitCar.TrackingStateType.Tracking_State_2)) {
    RingbitCar.freestyle(50, 0)
    basic.pause(100)
  }
  if (RingbitCar.tracking(RingbitCar.TrackingStateType.Tracking_State_1)) {
    RingbitCar.freestyle(0, 50)
    basic.pause(100)
  }
  RingbitCar.freestyle(100, 100)
})
```

Snímač čiary pre robota Ring: bit V2 má dva senzory, ľavý a pravý. Ak by ste si modul na sledovanie čiary vyrábali sami, mohli by ste si úlohu zjednodušiť a použiť len jeden senzor, čiže jeden pár svietiacej a snímacej infračervenej diódy umiestnený v strede. Vyskúšame aj tento variant s tým, že na snímači použijeme v aplikácii len jeden senzor. Aplikácia je veľmi jednoduchá, avšak pohyb robotického podvozku bude tak trochu viac „cik-cak“. Keďže teraz máme „akože“ iba jeden svetelný senzor, budeme sledovať hranu čiernej čiary, napríklad pravú. Predpokladajme, že pri štarte je senzor nad čiernou čiarou. Robot sa pohne dopredu a keďže potrebujeme sledovať hranu vždy keď je senzor nad čiernou čiarou necháme ho vychýliť doprava. Akonáhle sa dostane nad svetlú plochu začne sa otáčať doľava až kým sa nevráti nad čiaru. Tento cyklus sa opakuje, takže robot de facto nejde priamo po čiare, ale ide kmitavým pohybom okolo pravej hrany čiary.



JavaScript

```
RingbitCar.init_wheel(AnalogPin.P1, AnalogPin.P2)
RingbitCar.forward()
basic.forever(function () {
  if (RingbitCar.tracking(RingbitCar.TrackingStateType.Tracking_State_1)) {
    RingbitCar.freestyle(0, 50)
    basic.pause(100)
  } else {
    RingbitCar.freestyle(50, 0)
    basic.pause(100)
  }
})
```

Využiť môžete aj dve RGB LED diódy na pomocnej doštičke, ktoré sú nasmerované tak, aby svietili na podložku. Kombináciou modrého a červeného svetla môžete napodobniť maják. Prípadne môžete rozsvietením týchto LED signalizovať nejakú udalosť.

JavaScript

```
RingbitCar.init_wheel(AnalogPin.P1, AnalogPin.P2)
RingbitCar.forward()
let strip = neopixel.create(DigitalPin.P0, 2, NeoPixelMode.RGB)
basic.forever(function () {
  strip.showColor(neopixel.colors(NeoPixelColors.Red))
  basic.pause(100)
  strip.showColor(neopixel.colors(NeoPixelColors.Blue))
  basic.pause(100)
})
```

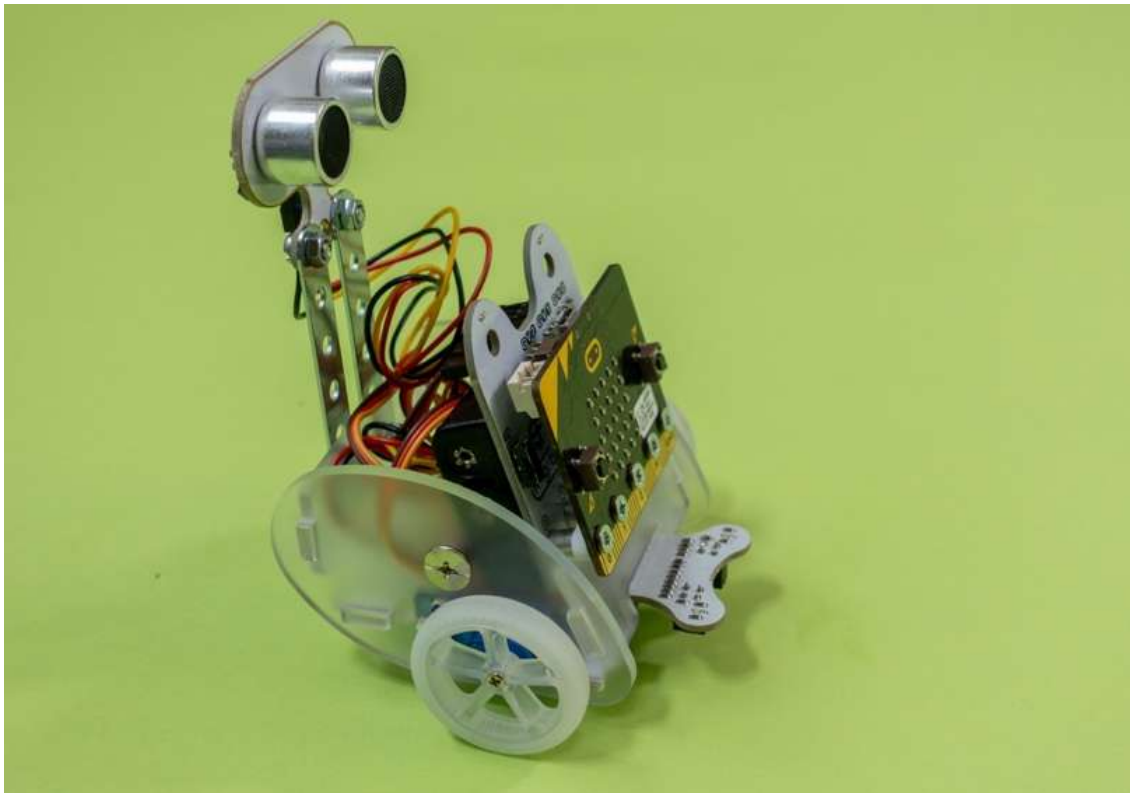
Na testovanie sme úmyselne použili dráhu na ktorej je čiernou čiarou na bielom podklade vykreslená veľká osmička. Vo videu vidíte rozdiel pri použití jedného a dvoch senzorov. Ak použijete dva senzory,

robotický podvozok bude jazdiť po celej osmičke. Ak použijete jeden senzor, ktorý sleduje vnútornú hranu, bude robot krúžiť iba v jednej časti osmičky. Ak robota umiestnite tak, aby sledoval vonkajšiu hranu osmičky bude jeho pohyb v mieste kde sa obidve slučky osmičkovej dráhy pretínajú väčšinou chaotický.

Ultrazvukový senzor na meranie vzdialenosti pre robot s Micro:bitom

Video <https://www.youtube.com/watch?v=U5onC5Vz2-w>

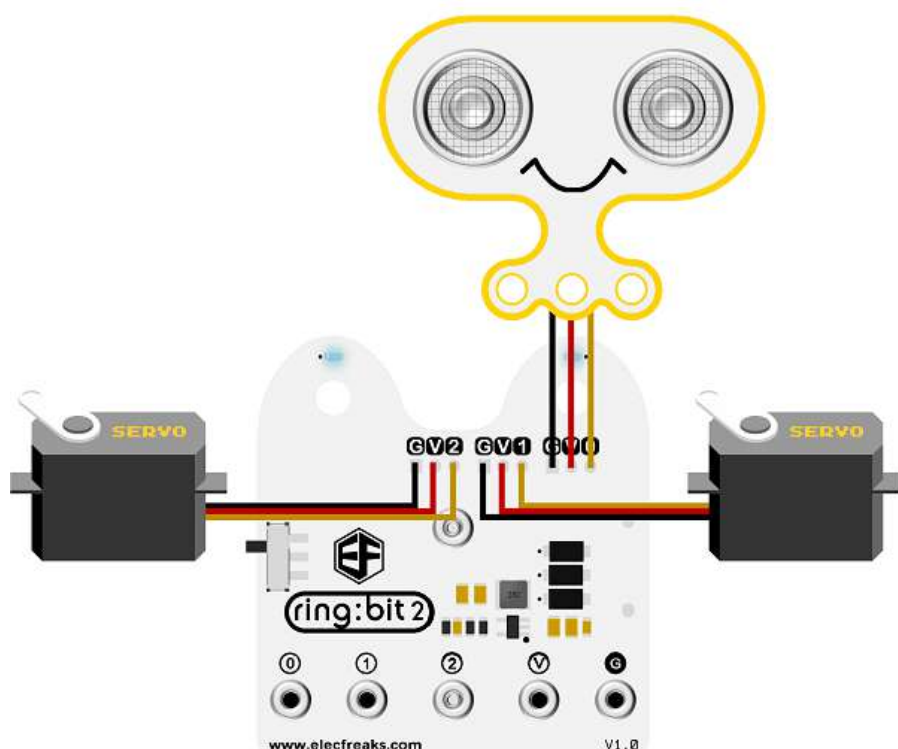
Ultrazvukový merač vzdialenosti Sensor:bit pre populárnu mikropočítačovú dosku BBC Micro:bit umožní merať vzdialenosti 4 ÷ 400 cm s presnosťou 1 cm. To umožňuje vášmu robotickému podvozku orientáciu v priestore, napríklad pri úlohách typu prechod labyrintom. Ak robot pomocou senzora zistí, že je pred ním prekážka, môže včas zmeniť smer.



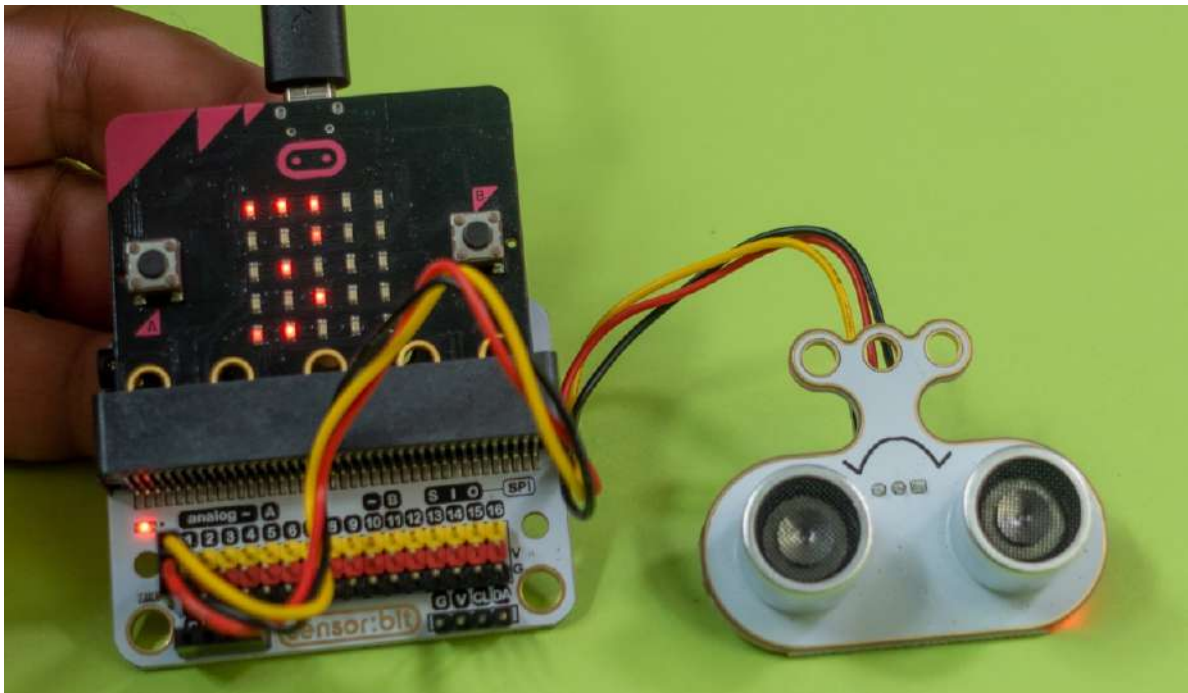
Senzor obsahuje akustický menič schopný vyslať ultrazvukový signál a mikrofón. Funguje tak, že vyšle ultrazvukový signál a čaká až sa odrazí od prekážky a vráti sa k prijímaciemu senzoru. Následne sa zmeria časový rozdiel medzi vyslaním a prijatím signálu a keďže rýchlosť šírenia zvuku je známa, podľa jednoduchého vzorca sa dá vypočítať vzdialenosť od prekážky. Tento výpočet robiť nemusíte, zariadenia to funkcie z knižnice, takže vaša aplikácia bude mať k dispozícii priamo výslednú vzdialenosť od prekážky. Pripomíname, že v online vývojovom prostredí <https://makecode.microbit.org> je potrebné do projektu importovať rozšírenia „ringbitcar“ a „sonarbit“.



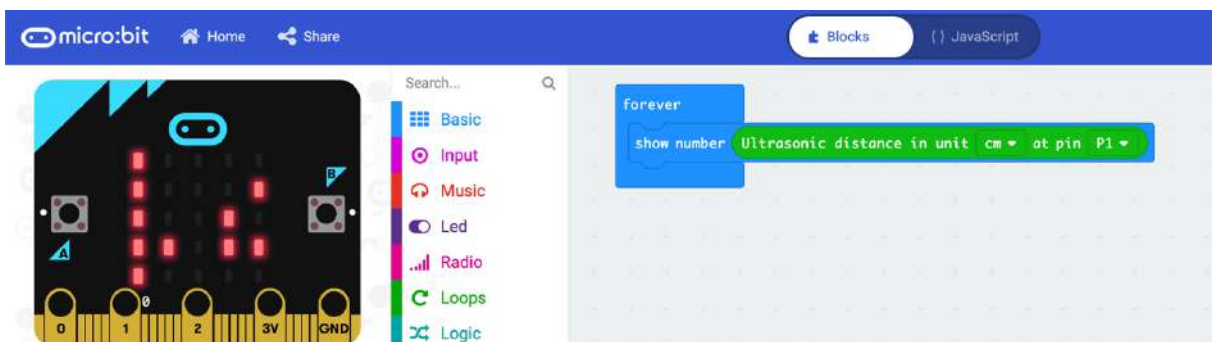
Senzor je okrem vodičov privádzajúcich napájacie napätie 3.3 V, čiže VCC a GND pripojený k Micro:bitu len jedným dátovým vodičom, čiže údaje sa prenášajú po trojvodičovom rozhraní založenom na zbernici 1-Wire. Väčšina ultrazvukových snímačov pre Arduino, či iné mikrokontroléry má totiž 4 vodiče (VCC, GND, TRIG, ECHO), takže pripojenie si vyžaduje dva porty. Senzor je univerzálny. Rozsah napájacieho napätia je 3 ÷ 5 V. To znamená, že merač vzdialenosti môžete použiť ako v 3,3V, tak v 5V systémoch.



V prvom jednoduchom príklade budeme na maticovom displeji microbitu zobrazovať nameranú vzdialenosť.



Na tento príklad nie je potrebný robotický podvozok, stačí pripojiť senzor cez dosku Sensor:bit, alebo využiť vodiče s pinmi určené do prepojovacieho poľa.

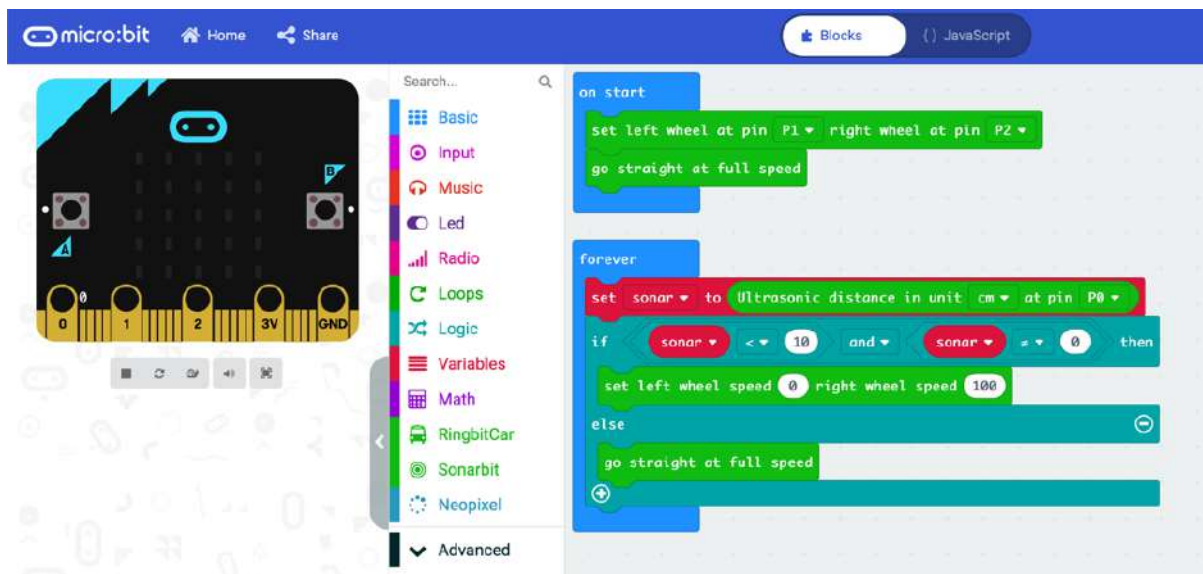


JavaScript

```
basic.forever(function () {  
    basic.showNumber(sonarbit.sonarbit_distance(Distance_Unit.Distance_Unit_cm,  
    DigitalPin.P1))  
})
```

K robotickému podvozku Ring:bit V2 sme ultrazvukový snímač upevnili pomocou dvoch líšt zo stavebnice Merkur, nakoľko sme nemali originálny plexisklový mechanický diel. Senzor sa zapojí na port, kde je inak pripojená spodná doska elektroniky, na ktorú sa upevňuje rôzne príslušenstvo, napríklad infračervené čidlá na sledovanie čiary, či nárazník s RGB LED diódami.

Kód je veľmi jednoduchý, ak senzor zistí, že vo vzdialenosti menšej ako 25 cm je prekážka, otočí sa doľava. Tých 25 cm je z dôvodu, že senzor nie je na robotickom podvozku umiestnený úplne vpredu a potrebuje aj nejaký priestor aby sa mohol otočiť. Parameter v príkaze pause() určuje o aký uhol sa podvozok pootočí.



JavaScript

```

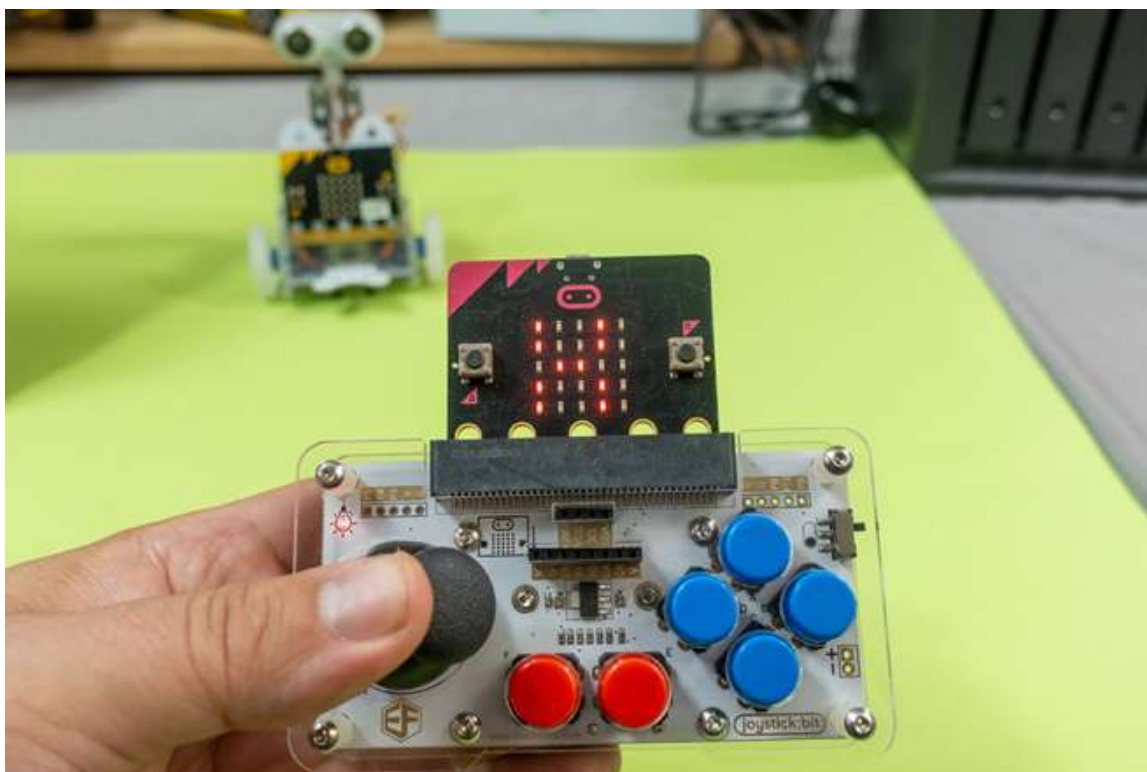
let sonar = 0
RingbitCar.init_wheel(AnalogPin.P1, AnalogPin.P2)
RingbitCar.forward()
basic.forever(function () {
    sonar = sonarbit.sonarbit_distance(Distance_Unit.Distance_Unit_cm,
    DigitalPin.P0)
    if (sonar < 25 && sonar != 0) {
        RingbitCar.freestyle(0, 100)
        basic.pause(500)
    } else {
        RingbitCar.forward()
    }
})

```

Joystick na ovládanie zariadení s Micro:bitom

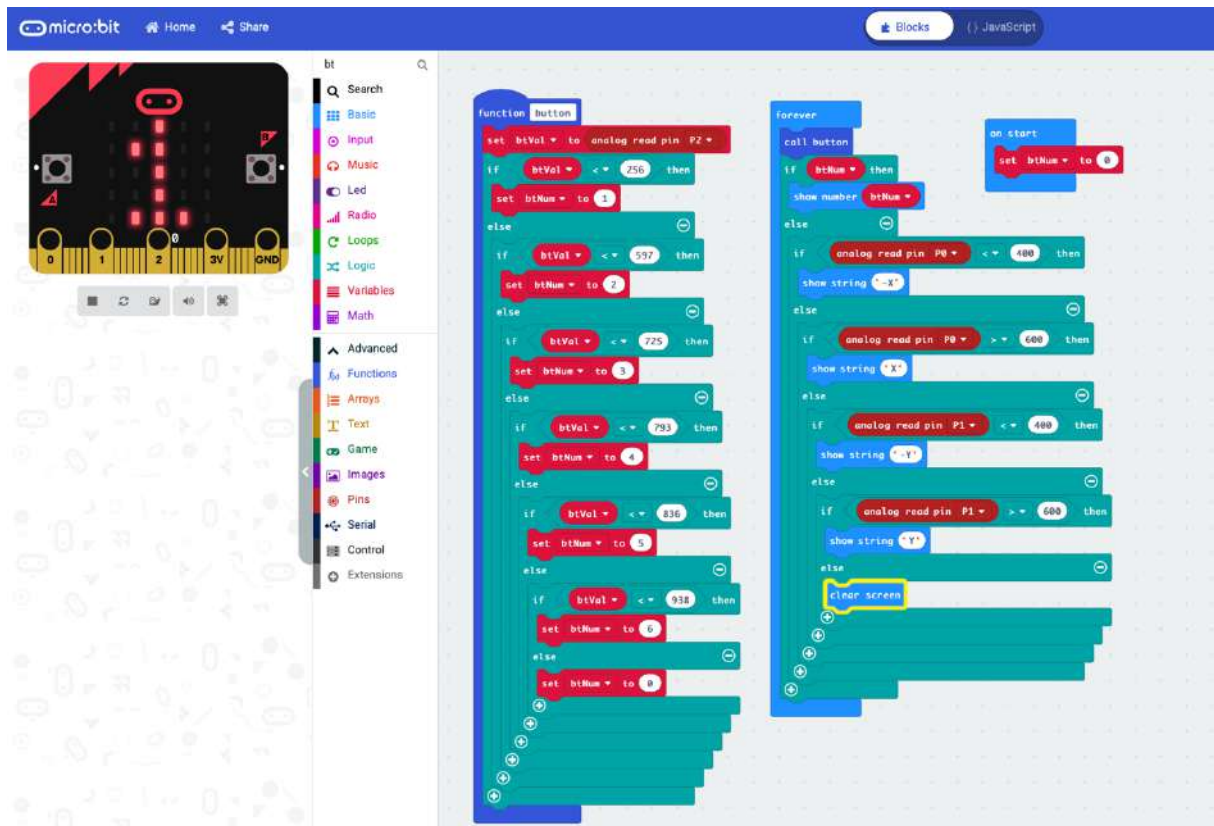
Video <https://www.youtube.com/watch?v=z2rIW3lp11I>

Vo viacerých príkladoch sme ukázali ako je možné ovládať robotické autíčko riadené doskou Micro:bit pomocou druhého Micro:bitu, ktorý držíte v ruke, pričom obidva Micro:bity sú navzájom prepojené. Na ovládanie ste mohli využiť tlačidlá A a B a nakláňanie dosky v ruke, pričom sa do riadiacej dosky robota posielali údaje z akcelerometra v osiach X a Y. Efektné, ale nie príliš presné. Oveľa presnejšie dokážete robotické autíčko, robotické rameno alebo akýkoľvek iný projekt s Micro:bitom ovládať pomocou joysticku. Modul joystick:bit má joystick pre pohyb v osiach X a Y a 6 tlačidiel. Má aj vlastný držiak batérií a tiež ochranné plexi. Všetkým ovládacím prvkom môžete v programe priradiť ľubovoľnú funkciu a vytvoriť bezdrôtový herný ovládač pre akýkoľvek projekt alebo robota s Micro:bitom.



Na doske joysticku je tiež držiak pre 3 batérie AAA, ktoré slúžia ako zdroj napätia. Nechýba ani hlavný vypínač a indikácia napájania. K dispozícii sú aj konektory s ďalšími GPIO signály micro: bit a komunikačné rozhranie I2C, SPI alebo UART. Niektoré piny sú vyvedené v rozložení GVS, ktoré uľahčuje pripojenie rozširujúcich modulov. GVS zodpovedajú signály GND (G), VCC (V), Signál (S). Na doske sú tiež spájkovacej plošky pre externé napájanie. V balení je montážny materiál a ochranné plexisklo.

V prvom príklade na displeji microbitu zasunutého v Joysticku zobrazíme číslo zatlačeného tlačidla, v prípade pohybu páčkou smer, čiže X a -X a Y a -Y. Tlačidlá nie sú pripojené na digitálne vstupy, ale premostujú rezistory deliča napätia, ktorý je pripojený na pin P2, ktorý je nakonfigurovaný ako analógový vstup. Podľa toho, ktoré tlačidlo je zatlačené sa na pin P2 privedie analógové napätie. Ak je hodnota načítaná na analógovom vstupe v intervale hodnôt 0 - 256 je zatlačené tlačidlo 1, v intervale 256 - 597 tlačidlo 2, hodnoty v intervale 598 - 725 indikujú tlačidlo 3 a tak ďalej. Intervaly sú v kóde. Hodnoty vychýlenia ovládacej páčky joysticku sú analógové hodnoty na pinoch P0 a P1



JavaScript

```
function button () {
  btVal = pins.analogReadPin(AnalogPin.P2)
  if (btVal < 256) {
    btNum = 1
  } else {
    if (btVal < 597) {
      btNum = 2
    } else {
      if (btVal < 725) {
        btNum = 3
      } else {
        if (btVal < 793) {
          btNum = 4
        } else {
          if (btVal < 836) {
            btNum = 5
          } else {
            if (btVal < 938) {
              btNum = 6
            } else {
              btNum = 0
            }
          }
        }
      }
    }
  }
}

let btVal = 0
let btNum = 0
btNum = 0
basic.forever(function () {
  button()
  if (btNum) {
    basic.showNumber(btNum)
  }
})
```

```

} else {
  if (pins.analogReadPin(AnalogPin.P0) < 400) {
    basic.showString("-X")
  } else {
    if (pins.analogReadPin(AnalogPin.P0) > 600) {
      basic.showString("X")
    } else {
      if (pins.analogReadPin(AnalogPin.P1) < 400) {
        basic.showString("-Y")
      } else {
        if (pins.analogReadPin(AnalogPin.P1) > 600) {
          basic.showString("Y")
        } else {
          basic.clearScreen()
        }
      }
    }
  }
}
}
})

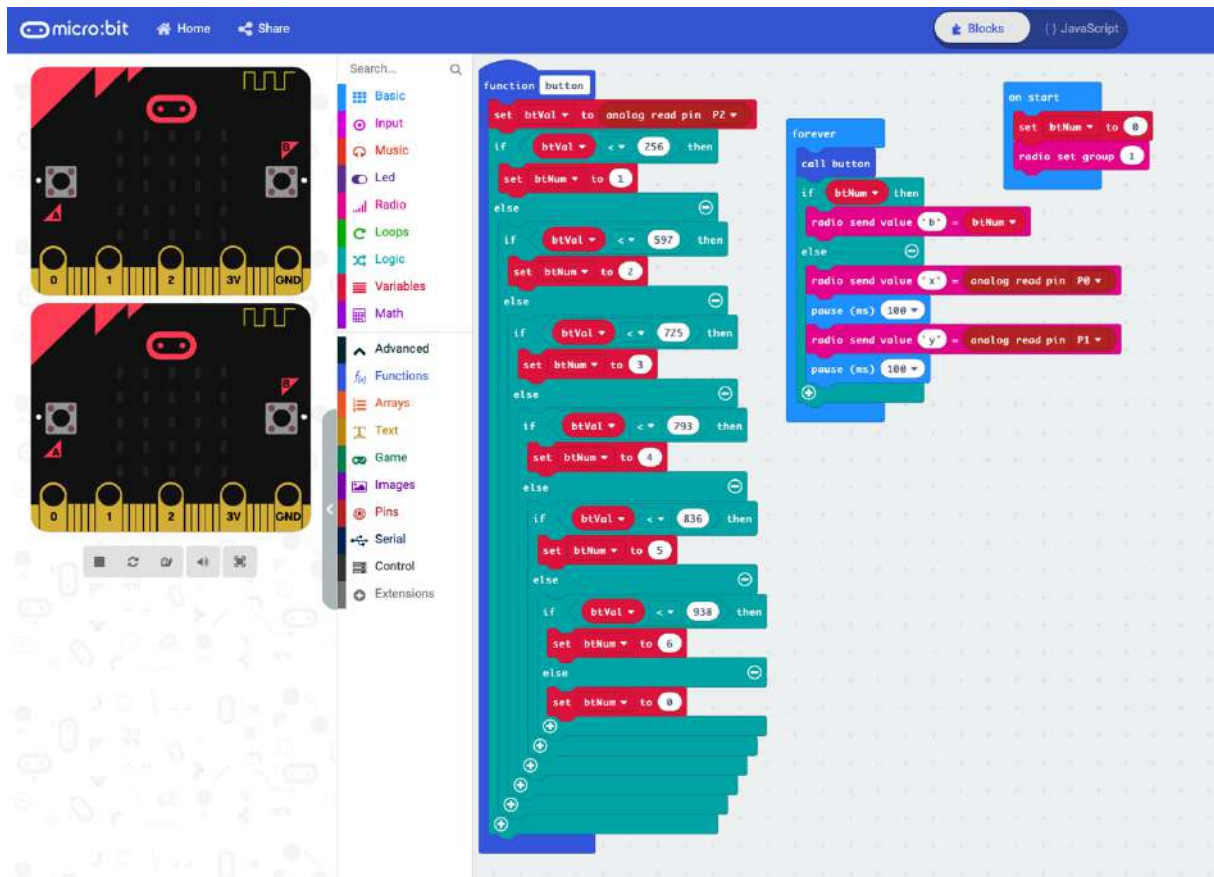
```

V druhom príklade ukážeme ovládanie robotického autíčka. Už nebudeme údaje o zatlačených tlačidlách, alebo polohe joysticku zobrazovať na displeji ale budú sa bezdrôtovo prenášať do druhého Micro:bitu, ktorý v tomto prípade riadi motorčeky a ostatný hardvér robotického autíčka

Micro:bit umožňuje vytvorenie rádiového prepojenia medzi dvomi, prípadne viacerými doskami, takže jedna doska môže riadiť hardvér robotického podvozku a pomocou druhej dosky môžete robotické vozidlo ovládať. Ešte zopakujeme princíp pohybu robotického podvozku s dvoma kolesami, pričom každé poháňa samostatný motor. Ak sa kolieska robotického podvozku otáčajú rovnakou rýchlosťou jedným, alebo druhým smerom, podvozok sa pohybuje priamo dopredu, alebo dozadu. Ak sa kolieska otáčajú rôznou rýchlosťou, robot zatáča do tej strany, na ktorej sa koliesko pohybuje pomalšie. Ak jedno z kolies úplne zastane a druhé sa otáča, tak robot sa začne otáčať na mieste, pričom stredom otáčania je koleso, ktoré sa nepohybuje a polomer otáčania je daný vzdialenosťou medzi kolesami. Ak potrebujete robotický podvozok otočiť na mieste ale na menšom priemere, tak kolieska sa musia otáčať opačným smerom. Vtedy bude polomer otáčania rovný polovičke vzdialenosti medzi kolesami a os otáčania bude v strede pomyslenej spojnice kolies.

Do mikropočítača ovládajúceho autíčko budeme posielat údaje o smere a veľkosti vychýlenia joysticku a tiež údaj o tom aké tlačidlo bolo prípadne stlačené. V inicializačnom bloku je potrebné nastaviť rádiový kanál, konkrétne číslo skupiny. Toto číslo musí byť rovnaké pre obidve dosky, ktoré majú spolu komunikovať. V príklade je pomocou príkazu `setGroup()` ako číslo skupiny nastavené 1.

Kód pre Micro:bit v joysticku obsahuje aj vysielanie hodnôt zatlačených tlačidiel, takže je univerzálny.



JavaScript

```
function button () {
  btVal = pins.analogReadPin(AnalogPin.P2)
  if (btVal < 256) {
    btNum = 1
  } else {
    if (btVal < 597) {
      btNum = 2
    } else {
      if (btVal < 725) {
        btNum = 3
      } else {
        if (btVal < 793) {
          btNum = 4
        } else {
          if (btVal < 836) {
            btNum = 5
          } else {
            if (btVal < 938) {
              btNum = 6
            } else {
              btNum = 0
            }
          }
        }
      }
    }
  }
}

let btVal = 0
let btNum = 0
btNum = 0
radio.setGroup(1)
```

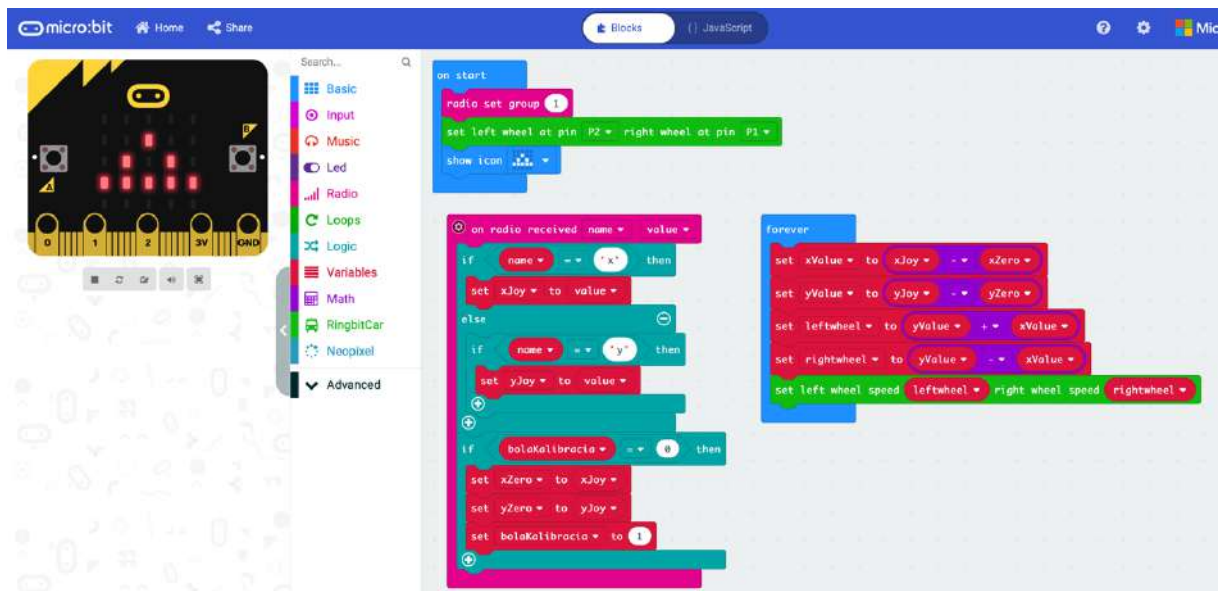
```

basic.forever(function () {
  button()
  if (btNum) {
    radio.sendValue("b", btNum)
  } else {
    radio.sendValue("x", pins.analogReadPin(AnalogPin.P0))
    basic.pause(100)
    radio.sendValue("y", pins.analogReadPin(AnalogPin.P1))
    basic.pause(100)
  }
})

```

V hlavnej slučke sa spracovávajú povely, ktoré obsahujú informácie z joysticku. Na ovládanie pohybu robota do strán sa použije hodnota naklonenia páčky joysticku v osi X a na pohyb dopredu, či dozadu hodnota naklonenia páčky joysticku v osi Y

Kód pre Micro:bit v robotickom podvozku Ring:bit V2



JavaScript

```
radio.setGroup(1)
```

```

radio.onDataPacketReceived(function ({ receivedString: name, receivedNumber: value
}) {
  if (name == "x") {
    xJoy = value
  } else {
    if (name == "y") {
      yJoy = value
    }
  }

  if (bolaKalibracia == 0) {
    xZero = xJoy
    yZero = yJoy
    bolaKalibracia = 1
  }
})

```

```

let rightwheel = 0
let leftwheel = 0

```

```

let yJoy = 0
let xJoy = 0

let yValue = 0
let xValue = 0
RingbitCar.init_wheel(AnalogPin.P2, AnalogPin.P1)
basic.showIcon(IconNames.Triangle)
let xZero = 0
let yZero = 0

let bolaKalibracia = 0

basic.forever(function () {
  xValue = xJoy - xZero
  yValue = yJoy - yZero
  leftwheel = yValue + xValue
  rightwheel = yValue - xValue
  RingbitCar.freestyle(leftwheel, rightwheel)
})

```

Kód na prijímanie a vyhodnocovanie povelov je trochu zložitejší. V hlavnej slučke sa na základe prijatých povelov, ktoré obsahujú informácie o polohe joysticku, pričom pohyb v každej osi generuje hodnoty od 0 do 1023. To sú krajné hodnoty. Pokiaľ je joystick v kľudovej polohe v strede, obidve hodnoty by mali byť okolo 512. Treba to však vedieť presne, preto sa tieto hodnoty zisťujú pri kalibrácii. Preto je potrebné najskôr zapnúť joystick, nechať ovládaciu páku v strednej polohe a až potom zapnúť robotický podvozok. Vtedy sa vykoná kalibrácia. Tento kód sa spustí iba raz

```

let bolaKalibracia = 0
...
if (bolaKalibracia == 0) {
  xZero = xJoy
  yZero = yJoy
  bolaKalibracia = 1
}

```

Hodnoty zistené kalibráciou sa odpočítajú od aktuálne prijatých hodnôt z joysticka

```

xValue = xJoy - xZero
yValue = yJoy - yZero

```

Pohyb dopredu, dozadu a do strán je odvodený od aktuálnej polohy ovládacej páčky.

Predpokladajme zatiaľ, že naklonenie páčky v osi X je nulové. V takom prípade, ak je páčka ovládača naklonená dopredu potrebujete aby sa robot pohyboval dopredu a naopak. Preto ak je hodnota Y kladná budú sa kolesá otáčať takým smerom, aby sa robotický podvozok pohyboval dopredu a pokiaľ je hodnota Y záporná, tak dozadu. V oboch prípadoch sa budú kolesá otáčať rovnakou rýchlosťou. Na ovládanie pohybu robota do strán sa použije hodnota naklonenia páčky ovládača v osi X. Vtedy sa budú kolesá otáčať každé iným smerom. Pre ľavé koleso sa použije hodnota nameraná v osi X priamo a pre pravé hodnota s prevráteným znamienkom.

Z toho vyplývajú vzorce pre otáčky kolies:

```

leftwheel = yValue + xValue
rightwheel = yValue - xValue

```

Na základe znamienok hodnôt v jednotlivých osiach sa určí smer otáčania motorčekov poháňajúcich kolesá. Ak sú hodnoty kladné definuje sa smer otáčania „forward“ a ak sú záporné tak opačný smer „reverse“. V blokoch pre opačný chod sa počíta absolútna hodnota, pretože funkcie na ovládanie motorov vyžadujú pre rýchlosť ako parameter kladné číslo.

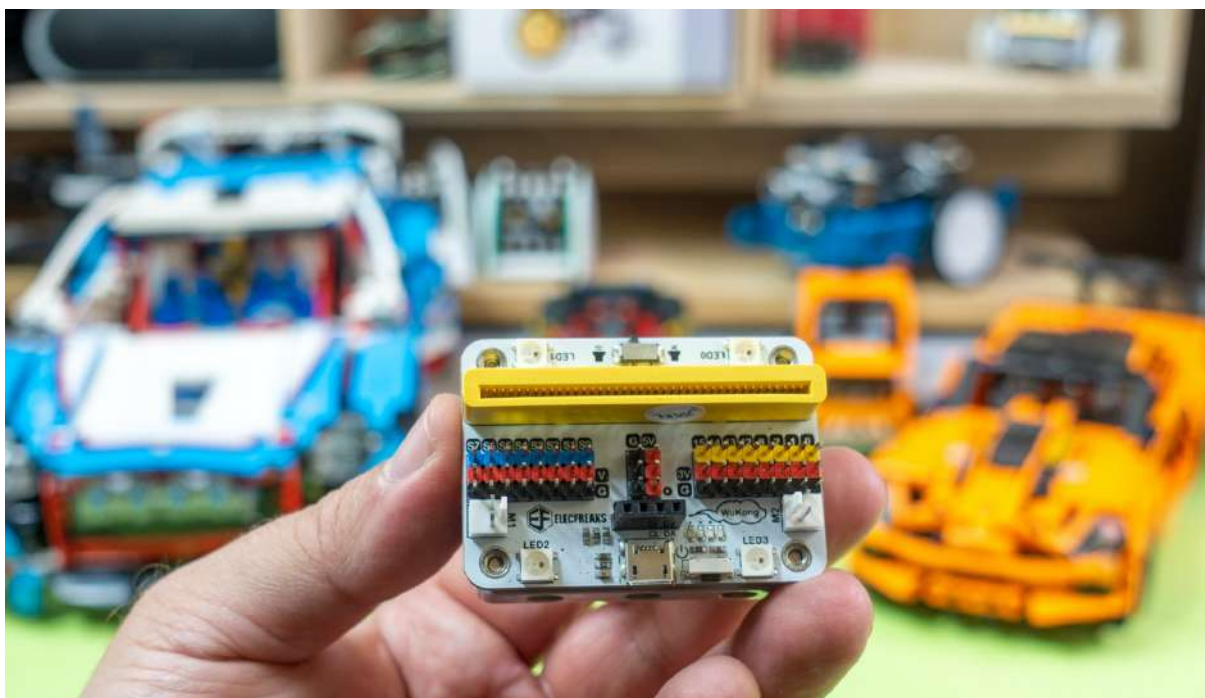
Ovládanie modelov Lego Technic mikropočítačom BBC Micro:bit

Video <https://www.youtube.com/watch?v=z2rIW3lp11l>

V súvislosti s pojmi LEGO a robotika si každý predstaví robotickú stavebnicu LEGO Mindstorms. Jej cena nad 350 eur je však pre mnohých rodičov ako vstupná brána do robotiky trochu riziková investícia, hlavne u detí, kde sa nedá odhadnúť, či ich stavba a programovanie robotov bude, alebo nebude baviť. Niektoré stavebnice LEGO Technic, napríklad modely áut, či iných mechanizmov sú však cenovo dostupnejšie – typicky pod 50 eur a keď sa model zo stavebnice doplní o Micro:bit ako riadiacu jednotku, servá, motory a senzory, môžu vzniknúť kreatívne konštrukcie. Tu vychádza cenová kalkulácia oveľa priaznivejšie. Napríklad stavebnica auta Corvette ZR0 (33 eur) + Micro:bit (14 eur) + modul ElecFreaks Wukong (27 eur), ktorý umožňuje riadiacu jednotku zakomponovať do Lego modelu. Potrebujete samozrejme aj motorčeky, alebo servá. Aj tie sú k dispozícii v takom konštrukčnom usporiadaní aby sa dali spájať s komponentmi stavebnice LEGO Technic a sú lacné.



Azda najzaujímavejšie bude využiť Micro:bit ako riadiacu jednotku modelov áut. Tie majú inú filozofiu ovládania, než robotické podvozky predstavené v miniseriáli Robotika. Tie mali len dve kolesá a robot menil smer zmenou rýchlosti a smeru ich otáčania. Modely z Lega fungujú ako klasické autá a menia smer zatáčaním predných kolies. Väčšina modelov z Lega má diferenciál, takže na pohon stačí jediný motor. Overili sme, že miniatúrne servá napájané z modulu Wukong, ktorý má vstavanú batériu s pomerne nízkym napätím 3.7 V schopné poháňať aj väčšie Lego modely a konštrukcie.

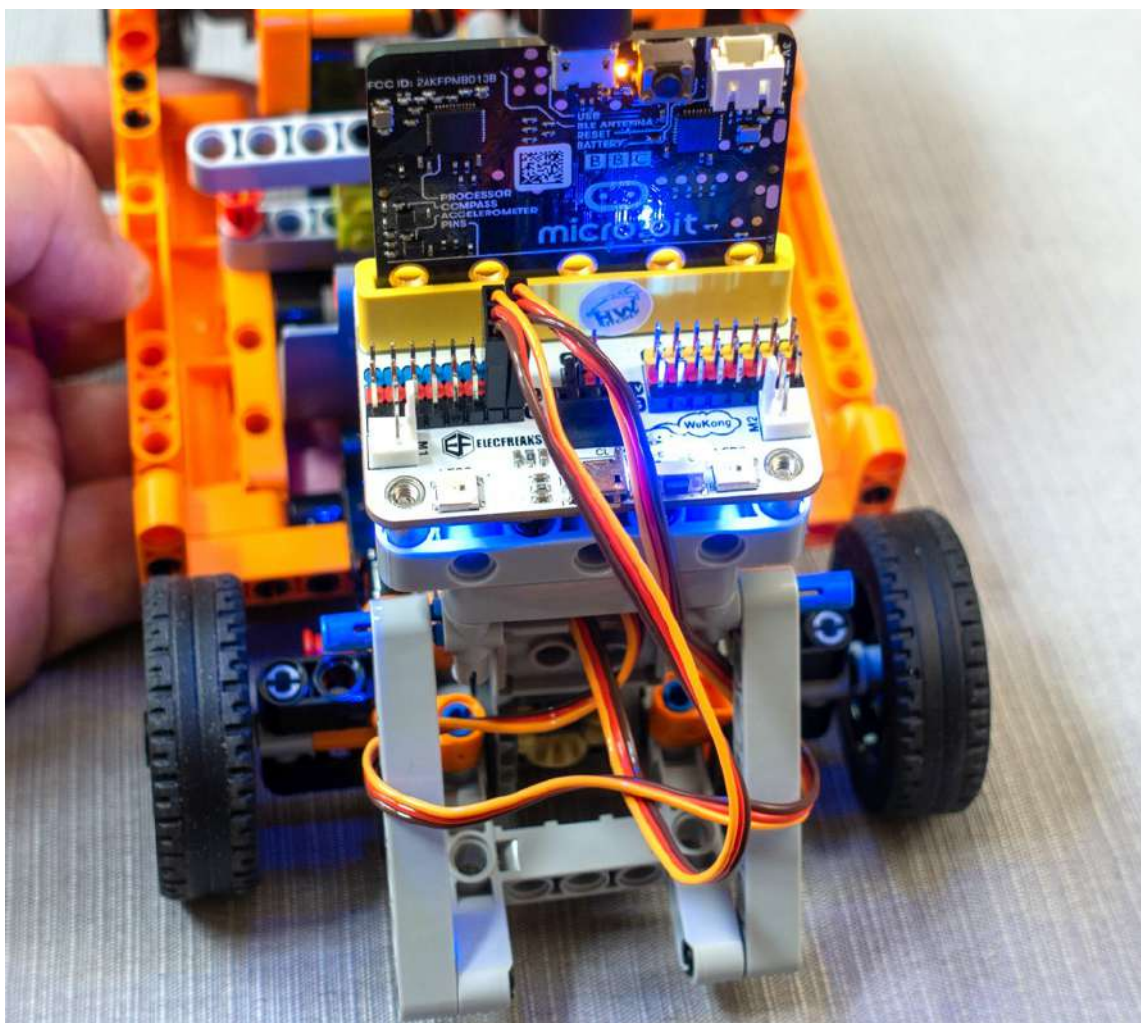


Pripomíname, že servo je k riadiacej jednotke pripojené pomocou troch vodičov. Na červenom vodiči je kladné napájacie napätie, na čiernom (niekedy hnedom) vodiči je zem a na oranžový vodič sa privádzajú signály. Oranžový vodič je vo vnútri serva pripojený na riadiacu elektroniku, nemá veľký prúdový odber, takže ho môžete pripojiť priamo na port. Ovláda sa pomocou PWM. Frekvencia impulzov privádzaných na oranžový vodič je 50 Hz, čo je čas trvania periódy 20 ms. Uhol natočenia sa ovláda šírkou impulzu. U klasického modelárskeho serva je šírka pulzu 0.5 až 2.5 ms a tejto hodnote zodpovedá uhol natočenia 0° až 180, prípadne 0 – 270 stupňov, podľa konštrukcie serva. Kontinuálne servomotory fungujú tak, že zmenou šírky impulzov sa mení rýchlosť otáčania.

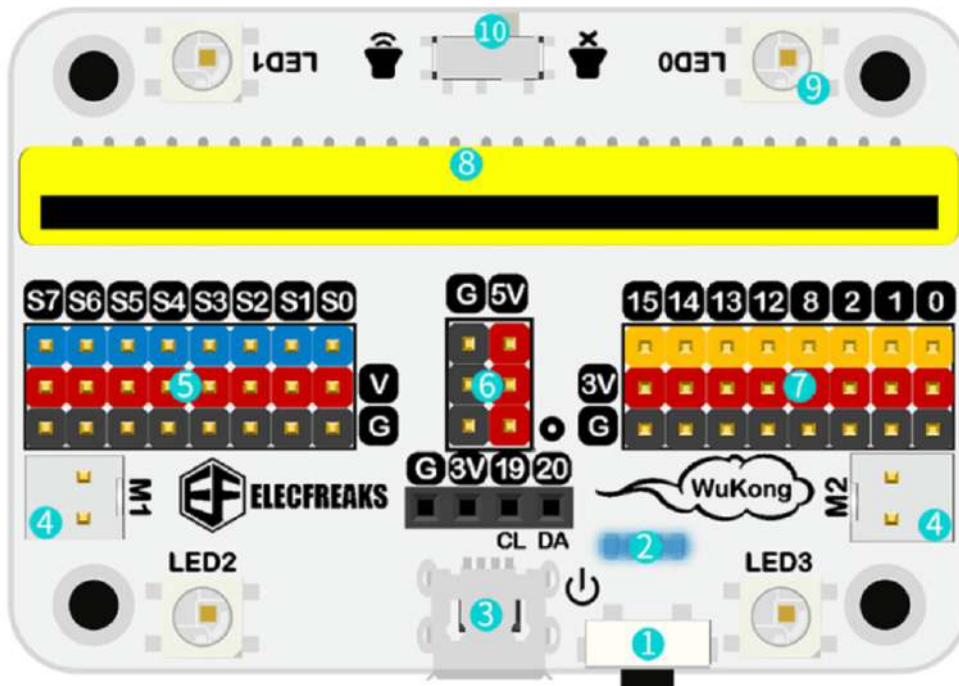
Príklad prototypu modelu LEGO Technic mikropočítačom BBC Micro:bit 2

Video <https://www.youtube.com/watch?v=0R24Gk-tKNY>

Rozširujúci modul Wukong nie je o mnoho väčší než samotný Micro:bit. Je konštruovaný tak, že sa dá zabudovať do platformy LEGO vybavené štandardným rámečkom 7 x 5. Vo vnútri platformy je 400 mAh akumulátor. Pri stavbe modulu dbajte na to, aby pokiaľ možno bol viditeľný maticový displej, na ktorom sa dajú zobrazovať rôzne informácie a boli dostupné tlačidlá. Mimochodom názov Wukong je odvodený od postavy Opičieho kráľa z čínskej románovej tvorby. Opičiak (podobne ako tento modul pre Micro:bit) vynikal rôznymi schopnosťami a ovládal tiež rôzne premeny a kúzla.



Wukong dokáže ovládať dva jednosmerné motory a osem serv. Môžete taktiež použiť ďalšie moduly s možnosťou pripojenia cez GVS, čiže tri vodiče: G – elektrická zem, V – napájacie napätie 3.3 V a S – signál z portu. Pripojiť môžete aj OLED modul, prípadne rôzne snímače, napríklad BME280. Priamo na doske Wukong sú 4 RGB LED diódy na vytváranie farebných efektov, 8 modrých LED na vytvorenie efektného podsvietenia a bzučiak napríklad pre funkciu trúbenie. K modulu Wukong sú k dispozícii LEGO kompatibilné aktuátory a snímače, ako DC motory, serva a rôzne snímače.

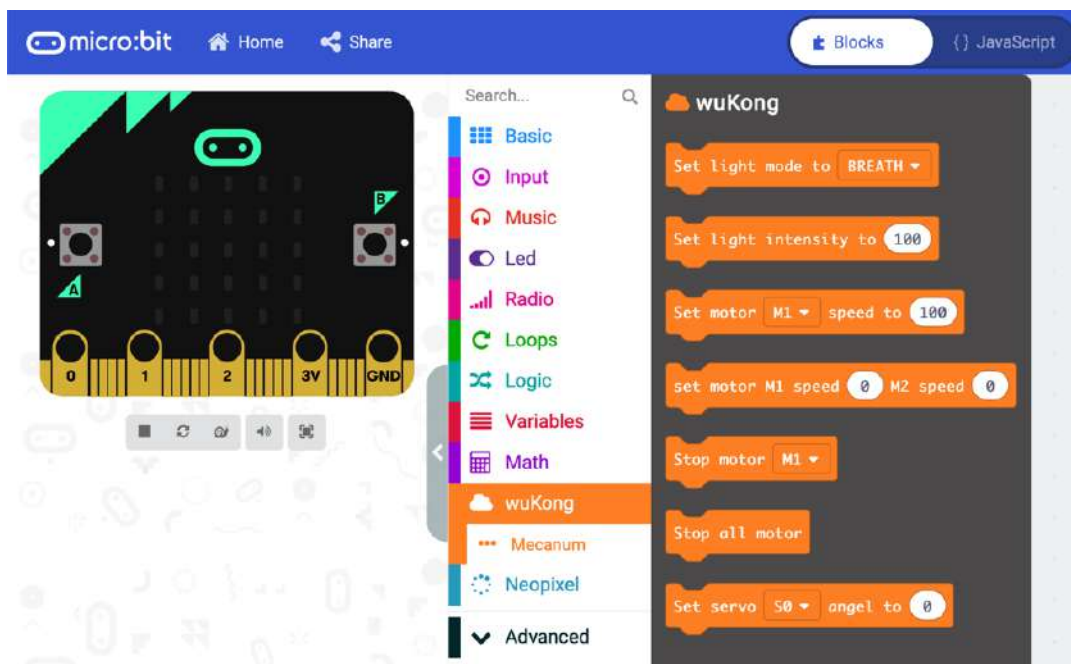


Zakomponovanie servomotorov je otázkou kreativity a bude u každého modelu iné. V našom prípade sme napojili kontinuálne servo na hriadeľ poháňajúci zadné kolesá cez diferenciál a klasické servo, ktoré sa otáča o definovaný uhol na hriadeľ mechanizmu natáčania predných kolies. Skôr, než sme servá pripojili dosku Wukong, vyskúšali sme použiť na ich ovládanie súpravu na diaľkové ovládanie modelov a servá sme pripojili na modul, ktorý sa inštaluje do modelu auta, lietadla, či lode. Vo videu môžete vidieť, že autíčko ovládané natáčaním predných kolies má menšie manévrovacie schopnosti ako robotický podvozok, čo je potrebné pri programovaní zohľadniť. Ak doplníte autíčko o senzor sledovania čiary, budú musieť byť oblúky väčšie - také, aby ich autíčko dokázalo prejsť.



Pripomenieme, že Servo je k riadiacej jednotke pripojené pomocou troch vodičov. Na červenom vodiči je kladné napájacie napätie, na čiernom (niekedy hnedom) vodiči je zem a na oranžový vodič sa privádzajú signály. Oranžový vodič je vo vnútri serva pripojený na riadiacu elektroniku, nemá veľký prúdový odber, takže ho môžete pripojiť priamo na port. Ovláda sa pomocou PWM. Frekvencia impulzov privádzaných na oranžový vodič je 50 Hz, čo je čas trvania periódy 20 ms. Uhol natočenia sa ovláda šírkou impulzu. U klasického modelárskeho serva je šírka pulzu 0.5 až 2.5 ms a tejto hodnote zodpovedá uhol natočenia 0° až 180, prípadne 0 – 270 stupňov, podľa konštrukcie serva. Kontinuálne servomotory fungujú tak, že zmenou šírky impulzov sa mení rýchlosť otáčania.

V našich príkladoch bude servo na ovládanie natočenia predných kolies pripojené na port S0 a kontinuálne servo na pohon zadných kolies na port S1. Do prostredia MakeCode, je potrebné doinštalovať rozširujúcu knižnicu funkcií Wukong.

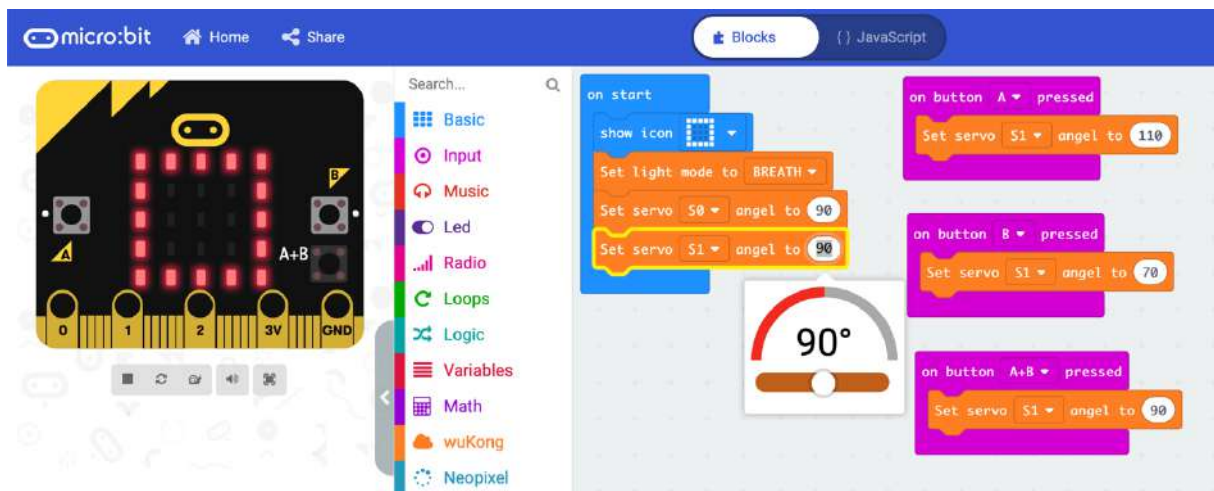


V prvom príklade nastavíme servo riadenia do východiskovej polohy, čiže do stredu. Nakoľko servo sa natáča v rozsahu 0 – 180° bude stredová poloha pri nastavenom uhle 90°. Po spustení programu sa servo pripojené na port S0 do tejto polohy natočí. Treba ho spojiť s hriadeľom tak, aby v tejto polohe smerovali kolesá rovno. Keďže modul Wukong má aj efektné modré LED diódy tak pre efekt ich nastavíme tak, aby sa v móde BREATH rozsviecovali a stmievali. Môžete taktiež nastaviť a meniť farby na štyroch RGB diódach v rohu dosky. Na tento účel použijete funkcie z knižnice Neopixel.

```
basic.showIcon(IconNames.Heart)
wukong.setLightMode(wukong.LightMode.BREATH)

input.onButtonPressed(Button.A, function () {
    wukong.setServoAngel(wukong.ServoList.S0, 90)
})
```

Aj druhý príklad bude na kalibráciu riadenia. Skúšajte nastavovať rozsah uhlov zodpovedajúci natočeniu prednej nápravy. Niektoré modely majú riadenie sprevádzané viac, iné menej. V našom modeli sa osvedčilo nastavenie rozsahu 20° na každú stranu, čiže od 70 do 100°. Tlačidlom A sa riadenie vychýli do jednej a tlačidlom B do druhej strany. Súčasným zatlačením tlačidiel A+B sa nastaví uhol 90°



```
input.onButtonPressed(Button.A, function () {
    wuKong.setServoAngel (wuKong.ServoList.S0, 110)
})
input.onButtonPressed(Button.B, function () {
    wuKong.setServoAngel (wuKong.ServoList.S0, 70)
})
input.onButtonPressed(Button.AB, function () {
    wuKong.setServoAngel (wuKong.ServoList.S0, 90)
})
```

Rovnaký kód použijeme aj na vyskúšanie rýchlosti. Kontinuálne servo na porte S1 v polohe 90° stojí a v závislosti od uhla natočenie smerom k 0°, alebo 90° sa menia otáčky smerom dopredu, alebo dozadu. Pri takto nastavených hodnotách sa bude motor otáčať relatívne pomaly

```
basic.showIcon (IconNames.Square)
wuKong.setLightMode (wuKong.LightMode.BREATH)
wuKong.setServoAngel (wuKong.ServoList.S0, 90)
wuKong.setServoAngel (wuKong.ServoList.S1, 90)

input.onButtonPressed(Button.A, function () {
    wuKong.setServoAngel (wuKong.ServoList.S1, 110)
})

input.onButtonPressed(Button.B, function () {
    wuKong.setServoAngel (wuKong.ServoList.S1, 70)
})

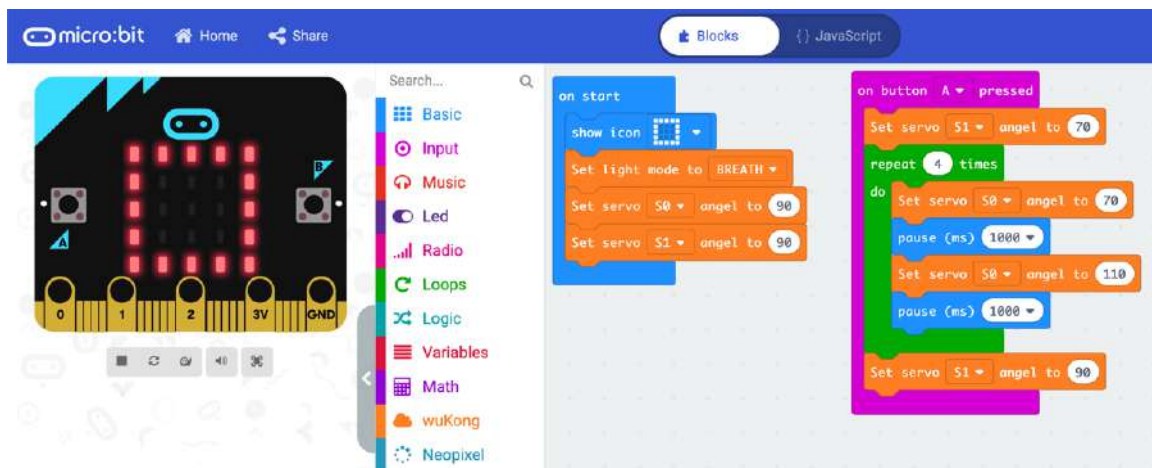
input.onButtonPressed(Button.AB, function () {
    wuKong.setServoAngel (wuKong.ServoList.S1, 90)
})
```

Maximálne otáčky bu ste dosiahli nastavením hodnôt

```
input.onButtonPressed(Button.A, function () {
    wuKong.setServoAngel (wuKong.ServoList.S1, 180)
})

input.onButtonPressed(Button.B, function () {
    wuKong.setServoAngel (wuKong.ServoList.S1, 0)
})
```

Skúsime jednoduchý program ktorý urobí pri pohybe dopredu niekoľko zatočení na jednu, alebo druhú stranu



```

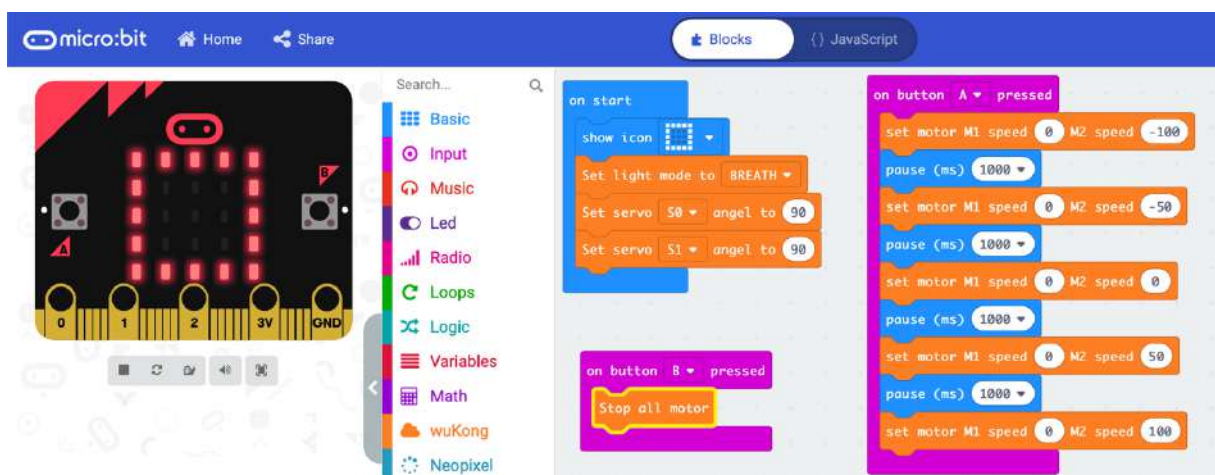
basic.showIcon(IconNames.Square)
wuKong.setLightMode(wuKong.LightMode.BREATH)
wuKong.setServoAngel(wuKong.ServoList.S0, 90)
wuKong.setServoAngel(wuKong.ServoList.S1, 90)

input.onButtonPressed(Button.A, function () {
  wuKong.setServoAngel(wuKong.ServoList.S1, 70)
  for (let i = 0; i < 4; i++) {
    wuKong.setServoAngel(wuKong.ServoList.S0, 70)
    basic.pause(1000)
    wuKong.setServoAngel(wuKong.ServoList.S0, 110)
    basic.pause(1000)
  }
  wuKong.setServoAngel(wuKong.ServoList.S1, 90)
})

```

Program môžete upraviť tak, aby autíčko „autonómne“ zacúvalo na parkovacie miesto vymedzené dvomi papierovými škatuľkami a podobne.

V modeli autíčka v príklade sme nepoužili jednosmerný motor. Ak by ste ho chceli použiť namiesto kontinuálneho serva, pripája sa na konektor určený pre motory. Rýchlosť otáčania sa riadi pulzne šírkovou moduláciou s parametrom 0 – 100 a smer otáčania sa ovláda znamienkom



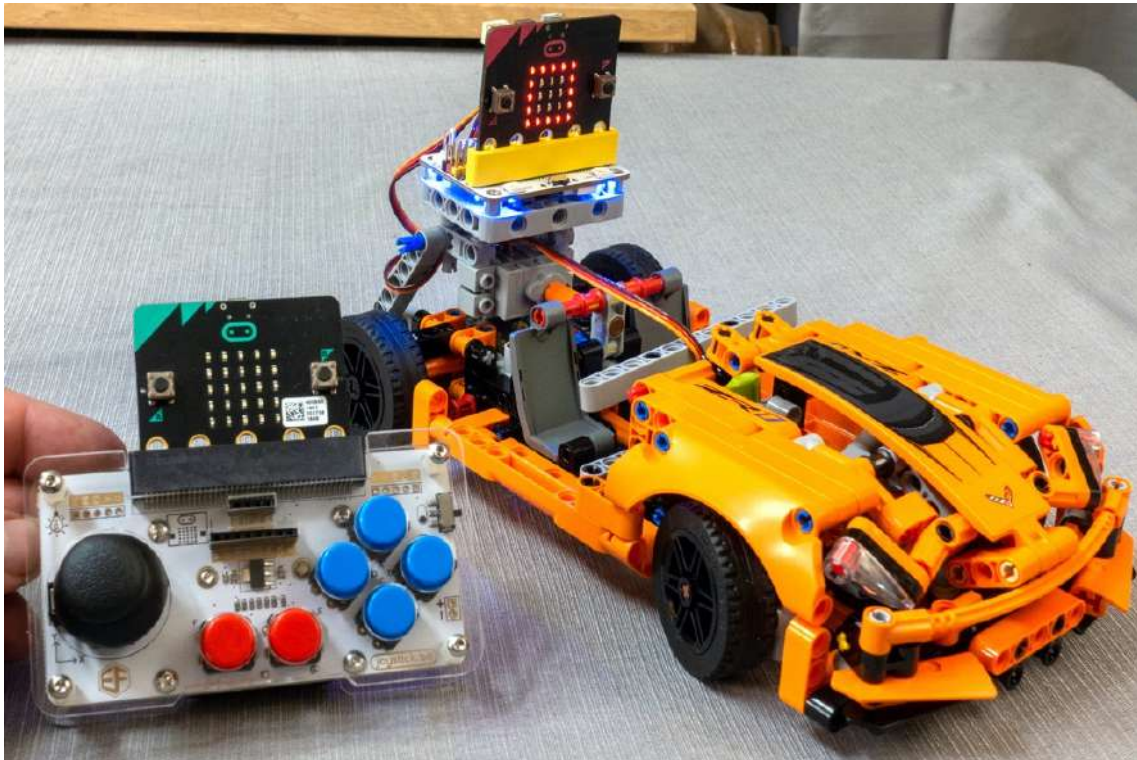
```

input.onButtonPressed(Button.A, function () {
  wuKong.setAllMotor(0, -100)
  basic.pause(1000)
  wuKong.setAllMotor(0, -50)
  basic.pause(1000)
  wuKong.setAllMotor(0, 0)
})

```

```
basic.pause(1000)
wuKong.setAllMotor(0, 50)
basic.pause(1000)
wuKong.setAllMotor(0, 100)
})
input.onButtonPressed(Button.B, function () {
  wuKong.stopAllMotor()
})
```

Veľmi zaujímavé je rádiové ovládanie modelu druhým Micro:bitom. Táto téma aj s príkladmi je podrobne popísaná v kapitole Joystick na ovládanie zariadení s Micro:bitom



Program pre Micro:bit v joysticku môže zostať bezo zmeny avšak vzhľadom na iný princíp pvládania modelu musíte modifikovať program Micro:bitu ovládajúci servomotory.



K modulu Wukong môžete pripájať akékoľvek príslušenstvo, ktoré sa pripája pomocou troch vodičov GVS. Napríklad zo súpravy Smarthome:kit.

Robotika – robot s kolesami Mecanum sa môže pohybovať všetkými smermi

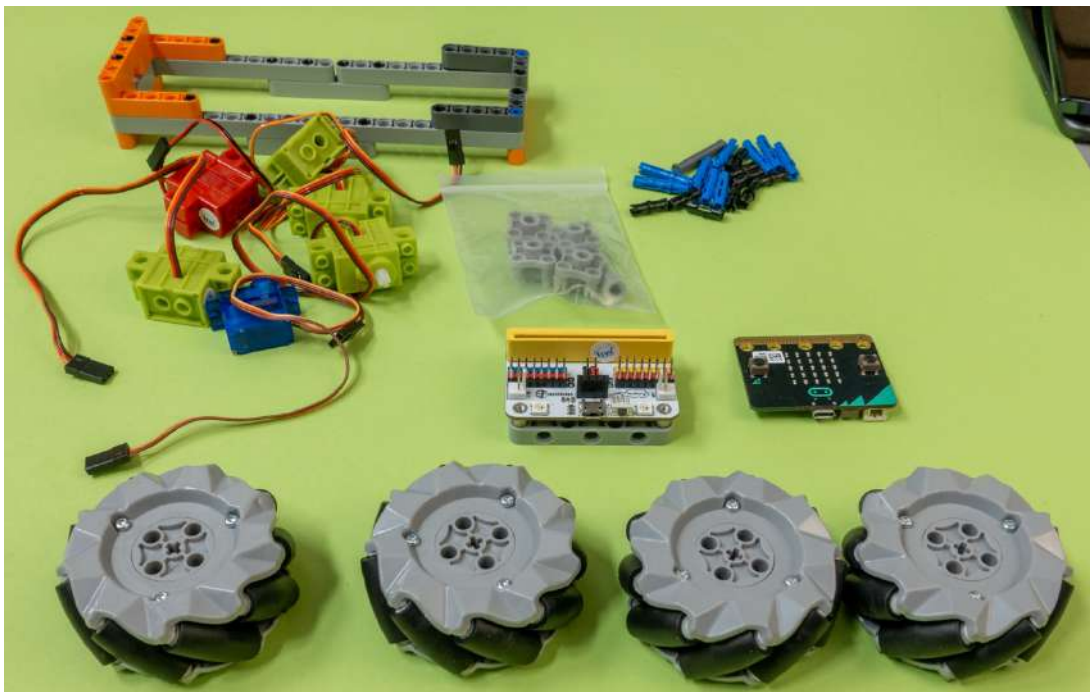
Video <https://www.youtube.com/watch?v=7Ezes6fPYXo>

Dosiaľ sme predstavili dva spôsoby manévrovania robotických podvozkov. Robotické podvozky ktoré majú dve samostatne poháňané kolesá + opornú guľičku a robot mení smer zmenou rýchlosti a smeru ich otáčania a autíčko z Lega doplnené o servá a Micro:bit, ktoré mení smer zatáčaním predných kolies.

K dispozícii sú však aj ďalšie možnosti, napríklad malé bágre majú štyri kolesá bez možnosti natáčania a smer menia zmenou smeru a rýchlosti otáčania, pričom niektoré z kolies preklzujú. To si však na niektorých povrchoch vyžaduje značný krútiaci moment. V článku ukážeme ešte zaujímavejší spôsob manévrovania štvorkolesového podvozku so štyrmi kolesami Mecanum, ktoré sú upevnené tak, že nemajú možnosť zatáčania a napriek tomu má podvozok bohaté možnosti manévrovania. Tieto kolesá si právom zaslúžia prívlastok všesmerové. Princíp najlepšie pochopíte z videa.



Kolesá Mecanum majú priemer 64 mm a po obvode majú šikmo umiestnené otočné valčeky. Kolesá sú kompatibilné so stavebnicou LEGO Technic, majú otvory pre hriadele v tvare kríža takže sa dajú nasadiť priamo na Geek servá určené pre stavebnicu LEGO . V balení sú však aj redukcie aby bolo kolesá možné poháňať jednosmernými motormi TT alebo kontinuálnymi modelárskymi servami SG90. Aby podvozok mohol manévrovať, potrebujete mať možnosť otáčať všetkými štyrmi kolesami a to obidvomi smermi.



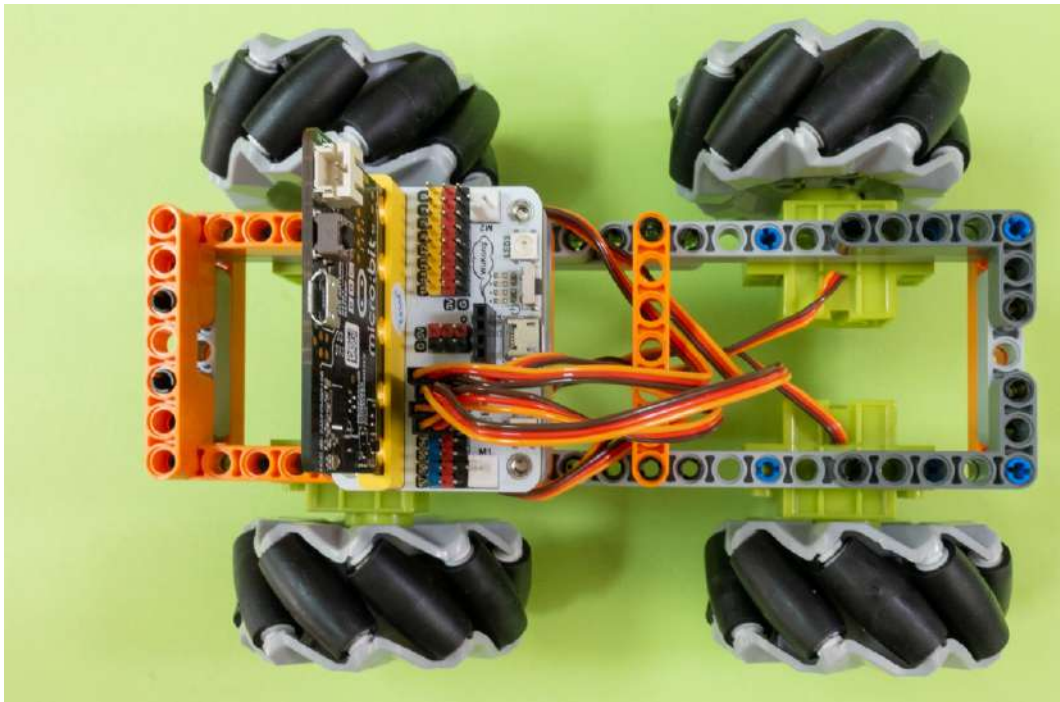
Koleso Mecanum vynašiel inžinier švédskej firmy Mecanum AB Bengt Erland Ilon a v roku 1972 bol princíp kolesa patentovaný v USA. Je to koleso s radom valcov pripevnených k jeho obvodu. Každý z týchto valcov má os rotácie pootočenú o 45°. Striedaním kolies s ľavými a pravými valcami tak, že každé koleso pôsobí silou zhruba kolmo na diagonálu rázvoru kolies je vozidlo stabilné a môže sa pohybovať v ľubovoľnom smere a otáčať sa zmenou rýchlosti a smeru otáčania každého kolesa. Pohyb

všetkých štyroch kolies v tom istom smere spôsobuje pohyb vpred alebo vzad, beh kolies na jednej strane v opačnom smere ako na druhej strane spôsobuje rotáciu vozidla a beh kolies na diagonále v opačnom smere ako na kolesách na druhej uhlopriečke spôsobuje pohyb do strany. Kombinácie týchto pohybov kolies umožňujú pohyb vozidla v ľubovoľnom smere s akoukoľvek rotáciou.

Na ovládanie využijeme rozširujúci modul Wukong popísaný v predchádzajúcej stati. Je konštruovaný tak, že sa dá zabudovať do platformy LEGO vybavené štandardným rámečkom 7 x 5. Vo vnútri platformy je 400 mAh akumulátor. Wukong dokáže ovládať dva jednosmerné motory a osem serv. Môžete taktiež použiť ďalšie moduly s možnosťou pripojenia cez GVS, čiže tri vodiče: G – elektrická zem, V – napájacie napätie 3.3 V a S – signál z portu. Pripojiť môžete aj OLED modul, prípadne rôzne snímače. Priamo na doske Wukong sú 4 RGB LED diódy navytváranie farebných efektov, 8 modrých LED na vytvorenie efektného podsvietenia a bzučiak napríklad pre funkciu trúbenie. Na pohon kolies Mecanum využijeme štyri kontinuálne 360° servá kompatibilné s Legom. Pripomenieme, že servo je k riadiacej jednotke pripojené pomocou troch vodičov. Na červenom vodiči je kladné napájacie napätie, na čiernom (niekedy hnedom) vodiči je zem a na oranžový vodič sa privádzajú signály. Oranžový vodič je vo vnútri serva pripojený na riadiacu elektroniku, nemá veľký prúdový odber, takže ho môžete pripojiť priamo na port. Ovláda sa pomocou PWM. Frekvencia impulzov privádzaných na oranžový vodič je 50 Hz, čo je čas trvania periódy 20 ms. Kontinuálne servomotory fungujú tak, že zmenou šírky impulzov v rozsahu 0.5 až 2.5 ms sa mení rýchlosť otáčania. Pri šírke 1.5 ms sa servo neotáča. Pri kratších impulzoch sa otáča na jednu a pri dlhších impulzoch na druhú stranu.

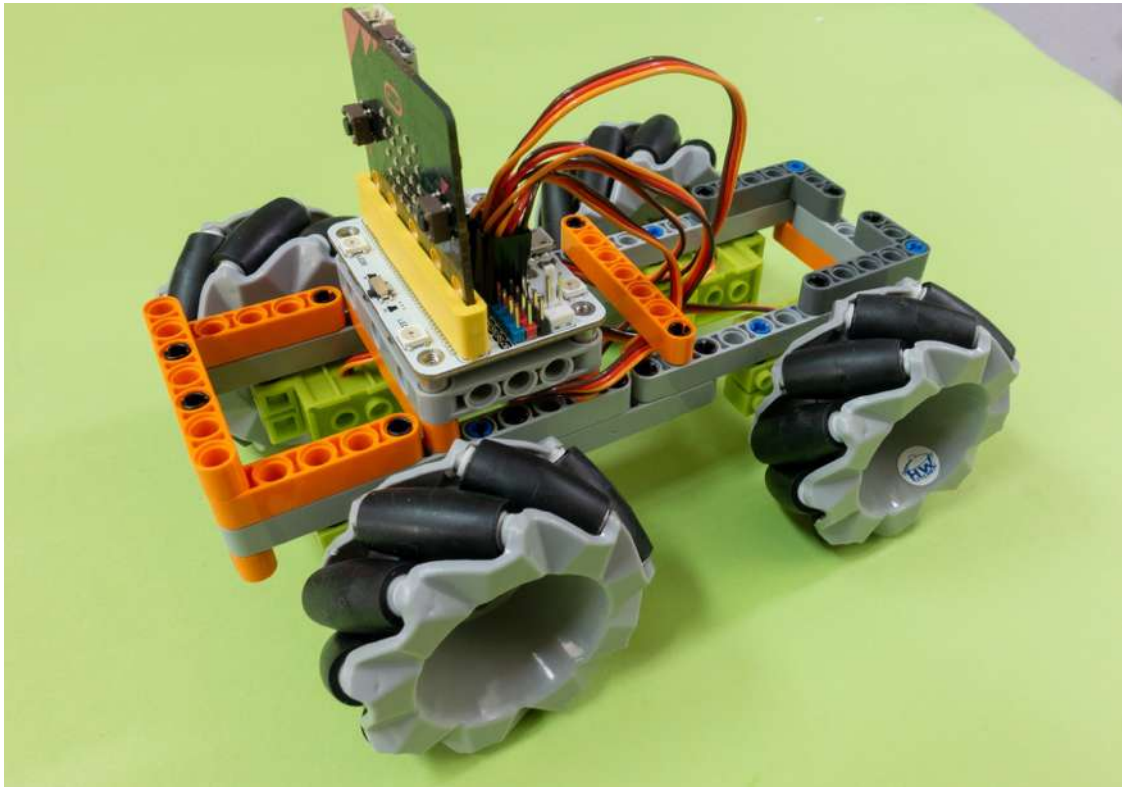
Servomotory kolies sú v našej konštrukcii k doske Wukong pripojené takto:

- Ľavé predné koleso – S0
- Pravé predné koleso – S1
- Ľavé zadné koleso – S2
- Pravé zadné koleso – S3



Do prostredia MakeCode, je potrebné doinštalovať rozširujúcu knižnicu funkcií Wukong. Najnovšia verzia knižnice Wukong má aj podzložku Mecanum a v nej funkcie na ovládanie kolies. V ďalších príkladoch budeme tieto funkcie používať, ale prvý príklad urobíme všeobecný tak, že budeme

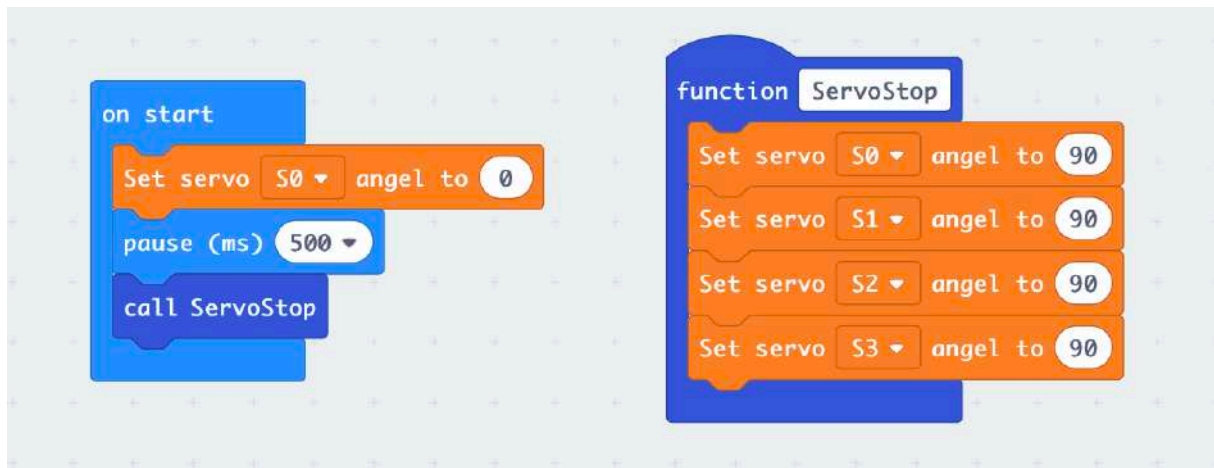
ovládať priamo servá, aby kolesá Mecanum mohli použiť aj tí, čo použijú inú dosku, ktorá má GVS výstupy na servá, napríklad Motor:bit, Robit, ale aj dosku zo súpravy Smart Home Kit



Príklad s priamym ovládaním servomotorov

V prvom príklade sa budú všetky kolesá otáčať rovnakým smerom, aby sa robotický podvozok pohyboval dopredu. Kontinuálne servo sa z hľadiska knižničných funkcií ovláda ako normálne, čiže sa zadáva uhol v rozsahu 0 – 180°. Kludový stav bude pri nastavenom uhle 90°. Pri uhloch väčších ako 90° sa bude servo otáčať na jednu stranu a čím bude uhol väčší, tým bude pohyb rýchlejší. Pri hodnote 180° sa bude servo otáčať najrýchlejšie. Pri uhloch menších ako 90° sa bude servo otáčať na jednu stranu a čím bude uhol menší, tým bude pohyb rýchlejší. Pri hodnote 0° sa bude servo otáčať najrýchlejšie.

Nakoľko smer otáčania hriadeľa serva závisí na tom ako a kde je servo namontované, bude potrebné zistiť pri akých hodnotách sa otáča každé koleso dopredu a pri akých dozadu a podľa toho napísať funkciu na ovládanie motorov. Funkciu na zastavenie všetkých štyroch motorov môžete napísať hneď, stačí pre všetky servomotory nastaviť uhol 90°. Pomocou jednoduchej programovej konštrukcie, kde necháme servo otáčať pri zadaní uhla napríklad 45°, aby sa neotáčalo príliš rýchlo vyplníme tabuľku



```
function ServoStop () {
    wuKong.setServoAngel (wuKong.ServoList.S0, 90)
    wuKong.setServoAngel (wuKong.ServoList.S1, 90)
    wuKong.setServoAngel (wuKong.ServoList.S2, 90)
    wuKong.setServoAngel (wuKong.ServoList.S3, 90)
}
```

```
wuKong.setServoAngel (wuKong.ServoList.S0, 43)
basic.pause (500)
ServoStop ()
```

- Ľavé predné koleso - 180° maximum dopredu, 0° maximum dozadu
- Pravé predné koleso - 0° maximum dopredu, 180° maximum dozadu
- Ľavé zadné koleso - 180° maximum dopredu, 180° maximum dozadu
- Pravé zadné koleso - 0° maximum dopredu, 180° maximum dozadu

Podľa toho napíšeme funkcie a príklad ich volania, aby sa všetky kolesá pohybovali smerom dopredu



```
function PohybMotorov (lp: number, pp: number, lz: number, pz: number) {
    wuKong.setServoAngel (wuKong.ServoList.S0, 180 - lp)
    wuKong.setServoAngel (wuKong.ServoList.S1, pp)
    wuKong.setServoAngel (wuKong.ServoList.S2, 180 - lz)
    wuKong.setServoAngel (wuKong.ServoList.S3, pz)
}
```

```
function ServoStop () {
    wuKong.setServoAngel (wuKong.ServoList.S0, 90)
    wuKong.setServoAngel (wuKong.ServoList.S1, 90)
    wuKong.setServoAngel (wuKong.ServoList.S2, 90)
    wuKong.setServoAngel (wuKong.ServoList.S3, 90)
}
```

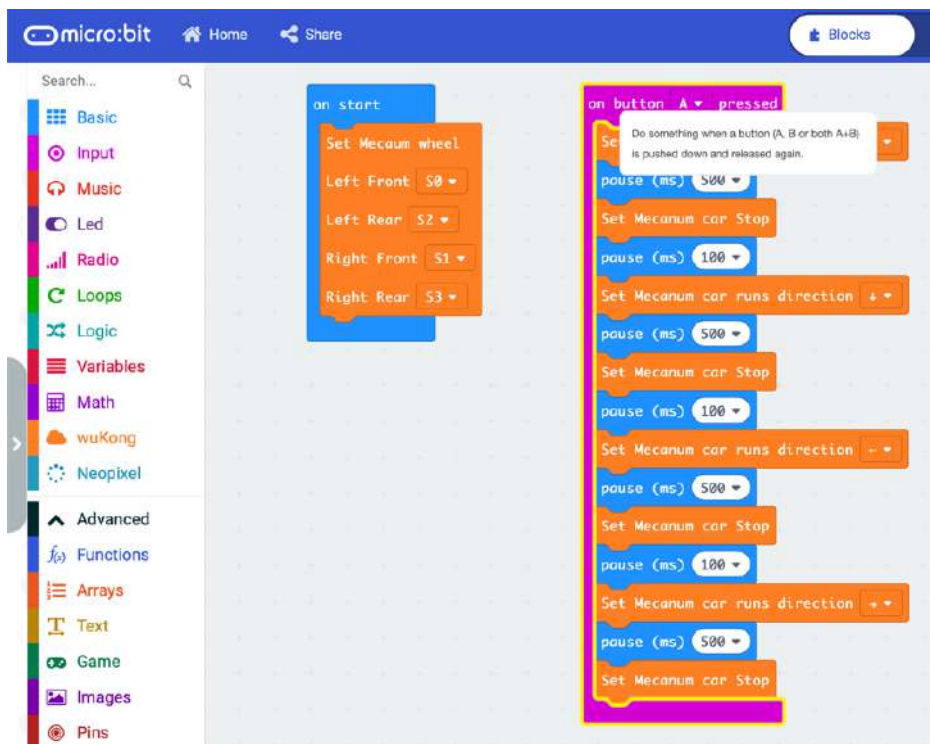
```
PohybMotorov (45, 45, 45, 45)
basic.pause (500)
ServoStop ()
```

Príklady využívajúce knižnicu Wukong s podporou pre kolesá Mecanum

Pripomenieme, že servomotory kolies sú v našej konštrukcii k doske Wukong pripojené takto:

- Ľavé predné koleso – S0
- Pravé predné koleso – S1
- Ľavé zadné koleso – S2
- Pravé zadné koleso – S3

Podľa toho je potrebné nastaviť inicializačnú funkciu v bloku on start. Vyskúšame pohyb všetkými smermi. Po spustení zistíte, že robotický podvozok pri použití funkcií left a right sa natočí požadovaným smerom



```
wuKong.mecanumWheel (  
wuKong.ServoList.S0,  
wuKong.ServoList.S2,  
wuKong.ServoList.S1,  
wuKong.ServoList.S3  
)
```

```
input.onButtonPressed(Button.A, function () {  
    wuKong.mecanumRun (wuKong.RunList.Front)  
    basic.pause (500)  
    wuKong.mecanumStop ()  
    basic.pause (100)  
    wuKong.mecanumRun (wuKong.RunList.rear)  
    basic.pause (500)  
    wuKong.mecanumStop ()  
    basic.pause (100)  
    wuKong.mecanumRun (wuKong.RunList.left)  
    basic.pause (500)  
    wuKong.mecanumStop ()  
    basic.pause (100)  
    wuKong.mecanumRun (wuKong.RunList.right)  
    basic.pause (500)  
    wuKong.mecanumStop ()  
})
```

Podľa toho aká je vaša konštrukcia veľká a ťažká je potrebné nastaviť pauzu po zastavení pohybu určitým smerom, aby sa podvozok prestal pohybovať zotrvačnosťou. Len vtedy budú na seba nadväzujúce pohyby presné. Na zvýšenie presnosti odporúčam občas očistiť gumené valčeky na kolesách.

V nasledujúcom príklade nahradíme funkcie left a right funkciami spin s parametrami left a right. Teras sa robotický podvozok bude pohybovať kolmo doprava a doľava, čiže pohybom, ktorý je snom všetkých, čo chcú zaparkovať do medzeru kam sa vaše auto zmestí s toleranciou niekoľkých centimetrov



```
wuKong.mecanumWheel (
wuKong.ServoList.S0,
wuKong.ServoList.S2,
wuKong.ServoList.S1,
wuKong.ServoList.S3
)

input.onButtonPressed(Button.A, function () {
    wuKong.mecanumSpin (wuKong.TurnList.Left)
    basic.pause (500)
    wuKong.mecanumStop ()
    basic.pause (100)
    wuKong.mecanumSpin (wuKong.TurnList.Right)
    basic.pause (500)
    wuKong.mecanumStop ()
})
```

V ďalšom príklade nahradíme funkciu spin funkciou drift, opäť ju spustíme postupne s parametrami left a right. Tentoraz sa bude robotický podvozok otáčať na mieste

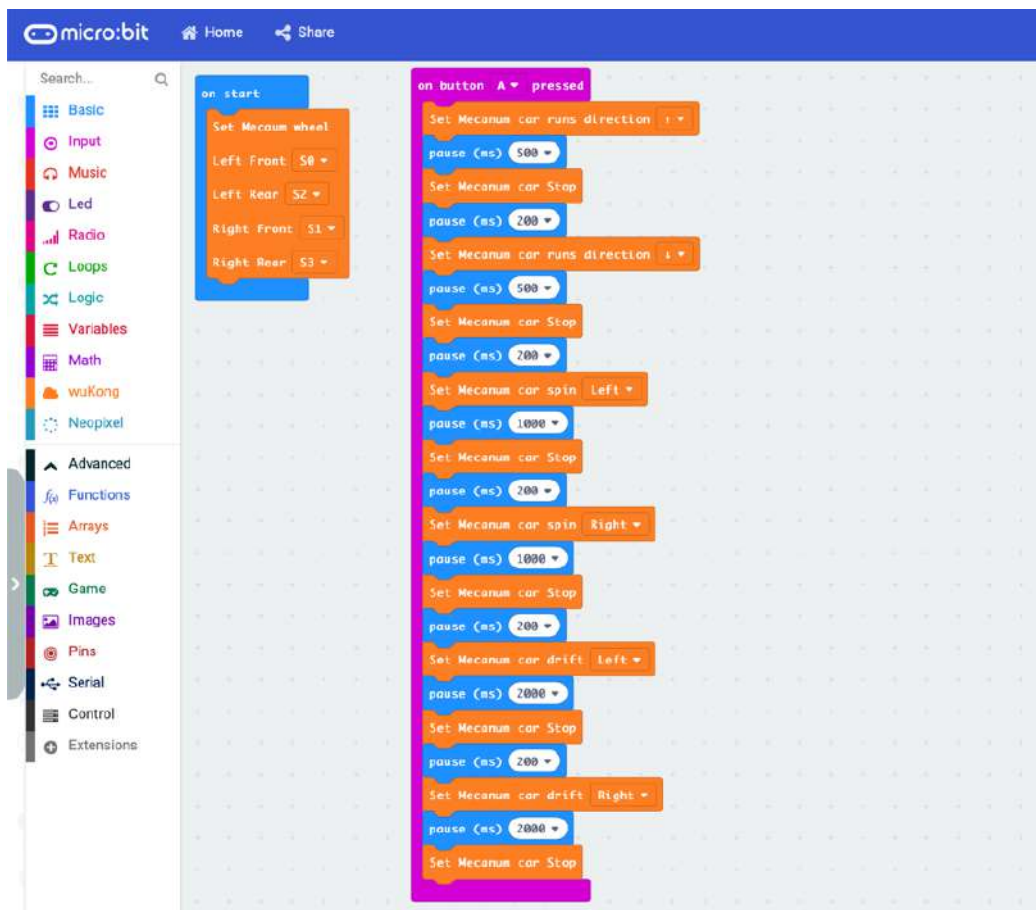


```

input.onButtonPressed(Button.A, function () {
  wuKong.mecanumDrift(wuKong.TurnList.Left)
  basic.pause(2000)
  wuKong.mecanumStop()
  basic.pause(200)
  wuKong.mecanumDrift(wuKong.TurnList.Right)
  basic.pause(2000)
  wuKong.mecanumStop()
})
wuKong.mecanumWheel(
wuKong.ServoList.S0,
wuKong.ServoList.S2,
wuKong.ServoList.S1,
wuKong.ServoList.S3
)

```

V záverečnom príklade skombinujeme pohyb dopredu, dozadu, kolmo do obidvoch strán a napokon rotáciu podvozku na mieste okolo svojej osi obidvomi smermi.



```

input.onButtonPressed(Button.A, function () {
  wuKong.mecanumRun (wuKong.RunList.Front)
  basic.pause (500)
  wuKong.mecanumStop ()
  basic.pause (200)
  wuKong.mecanumRun (wuKong.RunList.rear)
  basic.pause (500)
  wuKong.mecanumStop ()
  basic.pause (200)
  wuKong.mecanumSpin (wuKong.TurnList.Left)
  basic.pause (1000)
  wuKong.mecanumStop ()
  basic.pause (200)
  wuKong.mecanumSpin (wuKong.TurnList.Right)
  basic.pause (1000)
  wuKong.mecanumStop ()
  basic.pause (200)
  wuKong.mecanumDrift (wuKong.TurnList.Left)
  basic.pause (2000)
  wuKong.mecanumStop ()
  basic.pause (200)
  wuKong.mecanumDrift (wuKong.TurnList.Right)
  basic.pause (2000)
  wuKong.mecanumStop ()
})
wuKong.mecanumWheel (
wuKong.ServoList.S0,
wuKong.ServoList.S2,
wuKong.ServoList.S1,
wuKong.ServoList.S3
)

```

V našich jednoduchých príkladoch boli ukázané možnosti pohybu robotického podvozku s kolesami Mecanum. V príklade sme úmyselne nepoužili funkcie, ani cykly, to necháme na vašu tvorivosť keď si osvojíte základné pohyby. V budúcom pokračovaní ukážeme diaľkové ovládanie podvozku s kolesami Mecanum pomocou druhého Micro:bitu, ktorý bude komunikovať rádiovým s Micro:bitom v module Wukong, ktorý riadi robotický podvozok.

Micro:bit ako riadiaca jednotka populárneho robota mBot

Video <https://www.youtube.com/watch?v=JT2gJBT2jMU>

Najskôr sa budeme venovať originálnej zostave robota mBot s proprietárnou riadiacou jednotkou mCore, ktorá je kompatibilná s Arduino UNO, takže aplikácie pre robota môžete vytvárať v prostredí mBlock, alebo Arduino IDE. Na pohon robota slúžia dva jednosmerné motorčeky s prevodovkami. Motorčeky nemajú spätnú väzbu, takže riadiaca jednotka nedokáže zistiť uhol pootočenia. Na riadiacej jednotke sú dva konektory typu JST na pripojenie motorov a štyri konektory typu RJ-25 aké sa používajú aj na pripojenie telefónnej linky. Riadiaca jednotka má aj konektor na pripojenie komunikačného modulu. Stavebnica sa dodáva v dvoch variantoch s modulom Bluetooth, takže robot je možné ovládať a programovať zo smartfónu, alebo tabletu, alebo s modulom rádiovkej komunikácie v pásme 2,4 GHz, ktorý komunikuje s USB dongle, ktorý sa zasunie do PC, takže robot sa programuje z počítača. Súčasťou stavebnice je aj diaľkové ovládanie infračervený modul na sledovanie čiary a ultrazvukový merač vzdialenosti.

Riadiaca jednotka mCore je predprogramovaná na tri úlohy, ktoré sa aktivujú tlačidlami:

A – manuálne ovládanie (nedajú sa použiť dva tlačidlá smeru súčasne)

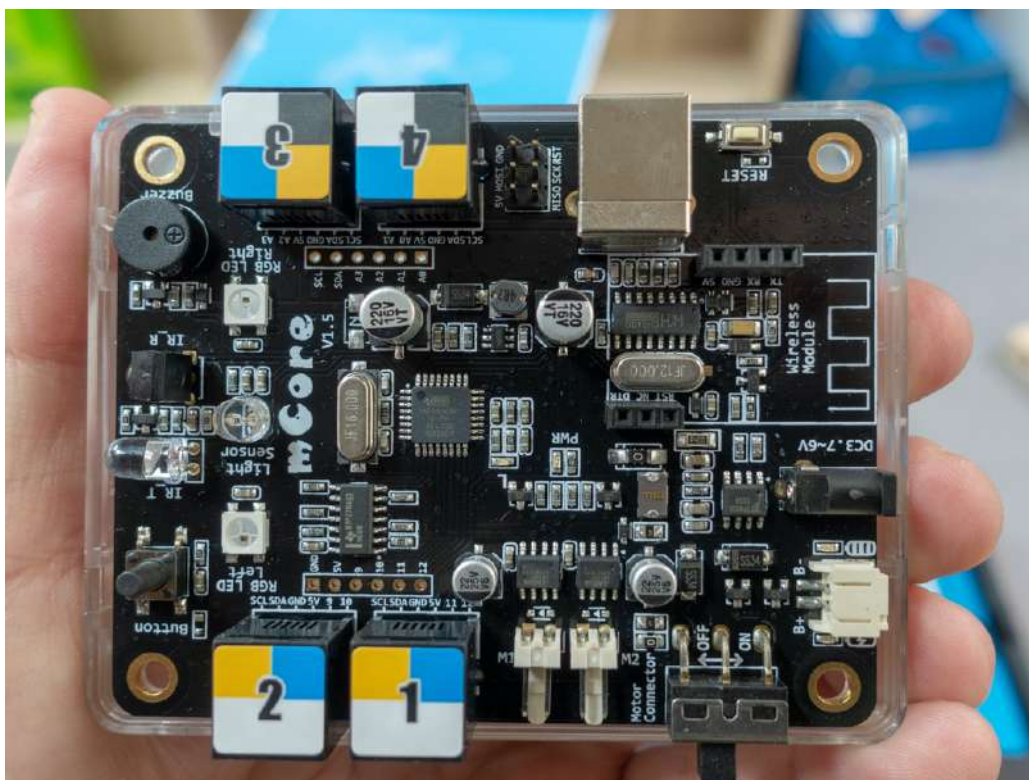
B – vyhýbanie sa prekážkam

C – sledovanie čiary (plagát s čiarou v tvare osmičky je pribalený)

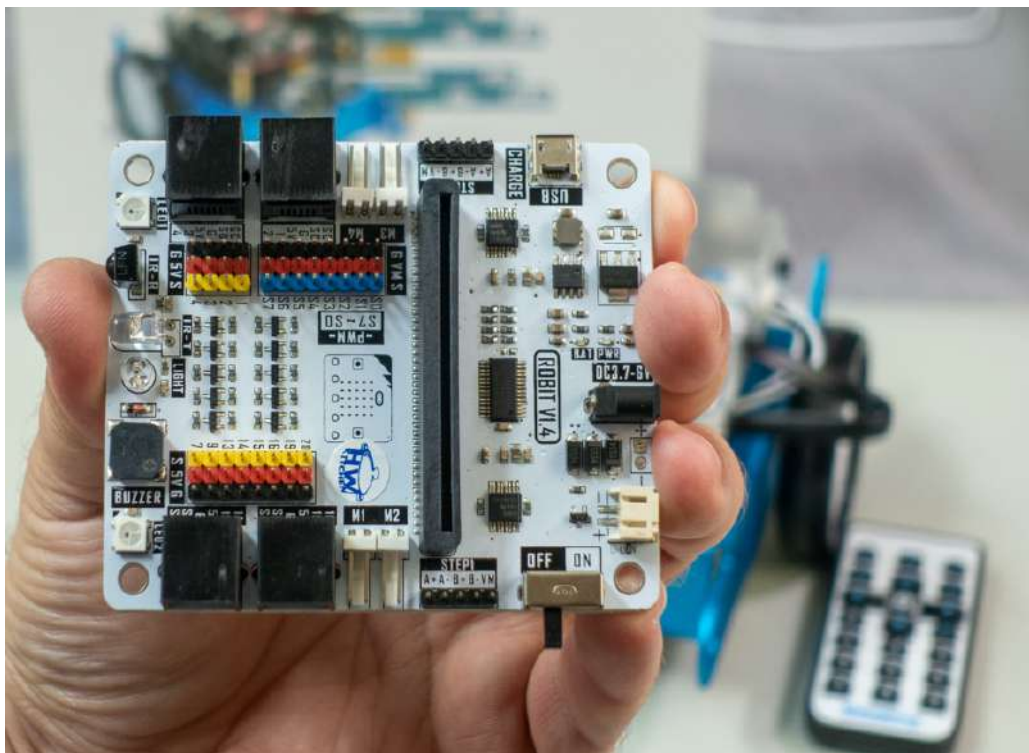
Po naprogramovaní vlastnej aplikácie sa táto preloží do binárneho kódu, ktorý sa zavedie do procesora.



Súčasťou robotického stavebnice mBot od firmy MakeBlock je riadiaca jednotka s procesorom ATmega 328 kompatibilná s Arduino.

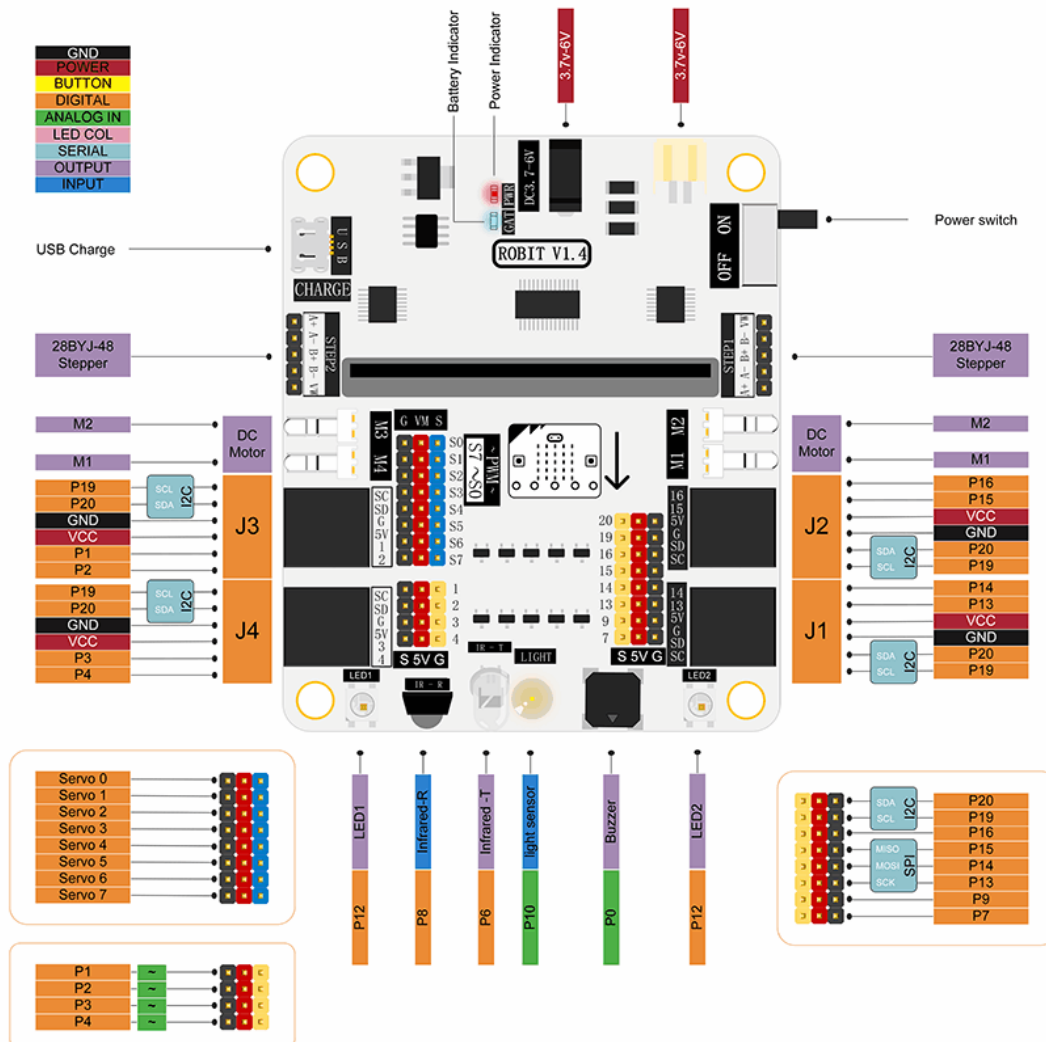


K dispozícii je aj modul, ktorý ako alternatívnu riadiacu jednotku umožňuje použiť modernú a veľmi populárnu mikropočítačovú dosku BBC Micro:bit, používanú v mnohých našich školách na výučbu programovania a robotiky



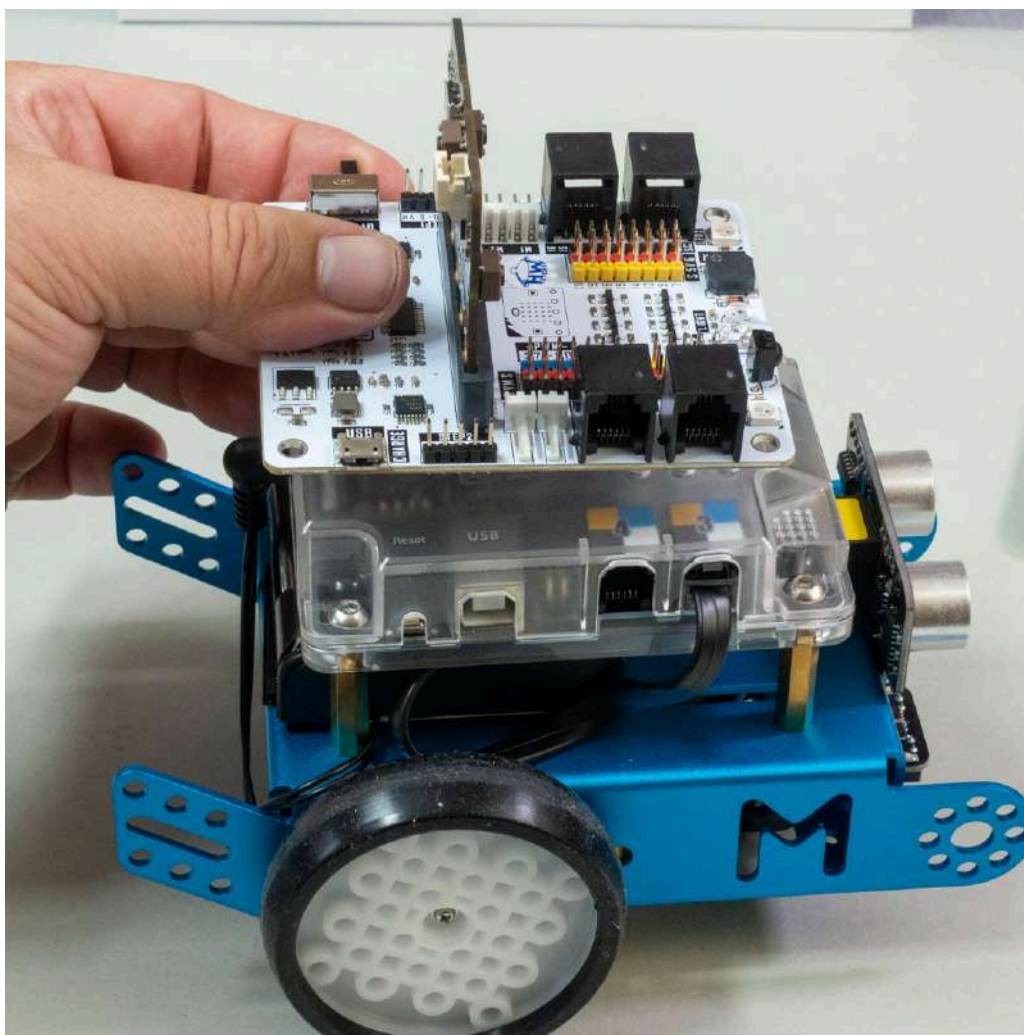
Pomocou dosky ROBIT od firmy Elecfreaks je možné použiť ako riadiacu jednotku dosku Micro.Bit, ktorá má mikrokontrolér s oveľa väčším výkonom a kapacitou pamäte. Dosku samozrejme môžete použiť aj ako riadiacu jednotku vlastného robota, prípadne robota z akejkoľvek stavebnice

Modul ROBIT má rovnaké konektory ako originálna riadiaca jednotka mCore a je s ňou aj mechanicky kompatibilný vrátane rozmerov a polohy upevňovacích otvorov. Navyše má dva ďalšie konektory pre jednosmerné motory, plus dva konektory pre krokové motorčeky. Ďalej modul obsahuje 8 konektorov typu GVS (Ground-Voltage-Signal), ktoré majú PWM signály na ovládanie servomotorov. Ďalších 8 GVS konektorov GVS je na pripojenie modulov a snímačov. Tieto porty umožňujú pripojenie aj 5 V logiky. Piny P1, P2, P3 a P4 sú špeciálne určené len na čítanie dát z 5 V analógových senzorov.



Porovnanie konektorov dosky ROBIT s originálnou riadiacou jednotkou mCore

Na doske je umiestnený aj reproduktor, RGB LEDky, snímač osvetlenia, IR prijímač / vysielač, napájací konektor a samozrejme konektor pre dosku Micro:bit. Tá sa do modulu ROBIT vkladá kolmo a jej LED displej, ktorý umožňuje aj snímanie intenzity osvetlenia smeruje dopredu.

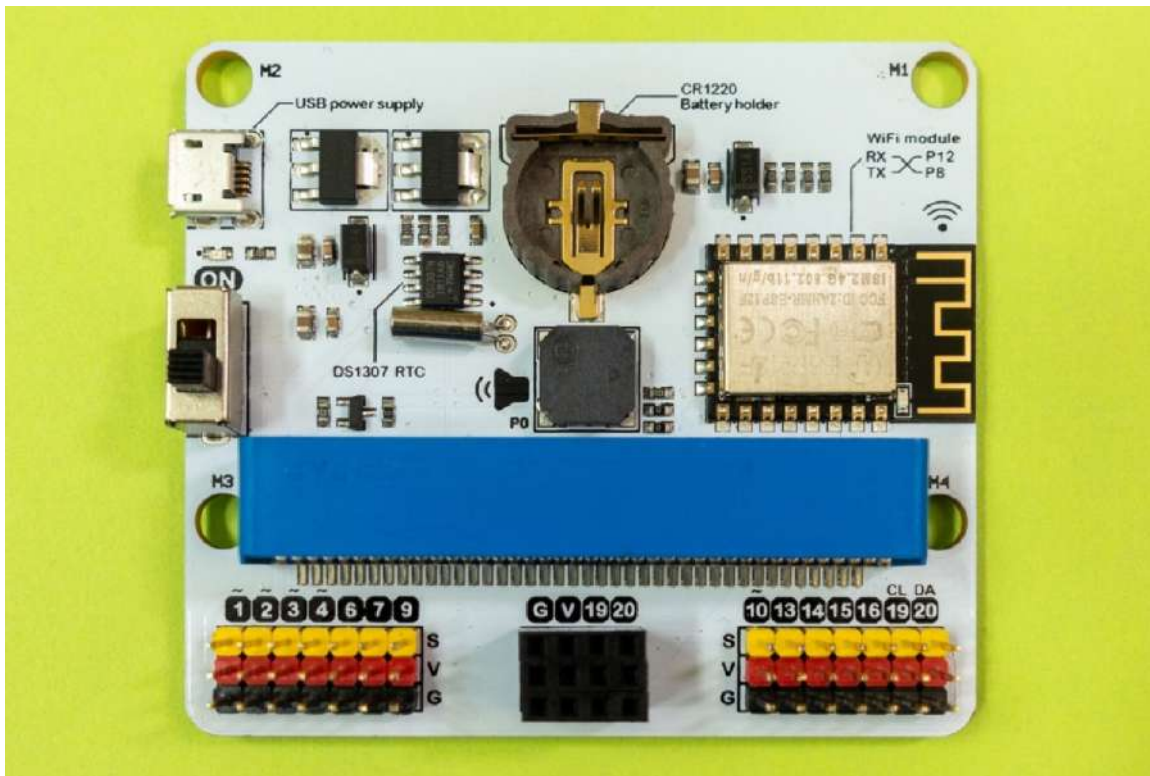


Robot sa programuje na rovnakom princípe ako dvojkoľosový robotický podvozok zo stavebnice Ring:bit

Pripojenie Micro:bitu na WiFi a komunikácia s cloudovou IoT službou

Populárna mikropočítačová doska Micro:bit má na čipe integrovaný modul na bezdrôtovú komunikáciu, avšak ten umožňuje „len“ prepojenie viacerých (aj niekoľko desiatok) dosiek Micro:bit medzi sebou, takže je možné napríklad vytvoriť viacznakový displej tak, že každý Micro:bit zobrazuje jeden znak. Prípadne môžete rádiový modul nakonfigurovať tak, aby ste mohli komunikovať zo smartfónom cez bluetooth. **Pripojením na WiFi doska Micro:bit nedisponuje.**

K dispozícii je však modul IoT: bit pre Micro:bit, ktorý umožňuje pripojenie k WiFi, takže z Micro:bitu sa stane plnohodnotná platforma pre internet vecí (IoT). Budete môcť na diaľku ovládať rôzne zariadenie alebo merať požadované veličiny a posilať ich do cloudu na ďalšie spracovanie.

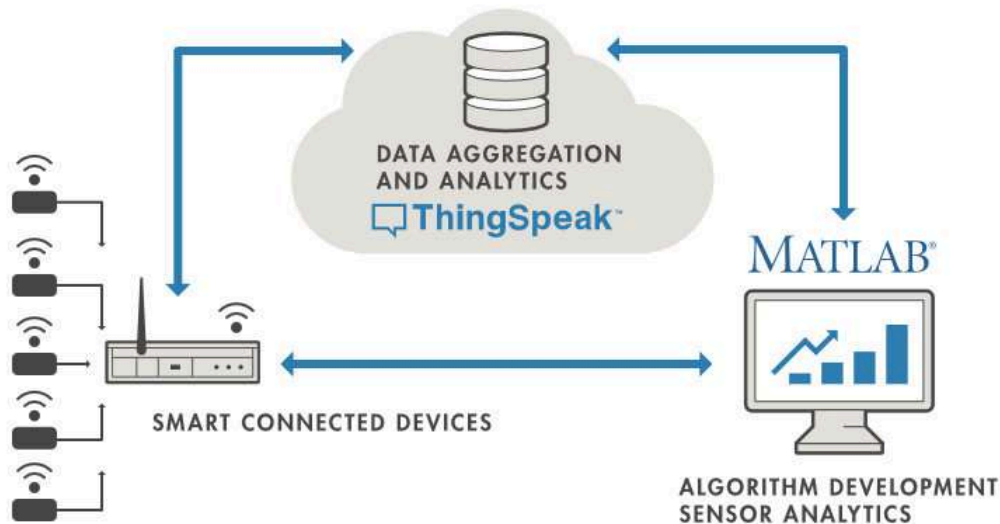


Takže v reálnom IoT riešení budete riešiť tri okruhy problémov: „čo budete posilať“, „ako to budete posilať“ a „kam to budete posilať“.

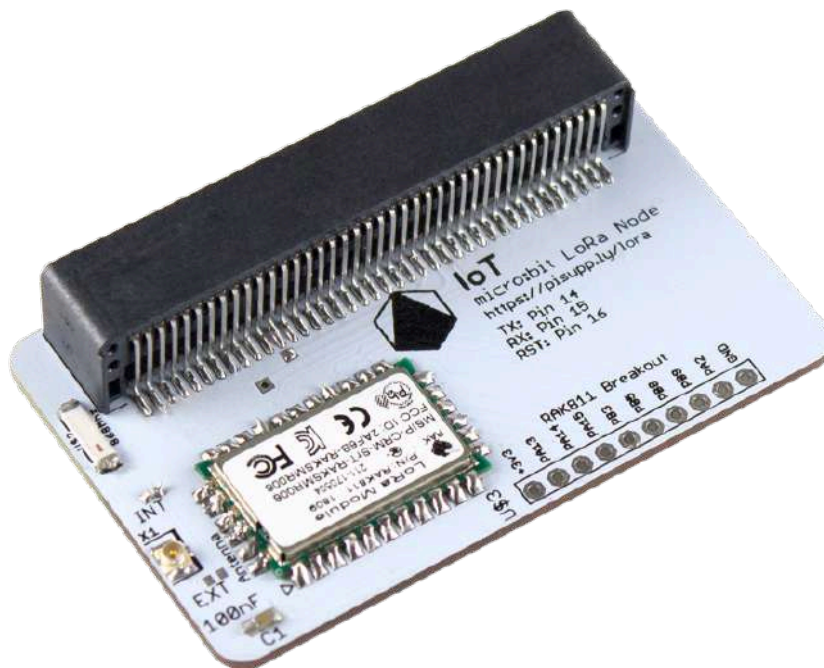
Čo – potrebujete získať požadované údaje, napríklad pomocou senzorov pripojených na zbernicu mikropočítačovej dosky.

Ako – potrebujete vytvoriť pripojenie mikropočítačovej dosky k internetu

Kam – potrebujete mať používateľský účet vo vhodnej cloudovej službe, kde sa budú údaje spracovávať, ukladať a analyzovať



V prvých príkladoch môžete pre jednoduchosť reálne údaje simulovať napríklad generátorom náhodných čísel nastaveným na vhodný rozsah. Po odladení môžete začať posilať reálne údaje napríklad z integrovaného akcelerometra, čidla na meranie teploty, ktoré meria teplotu mikrokontroléra, alebo z externých senzorov.



K dispozícii je aj modul pre IoT sieť LoRa

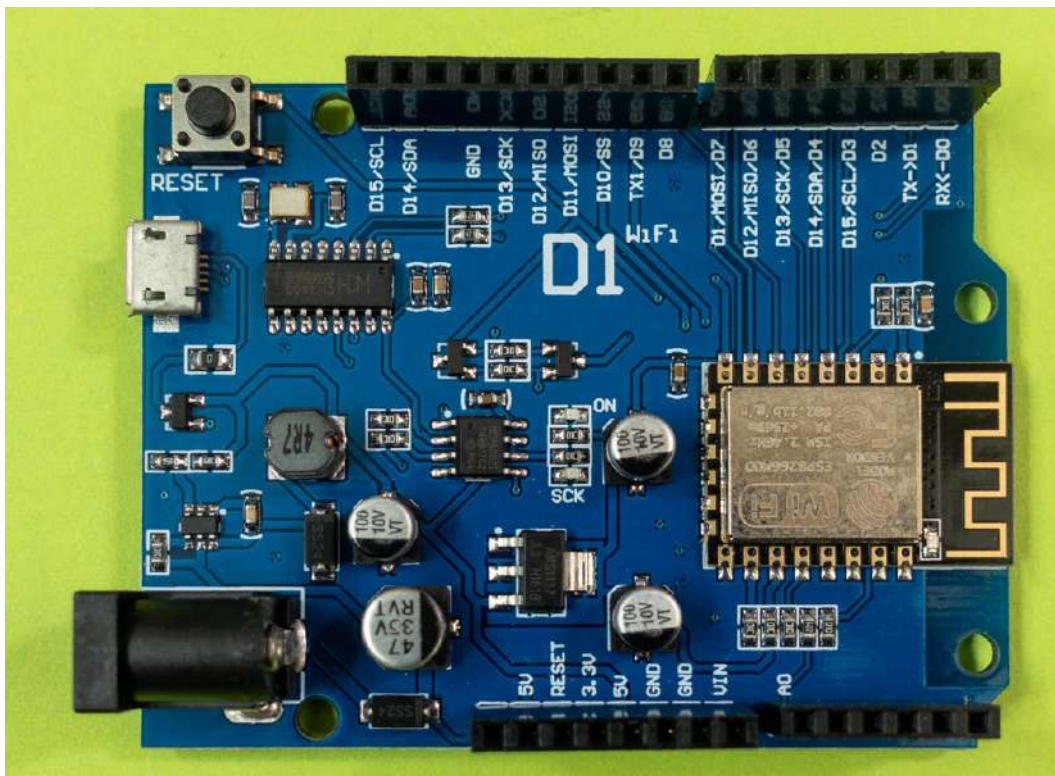
Na pripojenie do internetu použijeme modul IoT: bit pre Micro:bit, ktorý umožňuje pripojenie cez integrovaný WiFi modul ESP12F. Tento modul využíva populárny obvod ESP8266, ktorý mimochodom je možné samostatne programovať a využiť tak WiFi modul na riadenie jednoduchých zariadení.

V krátkosti pripomenieme, že ESP8266 je 32bitový mikrokontrolér s RISC jadrom, ktorý je primárne určený na doplnenie Wi-Fi konektivity k existujúcim zariadeniam riadeným inými mikroprocesormi,

ktoré komunikujú cez sériové rozhranie pomocou AT príkazov. To je presne náš prípad, kedy ESP8266 zabezpečuje WiFi konektivitu pre Micro:bit.

Platforma ESP

My to v príkladoch v tejto publikácii nevyužijeme, ale je užitočné vedieť, že ESP8266 vďaka svojim pinom rozhraní môže aj sám fungovať ako mikrokontrolér. Možnosť bezdrôtovej komunikácie, cena čipov pod jeden dolár a vývojových dosiek s cenou 3 – 5 dolárov a taktiež možnosť programovať tento mikrokontrolér vo vývojovom prostredí pre Arduino rýchlo vystrelilo tento čip na vrchol rebríčkov popularity u hobby konštruktérov. Stačí doplniť firmvér pre WiFi komunikáciu o ovládanie vstupno – výstupných pinov a môžete ovládať rôzne konštrukcie a zariadenia. Do ESP8265 sa dá nahráť aj interpretér jazyka Micro Python a potom môžete túto dosku programovať aj v Pythone. Čip ESP8266 využíva štandardne taktováciu frekvenciu 80 MHz, ale úspešne funguje aj pri takmer dvojnásobnom pretaktovaní, samozrejme za cenu vyššej spotreby. Tento čip však vie byť aj veľmi energeticky úsporný, samozrejme program musí správne využívať prechody do režimov s nízkou spotrebou. K dispozícii je 11 digitálnych vstupov a výstupov a 1 analógový vstup s 10 bitovým prevodníkom. Takmer všetky piny majú alternované funkcie pre komunikačné rozhrania. K dispozícii sú komunikačné rozhrania 2 x SPI, I2C, 2 x UART

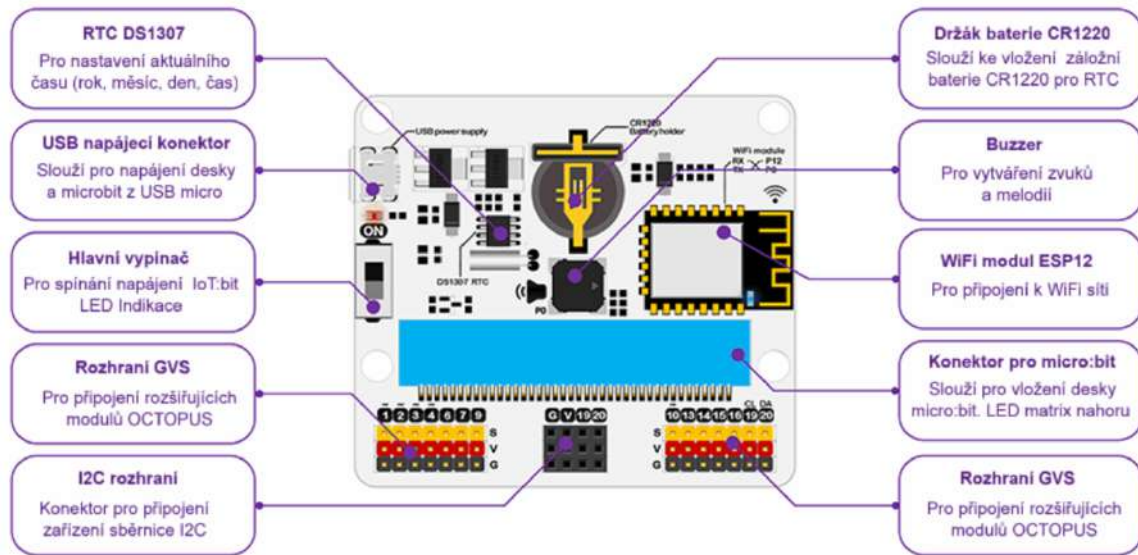


K dispozícii je veľa dosiek s obvodom ESP8266, napríklad aj klon Arduina s názvom D1

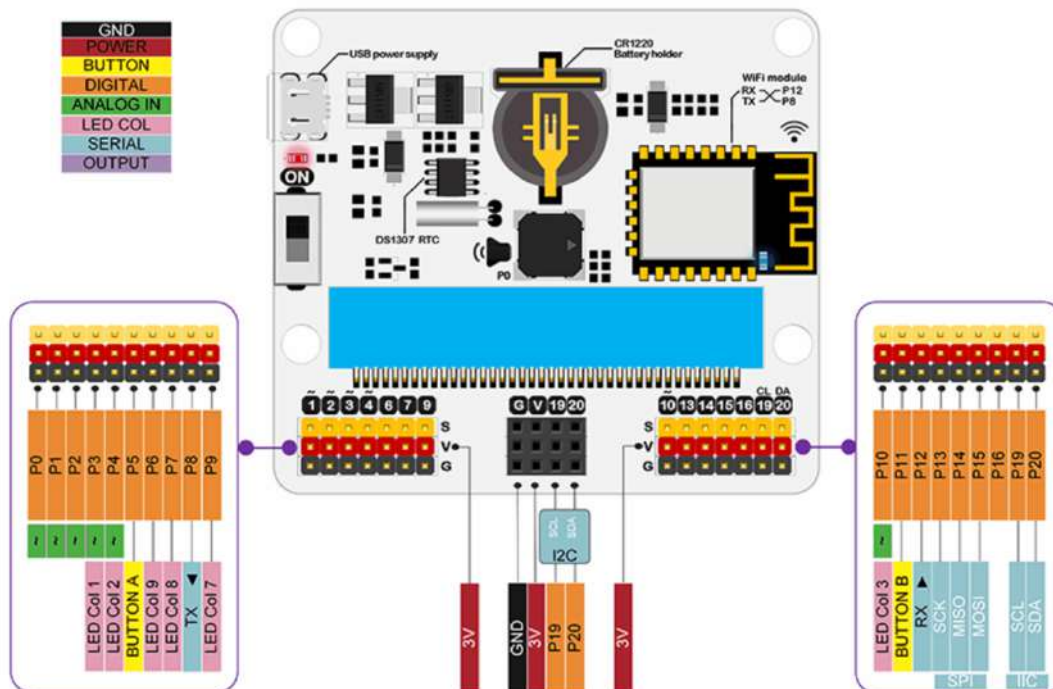
Na uloženie programu je k dispozícii kapacita pamäti typu flash 4 MB. Kapacita RAM je 96 kB, avšak na program sa dá využiť menej ako polovica, pretože zvyšok využíva firmvér pre WiFi. Aplikácie sa dajú programovať v jazyku C, Pythone, prípadne v programovacom jazyku Lua. Pri vývoji kódu si musíte uvedomiť, že sa o mikrokontrolér delíte s WiFi modulom preto je nutné stále odovzdávať riadenie programu firmvéru pre WiFi. Tento je implementovaný ako BLOB, čiže ako blok binárneho kódu, takže nemáte možnosť ho modifikovať. Od tejto starosti vás však aspoň spočiatku, kým programovanie ESP 8266 nezvládnete bravúrne odbremení knižnice.

Modul IoT: bit pre Micro:bit

Vráťme sa však k modulu IoT: bit pre Micro:bit, ktorý má ešte jednu dôležitú funkciu obvod reálneho času. Už pri prvom pohľade na dosku zistíte, že má držiak na miniatúrnu batériu. Tá nenapája Micro:bit, ale obvod reálneho času DS1307 RTC. Batéria CR1220 je potrebná len ak chcete zálohovať obvod reálneho času. Ostatné funkcie dosky vrátane WiFi pripojenia fungujú aj bez nej.



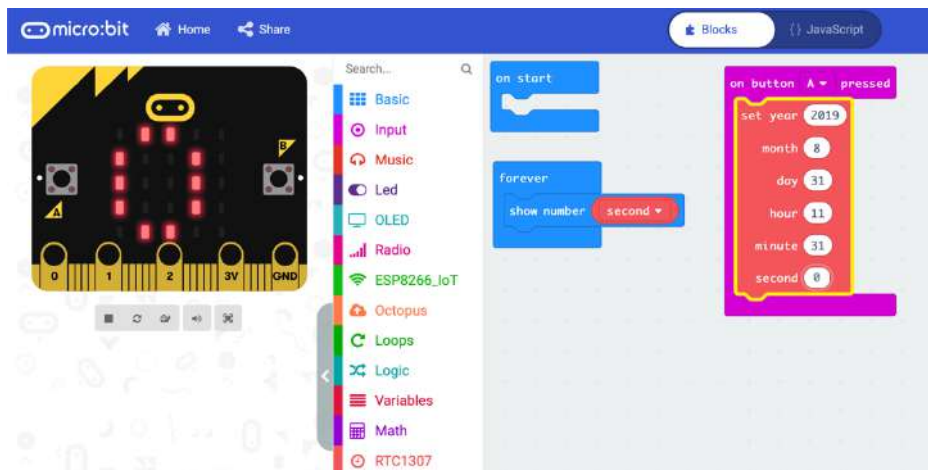
Na dosce je aj bzučiak, konektory GVS a taktiež konektor zbernice I2C na pripojenie displeja, snímača barometrického tlaku BME280, prípadne iných I2C zariadení. Montážne otvory na skrutky sú kompatibilné s platformou LEGO Technic 7 x 5.



Piradenie portov na GVS vstupy dosky IoT:bit

Aby ste mohli využiť funkcie modulu IoT: bit pre Micro:bit, je potrebné do online vývojového prostredia MakeCode doplniť rozširujúcu knižnicu Enviroment-and-Science-IoT.

V prvom príklade otestujeme modul reálneho času



```
input.onButtonPressed(Button.A, function () {
  RTC_DS1307.DateTime(
    2019,
    8,
    31,
    11,
    31,
    0
  )
})

basic.forever(function () {
  basic.showNumber(RTC_DS1307.getTime(RTC_DS1307.TimeType.SECOND))
})
```

Cloudová platforma ThingSpeak

Skôr než začnete posielať cez WiFi údaje, musíte si vytvoriť konto vo vhodnej cloudovej platforme, kam budete údaje posielať. Knižnica pre modul IoT :bit podporuje cloudovú platformu ThinkSpeak <https://thingspeak.com>.



 **Collect**
Send sensor data privately to the cloud.

 **Analyze**
Analyze and visualize your data with MATLAB.

 **Act**
Trigger a reaction.

Na bežné účely vám stačí bezplatné konto. To umožňuje poslať ročne až 3 milióny správ, čo je 8 200 správ denne. Interval na update správy je 15 sekúnd. Môžete využívať 4 kanály a 3 MQTT subskripcie. Na bežné projekty sú to viac než dostatočné parametre

	FREE For small non-commercial projects	HOME For personal use only ⁽¹⁾
Scalable for larger projects	✘ No. Annual usage is capped.	✔
Number of messages	3 million/year (~8 200/day) ⁽²⁾	33 million/year per unit (~90 000/day per unit) ⁽²⁾
Message update interval limit	Every 15 seconds	Every second
Number of channels	4	10 per unit
MATLAB Compute Timeout	20 seconds	20 seconds
Number of simultaneous MQTT subscriptions	Limited to 3	50 per unit
Private channel sharing	Limited to 3 shares	Unlimited
Technical Support	Forum	Forum

Porovnanie bezplatného konta a konta pre domácich používateľov. Cena závisí od objemu predplatených služieb

Po úspešnej registrácii si vytvoríte nový kanál a vhodne ho pomenujete, napríklad Microbit meranie 1. V rovnakom formulári ešte vyplníte názvy jednotlivých polí, napríklad: Teplota °C, Zrychlenie X a podobne. Po potvrdení sa zobrazí Write API key, ktorý použijete ako parameter v príkaze `setdata()`. Postup bude ukázaný v nasledujúcich príkladoch.

The screenshot shows the ThingSpeak website interface. At the top, there is a navigation bar with links for Channels, Apps, Community, Support, Commercial Use, How to Buy, Sign In, and Sign Up. The main heading is "Sign up for ThingSpeak". Below this, there is a paragraph explaining the free account options and a note about MathWorks accounts. The "Create MathWorks Account" form is visible, with fields for Email Address (llacko@pcrevue.sk), Location (Slovakia), First Name (Luboslav), and Last Name (Lacko). To the right of the form is a diagram illustrating the data flow: "SMART CONNECTED DEVICES" send data to a cloud labeled "DATA AGGREGATION AND ANALYTICS ThingSpeak". This cloud then connects to a "MATLAB" computer, which is used for "ALGORITHM DEVELOPMENT SENSOR ANALYTICS".

Príklady - meranie a posielanie údajov do cloudu ThingSpeak

Na praktických príkladoch ukážeme ako sa dajú merať rôzne veličiny a poselať údaje do cloudu. Využijeme interné senzory na doske Micro:bit. Tie umožňujú merať teplotu pomocou senzoru priamo na čipe, zrýchlenie pomocou akcelerometra a dokonca aj intenzitu osvetlenia s využitím diód maticového displeja. Podľa skúseností je teplota okolia spravidla o 4 °C vyššia než teplota okolia.

V prvom príklade budeme merať teplotu a intenzitu osvetlenia a následne poselať namerané údaje do cloudu, v druhom zrýchlenie v osiach X, Y a Z. Môžete samozrejme využiť aj externé senzory na presné meranie teploty, intenzity osvetlenia, prípadne ďalších parametrov. Môžete využiť napríklad senzory zo súpravy Smart Science IoT Kit, prípadne aj zo súpravy Smart home Kit, ktoré sa pomocou trojvodičového kábla pripájajú na porty GVS (Ground, Voltage, Signal) dosky IoT:bit. Na zobrazovanie použijeme OLED displej z niektorej zo spomínaných súprav, pripojený k doske IoT:bit cez rozhranie I2C. Pripojenie displeja nie je pre fungovanie príkladov potrebné.

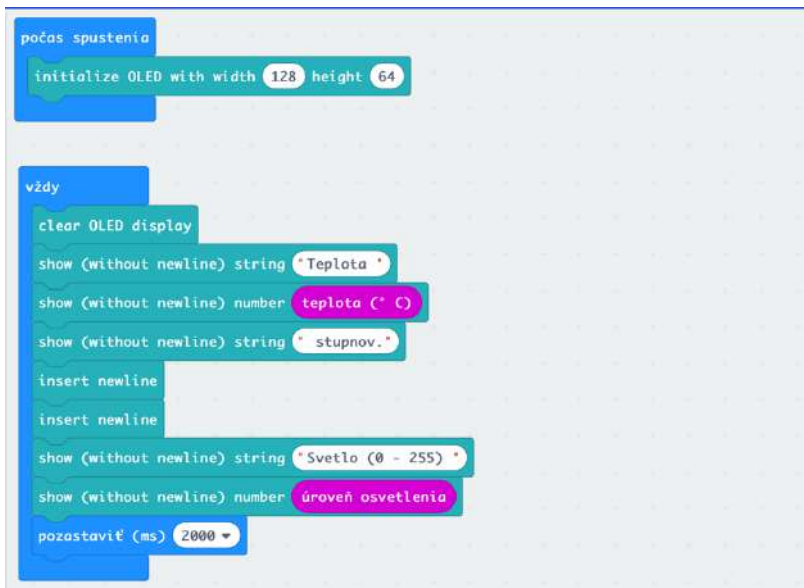


Pripomíname, že aby ste mohli využiť funkcie modulu IoT: bit pre Micro:bit, je potrebné do online vývojového prostredia MakeCode doplniť rozširujúcu knižnicu Enviroment-and-Science-IoT. Tá obsahuje aj podporu OLED displeja pripojeného cez I2C. Pri vyhľadávaní rozšírení stačí zadať IoT.

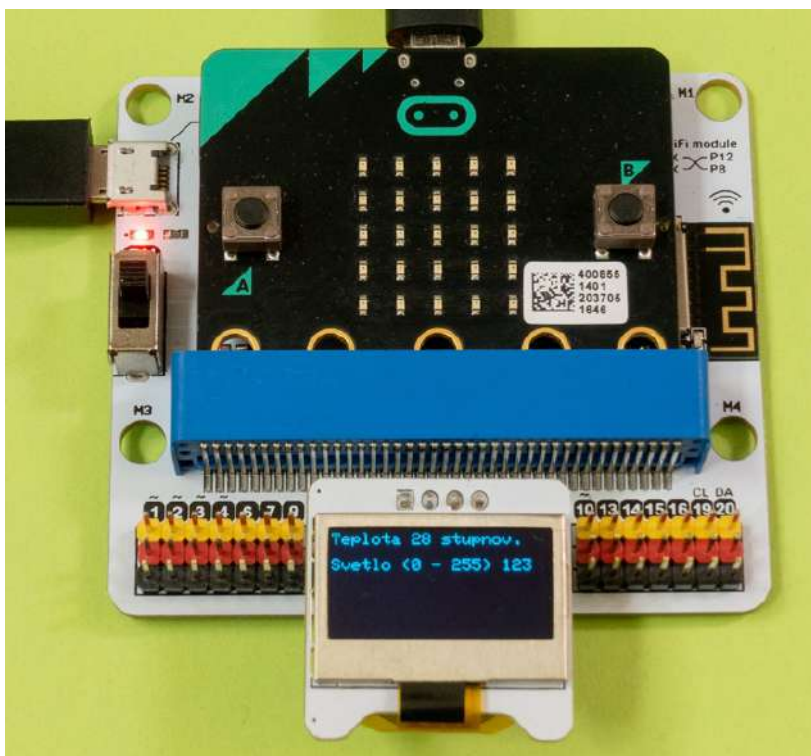
Zobrazovanie na OLED displeji (ak nemáte displej môžete vynechať)

Začneme meraním a rozšírením možnosti zobrazovania pri ladení. Maticový jednoznakový displej s možnosťou rolovania textu je efektný, no na ladenie nie príliš praktický. Preto ako prvý krok (ktorý je možné vynechať) pripojíme k doske IoT:bit displej cez rozhranie I2C. Použijeme displej zo súpravy Smart home Kit, zasunutý do štvorpinového konektora dosky IoT bit. Na generovanie signálu SCL sa využíva port P19 a na generovanie signálu SDA port P20.

Nasledujúci kód zobrazí na pripojenom OLED displeji údaje o teplote a intenzite osvetlenia získané zo senzorov, ktoré sú priamo na doske Micro:bit.



```
OLED.init(128, 64)
basic.forever(function () {
  OLED.clear()
  OLED.writeString("Teplota ")
  OLED.writeNum(input.temperature())
  OLED.writeString(" stupnov.")
  OLED.newLine()
  OLED.newLine()
  OLED.writeString("Svetlo (0 - 255) ")
  OLED.writeNum(input.lightLevel())
  basic.pause(2000)
})
```



Primárnym zámerom príkladov je meranie údajov a ich posielanie do cloudu. Zobrazovanie hodnôt na displeji je len sekundárnou funkcionalitou, ktorú ak nemáte k dispozícii OLED displej pripojiteľný cez zbernicu I2C nemusíte riešiť. Vytvoríme na tento účel funkciu, ktorej odovzdáme údaje o teplote a intenzite osvetlenia. Ak nemáte displej tak definíciu funkcie zobrazHodnoty a jej volanie v hlavnej sľučke

```
basic.forever(function () {
  ...
  zobrazHodnoty(nTeplota, nSvetlo)
  ...
})
```

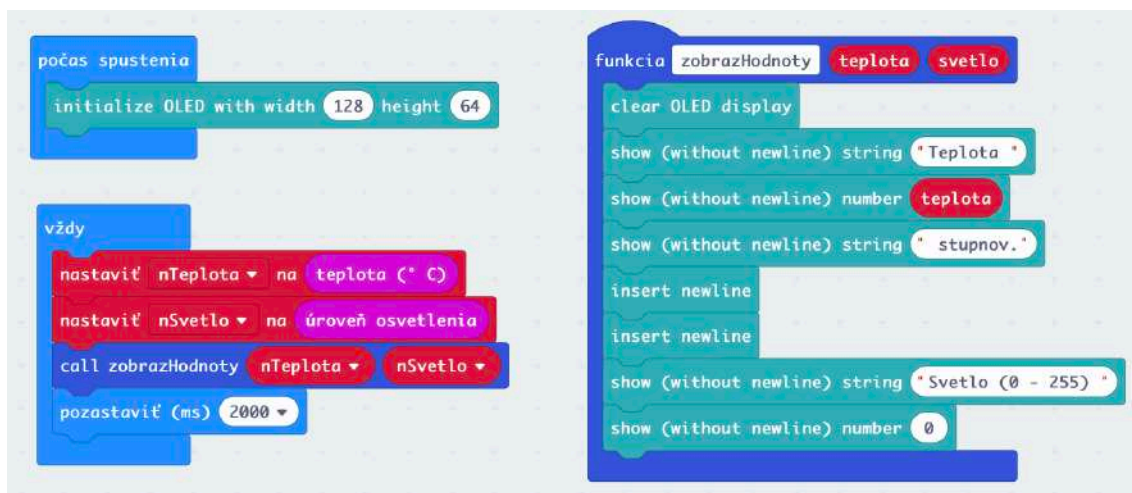
môžete vynechať

Edit Function



Dialóg na vytvorenie funkcie

Modifikácia predchádzajúceho príkladu s využitím funkcie na zobrazovanie nameraných údajov



```
function zobrazHodnoty (teplota: number, svetlo: number) {
  OLED.clear()
  OLED.writeString("Teplota ")
  OLED.writeNum(teplota)
  OLED.writeString(" stupnov.")
  OLED.newLine()
  OLED.newLine()
  OLED.writeString("Svetlo (0 - 255) ")
  OLED.writeNum(0)
}
```

```

let nSvetlo = 0
let nTeplota = 0
OLED.init(128, 64)

basic.forever(function () {
  nTeplota = input.temperature()
  nSvetlo = input.lightLevel()
  zobrazHodnoty(nTeplota, nSvetlo)
  basic.pause(2000)
})

```

Na displeji by sa mala zobraziť teplota v stupňoch Celzia približne o 4 stupne väčšia, než teplota v miestnosti a úroveň osvetlenia v intervale číselných hodnôt 0 – 255.

Posielanie údajov do cloudu

Po úspešnej registrácii v cloudovej platforme Thing Speak i si vytvoríte nový kanál a vhodne ho pomenujete, napríklad Microbit meranie interne. V rovnakom formulári ešte vyplníte názvy jednotlivých polí, v našom prípade Teplota °C a Svetlo (0 - 255)

ThingSpeak™ Channels Apps Support Commercial Use How to Buy Account Sign Out

New Channel

Name: Microbit meranie interne

Description:

Field 1: Teplota (°C)

Field 2: Svetlo (0-255)

Field 3:

Field 4:

Field 5:

Field 6:

Field 7:

Field 8:

Metadata:

Tags:

Help

Channels store all the data that a ThingSpeak application collects. Each channel includes eight fields that can hold any type of data, plus three fields for location data and one for status data. Once you collect data in a channel, you can use ThingSpeak apps to analyze and visualize it.

Channel Settings

- Channel Name:** Enter a unique name for the ThingSpeak channel.
- Description:** Enter a description of the ThingSpeak channel.
- Field#:** Check the box to enable the field, and enter a field name. Each ThingSpeak channel can have up to 8 fields.
- Metadata:** Enter information about channel data, including JSON, XML, or CSV data.
- Tags:** Enter keywords that identify the channel. Separate tags with commas.
- Link to External Site:** If you have a website that contains information about your ThingSpeak channel, specify the URL.
- Show Channel Location:**
 - Latitude:** Specify the latitude position in decimal degrees. For example, the latitude of the city of London is 51.5072.
 - Longitude:** Specify the longitude position in decimal degrees. For example, the longitude of the city of London is -0.1275.
 - Elevation:** Specify the elevation position meters. For example, the elevation of the city of London is 35.052.
- Video URL:** If you have a YouTube™ or Vimeo® video that displays your channel information, specify the full path of the video URL.
- Link to GitHub:** If you store your ThingSpeak code on GitHub®, specify the GitHub repository URL.

Po potvrdení sa prepnete do záložky API Keys. Tu nájdete identifikátor Write API key, ktorý použijete v programe pre Micro:bit ako parameter v príkaze setdata()

Microbit meranie interne

Channel ID: 862722
 Author: lubolavko
 Access: Private

Private View Public View Channel Settings Sharing API Keys Data Import / Export

Write API Key

Key ZHUTMJ252[REDACTED]

Generate New Write API Key

Read API Keys

Key SDU1GD3MVNQAQ3XP

Note

Help

API keys enable you to write data to a channel or read data from a private channel. API keys are auto-generated when you create a new channel.

API Keys Settings

- **Write API Key:** Use this key to write data to a channel. If you feel your key has been compromised, click **Generate New Write API Key**.
- **Read API Keys:** Use this key to allow other people to view your private channel feeds and charts. Click **Generate New Read API Key** to generate an additional read key for the channel.
- **Note:** Use this field to enter information about channel read keys. For example, add notes to keep track of users with access to your channel.

API Requests

Write a Channel Feed

```
GET https://api.thingspeak.com/update?api_key=ZHUTMJ2523RN29
NT&field1=0
```

Teraz vytvoríme základnú šablónu kódu na posielanie údajov do cloudovej platformy ThingSpeak. Ako prvý krok je potrebné iniciovať pripojenie Micro:bitu k WiFi sieti. Je potrebné zadať porty priradené signálom RX a TX. Priradenie nájdete v pravom hornom rohu dosky IoT:bit. Signál RX je pripojený na port P8 a TX na port P12.

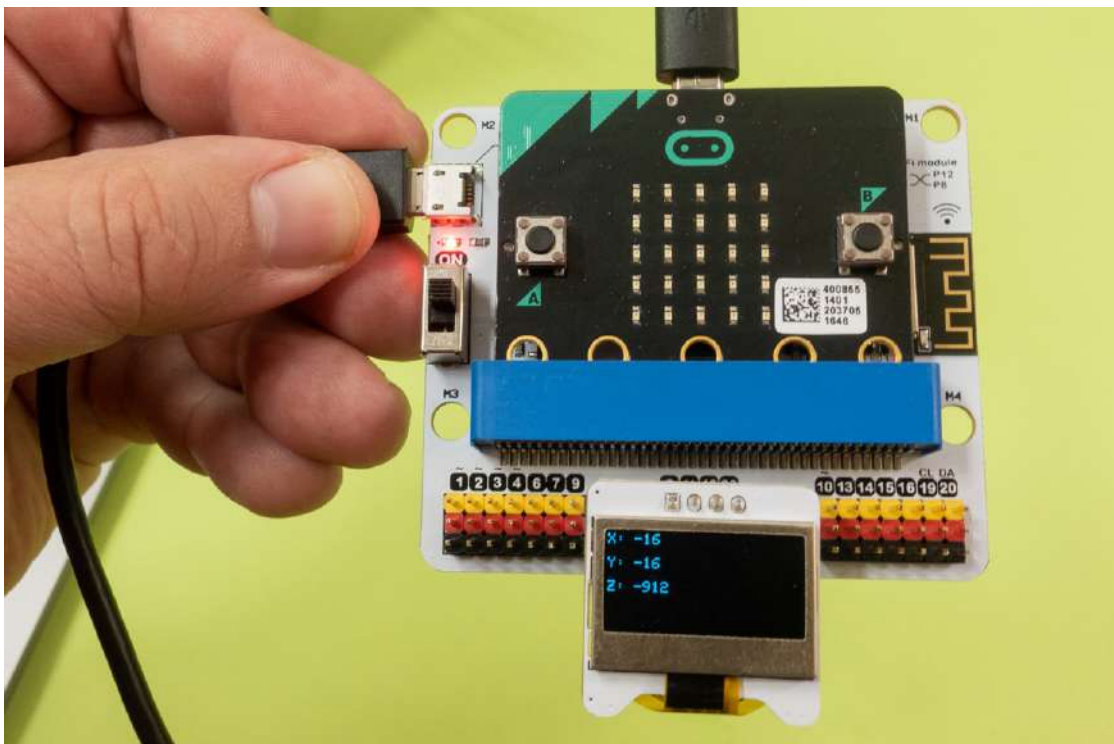
Ako parametre funkcie connectWifi sa zadáva identifikátor siete a prístupové heslo. Nasleduje pripojenie do cloudovej platformy ThingSpeak, vytvorenie správy a jej odoslanie. Správa obsahuje api_key a osem údajov. Samozrejme nemusíte využiť všetkých osem, môžete posilať len napríklad dve ako v tomto príklade



```

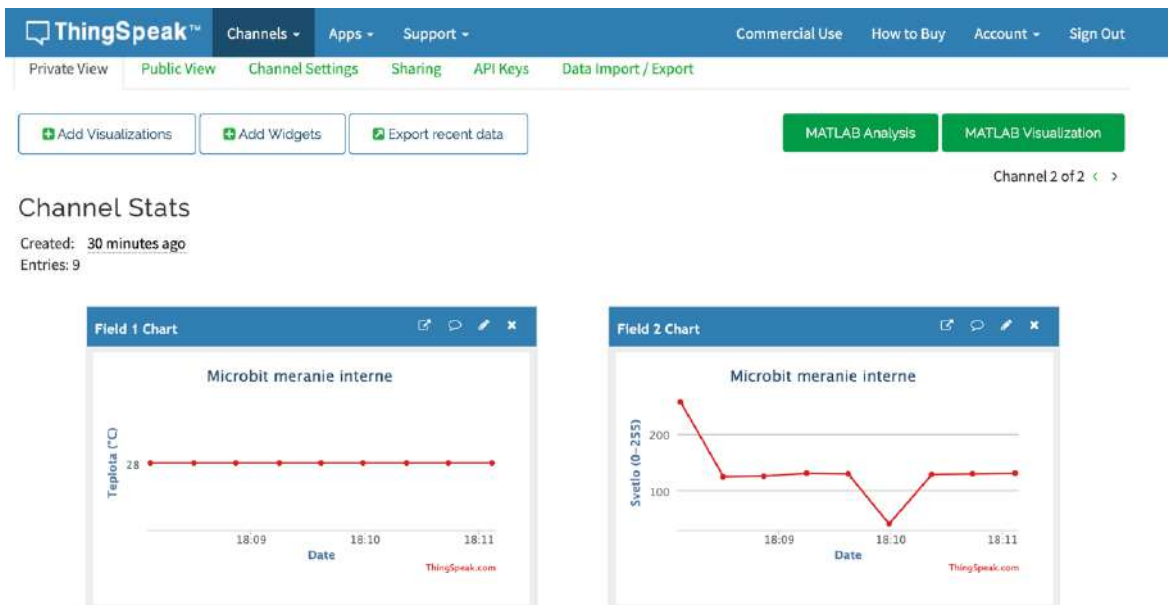
function zobrazHodnoty (teplota: number, svetlo: number) {
  OLED.clear()
  OLED.writeString("Teplota ")
  OLED.writeNum(teplota)
  OLED.writeString(" stupnov.")
  OLED.newLine()
  OLED.newLine()
  OLED.writeString("Svetlo (0 - 255) ")
  OLED.writeNum(svetlo)
}
let nSvetlo = 0
let nTeplota = 0
OLED.init(128, 64)
ESP8266_IoT.initWIFI(SerialPin.P8, SerialPin.P12, BaudRate.BaudRate115200)
ESP8266_IoT.connectWifi("TP-LINK_A471", "xxx")
basic.forever(function () {
  nTeplota = input.temperature()
  nSvetlo = input.lightLevel()
  zobrazHodnoty(nTeplota, nSvetlo)
  ESP8266_IoT.connectThingSpeak()
  ESP8266_IoT.setdata(
    "ZHUTMJ2523RNxxx",
    nTeplota,
    nSvetlo,
    0,
    0,
    0,
    0,
    0,
    0
  )
  ESP8266_IoT.uploadData()
  basic.pause(10000)
})

```

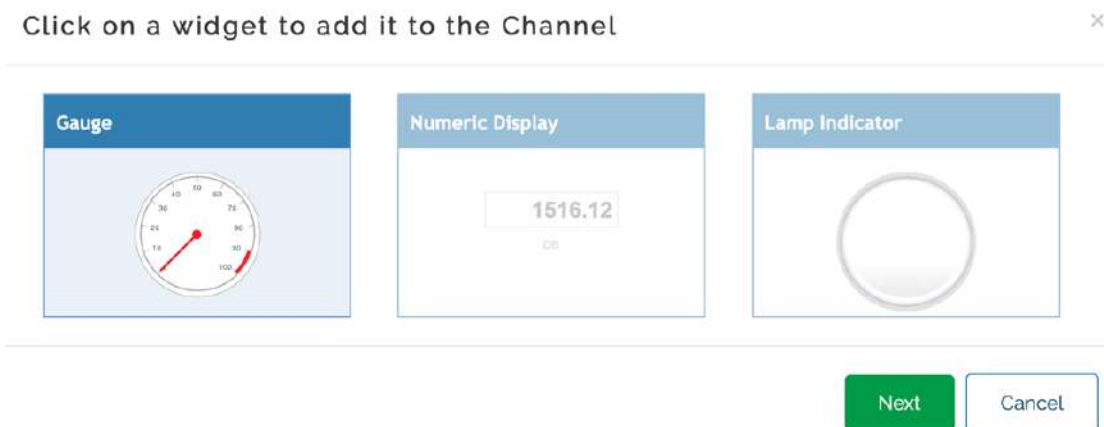


Zobrazovanie údajov v službe ThingSpeak

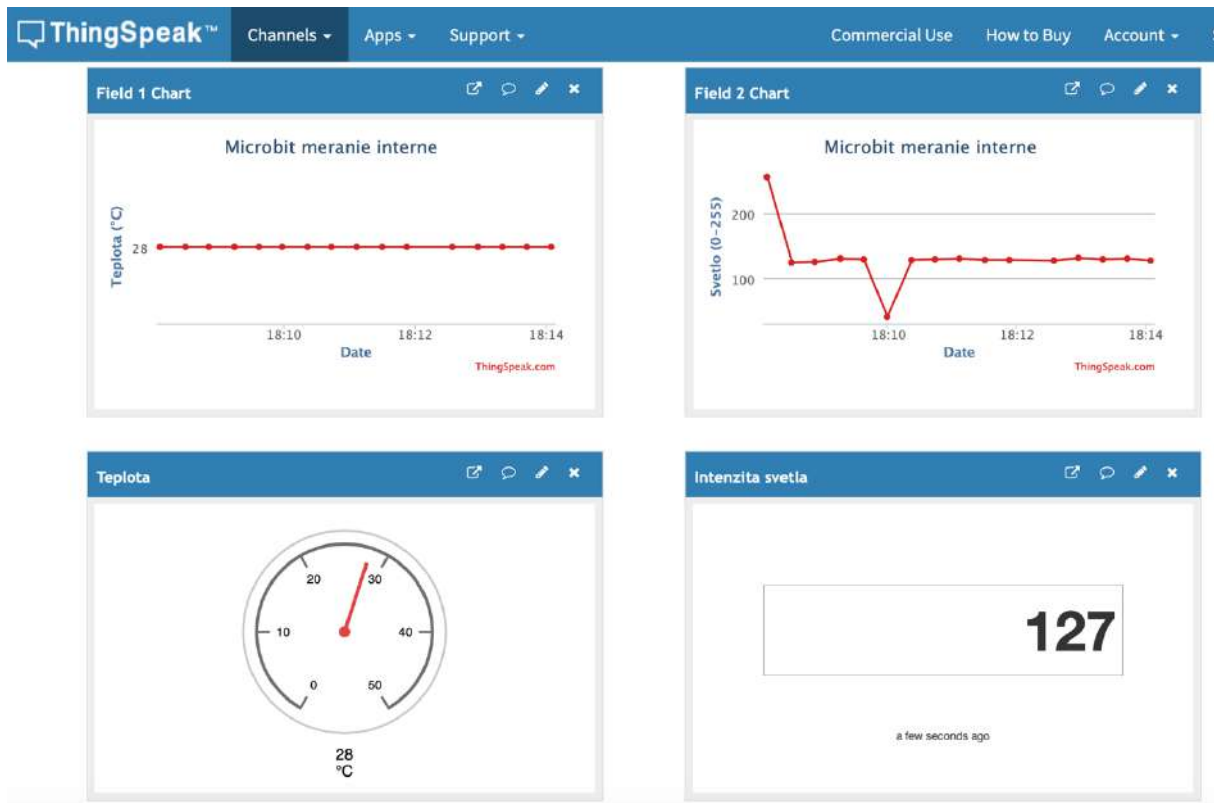
Na webovom portáli služby ThingSpeak sa prepnite do záložky **Private View**, kde môžete sledovať graf hodnôt posielaných z dosky Micro:bit do cloudovej služby



Okrem grafov môžete aktuálne namerané údaje zobrazovať aj formou analógových, či digitálnych ukazovateľov, prípadne formou kontrolky typu zelená – červená. V záložke Private View aktivujte tlačidlo **Add Widget**.



Následne môžete pridať vhodný typ ukazovateľa. V našom prípade analógový ukazovateľ pre teplotu a digitálny pre úroveň osvetlenia. Pre každý ukazovateľ nastavte interval hodnôt, napríklad pre izbovú teplotu rozsah 0 – 50 °C

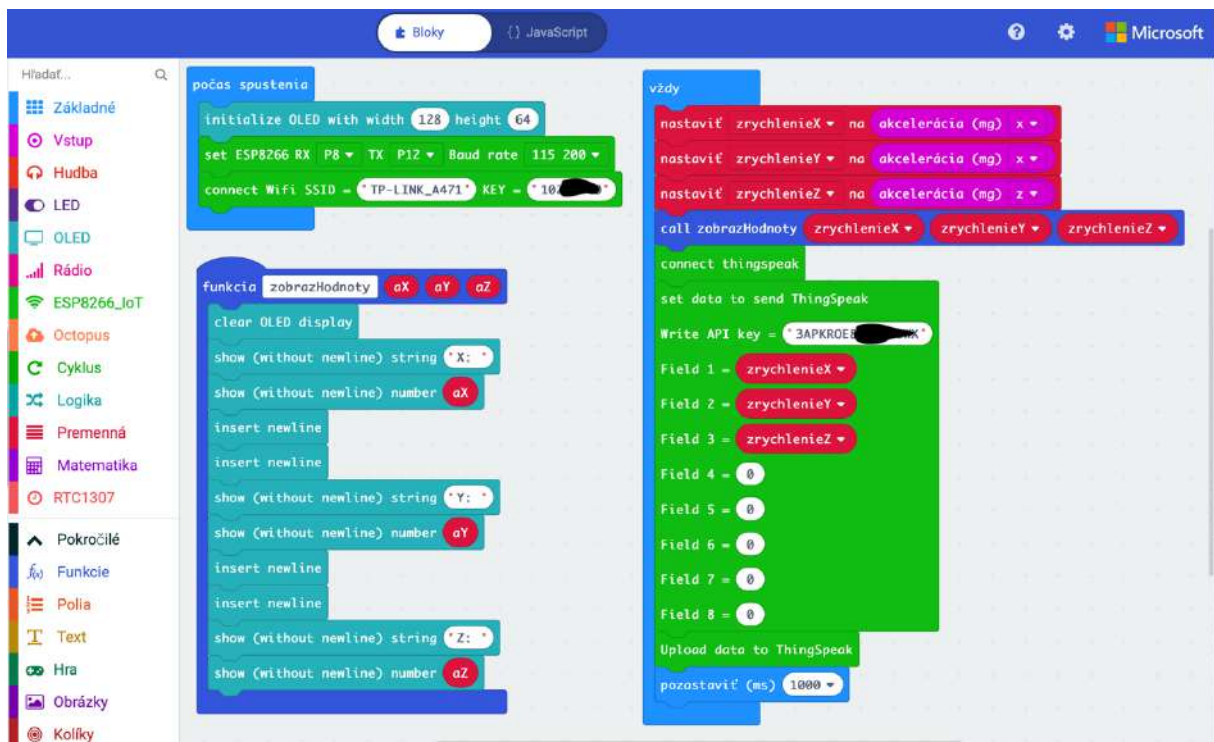


Meranie, posielanie do cloudu a zobrazovanie údajov z akcelerometra

V druhom príklade využijeme akcelerometer na doske Micro:bit a budeme merať zrýchlenie v osiach X, Y, a Z. Najskôr v cloudovej platforme vytvoríte nový kanál na posielanie údajov, v tomto prípade zrýchlenia v osiach X, Y a Z a skopírujte si do schránky jeho Write API Key, ktorý potom vložíte do kódu v Micro:bite.

The screenshot shows the 'New Channel' form in ThingSpeak. The 'Name' field is filled with 'Microbit zrýchlenie'. The 'Description' field is empty. Under 'Field Settings', three fields are listed: 'Zrýchlenie X', 'Zrýchlenie Y', and 'Zrýchlenie Z', each with a checked checkbox. To the right, a 'Help' section explains that channels store data and lists 'Channel Settings' including Channel Name, Description, Field#, and Metadata.

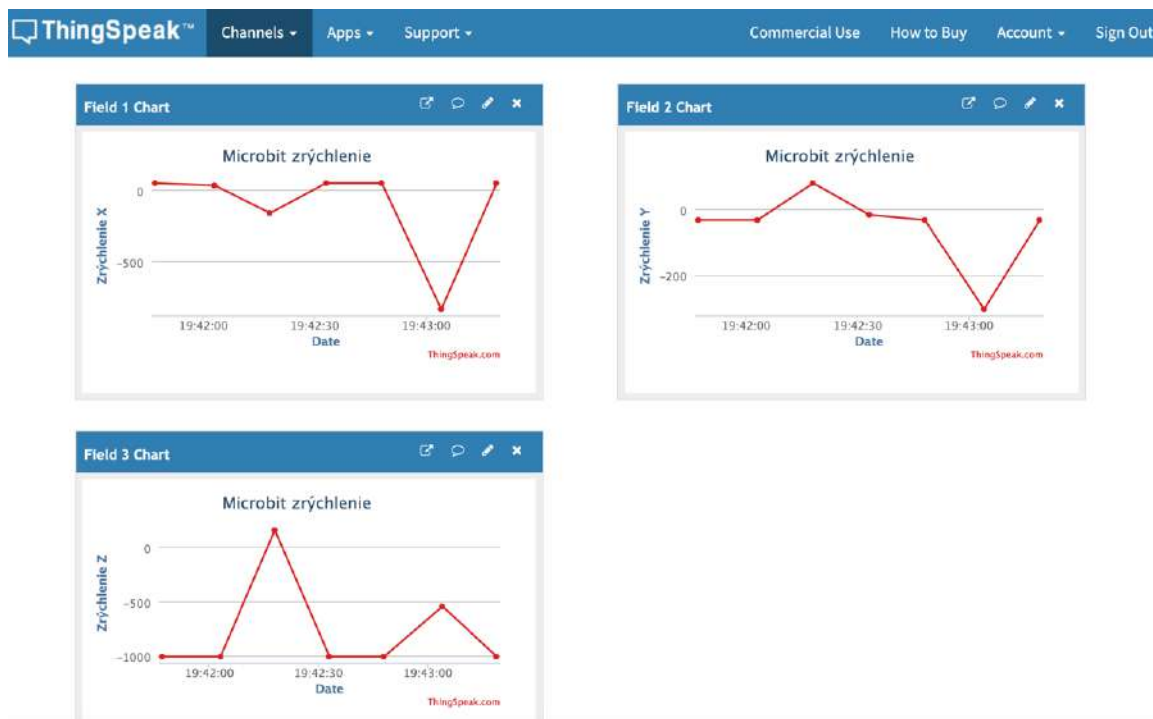
Upravíme aj funkciu na zobrazovanie hodnôt na OLED displeji a príkaz na vytvorenie bloku údajov, nakoľko sa budú zobrazovať a posilať tri hodnoty



```
function zobrazHodnoty (aX: number, aY: number, aZ: number) {
    OLED.clear()
    OLED.writeString("X: ")
    OLED.writeNum(aX)
    OLED.newLine()
    OLED.newLine()
    OLED.writeString("Y: ")
    OLED.writeNum(aY)
    OLED.newLine()
    OLED.newLine()
    OLED.writeString("Z: ")
    OLED.writeNum(aZ)
}
```

```
let zrychlenieZ = 0
let zrychlenieY = 0
let zrychlenieX = 0
OLED.init(128, 64)
ESP8266_IoT.initWIFI(SerialPin.P8, SerialPin.P12, BaudRate.BaudRate115200)
ESP8266_IoT.connectWifi("TP-LINK_A471", "10215449")
basic.forever(function () {
    zrychlenieX = input.acceleration(Dimension.X)
    zrychlenieY = input.acceleration(Dimension.X)
    zrychlenieZ = input.acceleration(Dimension.Z)
    zobrazHodnoty(zrychlenieX, zrychlenieY, zrychlenieZ)
    ESP8266_IoT.connectThingSpeak()
    ESP8266_IoT.setdata(
        "3APKROE8NG9MNIWX",
        zrychlenieX,
        zrychlenieY,
        zrychlenieZ,
        0,
        0,
        0,
        0,
        0
    )
    ESP8266_IoT.uploadData()
    basic.pause(1000)
})
```

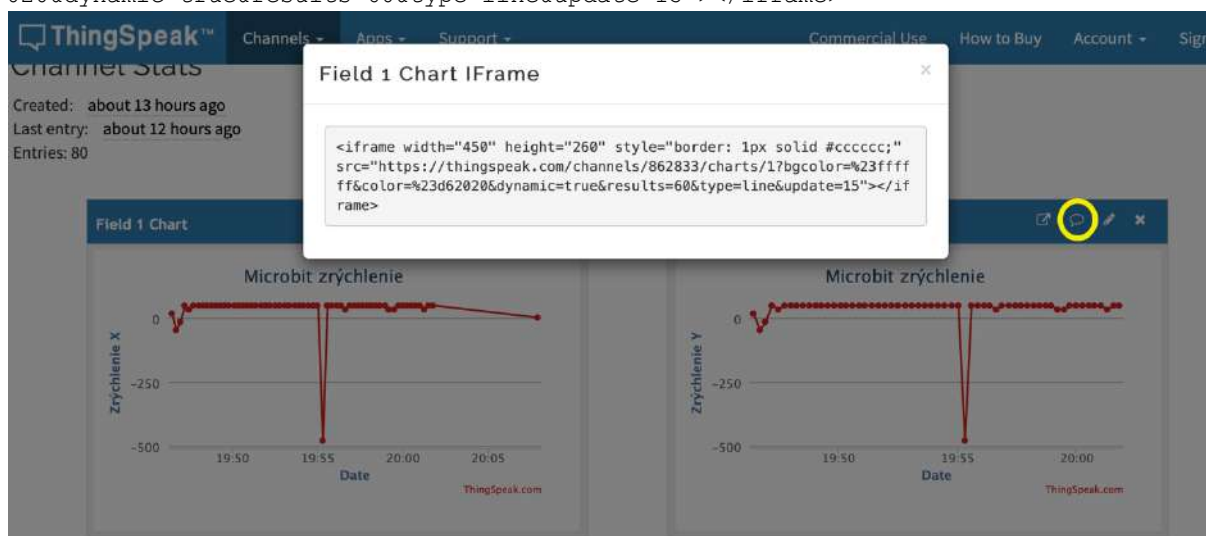
Interval posielania údajov sme skrátili na 1 sekundu. Teraz môžete pohybovať doskou microbit v rôznych smeroch a ak máte pripojený OLED displej tak na ňom sledovať hodnoty zrýchlenia. Časový priebeh zrýchlenia v jednotlivých osiach môžete sledovať na grafoch



Ako napovedá názov záložky Private View, údaje v tejto záložke môžete vidieť len vy. Ak ich chcete zdieľať, je potrebné parametre zdieľania nastaviť v záložke Sharing.

Ak chcete niektorý graf vložiť do svojej webovej stránky aktivujte v príslušnom grafe v záhlaví ikonu s bublinou a získate kód ktorý vložíte do stránky v tvare:

```
<iframe width="450" height="260" style="border: 1px solid #cccccc;"
src="https://thingspeak.com/channels/862833/charts/1?bgcolor=%23ffffff&color=%23d62020&dynamic=true&results=60&type=line&update=15"></iframe>
```



Údaje z príslušného kanála cloudovej platformy ThingSpeak môžete vyexportovať vo formátoch JSON, XML a CSV.

The screenshot shows the ThingSpeak web interface. At the top, there's a navigation bar with 'Channels', 'Apps', and 'Support' menus. The main header displays 'Microbit zrýchlenie' with channel ID 862833, author 'lubolavko', and private access. Below this, there are tabs for 'Private View', 'Public View', 'Channel Settings', 'Sharing', and 'API Keys'. The 'Data Import / Export' tab is active, showing two main sections: 'Import' and 'Export'. The 'Import' section has a 'File' input field (currently empty), a 'Time Zone' dropdown set to '(GMT+00:00) UTC', and an 'Upload' button. The 'Export' section has a 'Time Zone' dropdown set to '(GMT+00:00) UTC' and a 'Download' button. To the right, there's a 'Help' section with an 'Import' subsection explaining the correct format for data import, providing a 'CSV Import Format' example: 'date,time,field1,field2,field3,latitude,longitude,elevation' followed by a sample data row: '2019-01-01T10:11:12-05:00,11,33,44,88,10'. Below the help, there are links for 'Read Data' and 'Write Data'.

CSV je textový súbor, v ktorom sú hodnoty oddelené čiarkou. Názvy atribútov sú v prvom riadku. Je tam doplnený aj čas a ako zatiaľ prázdne atribúty sú zemepisné súradnice a stav.

```
created_at,entry_id,field1,field2,field3,latitude,longitude,elevation,status
2019-09-12 17:41:01 UTC,1,-112,464,-880,,,,
2019-09-12 17:41:17 UTC,2,176,352,-704,,,,
2019-09-12 17:41:32 UTC,3,32,-16,-1008,,,,
2019-09-12 17:41:47 UTC,4,48,-32,-1008,,,,
2019-09-12 17:42:03 UTC,5,32,-32,-1008,,,,
2019-09-12 17:42:18 UTC,6,-160,80,160,,,,
2019-09-12 17:42:33 UTC,7,48,-16,-1008,,,,
2019-09-12 17:42:48 UTC,8,48,-32,-1008,,,,
2019-09-12 17:43:04 UTC,9,-832,-304,-544,,,,
```

Tento formát dokáže načítať aj aplikácia Excel. Niekedy budete musieť atribúty oddelené čiarkou rozdeliť do stĺpcov manuálne.

The screenshot shows the Microsoft Excel interface with the imported data from the ThingSpeak CSV file. The data is displayed in a table with columns A through N. The first row (row 1) contains the headers: 'created_at', 'entry_id', 'field1', 'field2', 'field3', 'latitude', 'longitude', 'elevation', and 'status'. The subsequent rows (rows 2-17) contain the data points, with the 'elevation' column being empty for all entries. The 'created_at' column shows timestamps in UTC format, and the other columns contain numerical values as specified in the CSV data.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	created_at	entry_id	field1	field2	field3	latitude	longitude	elevation	status					
2	2019-09-12 17:41:01 UTC	1	-112	464	-880									
3	2019-09-12 17:41:17 UTC	2	176	352	-704									
4	2019-09-12 17:41:32 UTC	3	32	-16	-1008									
5	2019-09-12 17:41:47 UTC	4	48	-32	-1008									
6	2019-09-12 17:42:03 UTC	5	32	-32	-1008									
7	2019-09-12 17:42:18 UTC	6	-160	80	160									
8	2019-09-12 17:42:33 UTC	7	48	-16	-1008									
9	2019-09-12 17:42:48 UTC	8	48	-32	-1008									
10	2019-09-12 17:43:04 UTC	9	-832	-304	-544									
11	2019-09-12 17:43:19 UTC	10	48	-32	-1008									
12	2019-09-12 17:43:34 UTC	11	48	-32	-992									
13	2019-09-12 17:43:50 UTC	12	48	-32	-1008									
14	2019-09-12 17:44:05 UTC	13	48	-32	-1008									
15	2019-09-12 17:44:20 UTC	14	48	-32	-1008									
16	2019-09-12 17:44:36 UTC	15	48	-32	-1008									
17	2019-09-12 17:44:51 UTC	16	48	-16	-1008									

Jednoduchá automatizácia s využitím IFTTT

Námetom tohoto pokračovania je jednoduchá automatizácia s využitím IFTTT. Skratku v nadpise asi poznáte, znamená "If This Then That", čiže po našom „ak toto, potom tamto“ alebo inak povedané, ak nastane takáto situácia, urob takúto akciu. IFTTT je riešenie, ktoré vám umožní vytvárať si akési „recepty“ (recipes), prostredníctvom ktorých si jednoducho nadefinujete akciu a reakciu.

Akcia sa môže týkať nejakej veličiny, napríklad že teplota v danom priestore klesla pod 15 °C, prípadne to môže byť všeobecná akcia, napríklad vám príde e-mail od konkrétneho odosielateľa, dostanete do telefónu SMS správu a podobne.

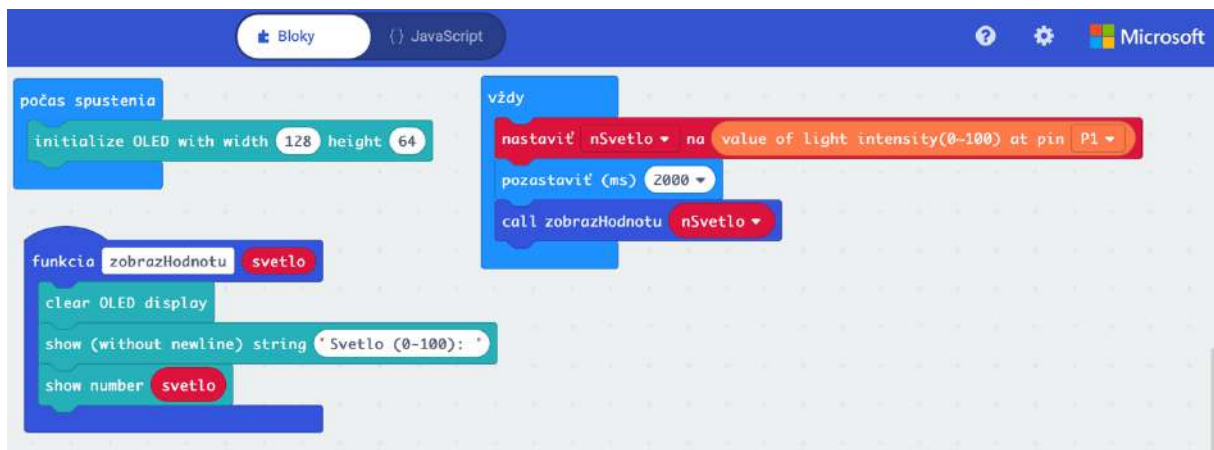
Reakcia môže byť poslanie e-mailu, alebo SMS, že hodnota sledovanej veličiny prekročila stanovenú úroveň, zapnutie osvetlenia, vykurovania, príkaz robotickému vysávaču aby povysával konkrétnu miestnosť a podobne.

Týchto akcií a reakcií je k dispozícii veľké množstvo. Ich aktuálny zoznam nájdete na oficiálnom webe <https://ifttt.com/services>. IFTTT je riešenie zamerané skôr na bežné, domáce používanie, či študentské projekty, než na priemyselnú automatizáciu. S tým súvisí aj jeho jednoduché rozhranie a skutočnosť, že služba je pre bežného používateľa zadarmo. Stačí si vytvoriť konto a aplikáciu je možné začať používať okamžite s tým, že ovládanie je veľmi intuitívne. K dispozícii je webové rozhranie, ako aj aplikácia pre Android a iPhone.

Takto môžete programovať naprieč IoT senzormi a akčnými členmi, cloudovými službami, internetovými, či mobilnými aplikáciami, mailovými službami, službami na posielanie správ a podobne. IFTTT je silným nástrojom s jednoduchým používaním, ktorý dokáže prepojiť desiatky internetových služieb, aplikácií... a ušetrí vám veľa opakovanej rutinné práce. Jednotlivé prepojenia sa nazývajú recepty a môžete si ich vytvoriť sami, alebo použiť recepty od iných užívateľov. Pomocou IFTTT služieb môžete automatizovať a ovládať prvky inteligentnej domácnosti Google Home, Apple Homekit, Amazon Echo ..

V príklade použijeme meranie intenzity svetla. Môžete použiť buď senzor na doske MicroBit, ktorý využíva vlastností LED diód maticového displeja, alebo ako v našom prípade externý senzor zo Smart Science IoT Kit, prípadne zo súpravy Smart home Kit, či akýkoľvek iný senzor, ktorý viete softvérovo obslúžiť. Externý senzor sme použili nielen preto že je presnejší, ale aby bolo video názornejšie. Námetom bude poslanie správy pri udalosti otvorenia skrinky, keď do nej prenikne svetlo.

Budeme postupovať systematicky. Najskôr vytvoríme a odladíme aplikáciu, ktorá zobrazuje intenzitu osvetlenia na displeji



```
function zobrazHodnotu (svetlo: number) {
```

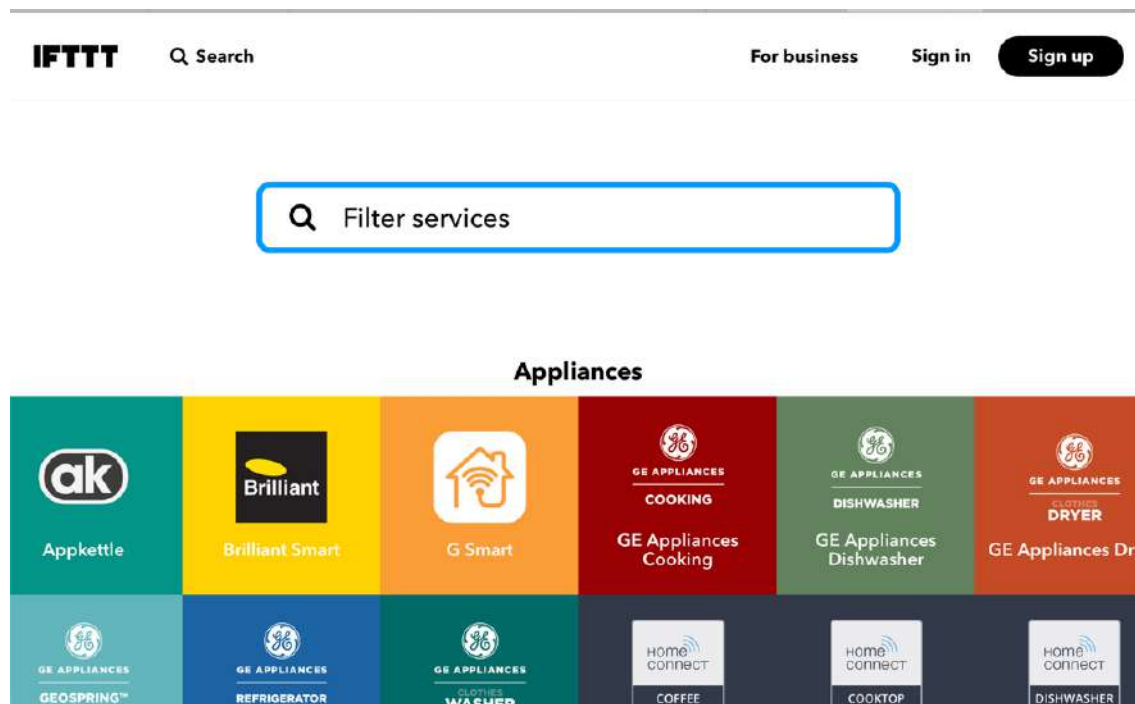


```

0,
0,
0
)
ESP8266_IoT.uploadData ()
basic.pause (5000)
})

```

Ak chcete realizovať IFTTT scenár, v našom prípade poslanie mailu v prípade ak úroveň osvetlenia (teploty, vlhkosti... jednoducho akejkoľvek veličiny) prekročí stanovanú úroveň potrebujete službu IFTTT. Prihláste sa do služby IFTTT <https://ifttt.com> Ak nemáte účet, musíte si ho vytvoriť. Prihlásiť sa môžete aj cez svoj Google, či Facebook účet.



Najskôr je potrebné vytvoriť aplet, čiže predpis pre akciu a reakciu. Kliknite na ikonu používateľa vľavo hore a z ponuky vyberte **Create**.



Start connecting your world.



- Account
- Activity
- My Applets
- My services
- Create
- Help
- Sign out

V dialógu If This Then That ktorý sa zobrazí, kliknite na slovo **This**.



Create your own

If This Then That

Build your own service on the **IFTTT** Platform [↗](#)

Nasledujúci postup je rozdelený do šiestich krokov.

V prvom kroku nájdite službu **webhooks**. Tá umožňuje reagovať na udalosti z iných webových služieb, v našom prípade zo služby ThingSpeak. Voči službe Webhooks sa autentifikujete pomocou mailovej adresy. Zadáte svoju mailovú adresu a služba vám mailom pošle PIN kód, aby bolo potvrdené, že je to skutočne vaša mailová adresa.

[Back](#)

Choose a service

Step 1 of 6

web hooks



V druhom kroku je potrebné vytvoriť trigger, čiže spúšť. Je to definovanie podmienky pri ktorej sa spustí nadefinovaná akcia, v našom prípade poslanie mailu. Udalosť príde z inej webovej, alebo cloudovej služby, v našom prípade z ThingSpeak. Takže kliknite na pole Receive web request.

[Back](#)



Choose trigger

Step 2 of 6

Receive a web request

This trigger fires every time the Maker service receives a web request to notify it of an event. For information on triggering events, go to your Maker service settings and then the listed URL (web) or tap your username (mobile)

Pomenujte trigger, napríklad microbit_svetlo_mail. Prepojenie na IFTTT nadefinujete neskôr cez portál služby ThingSpeak.

[Back](#)



Complete trigger fields

Step 2 of 6

Event Name

microbit_svetlo_mail

The name of the event, like "button_pressed" or "front_door_opened"

[Create trigger](#)

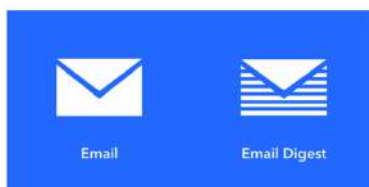
Teraz budete definovať akciu, čiže čo sa má stať ak nastane definovaná situácia. V dialógu If This Then That tentokrát kliknite na slovo **That**.

< Back

Choose action service

Step 3 of 6

Q email



V kroku 3 vyberte službu email.

< Back



Choose action

Step 4 of 6

Send me an email

This Action will send you an HTML based email. Images and links are supported.

Následne kliknite na pole **Send me an email**.

Zobrazí sa vám šablóna správy. V poli **Subject** zadajte názov udalosti, v našom prípade `microbit_svetlo_mail`. V poliach v zátvorkách `{{}}` je možné extrahovať údaje s rovnakými atribútmi, ktoré prídu z webovej služby, v našom prípade zo služby ThingSpeak.

< Back



Complete action fields

Step 5 of 6

Subject

The event named "`EventName`" occurred on the Maker Webhooks service

Add ingredient

Body

What: `EventName`

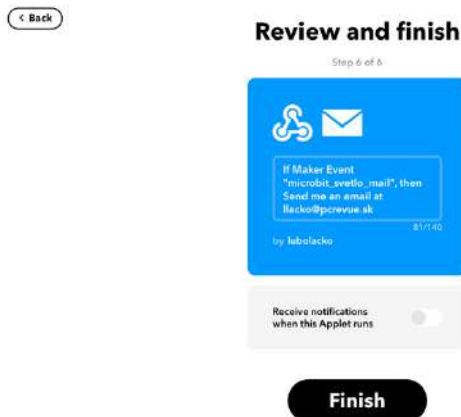
When: `OccurredAt`

Extra Data: `Value1`, `Value2`, `Value3`

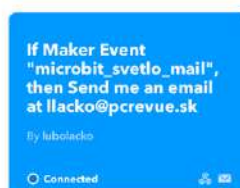
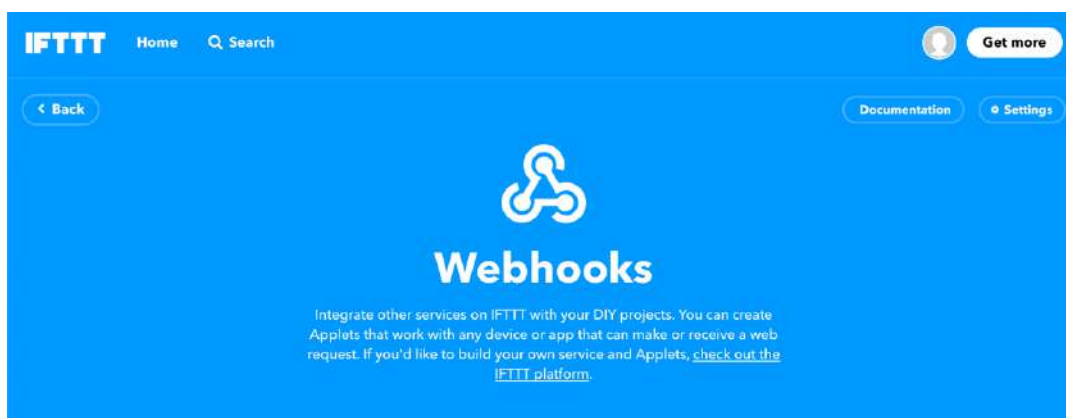
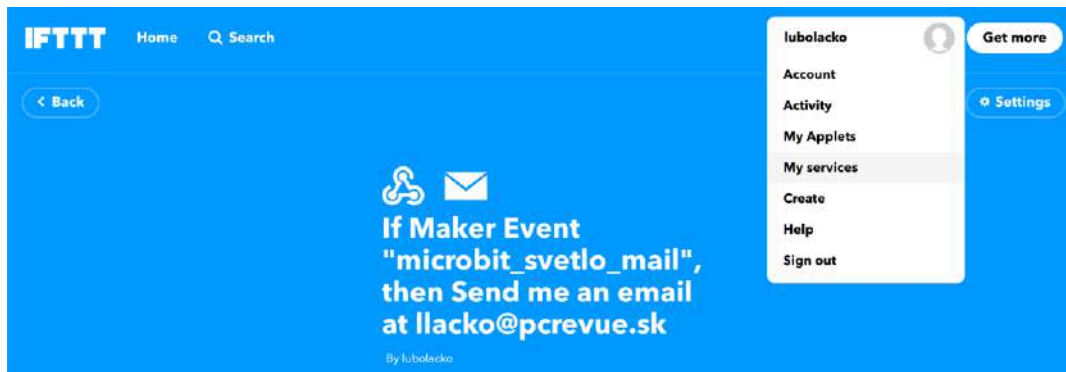
Add ingredient

Create action

V poslednom dialógu sa zobrazí sumarizácia. Zapnite prepínač **Receive notifications when this Applet runs**.



Na prepojenie so službou ThingSpeak potrebujete kľúč pre web request. Ten zistíte v dokumentácii o nastavení služby Webhooks. Dostanete sa tam cez menu svojho účtu – položku Services a následne vyberiete službu Webhooks.

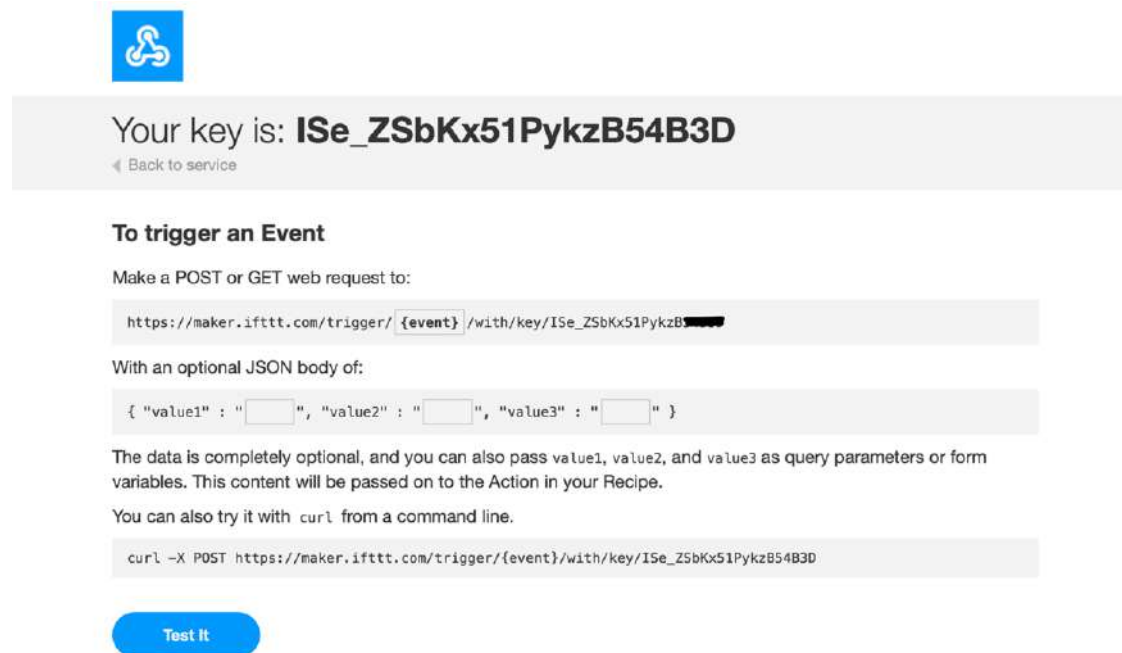


V našom prípade je web request

https://maker.ifttt.com/trigger/{event}/with/key/ISe_ZSbKx51PykzBxxxx

doplňte do reťazca názov udalosti microbit_svetlo_mail. Získate adresu v tvare

https://maker.ifttt.com/trigger/microbit_svetlo_mail/with/key/ISe_ZSbKx51PykzBxxxx



Your key is: ISe_ZSbKx51PykzB54B3D
◀ Back to service

To trigger an Event

Make a POST or GET web request to:

```
https://maker.ifttt.com/trigger/{event}/with/key/ISe_ZSbKx51PykzB54B3D
```

With an optional JSON body of:

```
{ "value1" : " ", "value2" : " ", "value3" : " " }
```

The data is completely optional, and you can also pass value1, value2, and value3 as query parameters or form variables. This content will be passed on to the Action in your Recipe.

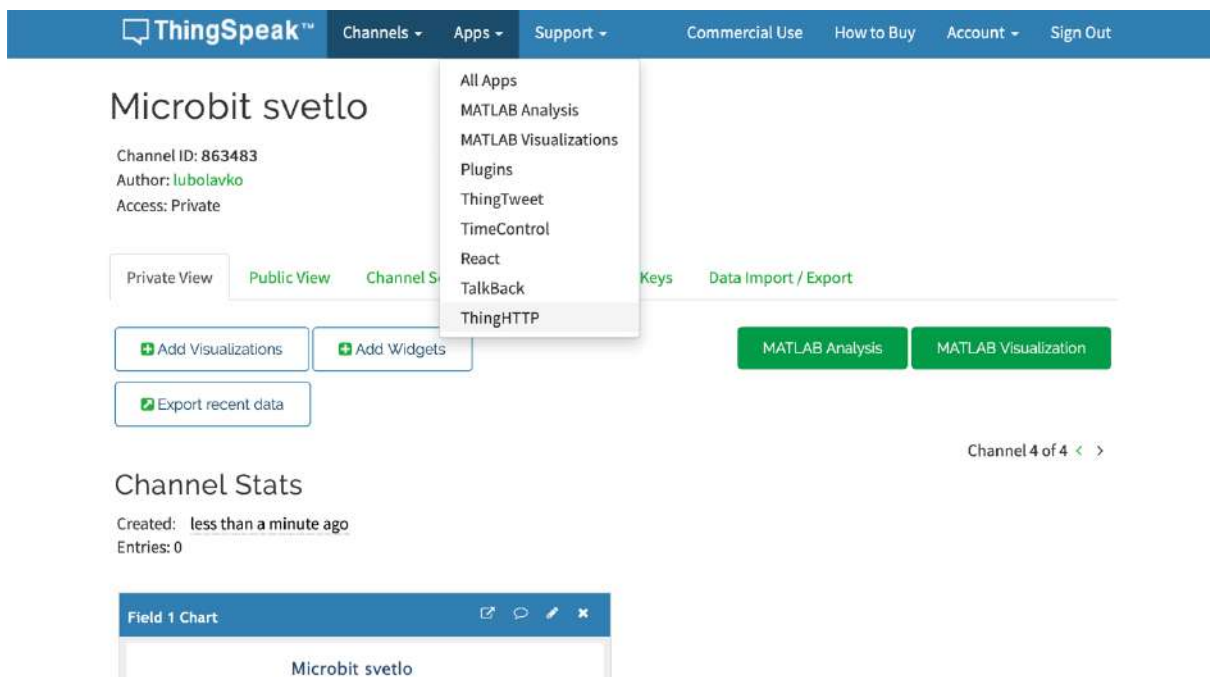
You can also try it with curl from a command line.

```
curl -X POST https://maker.ifttt.com/trigger/{event}/with/key/ISe_ZSbKx51PykzB54B3D
```

[Test It](#)

Textový reťazec s URL adresou v poli **Make a POST or GET web request to:** si skopírujte do schránky. Budete ho potrebovať pre ThingSpeak.

Vo webovom rozhraní služby ThingSpeak pre príslušný kanál V menu **Apps** aktivujte položku **ThingHTTP**.



ThingSpeak™ Channels - Apps - Support - Commercial Use How to Buy Account - Sign Out

Microbit svetlo

Channel ID: 863483
Author: lubolavko
Access: Private

Private View Public View Channel Settings Keys Data Import / Export

[Add Visualizations](#) [Add Widgets](#) [Export recent data](#)

[MATLAB Analysis](#) [MATLAB Visualization](#)

Channel 4 of 4 < >

Channel Stats

Created: less than a minute ago
Entries: 0

Field 1 Chart

Microbit svetlo

Do poľa ULR zadajte adresu zo schránky

V poli **Method** nastavte **POST**.

Do poľa **Content Type** zadajte: application/json. Verziu HTTP nechajte nastavenú na 1.1

{"value1":"%%channel_138112_field_1%%"}

The screenshot shows the ThingSpeak 'ThingHTTP Settings' page. The 'Name' field is 'SvetelnyAlarm'. The 'API Key' is 'BXT83S0NE02HZVE0'. The 'URL' is 'https://maker.ifttt.com/trigger/microbit_svetlo_n'. The 'Method' is 'POST'. The 'Content Type' is 'application/json'. The 'HTTP Version' is '1.1'. The 'Host' field is empty. The 'Body' field contains the JSON string: {"value1":"%%channel_863483_field_1%%"}. The 'Parse String' field is empty. The 'Save ThingHTTP' button is visible at the bottom.

ThingHTTP Settings

- **Name:** Enter a unique name for your ThingHTTP request.
- **API Key:** Auto generated API key for the ThingHTTP request.
- **URL:** Enter the address of the website you are requesting data from or writing data to starting with either http:// or https://.
- **HTTP Auth Username:** If your URL requires authentication, enter the username for authentication to access private channels or websites.
- **HTTP Auth Password:** If your URL requires authentication, enter the password for authentication to access private channels or websites.
- **Method:** Select the HTTP method required to access the URL.
- **Content Type:** Enter the MIME or form type of the request content. For example, application/x-www-form-urlencoded.
- **HTTP Version:** Select the version of HTTP on your server.
- **Host:** If your ThingHTTP request requires a host address, enter the domain name here. For example, api.thingspeak.com.
- **Headers:** If your ThingHTTP request requires custom headers, enter the information here. You must specify the name of the header and a value.
- **Body:** Enter the message you want to include in your request.
- **Parse String:** If you want to parse the response, enter the exact string to look for in the response data.

Example

Send an HTTP GET request to MarketWatch and parse the response for an element corresponding to the S&P 500 current price change for the day.

1. Enter `https://www.marketwatch.com` in the URL field
2. Enter `GET` in the Method field
3. In the Parse String field, enter `/html/body/section[1]/div[2]/div/div[1]/div[2]/div[1]/table/tbody/tr[2]/td[4]/bg-quote`
4. Save and attach the HTTP request to a reaction you set up in the React, TimeControl, or TweetControl apps.

[Learn More](#)

V menu **Apps** aktivujte položku **React**.

Zadajte názov, nastavte typ podmienky ako numerickú. Nastavíme, že trigger sa má aktivovať ak hodnota intenzity svetla prekročí 50. Nastavte aby sa akcia vykonala zakaždým, označte pole **Run action each time condition is met**.

ThingSpeak™ Channels Apps Support Commercial Use How to Buy Account Sign Out

Apps / React / New

React Name: Alarm

Condition Type: Numeric

Test Frequency: On Data Insertion

Condition: If channel: Microbit svetlo (863483)

field: 1 (Intenzita svetla)

is greater than: 50

Action: ThingHTTP

then perform ThingHTTP: SvetelnyAlarm

Help

React Settings

- **React Name:** Enter a unique name for your React.
- **Condition Type:** Select a condition type corresponding with your data. A channel can hold numeric sensor data, text, strings, status updates, or geographic location information.
- **Test Frequency:** Choose whether to test your condition every time data enters the channel or on a periodic basis.
- **Condition:** Select a channel, a field and the condition for your React.
- **Action:** Select ThingTweet, ThingHTTP, or MATLAB Analysis to run when the condition is met.
- **Options:** Select when the React runs.

[Learn More](#)

Teraz spustite aplikáciu pre Micro:bit, ktorá bude merať intenzitu svetla a posilať hodnoty do cloudu ThingSpeak.

Keď intenzita svetla prekročí hodnotu 50, alebo hodnota inej veličiny prekročí hodnotu nastavenú v podmienke, budete na túto skutočnosť upozornení mailovou správou.

ThingSpeak™ Channels Apps Support Commercial Use How to Buy Account Sign Out

Add Visualizations Add Widgets Export recent data

MATLAB Analysis MATLAB Visualization

Channel 4 of 4 < >

Channel Stats

Created: 39 minutes ago
 Last entry: less than a minute ago
 Entries: 35

Field 1 Chart

Date	Intenzita svetla
13:18:00	0
13:18:30	0
13:19:00	0
13:19:30	0
13:20:00	0
13:20:15	20
13:20:30	30
13:20:45	40
13:21:00	45
13:21:15	65
13:21:30	70
13:21:45	70
13:22:00	70
13:22:15	70
13:22:30	70
13:22:45	70
13:23:00	70
13:23:15	70
13:23:30	70
13:23:45	70
13:24:00	70
13:24:15	70
13:24:30	70
13:24:45	70
13:25:00	70
13:25:15	70
13:25:30	70
13:25:45	70
13:26:00	70
13:26:15	70
13:26:30	70
13:26:45	70
13:27:00	70
13:27:15	70
13:27:30	70
13:27:45	70
13:28:00	70
13:28:15	70
13:28:30	70
13:28:45	70
13:29:00	70
13:29:15	70
13:29:30	70
13:29:45	70
13:30:00	70
13:30:15	70
13:30:30	70
13:30:45	70
13:31:00	70
13:31:15	70
13:31:30	70
13:31:45	70
13:32:00	70
13:32:15	70
13:32:30	70
13:32:45	70
13:33:00	70
13:33:15	70
13:33:30	70
13:33:45	70
13:34:00	70
13:34:15	70
13:34:30	70
13:34:45	70
13:35:00	70
13:35:15	70
13:35:30	70
13:35:45	70
13:36:00	70
13:36:15	70
13:36:30	70
13:36:45	70
13:37:00	70
13:37:15	70
13:37:30	70
13:37:45	70
13:38:00	70
13:38:15	70
13:38:30	70
13:38:45	70
13:39:00	70
13:39:15	70
13:39:30	70
13:39:45	70
13:40:00	70
13:40:15	70
13:40:30	70
13:40:45	70
13:41:00	70
13:41:15	70
13:41:30	70
13:41:45	70
13:42:00	70
13:42:15	70
13:42:30	70
13:42:45	70
13:43:00	70
13:43:15	70
13:43:30	70
13:43:45	70
13:44:00	70
13:44:15	70
13:44:30	70
13:44:45	70
13:45:00	70
13:45:15	70
13:45:30	70
13:45:45	70
13:46:00	70
13:46:15	70
13:46:30	70
13:46:45	70
13:47:00	70
13:47:15	70
13:47:30	70
13:47:45	70
13:48:00	70
13:48:15	70
13:48:30	70
13:48:45	70
13:49:00	70
13:49:15	70
13:49:30	70
13:49:45	70
13:50:00	70
13:50:15	70
13:50:30	70
13:50:45	70
13:51:00	70
13:51:15	70
13:51:30	70
13:51:45	70
13:52:00	70
13:52:15	70
13:52:30	70
13:52:45	70
13:53:00	70
13:53:15	70
13:53:30	70
13:53:45	70
13:54:00	70
13:54:15	70
13:54:30	70
13:54:45	70
13:55:00	70
13:55:15	70
13:55:30	70
13:55:45	70
13:56:00	70
13:56:15	70
13:56:30	70
13:56:45	70
13:57:00	70
13:57:15	70
13:57:30	70
13:57:45	70
13:58:00	70
13:58:15	70
13:58:30	70
13:58:45	70
13:59:00	70
13:59:15	70
13:59:30	70
13:59:45	70
14:00:00	70
14:00:15	70
14:00:30	70
14:00:45	70
14:01:00	70
14:01:15	70
14:01:30	70
14:01:45	70
14:02:00	70
14:02:15	70
14:02:30	70
14:02:45	70
14:03:00	70
14:03:15	70
14:03:30	70
14:03:45	70
14:04:00	70
14:04:15	70
14:04:30	70
14:04:45	70
14:05:00	70
14:05:15	70
14:05:30	70
14:05:45	70
14:06:00	70
14:06:15	70
14:06:30	70
14:06:45	70
14:07:00	70
14:07:15	70
14:07:30	70
14:07:45	70
14:08:00	70
14:08:15	70
14:08:30	70
14:08:45	70
14:09:00	70
14:09:15	70
14:09:30	70
14:09:45	70
14:10:00	70
14:10:15	70
14:10:30	70
14:10:45	70
14:11:00	70
14:11:15	70
14:11:30	70
14:11:45	70
14:12:00	70
14:12:15	70
14:12:30	70
14:12:45	70
14:13:00	70
14:13:15	70
14:13:30	70
14:13:45	70
14:14:00	70
14:14:15	70
14:14:30	70
14:14:45	70
14:15:00	70
14:15:15	70
14:15:30	70
14:15:45	70
14:16:00	70
14:16:15	70
14:16:30	70
14:16:45	70
14:17:00	70
14:17:15	70
14:17:30	70
14:17:45	70
14:18:00	70
14:18:15	70
14:18:30	70
14:18:45	70
14:19:00	70
14:19:15	70
14:19:30	70
14:19:45	70
14:20:00	70
14:20:15	70
14:20:30	70
14:20:45	70
14:21:00	70
14:21:15	70
14:21:30	70
14:21:45	70
14:22:00	70
14:22:15	70
14:22:30	70
14:22:45	70
14:23:00	70
14:23:15	70
14:23:30	70
14:23:45	70
14:24:00	70
14:24:15	70
14:24:30	70
14:24:45	70
14:25:00	70
14:25:15	70
14:25:30	70
14:25:45	70
14:26:00	70
14:26:15	70
14:26:30	70
14:26:45	70
14:27:00	70
14:27:15	70
14:27:30	70
14:27:45	70
14:28:00	70
14:28:15	70
14:28:30	70
14:28:45	70
14:29:00	70
14:29:15	70
14:29:30	70
14:29:45	70
14:30:00	70
14:30:15	70
14:30:30	70
14:30:45	70
14:31:00	70
14:31:15	70
14:31:30	70
14:31:45	70
14:32:00	70
14:32:15	70
14:32:30	70
14:32:45	70
14:33:00	70
14:33:15	70
14:33:30	70
14:33:45	70
14:34:00	70
14:34:15	70
14:34:30	70
14:34:45	70
14:35:00	70
14:35:15	70
14:35:30	70
14:35:45	70
14:36:00	70
14:36:15	70
14:36:30	70
14:36:45	70
14:37:00	70
14:37:15	70
14:37:30	70
14:37:45	70
14:38:00	70
14:38:15	70
14:38:30	70
14:38:45	70
14:39:00	70
14:39:15	70
14:39:30	70
14:39:45	70
14:40:00	70
14:40:15	70
14:40:30	70
14:40:45	70
14:41:00	70
14:41:15	70
14:41:30	70
14:41:45	70
14:42:00	70
14:42:15	70
14:42:30	70
14:42:45	70
14:43:00	70
14:43:15	70
14:43:30	70
14:43:45	70
14:44:00	70
14:44:15	70
14:44:30	70
14:44:45	70
14:45:00	70
14:45:15	70
14:45:30	70
14:45:45	70
14:46:00	70
14:46:15	70
14:46:30	70
14:46:45	70
14:47:00	70
14:47:15	70
14:47:30	70
14:47:45	70
14:48:00	70
14:48:15	70
14:48:30	70
14:48:45	70
14:49:00	70
14:49:15	70
14:49:30	70
14:49:45	70
14:50:00	70
14:50:15	70
14:50:30	70
14:50:45	70
14:51:00	70
14:51:15	70
14:51:30	70
14:51:45	70
14:52:00	70
14:52:15	70
14:52:30	70
14:52:45	70
14:53:00	70
14:53:15	70
14:53:30	70
14:53:45	70
14:54:00	70
14:54:15	70
14:54:30	70
14:54:45	70
14:55:00	70
14:55:15	70
14:55:30	70
14:55:45	70
14:56:00	70
14:56:15	70
14:56:30	70
14:56:45	70
14:57:00	70
14:57:15	70
14:57:30	70
14:57:45	70
14:58:00	70
14:58:15	70
14:58:30	70
14:58:45	70
14:59:00	70
14:59:15	70
14:59:30	70
14:59:45	70
15:00:00	70
15:00:15	70
15:00:30	70
15:00:45	70
15:01:00	70
15:01:15	70
15:01:30	70
15:01:45	70
15:02:00	70
15:02:15	70
15:02:30	70
15:02:45	70
15:03:00	70
15:03:15	70
15:03:30	70
15:03:45	70
15:04:00	70
15:04:15	70
15:04:30	70
15:04:45	70
15:05:00	70
15:05:15	70
15:05:30	70
15:05:45	70
15:06:00	70
15:06:15	70
15:06:30	70
15:06:45	70
15:07:00	70
15:07:15	70
15:07:30	70
15:07:45	70
15:08:00	70
15:08:15	70
15:08:30	70
15:08:45	70
15:09:00	70
15:09:15	70
15:09:30	70
15:09:45	70
15:10:00	70
15:10:15	70
15:10:30	70
15:10:45	70
15:11:00	70
15:11:15	70
15:11:30	70
15:11:45	70
15:12:00	70
15:12:15	70
15:12:30	70
15:12:45	70
15:13:00	70
15:13:15	70
15:13:30	70
15:13:45	70
15:14:00	70
15:14:15	70