název: **Rotující Šipka**  
kategorie: A začátečníci  
ref.číslo: AZ4  
verze: 01, 2017-06-14  
autor: T.Krisl, Gymnázium Lanškroun

**Zadání:**

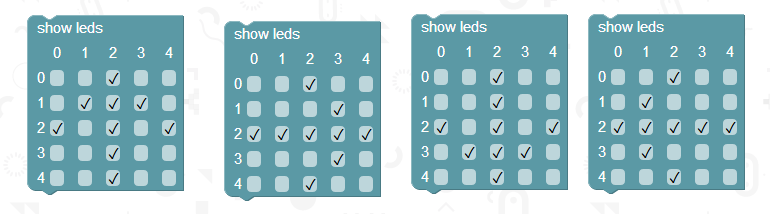
Vytvoř prográmek, kdy se bude na displeji zobrazovat šipka, která se točí dokola.

Tlačítkem A její pohyb zastavíme a pokud už bude stát, tak se roztočí.

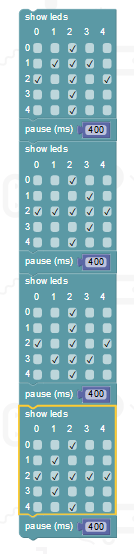
Vylepšení programu bude to, že při zmáčknutí druhého tlačítka se začne točit na opačnou stranu.

**Řešení:**

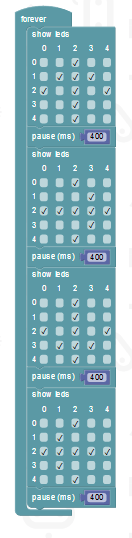
Jak udělat šipku, která se bude točit? Budu za sebou ukazovat na displeji obrázky šipek, které jsou vždy o kousek otočené. Ty si připravím pomocí bloku ***show leds***. Mohou stačit čtyři šipky - nahoru, vpravo, dolů a vlevo.



Když je spustím za sebou s malým časem mezi střídáním, aby to nebylo moc rychlé na sledování, budu mít rotující šipku.



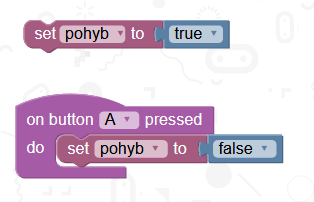
Samozřejmě bude třeba tuto jednu otočku opakovat pomocí nějakého cyklu. Protože budeme pořád dokola točit, můžeme použít ***forever*** anebo ***while***.



Nyní již máme šipku, která se točí dokola.

Teď musíme vyřešit to, že jí tlačítkem A budeme zastavovat a zase roztáčet. Zastavení by mělo vypadat tak, že při zmáčknutí A se šipka zastaví v poloze, jaké zrovna je. Roztočení při dalším stisknutí opět tlačítka A znamená, že bude pokračovat dalším obrázkem v pořadí.

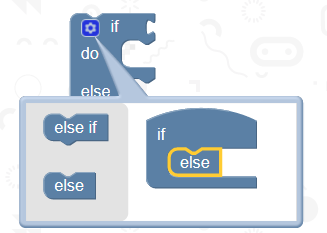
Možností, jak tento problém vyřešit je určitě více. My vyzkoušíme využít proměnou, která bude v sobě uchovávat informaci, jestli se máme točit anebo stát. Taková proměnná se používá často, jedná se o logickou proměnnou - bude obsahovat buď true nebo false (klidně bychom to mohli udělat pomocí 0 a 1). Pojmenuji jí **pohyb**, když v ní bude true, značí to, že se má šipka točit, při false bude stát. Při spuštění programu je třeba samozřejmě nastavit do této proměnné true, aby se šipka točila.



Takto vypadá nastavení proměnné **pohyb**. Bloky true a false najdete v sekci Logic. Takto nastavím proměnnou pohyb pro stisku tlačítka A na false, ale já ji potřebuji nastavit na false, když měla hodnotu true a na true, pokud měla hodnotu false. To uděláme pomocí bloku if, do kterého ale přidáme možnost mít dva výsledky:



Klikněte na kolečko.



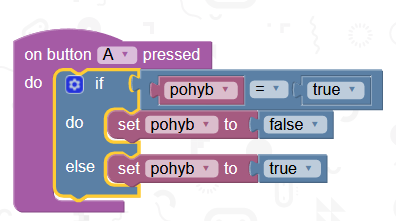
Znovu kliknu na kolečko a mám blok připraven:



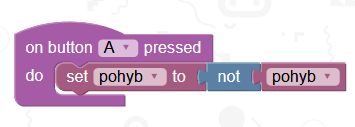
Za ***if*** sestavím podmínku - zjistím, zda proměnná pohyb je nastavena na true.

Za ***do*** napíšu, co se má stát, když je podmínka splněna - v **pohyb** je nastaveno true a já to potřebuji změnit, takže nastavím pohyb na false.

Za ***else*** napíšu, co se má stát, když podmínka není splněna - v pohyb není true (to znamená, že tam je false), takže tam nastavím true

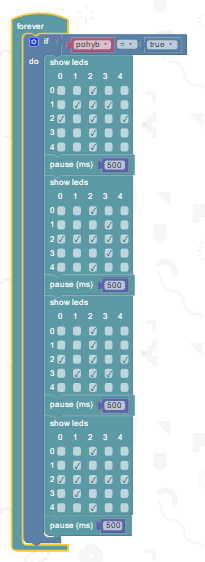


Poznámka pro znalce - lze to také udělat bez if tímto způsobem:



Už máme vyřešenou proměnnou, která má určovat pohyb a zastavení, ale ještě jsme ji nepoužili k zastavení a rozpohybování.

Budeme ji muset zabudovat do opakování, ve kterém se střídají obrázky šipek. Pohybování šipky bude probíhat pouze v tom případě, že v proměnná **pohyb** má hodnotu true.



Vyzkoušejte si, jak se program chová. Když zmáčknete A, tak se šipka nezastaví hned, ale až když je natočená vlevo, že? No vždyť přesně to máme v programu. My potřebujeme šipku zastavit hned. A navíc po spuštění se začne pohybovat hned na další pozici a ne až od začátku.

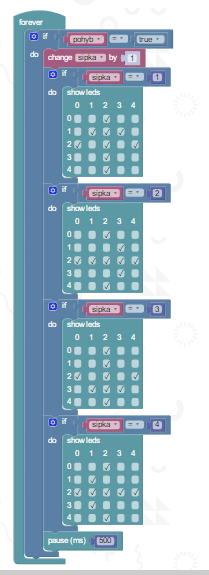
Musím to udělat tak, aby program kontroloval zastavení před každou změnou šipky a také si pamatoval, jak byla šipka zastavena.

Proto si vytvořím novou proměnnou **sipka**. Tato proměnná bude mít hodnotu 1, 2, 3 nebo 4 - pro každý obrázek šipky jednu. V cyklu forever budu nejdříve zjišťovat, jestli se má šipka pohybovat. Pokud ano, zvětším hodnotu proměnné **sipka** o 1, tím budu mít zajištěnu změnu obrázku na další. To provedu blokem



Tento blok změní hodnotu vybrané proměnné o tolik, kolik zadáme.

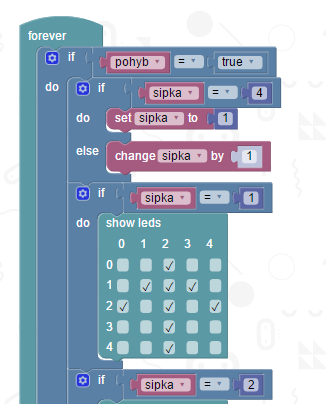
A pak ukážu jen jeden obrázek - ten, který je na řadě (to je přeci číslo v proměnné **sipka**). To samozřejmě musím přidat ke každému obrázku jako podmínku.



Pauzu jsem dal pouze jednu na konec, protože se vlastně provede v každém opakování cyklu pouze jedna změna obrázku, je tedy zbytečné ji mít u každého zvlášť.

Když si zkusíte program, tak se šipka otáčí, lze ji stopnout a zase rozpohybovat tlačítkem A, ale otočí se jen jednou dokola.

To musí být samozřejmě tím, že se nám proměnná šipka pořád zvyšuje o 1. Když se dostanu na číslo 4, tak další je 5, ale my potřebujeme 1. Musíme tedy na místo, kde se mění hodnota proměnné **sipka** přidat podmínku, která mám hodnotu 4 změní na 1.



Jinak se nic nemění.

Vyzkoušíme program a funguje to :-)

**Rozšíření:**

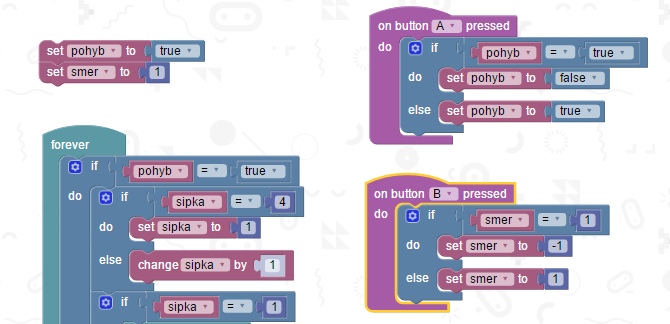
Pokud chcete rozšířit program o možnost změnit směr otáčení stiskem tlačítka B, budete potřebovat proměnnou, která do které si tuto informaci uložíte.

Vytvořím si novou proměnnou **smer**. Nebudu do ní ukládat hodnoty true a false, ale 1 a -1.

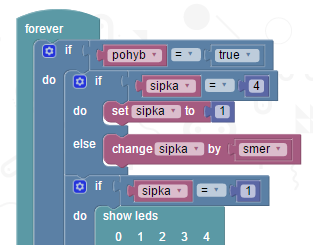
Udělám to z toho důvodu, že v programu měním číslo v proměnné sipka o 1, tím ho přičítám. Když ale dám hodnotu -1, tak se vlastně vrátím o obrázek dopředu a tím změním směr otáčení. (Jde to samozřejmě udělat i s true a false a není to o moc složitější)

Na začátku si nastavím **smer** na 1.

Naprogramuji také tlačítko B podobně jako A jen proměnnou **smer** a hodnotami 1 a -1.



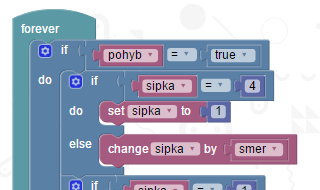
V cyklu forever nyní budu měnit hodnotu proměnné šipka ne o 1, ale o hodnotu v proměnné **smer**.



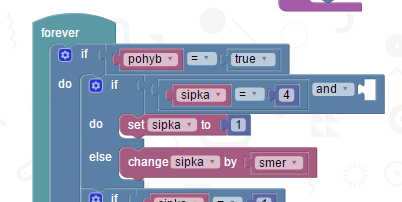
Program funguje ale jen do chvíle, kdy šipka rotuje proti směru hodinových ručiček do polohy vzhůru. Pak totiž dojde k tomu, že se hodnota v **sipka** změní na 0.

Je potřeba přidělat podmínku, která řeší přechod z 1 na 4 při točení proti směru hodinových ručiček. A když jsme u toho, je třeba si všimnout, že už máme podmínku, která řeší přechod z 4 na 1. Ta by měla probíhat pouze v případě, když se točíme ve směru hodinových ručiček.

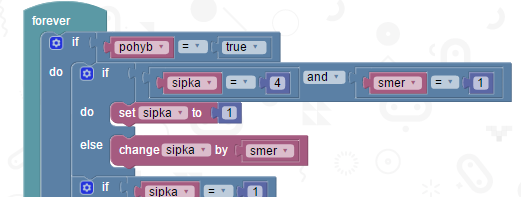
Nejdříve předěláme podmínku, kterou už máme:



Tento přechod ze 4 na 1 může nastat jen když se točíme ve směru hodinových ručiček, tedy když je v proměnné **smer** hodnota 1. Potřebujeme vlastně podmínku, ve které jsou dvě podmínky, které mají platit současně - na to máme blok **and** v sekci Logic.



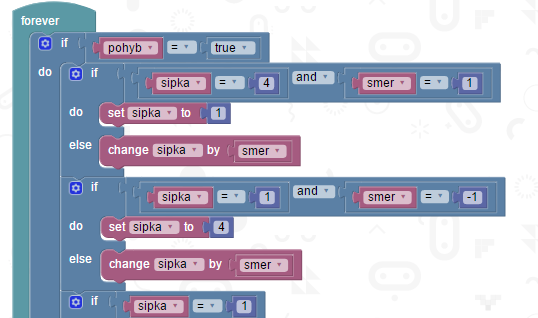
Doplním druhou podmínku pro proměnnou **smer**.



Program funguje pořád stejně - tedy při couvání pořád nefunguje.

Tak ještě přechod z 1 na 4 při točení proti směru hodinových ručiček - to je v proměnné **smer** hodnota -1.

Přidáme za podmínku, kterou jsme právě předělali skoro stejnou podmínku, jen přecvakneme pár věcí:

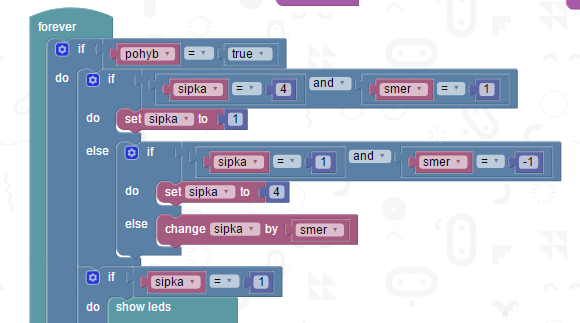


Uf. Vyzkoušíme a ono to zase dělá něco divného - jen se přepíná šipka vlevo a vpravo.

To je tím, že mám za sebou dvě podmínky, které v případě, že nejsou ani jedna splněny vlastně dvakrát změní hodnotu proměnné **sipka**.

Je třeba zjisti, jestli platí první podmínka. Pokud ne, musím zjistit, jestli platí druhá podmínka a pokud ani ta není splněna, pak teprve změním hodnotu proměnné **sipka** automaticky.

Podívejte, jak se to udělá:



Podmínka je strčená v podmínce.

A teď už by mělo vše fungovat tak, jak má.