

Název úlohy	Poplach, poplach!
Třída	5. třída
Úloha splňuje rámce	<ul style="list-style-type: none"><li>• ALGORITMIZACE A PROGRAMOVÁNÍ – řešení problému krokováním, programování, kontrola řešení</li></ul>
Propojení s RVP výstupy	<ul style="list-style-type: none"><li>• <b>I-5-2-03</b> - Žákyně/žák v blokově orientovaném programovacím jazyce sestaví program; rozpozná opakující se vzory, používá opakování a připravené podprogramy</li></ul>
Propojení s ŠVP výstupy	<ul style="list-style-type: none"><li>• Žákyně/žák v blokově orientovaném programovacím jazyce sestaví program řídící chování micro:bitu</li></ul>
Časová náročnost	45 minut (1 vyučovací hodina)
Stručný popis úlohy	Skupinová aktivita na bázi řetězové reakce
Odkaz na rozšíření	<a href="https://github.com/microbit-cz/pxt-alarm-extension">https://github.com/microbit-cz/pxt-alarm-extension</a>
Odkaz na řešení	<a href="https://github.com/microbit-cz/pxt-alarm-demo-easy">https://github.com/microbit-cz/pxt-alarm-demo-easy</a>

## Poplach, poplach!

### Začátek

Tato úloha funguje jako alarm, který se spustí po zmáčknutí tlačítka (nebo provedení jakéhokoli jiného vstupu). Funguje cyklicky, což znamená, že pokud máme v dosahu microbity se správným programem (kódem), zapnutí alarmu jim pošle příkaz, aby také začali houkat. Tyto microbity to opět pošlou dál. Tímto způsobem se vytvoří řetězová reakce.

Vypnutí se poté dá provést z jakéhokoli microbitu. Ten pak stejným způsobem pošle dál příkaz o přerušení alarmu, který se bude dostávat dál a dál, než se vypnou všechny microbity.



### Žák/Žákyně se v této úloze naučí/procvičí



- Využití sekce „Radio“
- Vytvoření a použití proměnné
- Vytvoření úplného podmíněného příkazu (if a else)
- Využití zvukové (a obrazové) signalizace

### Co budete potřebovat

- PC s přístupem k [MakeCode](#)
- Propojovací USB kabel s micro USB koncovkou
- Micro:bit (nejméně 2-3)
- Větší prostor (např. školní chodba)

### Rozšíření

Rozšíření	Popis rozšíření
<b>Spustí alarm a pošli pokyn</b> 	<ul style="list-style-type: none"><li>• spustí alarm a pošle všem zařízením v okolí pokyn ke spuštění alarmu (blok se dá rozšířit o vlastní parametr)</li><li>• parametry<ul style="list-style-type: none"><li>◦ message (nullable string)</li></ul></li><li>• bez návratové hodnoty</li></ul>
<b>Vypni alarm a pošli pokyn</b> 	<ul style="list-style-type: none"><li>• vypne alarm a pošle všem zařízením v okolí pokyn k vypnutí alarmu (blok se dá rozšířit o vlastní parametr)</li><li>• parametry<ul style="list-style-type: none"><li>◦ message (nullable string)</li></ul></li><li>• bez návratové hodnoty</li></ul>

<p><b>Přijmout pokyn</b></p> 	<ul style="list-style-type: none"> <li>• přijme a zpracuje pokyn od jiného microbitu</li> <li>• do parametrů se jednoduše přetáhnou argumenty z bloku „když je přijata hodnota“</li> <li>• parametry: <ul style="list-style-type: none"> <li>○ name (text)</li> </ul> </li> <li>• návratová hodnota <ul style="list-style-type: none"> <li>○ stav alarmu po přijetí pokynu (bool)</li> </ul> </li> </ul>
<p><b>Při zapnutí alarmu</b></p> 	<ul style="list-style-type: none"> <li>• provede akci v moment, kdy je zapnutý alarm</li> <li>• parametry <ul style="list-style-type: none"> <li>○ action (delegát)</li> </ul> </li> <li>• bez návratové hodnoty</li> </ul>

## Možný postup v úloze

*Žáci by mohli z popisu zadání poznat, že se v této úloze opět využije rádio a tím pádem by se jich učitel mohl zeptat, co se musí nastavit, aby rádio správně fungovalo. Na mysli máme rádiovou skupinu.*

### 1. Vytvoření kódu v bloku *při startu*

Jako nepovinný krok můžeme na microbitu ukázat nějaký tvar, v našem případě jsme zvolili zvonek, který má dvě podoby – zvoní a nezvoní.



*zvoní – alarm je zapnutý*



*nezvoní – alarm je vypnutý*

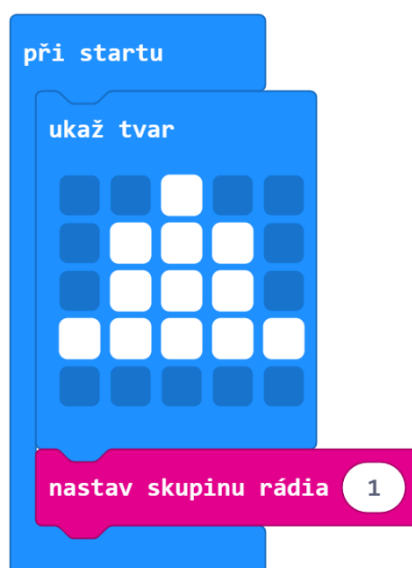
Při počátečním zapnutí microbit nikdy nebude houkat, tudíž nastavujeme obrázek nezvonícího zvonku.

Nejdříve si nastavíme rádiovou skupinu.



*Skupina komunikujících microbitů musí mít stejné číslo skupiny.*

*Pokud bychom chtěli třídu rozdělit do více týmů a nechat spolu komunikovat pouze určité microbity, nastavíme každému týmu jiné číslo skupiny rádia.*



## 2. Zprovoznění vstupů (např. tlačítek)

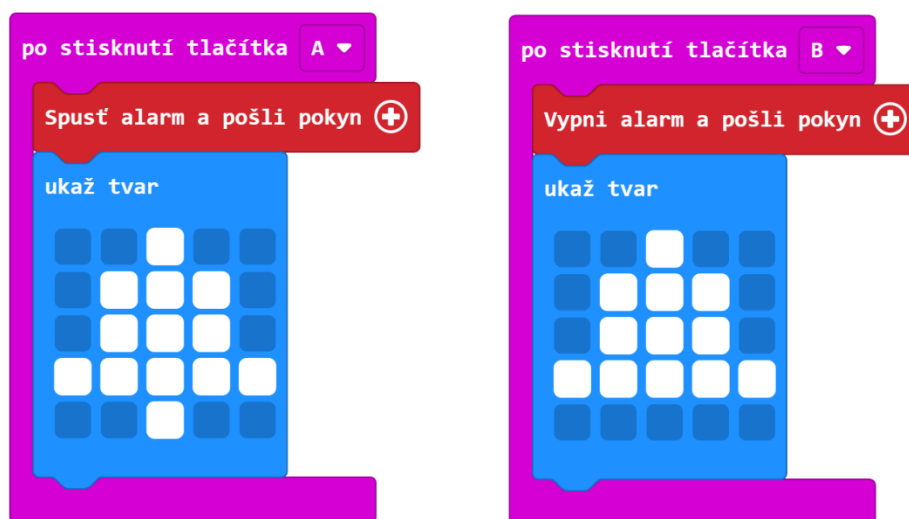
Nyní si zvolíme, jak chceme alarm zapínat a vypínat. V našem příkladu jsou jako vstupy zvoleny tlačítka.

V příkladu je *tlačítko A* zvoleno na zapínání alarmu. Vložíme do něj tedy vhodný blok.

*Pokud jsme se rozhodli zobrazovat na microbitu tvary, doplníme do bloku ten, který znázorňuje zapnutí.*

Do *tlačítka B*, nebo jakýkoliv jiný vstup, který jsme si zvolili na vypnutí alarmu, znovu vložíme patřičný blok na vypínání alarmu.

*Pokud jsme se rozhodli zobrazovat na microbitu tvary, doplníme do bloku ten, který znázorňuje vypnutí.*



## 3. Vytvoření podmínky o přijetí pokynu

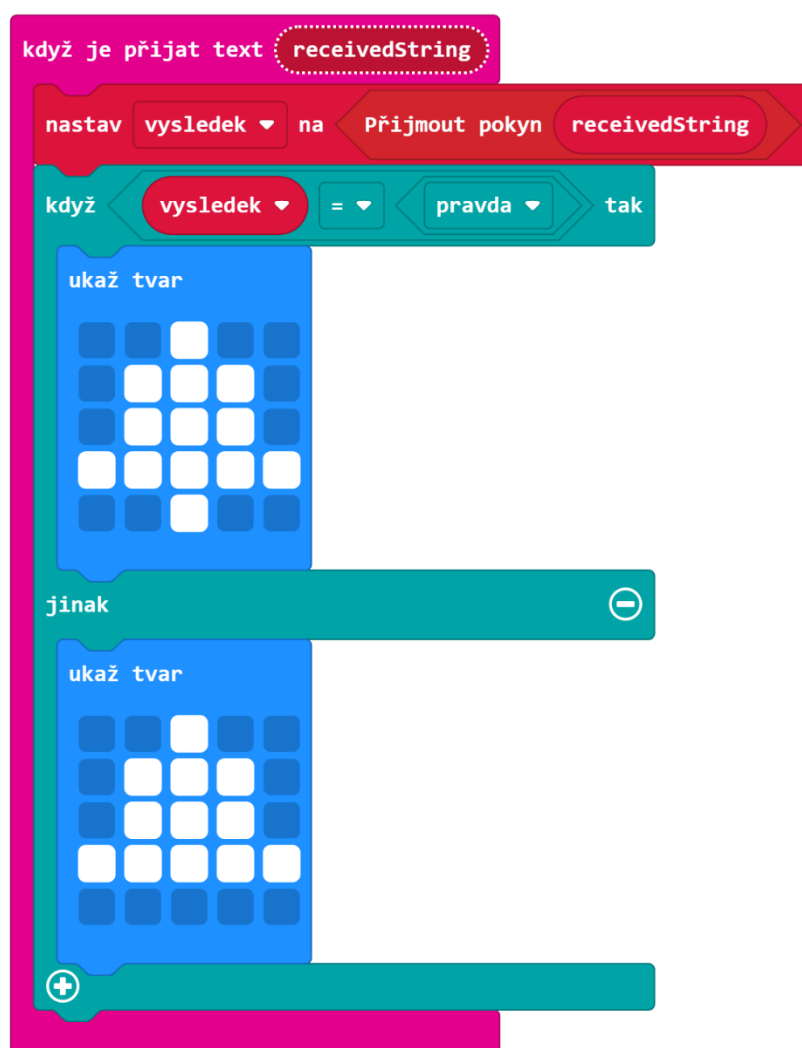
Jako další věc musíme zajistit, že microbit bude přijímat pokyny od ostatních microbitů. K tomu použijeme blok ze sekce *Rádio*.

Vytvoříme si proměnnou, kterou nazveme *výsledek*.

Proměnnou *výsledek* potom nastavíme na hodnotu, kterou získáme tak, že si zavoláme blok *Přijmout pokyn*. Tak se nám do proměnné uloží true nebo false (pravda/nepravda).

Nyní vytvoříme podmínku if a else. Tu vezmeme ze sekce *Logika*. V podmínce zjistíme hodnotu proměnné *výsledek*. Pokud bude mít *výsledek* hodnotu true (pravda), zobrazíme tvar zvonícího zvonku, pokud bude mít jakoukoli jinou hodnotu, zobrazíme tvar nezvonícího zvonku.

# Poplach, poplach!



## 4. Doplnění zvukového signálu při spuštění alarmu

Jako poslední už jenom do bloku *Při zapnutí alarmu* vložíme jakýkoliv zvuk, který chceme, aby se přehrál v momentu zapnutí alarmu.

