

Název úlohy	Poplach, poplach!
Třída	6. třída
Úloha splňuje rámce	<ul style="list-style-type: none">• ALGORITMIZACE A PROGRAMOVÁNÍ – řešení problému krokováním, programování, kontrola řešení
Propojení s RVP výstupy	<ul style="list-style-type: none">• I-9-2-05 - Žákyně/žák v blokově orientovaném programovacím jazyce vytvoří přehledný program s ohledem na jeho možné důsledky a svou odpovědnost za něj; program vyzkouší a opraví v něm případné chyby; používá opakování, větvení programu, proměnné
Propojení s ŠVP výstupy	<ul style="list-style-type: none">• Žákyně/žák v blokově orientovaném programovacím jazyce sestaví program pro ovládání micro:bitu, v programu najde a opraví chyby
Časová náročnost	45 minut (1 vyučovací hodina)
Stručný popis úlohy	Skupinová aktivita na bázi řetězové reakce
Odkaz na rozšíření	https://github.com/microbit-cz/pxt-alarm-extension
Odkaz na řešení	https://github.com/microbit-cz/pxt-alarm-demo-hard

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
- Porovnávání vlastních proměnných
- Konstanty
- 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

Popis rozšíření

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

Poplach, poplach!

Při zapnutí alarmu

Při zapnutí alarmu



- provede akci v moment, kdy je zapnutý alarm
- parametry
 - action (delegát)
- bez návratové hodnoty

Možný postup v úloze

1. Vytvoření kódu v bloku *při startu*

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.

Jako další si vytvoříme dvě proměnné – v příkladu pojmenované *onText* a *offText*.

Do nich uložíme text – v příkladu „alarm_on“ a „alarm_off“. Tyto zprávy budeme v dalším kroku posílat jako pokyn dalším microbitům.



Do proměnných se ve výchozím stavu dají vložit pouze čísla. Abychom do nich mohli vložit zmíněný text, musíme ze sekce Text přetáhnout do bloku s proměnnou na místo čísla hned první blok, ve kterém jsou pouze uvozovky.



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

Všichni žáci tedy musí mít stejný přijímací a odesílací text.

Poplach, poplach!



2. Zprovoznění vstupů (např. tlačítek)

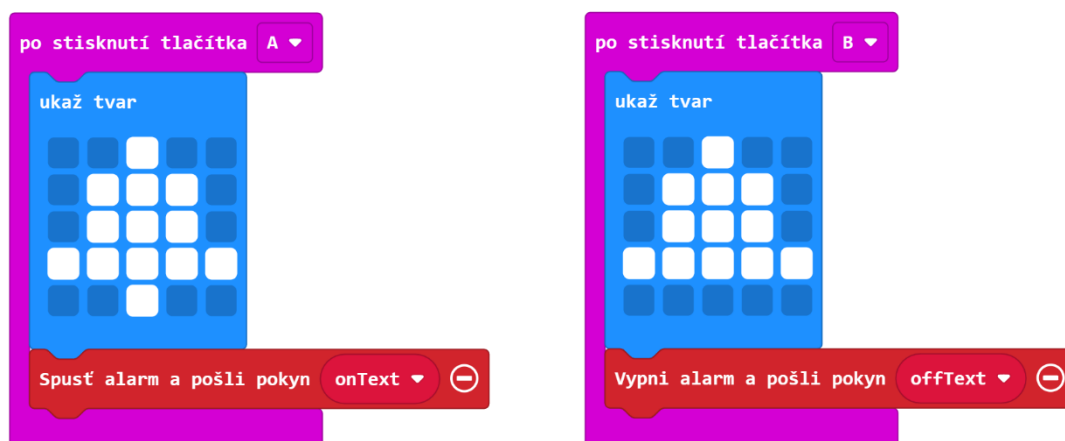
Nyní si zvolíme, jak chceme alarm zapínat a vypínat. V našem příkladu jsou jako vstupy zvoleny tlačítka.

V příkladu je *tlačítko A* zvoleno na zapínání alarmu. Vložíme do něj tedy vhodný blok a pošleme patřičnou proměnnou (při spouštění alarmu proměnnou se zprávou „alarm_on“ a při vypnutí „alarm_off“).

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 se správnou proměnnou.

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 pro přijímání textu ze sekce *Rádio*.

Přijatý text *receivedString* v podmíněném větvení porovnáme s naší proměnnou *onText* a *offText*. Tentokrát bude podmínka místo *else* (jinak) obsahovat *else if* (jinak když).

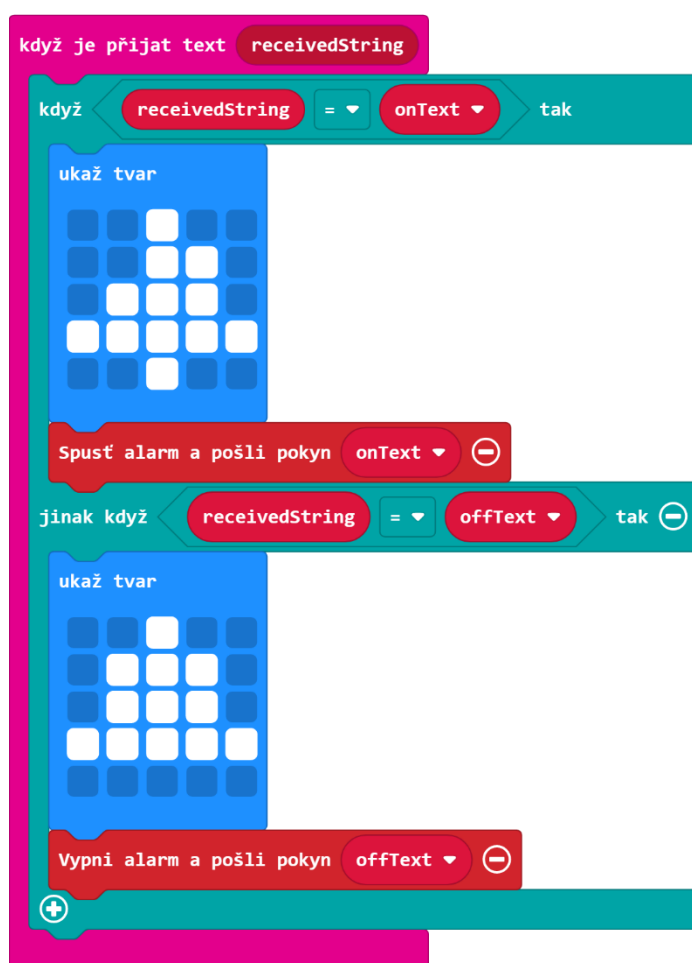


Podmínku else if získáme tak, že si vezmeme blok s podmínkou if a else a klikneme na tlačítko plus v pravém dolním rohu. Tím si přidáme možnost else if. Else pak můžeme jednoduše odstranit kliknutím na mínus vedle podmínky a máme to.

Na microbit nám může přijít cokoli, nás ale zajímají zprávy, které mají námi definovaný text („alarm_on/off“). Pokud dostaneme „alarm_on“, zapneme alarm a pošleme ostatním microbitům zprávu v proměnné *onText*, aby začaly houkat.

Pokud dostaneme „alarm_off“, alarm vypneme a také dáme vědět ostatním zprávou v proměnné *offText*.

Pokud zobrazujeme tvary, vložíme každý do patřičné části větvení.



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.

