

# Metodická příručka

Programové bloky v prostředí LEGO Mindstorms EV3



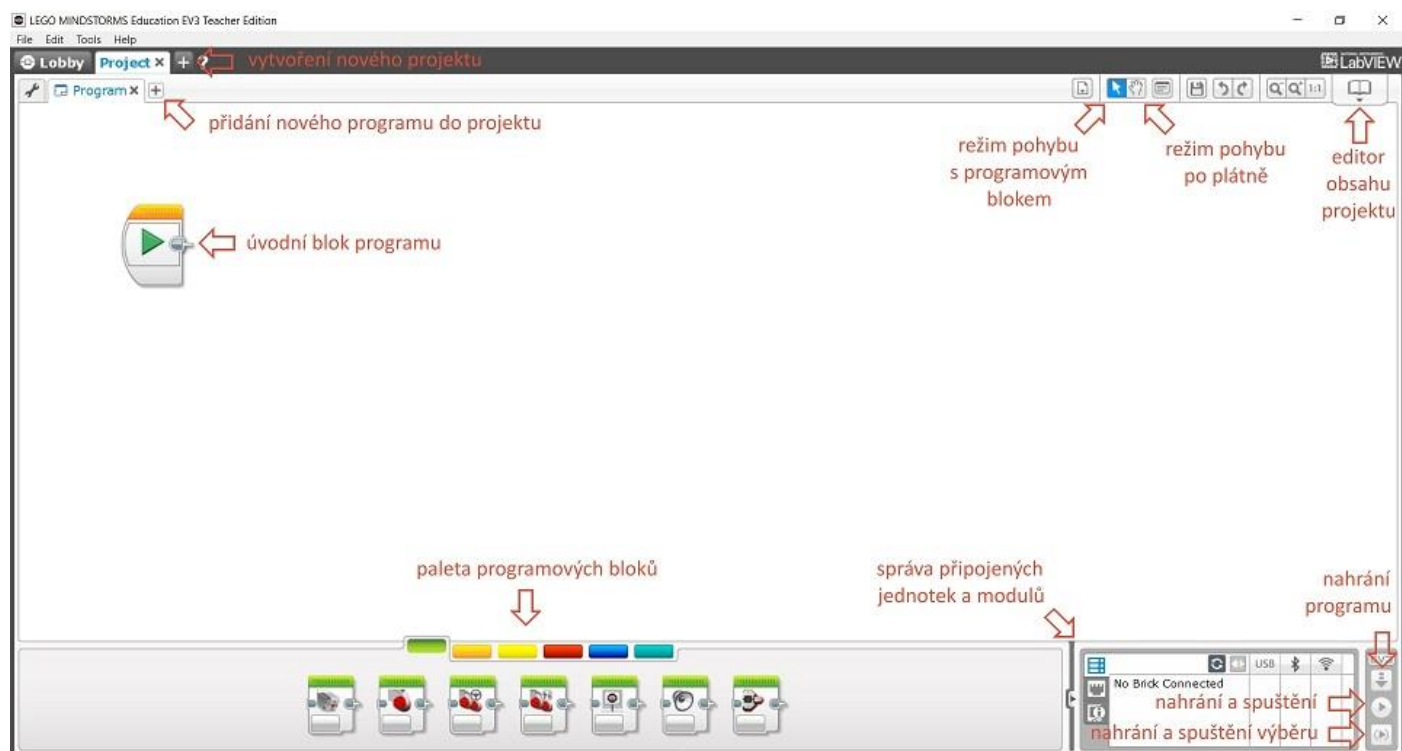
Mgr. Jan Batko

KVD FPE ZČU

# Obsah

Základní orientace v EV3.....	2
Obecný popis programového bloku .....	3
Medium Motor (lineární motor).....	4
Large Motor (řízení jednoho servomotoru).....	6
Move Steering (řízení dvou servomotorů u kolového vozidla) .....	8
Move Tank (řízení dvou servomotorů u pásového vozidla) .....	10
Display (vykreslování a vypisování na displej) .....	12
Sound (zvukový výstup) .....	16
Brick Status Light (světelná signalizace řídicí jednotky) .....	18
Wait (čekání/oddálení vykonávání dalšího příkazu).....	19
Loop (cyklus) .....	21
Switch (podmínka) .....	23
Color Sensor (barevný senzor).....	25
Motor Rotation (měřič otáček motoru).....	28
Timer (časovač).....	30
Touch Sensor (dotykový senzor).....	31
Ultrasonic Sensor (ultrazvukový senzor) .....	32
Variable (proměnná).....	34
Math (matematické operace).....	36
Compare (porovnání hodnot).....	37
Text (sloučení textových řetězců).....	38
Random (generování náhodného čísla).....	39
Tvorba vlastních metod .....	40

# Základní orientace v EV3



Obrázek 1 - Umístění základních funkcí v programovacím prostředí LEGO Mindstorms EV3.

Po spuštění programovacího prostředí EV3 se **vytvoří první projekt**. Jeho součástí může být libovolný počet programů a experimentů. Přidávat nové programy a experimenty nebo vytvořit nový projekt můžeme pomocí tlačítek **+** v levé horní části programu (viz obrázek).

**Úvodní blok** uvozující začátek každého programu. K němu se připojují další programové bloky. Úvodní blok obsahuje tlačítko pro nahrání programu do řídicí jednotky robota a jeho okamžité spuštění.

**Přesun programových bloků** je možné provádět pouze po přepnutí do příslušného režimu pomocí tlačítka s šipkou umístěného v pravé horní části EV3. 4

**Režim posunu po plátně** se aktivuje tlačítkem ve tvaru ruky. Usnadňuje orientaci a pohyb v rozsáhlejšímu programu.

**Editor obsahu projektu** lze naplňovat obrázky, videi, zvuky nebo textem. Vše je poté součástí uloženého projektu.

**Programové bloky** jsou rozděleny do pěti logických kategorií rozlišených barevně. Do poslední kategorie se umísťují **vlastní vytvořené bloky**.

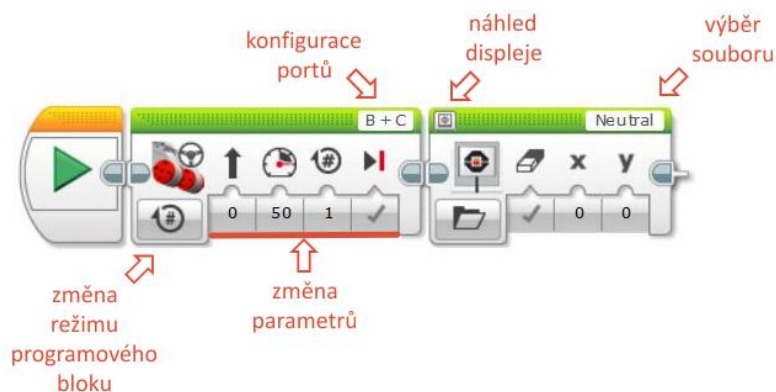
Secke umístěná v pravé dolní části je rozdělena do tří částí:

**Správa připojených jednotek** – obsahuje přehled připojených řídicích jednotek a způsob jejich připojení (USB, Bluetooth nebo Wifi).

**Přehled portů** – obsahuje informace o připojených motorech a senzorech jako je číslo portu řídicí jednotky, připojené zařízení, aktuální hodnota natočení (u motorů), aktuálně naměřená hodnota (u senzorů).

**Informace o řídicí jednotce** – obsahuje informaci o verzi firmware řídicí jednotky, způsobu připojení a míře zaplnění interní paměti. Pomocí tlačítek se můžeme přepnout do správy bezdrátových připojení nebo správy paměti.

# Obecný popis programového bloku



Obrázek 2 - Obecný popis částí běžného programového bloku.

## Běžné části programového bloku

Následující tři části obsahují všechny běžné programové bloky ovládající vstupní (senzory) či výstupní zařízení (motory).

**B + C** **Konfigurace portů** – slouží ke specifikaci vstupních nebo výstupních portů řídicí jednotky, ke kterým jsou motory nebo senzory připojeny.

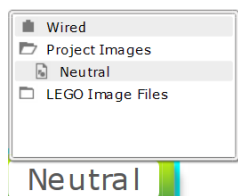
**Změna režimu programového bloku** – část, kterou obsahují všechny programové bloky. S její pomocí je možné měnit režim programového bloku, ve kterém blok aktuálně pracuje.

**Změna parametrů** – chování programového bloku ovlivňují parametry

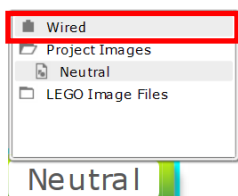
## Specifické části programového bloku

Specifické programové bloky obsahují některé netradiční části, které umožňují další funkcionalitu bloku.

**Náhled displeje** – programový blok *Display* obsahuje v levém horním rohu tlačítko pro zobrazení náhledu. Díky tomu si můžeme ještě před spuštěním programu ověřit, na jakém místě a v jaké podobě se bude požadovaný výstup (výpis) zobrazovat.



**Výběr souboru** – programový blok *Display* umožňuje vykreslit grafický soubor uložený v paměti řídicí jednotky na plochu displeje. Po kliknutí na ikonku v pravém horním rohu se zobrazí seznam dostupných adresářů s grafickým soubory.



**Zobrazení vstupního portu** – po kliknutí do pravého horního rohu bloku (u motorů a senzorů do pole pro změnu portů řídicí jednotky) je možné do bloku přidat vstupní port. S jeho pomocí je poté možné přivádět data na vstup. Slouží k tomu položka označená *Wired*.

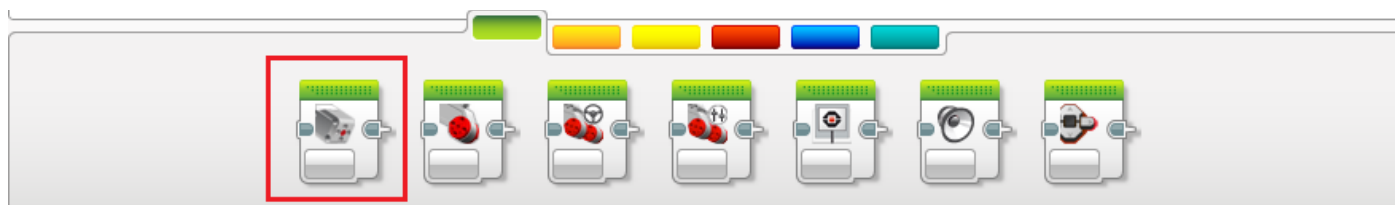
Zcela odlišnou strukturu a vzhled mají bloky pro kontrolu toku dat (*Flow Control*). Jedná se například o blok podmíněného vykonávání (*Switch*) nebo cyklus (*Loop*). Těmto blokům se věnujeme v samostatné kapitole.

# Medium Motor (lineární motor)

Kde blok v učebnici využíváme?

- Kapitola 5. mixér (pohon mixéru)
- Kapitola 6. závora (ovládání závory)

Umístění programového bloku



## Režimy programového bloku

Programový blok slouží k ovládání lineárního motoru. Jeho identifikace probíhá pomocí pole pro výběr portu (A-D). Blok může pracovat v následujících režimech:

### OFF (VYPNUTO)



▶ **Break at End** – způsob zastavení motoru. Okamžité zastavení (**Break**) reprezentuje logická hodnota True a pozvolné dojetí (**Coast**) hodnota False.

### ON (ZAPNUTO)



🌀 **Power** – rychlost otáčení motoru udávaná v procentech na stupnici od 100 (vpřed) do -100 (vzad).

### ON FOR SECONDS (OTÁČENÍ PO ZADANÝ POČET VTEŘIN)



🌀 **Power** – rychlost otáčení motoru udávaná v procentech na stupnici od 100 (vpřed) do -100 (vzad).

🕒 **Seconds** – doba otáčení motoru udávaná ve vteřinách.

▶ **Break at End** – způsob zastavení motoru. Okamžité zastavení (**Break**) reprezentuje logická hodnota True a pozvolné dojetí (**Coast**) hodnota False.

### ON FOR DEGREES (OTOČENÍ O ZADANÝ ÚHEL)






🌀 **Power** – rychlost otáčení motoru udávaná v procentech na stupnici od 100 (vpřed) do -100 (vzad).

🕒 **Degrees** – míra natočení zadávaná v úhlových stupních od 0 do 360°.

▶ **Break at End** – způsob zastavení motoru. Okamžité zastavení (**Break**) reprezentuje logická hodnota True a pozvolné dojetí (**Coast**) hodnota False.

## ON FOR ROTATIONS (OTOČENÍ O POČET OTÁČEK)



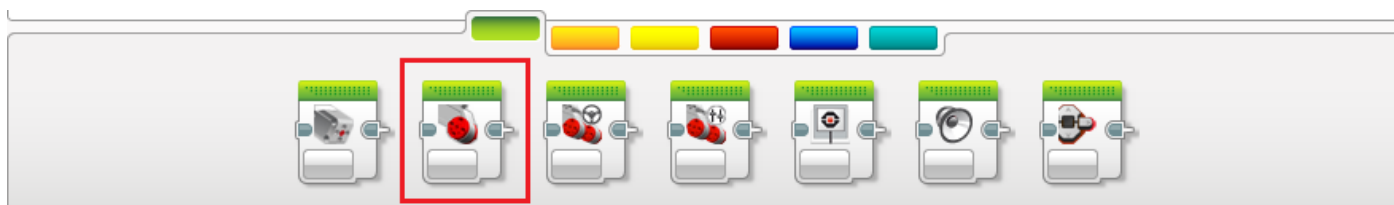
-  **Power** – rychlost otáčení motoru udávaná v procentech na stupnici od 100 (vpřed) do -100 (vzad).
-  **Rotations** – počet prováděných otáček motoru.
-  **Break at End** – způsob zastavení motoru. Okamžité zastavení (**Break**) reprezentuje logická hodnota True a pozvolné dojetí (**Coast**) hodnota False.

# Large Motor (řízení jednoho servomotoru)

Kde blok v učebnici využíváme?

- Kapitola 2. oživení robota (zatáčení)
- Kapitola 3. robot ve městě (zatáčení)
- Kapitola 9. inteligentní robot (kmitavý pohyb po čáře)
- Kapitola 10. parkovací asistent (parkování)

Umístění programového bloku



## Režimy programového bloku

Programový blok slouží k ovládání jednoho servomotoru. Jeho identifikace probíhá pomocí pole pro výběr portu (A-D). Blok může pracovat v následujících režimech:

### OFF (VYPNUTO)



► **Break at End** – způsob zastavení motoru. Okamžité zastavení (**Break**) reprezentuje logická hodnota True a pozvolné dojetí (**Coast**) hodnota False.

### ON (ZAPNUTO)



⚙️ **Power** – rychlost otáčení motoru udávaná v procentech na stupnici od 100 (vpřed) do -100 (vzad).

### ON FOR SECONDS (OTÁČENÍ PO ZADANÝ POČET VTEŘIN)



⚙️ **Power** – rychlost otáčení motoru udávaná v procentech na stupnici od 100 (vpřed) do -100 (vzad).

⌚ **Seconds** – doba otáčení motoru udávaná ve vteřinách.

► **Break at End** – způsob zastavení motoru. Okamžité zastavení (**Break**) reprezentuje logická hodnota True a pozvolné dojetí (**Coast**) hodnota False.

### ON FOR DEGREES (OTOČENÍ O ZADANÝ ÚHEL)





⚙️ **Power** – rychlost otáčení motoru udávaná v procentech na stupnici od 100 (vpřed) do -100 (vzad).

⦿ **Degrees** – míra natočení zadávaná v úhlových stupních od 0 do 360°.

- ▶ **Break at End** – způsob zastavení motoru. Okamžité zastavení (**Break**) reprezentuje logická hodnota True a pozvolné dojetí (**Coast**) hodnota False.

## ON FOR ROTATIONS (OTOČENÍ O POČET OTÁČEK)



-  **Power** – rychlost otáčení motoru udávaná v procentech na stupnici od 100 (vpřed) do -100 (vzad).
-  **Rotations** – počet prováděných otáček motoru.
- ▶ **Break at End** – způsob zastavení motoru. Okamžité zastavení (**Break**) reprezentuje logická hodnota True a pozvolné dojetí (**Coast**) hodnota False.

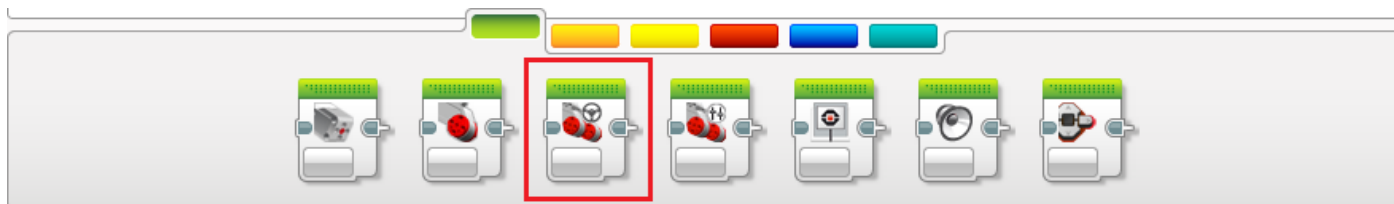


# Move Steering (řízení dvou servomotorů u kolového vozidla)

## Kde blok v učebnici využíváme?

- Kapitola 1. stavíme robota (první pohyb robota)
- Kapitola 2. oživení robota (pohyb robota vpřed, zatáčení)
- Kapitola 3. robot ve městě (pohyb robota vpřed, zatáčení)
- Kapitola 8. adaptivní tempomat (pohyb robota vpřed)
- Kapitola 9. inteligentní robot (pohyb robota vpřed)
- Kapitola 10. parkovací asistent (pohyb robota vpřed, parkování)

## Umístění programového bloku



## Režimy programového bloku

Programový blok slouží k ovládání dvou servomotorů v režimu řízení robota poháněného pomocí kol. Jeho identifikace probíhá pomocí pole pro výběr portu (A-D). Blok může pracovat v následujících režimech:

### OFF (VYPNUTO)



▶ **Break at End** – způsob zastavení motoru. Okamžité zastavení (**Break**) reprezentuje logická hodnota True a pozvolné dojetí (**Coast**) hodnota False.

### ON (ZAPNUTO)



↑ **Steering** – řízení směru natáčení robota.  
⤿ **Power** – rychlost otáčení motorů udávaná v procentech na stupnici od 100 (vpřed) do -100 (vzad).




### ON FOR SECONDS (OTÁČENÍ PO ZADANÝ POČET VTEŘIN)



↑ **Steering** - řízení směru natáčení robota.  
⤿ **Power** – rychlost otáčení motoru udávaná v procentech na stupnici od 100 (vpřed) do -100 (vzad).  
⌚ **Seconds** – doba otáčení motoru udávaná ve vteřinách.  
▶ **Break at End** – způsob zastavení motoru. Okamžité zastavení (**Break**) reprezentuje logická hodnota True a pozvolné dojetí (**Coast**) hodnota False.




## ON FOR DEGREES (OTOČENÍ O ZADANÝ ÚHEL)



- ↑ **Steering** - řízení směru natáčení robota.
-  **Power** – rychlost otáčení motoru udávaná v procentech na stupnici od 100 (vpřed) do -100 (vzad).
-  **Degrees** – míra natočení zadávaná v úhlových stupních od 0 do 360°.
-  **Break at End** – způsob zastavení motoru. Okamžité zastavení (**Break**) reprezentuje logická hodnota True a pozvolné dojetí (**Coast**) hodnota False.

## ON FOR ROTATIONS (OTOČENÍ O POČET OTÁČEK)



- ↑ **Steering** - řízení směru natáčení robota.
-  **Power** – rychlost otáčení motoru udávaná v procentech na stupnici od 100 (vpřed) do -100 (vzad).
-  **Rotations** – počet prováděných otáček motoru.
-  **Break at End** – způsob zastavení motoru. Okamžité zastavení (**Break**) reprezentuje logická hodnota True a pozvolné dojetí (**Coast**) hodnota False.

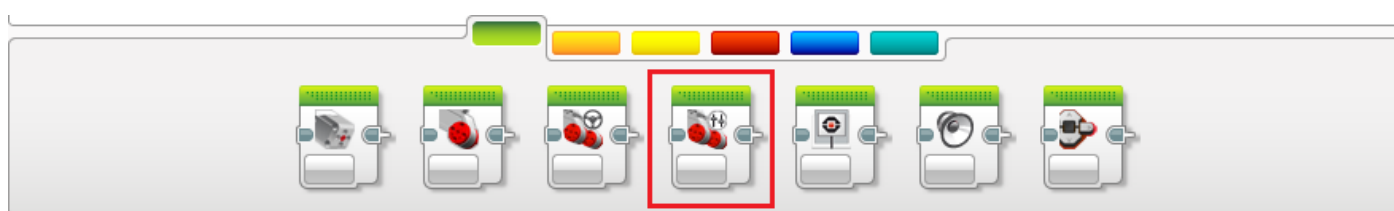
# Move Tank (řízení dvou servomotorů u pásového vozidla)

**Upozornění:** Programový blok primárně slouží k ovládní robotů opatřených pásy. Žáci ho ale velice snadno mohou použít i pro ovládní robota poháněného motory, a to zejména pro pohyb vpřed. Natáčení se s jeho pomocí realizuje velice těžko. Proto doporučujeme využívat primárně blok *Move Steering*.

## Kde by se dal blok v učebnici využít?

- Kapitola 1. stavíme robota (první pohyb robota)
- Kapitola 2. oživení robota (pohyb robota vpřed)
- Kapitola 3. robot ve městě (pohyb robota vpřed)
- Kapitola 8. adaptivní tempomat (pohyb robota vpřed)
- Kapitola 9. inteligentní robot (pohyb robota vpřed)
- Kapitola 10. parkovací asistent (pohyb robota vpřed)

## Umístění programového bloku



## Režimy programového bloku

Programový blok slouží k ovládní dvou servomotorů v režimu řízení pásového vozidla. Jeho identifikace probíhá pomocí pole pro výběr portu (A-D). Blok může pracovat v následujících režimech:

### OFF (VYPNUTO)



▶ **Break at End** – způsob zastavení motoru. Okamžité zastavení (**Break**) reprezentuje logická hodnota True a pozvolné dojetí (**Coast**) hodnota False.

### ON (ZAPNUTO)



⚙ **Power** – rychlost otáčení motorů udávaná v procentech na stupnici od 100 (vpřed) do -100 (vzad).

### ON FOR SECONDS (OTÁČENÍ PO ZADANÝ POČET VTEŘIN)






⚙ **Power** – rychlost otáčení motoru udávaná v procentech na stupnici od 100 (vpřed) do -100 (vzad).

⌚ **Seconds** – doba otáčení motoru udávaná ve vteřinách.

▶ **Break at End** – způsob zastavení motoru. Okamžité zastavení (**Break**) reprezentuje logická hodnota True a pozvolné dojetí (**Coast**) hodnota False.




## ON FOR DEGREES (OTOČENÍ O ZADANÝ ÚHEL)



-  **Power** – rychlost otáčení motoru udávaná v procentech na stupnici od 100 (vpřed) do -100 (vzad).
-  **Degrees** – míra natočení zadávaná v úhlových stupních od 0 do 360°.
-  **Break at End** – způsob zastavení motoru. Okamžité zastavení (**Break**) reprezentuje logická hodnota True a pozvolné dojetí (**Coast**) hodnota False.

## ON FOR ROTATIONS (OTOČENÍ O POČET OTÁČEK)



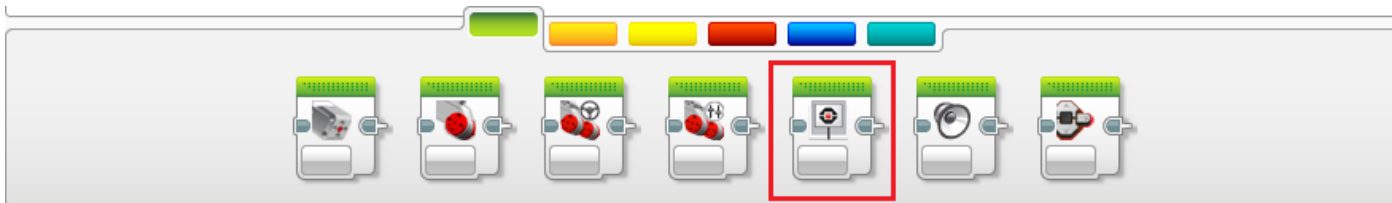
-  **Power** – rychlost otáčení motoru udávaná v procentech na stupnici od 100 (vpřed) do -100 (vzad).
-  **Rotations** – počet prováděných otáček motoru.
-  **Break at End** – způsob zastavení motoru. Okamžité zastavení (**Break**) reprezentuje logická hodnota True a pozvolné dojetí (**Coast**) hodnota False.

# Display (vykreslování a vypisování na displej)

## Kde blok v učebnici využíváme?

- Kapitola 4. zvuk a displej (práce s displejem)
- Kapitola 7. automatická závora (zobrazení hodnoty ultrazvukového senzoru na displej)
- Kapitola 8. adaptivní tempomat (výpis barvy zjištěné barevným senzorem na displej)
- Kapitola 10. parkovací asistent (výpis počtu stisků dotykového senzoru na displej)
- Kapitola 11 hra (výpis informací, generované hodnoty a počtu otáček na displej)

## Umístění programového bloku




## Režimy programového bloku

Programový blok slouží k ovládání displeje řídicí jednotky a umožňuje na něj vykreslovat body, křivky, jednoduché grafické soubory či vypisovat text. Blok může pracovat v následujících režimech:


### TEXT - PIXELS


#### (VYPSÁNÍ TEXTU NA POZICI KONKRÉTNÍHO PIXELU DISPLEJE)



 **Clear Screen** – mazání displeje. Pokud je logická hodnota nastavena na True (pravda), dojde ke smazání displeje. V případě, že je nastavena na hodnotu False (nepravda), ke smazání nedojde.


**x** a **y** - souřadnice displeje určující počáteční bod výpisu zvoleného textu. Souřadnice určují levý horní roh zobrazovaného textu. Hodnota **x** se pohybuje v rozmezí od -177 do 177 a hodnota **y** v rozmezí od -127 do 127.


 **Color** - volba způsobu výpisu na displej. Zvolíme-li možnost True (pravda), bude se text vypisovat bílou barvou na černé pozadí, pokud zvolíme False (nepravda), bude se vypisovat černou barvou na bílém pozadí.

 **Font** - volba fontu vypisovaného textu. Na výběr máme ze tří možností (Normal, Bold, Large).

### TEXT - GRID (VYPSÁNÍ TEXTU NA KONKRÉTNÍ POZICI V MŘÍŽCE)



 **Clear Screen** – mazání displeje. Pokud je logická hodnota nastavena na True (pravda), dojde ke smazání displeje. V případě, že je nastavena na hodnotu False (nepravda), ke smazání nedojde.

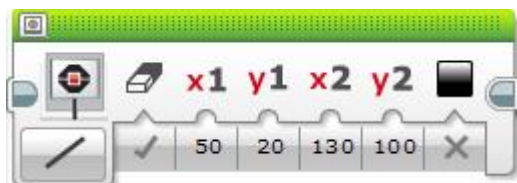
 **Column** – označuje počáteční sloupec pro výpis textu do mřížky. Sloupce jsou očíslovány 0–21 zleva doprava.

**Y** **Row** – označuje počáteční řádek pro výpis textu do mřížky. Řádky jsou číslovány 0–11 shora dolů.

**Color** – volba způsobu výpisu na displej. Zvolíme-li možnost True (pravda), bude se text vypisovat bílou barvou na černé pozadí, pokud zvolíme False (nepravda), bude se vypisovat černou barvou na bílém pozadí.

**Font** – volba fontu vypisovaného textu. Na výběr máme ze tří možností (Normal, Bold, Large).

### SHAPES - LINE (VYKRESLENÍ ČÁRY NA DISPLEJ)



**Clear Screen** – mazání displeje. Pokud je logická hodnota nastavena na True (pravda), dojde ke smazání displeje. V případě, že je nastavena na hodnotu False (nepravda), ke smazání nedojde.

**x1 a y1** – souřadnice počátečního bodu vykreslované čáry v rovinné kartézské soustavě souřadnic.

**x2 a y2** – souřadnice koncového bodu vykreslované čáry v rovinné kartézské soustavě souřadnic.

**Color** – volba způsobu výpisu na displej. Zvolíme-li možnost True (pravda), bude se text vypisovat bílou barvou na černé pozadí, pokud zvolíme False (nepravda), bude se vypisovat černou barvou na bílém pozadí.

### SHAPES - CIRCLE (VYKRESLENÍ KRUŽNICE NA DISPLEJ)



**Clear Screen** – mazání displeje. Pokud je logická hodnota nastavena na True (pravda), dojde ke smazání displeje. V případě, že je nastavena na hodnotu False (nepravda), ke smazání nedojde.

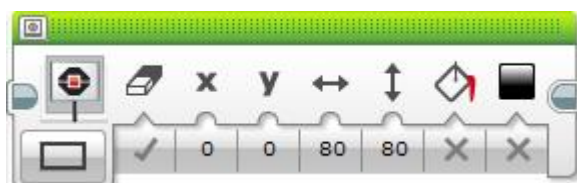
**x a y** – souřadnice středu kružnice v rovinné kartézské soustavě souřadnic zadávané v pixelech.

**Radius** – poloměr kružnice udávaný v pixelech.

**Fill** – výplň vykreslované kružnice. Pokud je hodnota nastavena na True (Pravda), vykresluje se kružnice s výplní, v opačném případě bez výplně.

**Color** – volba způsobu výpisu na displej. Zvolíme-li možnost True (pravda), bude se text vypisovat bílou barvou na černé pozadí, pokud zvolíme False (nepravda), bude se vypisovat černou barvou na bílém pozadí.

### SHAPES - RECTANGLE (VYKRESLENÍ OBDÉLNÍKU NEBO ČTVERCE NA DISPLEJ)



**Clear Screen** – mazání displeje. Pokud je logická hodnota nastavena na True (pravda), dojde ke smazání displeje. V případě, že je nastavena na hodnotu False (nepravda), ke smazání nedojde.



**X a Y** – souřadnice levého horního rohu vykreslovaného obdélníku nebo čtverce v pixelech.

**Width** – šířka obdélníku nebo čtverce v pixelech.

**Height** – výška obdélníku nebo čtverce v pixelech.

**Fill** – výplň vykreslované kružnice. Pokud je hodnota nastavena na True (Pravda), vykresluje se kružnice s výplní, v opačném případě bez výplně.

**Color** – volba způsobu výpisu na displej. Zvolíme-li možnost True (pravda), bude se text vypisovat bílou barvou na černé pozadí, pokud zvolíme False (nepravda), bude se vypisovat černou barvou na bílém pozadí.

## SHAPES - POINT (VYKRESLENÍ KONKRÉTNÍHO BODU (PIXELU) NA DISPLEJI)



**Clear Screen** – mazání displeje. Pokud je logická hodnota nastavena na True (pravda), dojde ke smazání displeje. V případě, že je nastavena na hodnotu False (nepravda), ke smazání nedojde.

**X a Y** – souřadnice konkrétního bodu (pixelu) na displeji.

**Color** – volba způsobu výpisu na displej. Zvolíme-li možnost True (pravda), bude se text vypisovat bílou barvou na černé pozadí, pokud zvolíme False (nepravda), bude se vypisovat černou barvou na bílém pozadí.

## IMAGE (VYKRESLENÍ OBRÁZKU NA DISPLEJI)



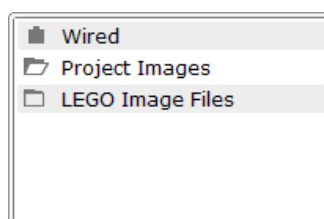
**Clear Screen** – mazání displeje. Pokud je logická hodnota nastavena na True (pravda), dojde ke smazání displeje. V případě, že je nastavena na hodnotu False (nepravda), ke smazání nedojde.

**X a Y** – souřadnice levého horního rohu vykreslovaného obrázku.

### Výběr obrázku z paměti řídicí jednotky



Výběr obrázku z galerie umístěné v paměti řídicí jednotky se provádí pomocí bílého pole v pravém horním rohu programovacího bloku (červeně zvýrazněno).



Po kliknutí na bílé pole se zobrazí nové okno, ve kterém buďto provedeme výběr obrázku z galerie umístěné v paměti řídicí jednotky (složka **LEGO Image Files**) nebo zvolíme vlastní připravené obrázek ze složky **Project Image**.

## RESET SCREEN (SMAZÁNÍ PLOCHY DISPLEJE)



Režim umožňující kompletní smazání plochy displeje.



# Sound (zvukový výstup)

## Kde blok v učebnici využíváme?

- Kapitola 4. zvuk a displej (práce se zvukem)
- Kapitola 6. závora (zvuková signalizace)
- Kapitola 7. automatická závora (zvuková signalizace)
- Kapitola 8. adaptivní tempomat (zvuková signalizace)

## Umístění programového bloku



## Režimy programového bloku

Programový blok slouží k ovládání zvukového výstupu řídicí jednotky. Umožňuje přehrávat tóny, noty nebo zvukové soubory. Blok může pracovat v následujících režimech:


### STOP (ZASTAVENÍ)



Režim programového bloku sloužící k zastavení aktuálně přehrávaného zvuku.




### PLAY FILE (PŘEHRÁNÍ SOUBORU)



 **Volume Control** – hlasitost přehrávaného zvuku udávaná na stupnici od 0 do 100.

 **Play Type** – způsob přehrání zvoleného zvukového souboru.

Přehrání je možné realizovat třemi způsoby:

-  **Wait for Completion** – zvuk je jedenkrát přehrán a program pokračuje až po dokončení přehrávání.
-  **Play Once** – zvuk je jedenkrát přehrán a program okamžitě pokračuje dále.
-  **Repeat** – zvuk je přehráván stále dokola, dokud není přerušen jiným blokem Sound a program okamžitě pokračuje dále.

### Výběr zvukového souboru z paměti řídicí jednotky

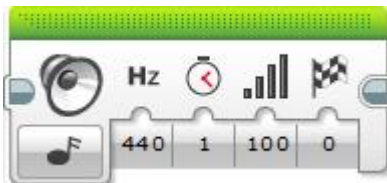


Výběr zvukového souboru z paměti v paměti řídicí jednotky se provádí pomocí bílého pole v pravém horním rohu programovacího bloku (červeně zvýrazněno).



Na výběr máme předpřipravené zvuky **LEGO Sound Files** nebo vlastní zvuky importované do projektu a umístěné ve složce **Project Sounds**.

## PLAY TONE (PŘEHRÁNÍ TÓNU)



**Hz Frequency** – frekvence zvoleného zvuku v Herzích (možné rozpětí hodnot je od 300 do 10 000).

**Duration** – délka přehrávaného zvuku ve vteřinách.

**Volume Control** – hlasitost přehrávaného zvuku udávaná na stupnici od 0 do 100.

**Play Type** – způsob přehrání zvoleného zvuku. Přehrání je možné realizovat třemi způsoby:

- **Wait for Completion** – zvuk je jedenkrát přehrán a program pokračuje až po dokončení přehrávání.
- **Play Once** – zvuk je jedenkrát přehrán a program okamžitě pokračuje dále.
- **Repeat** – zvuk je přehráván stále dokola, dokud není přerušen jiným blokem Sound a program okamžitě pokračuje dále.

## PLAY NOTE (PŘEHRÁNÍ TÓNU)



**Note** – výběr zvukového tónu.

**Duration** – délka přehrávaného tónu ve vteřinách.

**Volume Control** – hlasitost přehrávaného tónu udávaná na stupnici od 0 do 100.

**Play Type** – způsob přehrání zvoleného tónu. Přehrání je možné realizovat třemi způsoby:

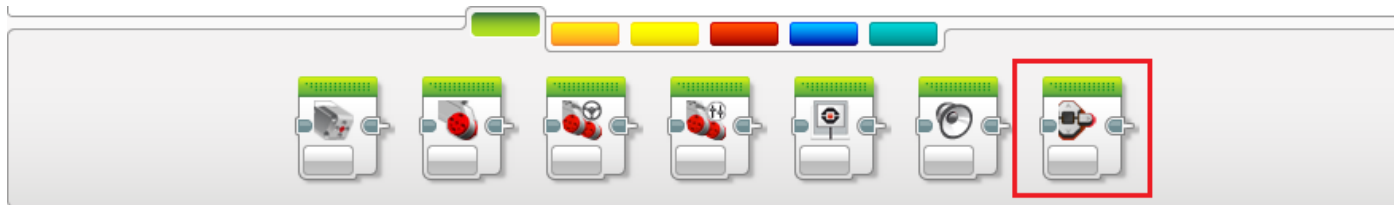
- **Wait for Completion** – tón je jedenkrát přehrán a program pokračuje až po dokončení přehrávání.
- **Play Once** – tón je jedenkrát přehrán a program okamžitě pokračuje dále.
- **Repeat** – tón je přehráván stále dokola, dokud není přerušen jiným blokem Sound a program okamžitě pokračuje dále.

# Brick Status Light (světelná signalizace řídicí jednotky)

Kde blok v učebnici využíváme?

- Kapitola 6. závora (světelná signalizace)

Umístění programového bloku



Režimy programového bloku

Programový blok slouží k ovládnání světelné signalizace řídicí jednotky. Umožňuje rozsvítit světelnou signalizaci (podsvícení tlačítek řídicí jednotky) ve spektru tří barev (zelená, žlutá, červená). Blok může pracovat v následujících režimech:


## OFF (VYPNUTO)

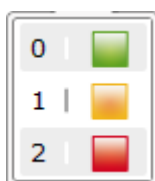



Režim, pomocí kterého je možné vypnout aktivní světelnou signalizaci řídicí jednotky.

## ON (ZAPNUTO)



 **Colour** – výběr barvy světelné signalizace. Na výběr máme ze tří barev (viz obrázek níže).



 **Pulse** – pokud je hodnota nastavena na True (pravda) světlo pulzuje, v opačném případě nikoliv.

## RESET (RESETOVÁNÍ)



Režim, který znovu aktivuje výchozí světelnou signalizaci chodu programu (zelené blikající světlo).

# Wait (čekání/oddálení vykonávání dalšího příkazu)

## Kde blok v učebnici využíváme?

- Kapitola 2 oživení pojízdného robota (čekání určitý počet vteřin)
- Kapitola 4. zvuk a displej (čekání určitý počet vteřin)
- Kapitola 5. mixér (čekání určitý počet vteřin, čekání na stisk tlačítka dotykového senzoru)
- Kapitola 7. automatická závora (čekání určitý počet vteřin)
- Kapitola 9. inteligentní robot (čekání na stisk tlačítka řídicí jednotky)
- Kapitola 10. parkovací asistent (čekání určitý počet vteřin, čekání na naměření určité vzdálenosti ultrazvukovým senzorem)
- Kapitola 11. hra (čekání určitý počet vteřin)

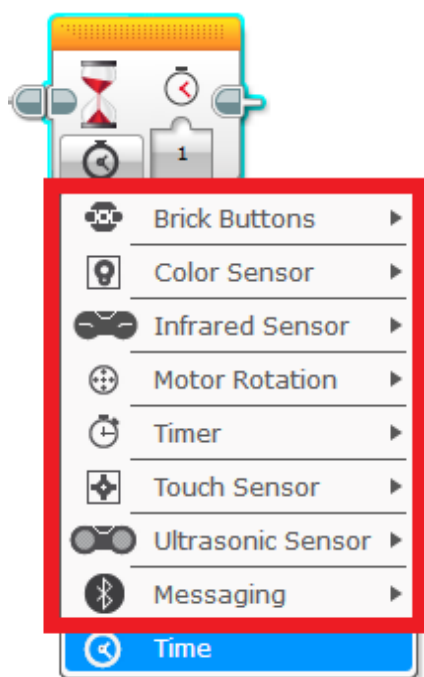
## Umístění programového bloku




## Režimy programového bloku


Programový blok slouží k oddálení vykonávání dalšího příkazu programu. Čekání může být prováděno po zadanou dobu ve vteřinách. Dalšími nejčastějšími variantami jsou čekání na dosažení požadované hodnoty naměřené senzorem nebo čekání na uživatelský vstup (stisk tlačítka řídicí jednotky, stisk tlačítka dotykového senzoru). Blok může pracovat v následujících režimech:


### REAKCE NA UDÁLOST




 **Brick Buttons** – čekání na stisk tlačítka řídicí jednotky.


 **Color Sensor** – čekání na detekci konkrétní barvy, úroveň intenzity okolního či odraženého světla.


 **Infrared Sensor** – čekání na úroveň nebo například směr infračerveného signálu.

 **Motor Rotation** – čekání na konkrétní počet vykonaných otáček motoru.

 **Timer** – čekání na čas naměřený časovačem.


 **Touch Sensor** – čekání na stisk nebo uvolnění tlačítka dotykového senzoru.

 **Ultrasonic Sensor** – čekání na vzdálenost naměřenou ultrazvukovým senzorem.

 **Messaging** – čekání (reakce) na zprávu zaslou přes Bluetooth.

## TIME (ČEKÁNÍ PO PEVNĚ STANOVENOU DOBU)



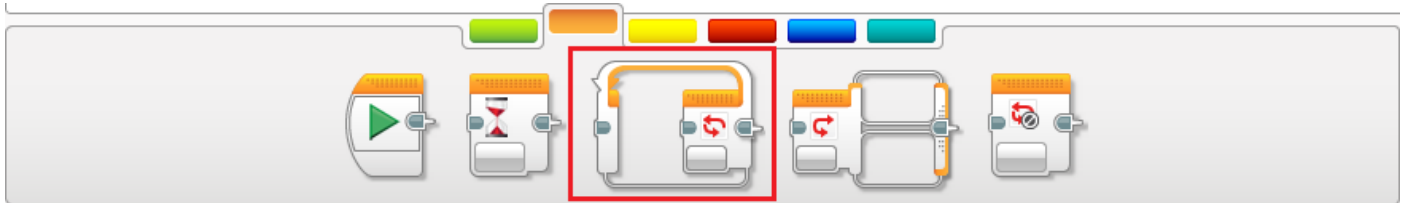
 **Seconds** – oddálení vykonávání dalšího příkazu je v tomto režimu prováděno o zadaný počet vteřin.

# Loop (cyklus)

## Kde blok v učebnici využíváme?

- Kapitola 3. robot ve městě (opakování při objíždění nemocnice – úkol C Hledání parkoviště)
- Kapitola 5. mixér (nekonečné opakování chodu mixéru)
- Kapitola 8. adaptivní tempomat (neustálé vyhodnocování naměřené vzdálenosti)
- Kapitola 9. inteligentní robot (neustálé opakování jízdy po čáře)
- Kapitola 10. parkovací asistent (neustálé opakování detekce volného místa)
- Kapitola 11. hra (zajištění opakovaného běhu hry)

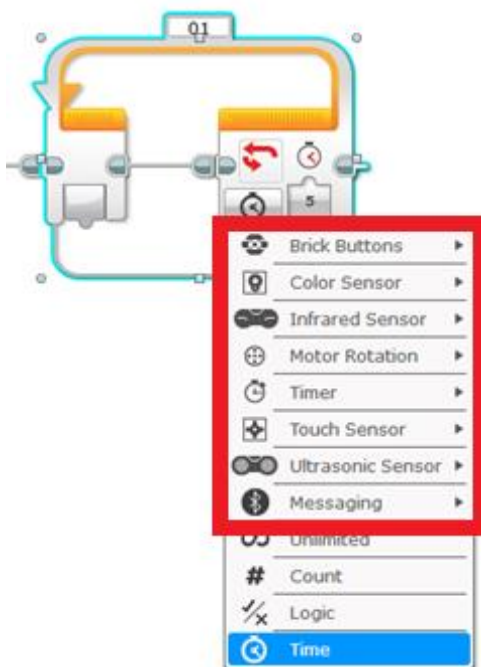
## Umístění programového bloku





## Režimy programového bloku


Programový blok **Loop** (cyklus) slouží k opakovanému vykonávání programových bloků umístěných do těla cyklu. Cyklus může být vykonáván do té doby, než uživatel provede určitý vstup do programu nebo bude detekována nějaká veličina. Je možné použít také nekonečný cyklus či cyklus s pevným počtem průchodů. Programový blok **Loop** může pracovat v následujících režimech:


### CYKLUS S PODMÍNKOU NA KONCI ŘÍZENÝ URČITOU UDÁLOSTÍ





 **Brick Buttons** – cyklus vykonávaný do té doby, dokud není stisknuto některé tlačítko řídicí jednotky.


 **Color Sensor** – cyklus vykonávaný do té doby, dokud není detekována konkrétní barva, úroveň intenzity okolního či odraženého světla.


 **Infrared Sensor** – cyklus vykonávaný do té doby, dokud infračervený signál přichází z určitého směru nebo určité vzdálenosti.

 **Motor Rotation** – cyklus vykonávaný do té doby, dokud není proveden určitý počet otáček motoru.

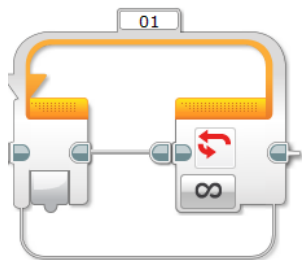
 **Timer** – cyklus vykonávaný do doby, než časovač dosáhne určité naměřené hodnoty.

 **Touch Sensor** – cyklus vykonávaný do té doby, než je stisknuto, uvolněno či stisknuto a uvolněno tlačítko dotykového senzoru.

 **Ultrasonic Sensor** – cyklus vykonávaný do té doby, než je dosažena (naměřena) požadovaná hodnota pomocí ultrazvukového senzoru.

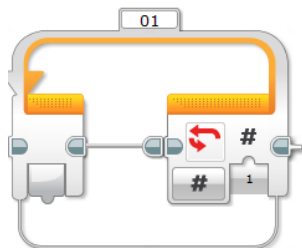
 **Messaging** – cyklus vykonávaný až do přijetí zprávy zasláné přes Bluetooth.

## UNLIMITED (NEKONEČNÝ CYKLUS)



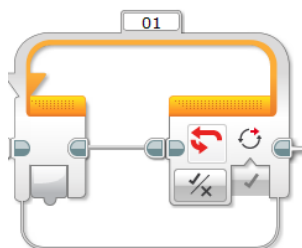
∞ **Unlimited** – nekonečné provádění cyklu.

## COUNT (CYKLUS S PEVNÝM POČTEM PRŮCHODŮ)



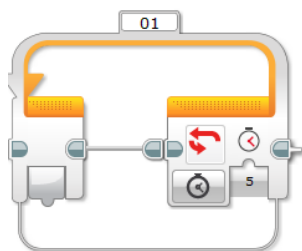
# **Count** – cyklus s pevným počtem průchodů. Číslo udává počet opakování cyklu.

## LOGIC (CYKLUS ŘÍZENÝ LOGICKOU HODNOTOU)



↻ **Until True** – cyklus je vykonáván do té doby, než je dosažena hodnota *True*.

## TIME (CYKLUS ŘÍZENÝ ČASEM)



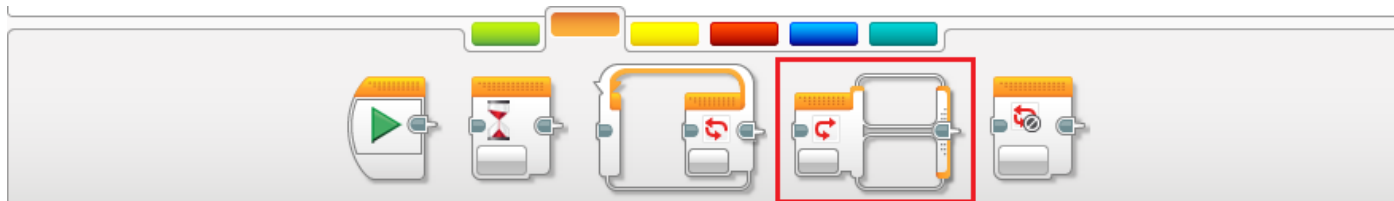
🕒 **Seconds** – cyklus je vykonáván po určitou dobu. Číslo udává dobu vykonávání ve vteřinách.

# Switch (podmínka)

## Kde blok v učebnici využíváme?

- Kapitola 5. mixér (podmínka řízená dotykovým senzorem)
- Kapitola 8. adaptivní tempomat (podmínka řízená ultrazvukovým senzorem)
- Kapitola 9. inteligentní robot (podmínka řízená světelným senzorem)
- Kapitola 10. parkovací asistent (podmínka řízená stiskem dotykového senzoru, podmínka řízená logickou hodnotou)
- Kapitola 11. hra (podmínka řízená hodnotou naměřenou časovačem)

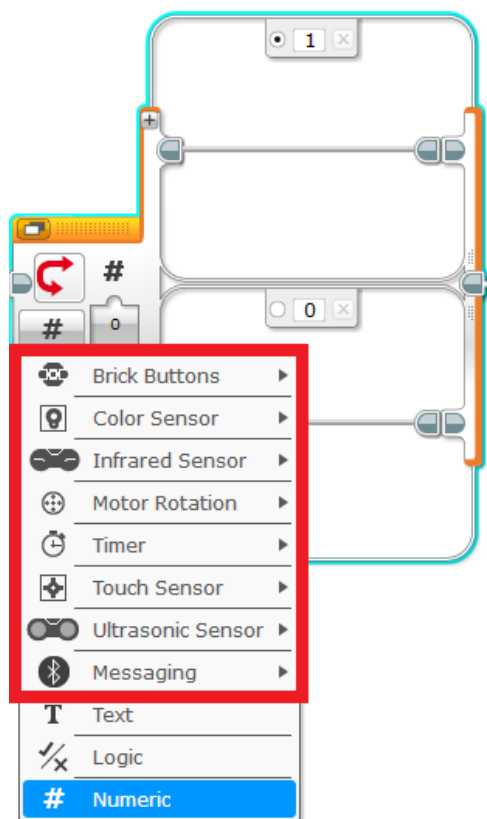
## Umístění programového bloku





## Režimy programového bloku


Programový blok **Switch** slouží k vykonávání podmíněných příkazů. Podmínka může být řízena několika možnostmi (reakce na událost, reakce na hodnotu zjištěnou senzorem, číslo nebo čas). Programový blok může pracovat v následujících režimech:


### REAKCE NA UDÁLOST





 **Brick Buttons** – podmínka řízená stiskem některého z tlačítek řídicí jednotky.


 **Color Sensor** – podmínka reagující na detekci konkrétní barvy, úroveň intenzity okolního či odraženého světla.


 **Infrared Sensor** – podmínka řízená úrovní nebo například směrem infračerveného signálu.

 **Motor Rotation** – podmínka řízená počtem vykonaných otáček motoru.

 **Timer** – podmínka řízená časem naměřeným časovačem.

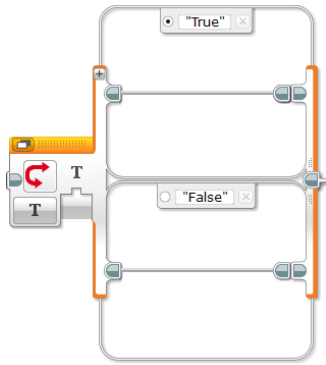
 **Touch Sensor** – podmínka reagující na stisk nebo uvolnění tlačítka dotykového senzoru.

 **Ultrasonic Sensor** – podmínka řízená vzdáleností naměřenou ultrazvukovým senzorem.

 **Messaging** – podmínka reagující na zprávu zaslou přes Bluetooth.

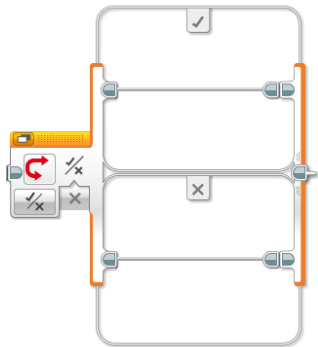


## TEXT (PODMÍNKA ŘÍZENÁ ŘETĚZCEM TEXTU)



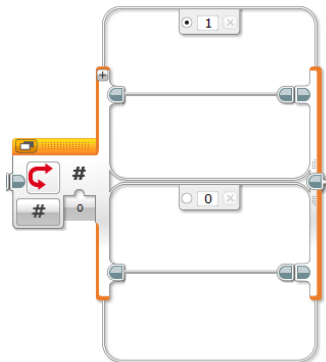
**T String** – podmínka je řízena řetězcem textu přivedeným na vstupní port.

## LOGIC (PODMÍNKA ŘÍZENÁ LOGICKOU HODNOTOU)



**✓/✗ Logic** – podmínka je řízena logickou hodnotou (true nebo false) přivedenou na vstupní port.

## NUMERIC (PODMÍNKA ŘÍZENÁ ČÍSELNOU HODNOTOU)



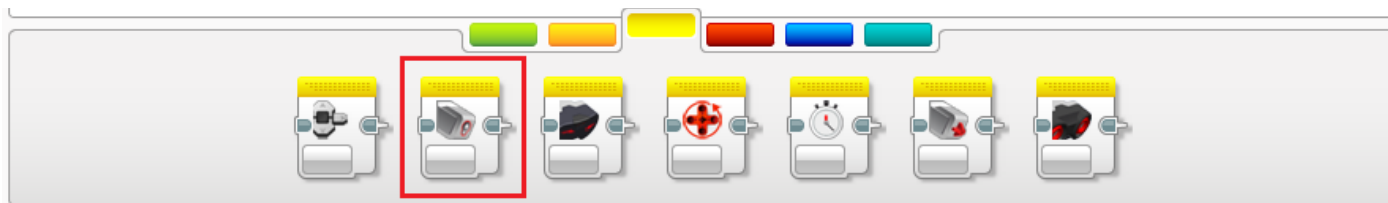
**# Number** – podmínka je řízena číselnou hodnotou přivedenou na vstupní port.

# Color Sensor (barevný senzor)

Kde blok v učebnici využíváme?

- Kapitola 7. automatická závora (mýtná brána)
- Kapitola 9. inteligentní robot (pohyb po čáře)

Umístění programového bloku




Režimy programového bloku

Programový blok slouží k ovládání barevného senzoru, který umožňuje měřit a porovnávat základní detekované barvy, intenzitu odraženého světla a intenzitu okolního světla. Blok může pracovat v následujících režimech:

## MEASURE - COLOUR (MĚŘENÍ BARVY)




 **Colour** – výstupní port vracejíčí číselnou návratovou hodnotu odpovídající zjištěné barvě. Návratové hodnoty mohou být následující:

- 0 = žádná barva
- 1 = černá
- 2 = modrá
- 3 = zelená
- 4 = žlutá
- 5 = červená
- 6 = bílá
- 7 = hnědá


## MEASURE - REFLECTED LIGHT INTENSITY (MĚŘENÍ INTENZITY ODRAŽENÉHO SVĚTLA)



 **Light** – návratovou hodnotou výstupního portu je číselná hodnota odpovídající intenzitě odráženého světla snímaného povrchu.

## MEASURE - AMBIENT LIGHT INTENSITY (MĚŘENÍ INTENZITY OKOLNÍHO SVĚTLA)



 **Light** – návratovou hodnotou výstupního portu je číselná hodnota odpovídající intenzitě světla snímaného z okolního prostředí.

## COMPARE – COLOUR (POROVNÁNÍ ZJIŠTĚNÉ BARVY SE ZADANOU)

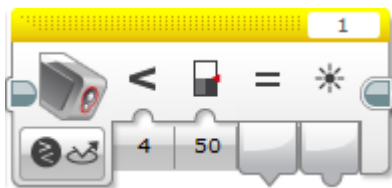


**Set of Colours** – vstupní port sloužící k zadání jedné či více barev. Při snímání se bude zjišťovat, zda snímaný povrch odpovídá zadané barvě/ám.

= **Compare Result** – výstupní port vracející logickou návratovou hodnotu podle toho, zda byla zadaná barva/y detekována/y (True) nebo nikoliv (False).

● **Colour** – výstupní port vracející číselnou návratovou hodnotu odpovídající zjištěné barvě.

## COMPARE - REFLECTED LIGHT INTENSITY (POROVNÁNÍ INTENZITY ODRAŽENÉHO SVĚTLA SE ZADANOU HODNOTOU)



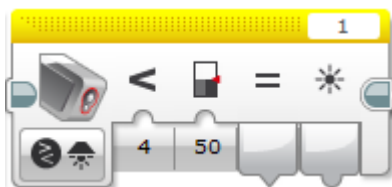
< **Compare Type** – výběr způsobu porovnání zjištěné hodnoty s prahovou. Vybíráme, zda chceme zjistit, jestli je snímaná hodnota menší, větší nebo rovna prahové.

■ **Threshold Value** – prahová hodnota, s kterou se porovnává intenzita odraženého světla zjištěná senzorem.

= **Compare Result** – výstupní port vracející logickou návratovou hodnotu v závislosti na tom, zda zjištěná hodnota byla v námi požadované intenzitě vůči prahové (True) či nikoliv (False).

\* **Light** – návratovou hodnotou výstupního portu je číselná hodnota odpovídající intenzitě světla odraženého od snímaného povrchu.

## COMPARE - AMBIENT LIGHT INTENSITY (POROVNÁNÍ INTENZITY OKOLNÍHO SVĚTLA SE ZADANOU HODNOTOU)



< **Compare Type** – výběr způsobu porovnání zjištěné hodnoty s prahovou. Vybíráme, zda chceme zjistit, jestli je snímaná hodnota menší, větší nebo rovna prahové.

■ **Threshold Value** – prahová hodnota, s kterou se porovnává intenzita snímaného světla z okolí zjištěná senzorem.

= **Compare Result** – výstupní port vracející logickou návratovou hodnotu v závislosti na tom, zda zjištěná hodnota byla v námi požadované intenzitě vůči prahové (True) či nikoliv (False).

\* **Light** – návratovou hodnotou výstupního portu je číselná hodnota odpovídající intenzitě snímaného světla v okolí senzoru.

## CALIBRATE REFLECTED LIGHT INTENSITY – MINIMUM (ZADÁNÍ MINIMA PŘI KALIBRACI SENZORU)



\* **Value** – hodnota zadávaná při kalibraci senzoru. Dále bude při snímání chápána jako minimum.

## CALIBRATE REFLECTED LIGHT INTENSITY – MAXIMUM (ZADÁNÍ MAXIMA PŘI KALIBRACI SENZORU)



\* **Value** – hodnota zadávaná při kalibraci senzoru. Dále bude při snímání chápána jako maximum.

## CALIBRATE REFLECTED LIGHT INTENSITY – RESET (UVEDENÍ SENZORU DO VÝCHOZÍHO NASTAVENÍ – ZRUŠENÍ NASTAVENÍ PROVEDENÉHO KALIBRACÍ)



Režim sloužící ke zrušení všech změn provedených při kalibraci, a tudíž navrácení senzoru opět do výchozího nastavení.

# Motor Rotation (měřič otáček motoru)

Kde blok v učebnici využíváme?

- Kapitola 10. parkovací asistent (měření počtu otáček)

Umístění programového bloku




Režimy programového bloku

Programový blok **Motor Rotation** slouží k ovládání čítače otáček motoru (ve stupních nebo v otáčkách) nebo ke zjištění aktuální rychlosti otáčení motoru. Blok může pracovat v následujících režimech:


## MEASURE - DEGREES (MĚŘENÍ OTÁČEK MOTORU VE STUPNÍCH)



-  **Degrees** – port vrací počet stupňů, o které se motor otočil.

## MEASURE – ROTATIONS (MĚŘENÍ OTÁČEK MOTORU V CELÝCH OTÁČKÁCH)



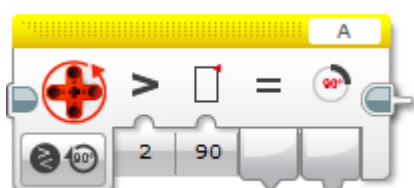
-  **Rotations** – port vrací počet otáček, o které se motor otočil.



## MEASURE – CURRENT POWER (MĚŘENÍ AKTUÁLNÍ RYCHLOSTI OTÁČENÍ MOTORU)



-  **Current Power** – port vrací aktuální rychlost otáčení motoru.



## COMPARE - DEGREES (POROVNÁNÍ HODNOT V ÚHLOVÝCH STUPNÍCH)



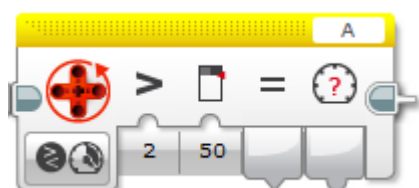
- > **Compare Type** – volba znaménka pro porovnání dvou hodnot.
-  **Threshold Value** – fixně zadaná hodnota porovnávaná s hodnotou zjištěnou čítačem otáček motoru.
- = **Compare Result** – výsledek porovnání dvou hodnot (true nebo false).
-  **Degrees** – port vrací počet stupňů, o které se motor otočil.



## COMPARE - ROTATIONS (POROVNÁNÍ ZJIŠTĚNÉHO POČTU OTÁČEK SE ZADANOU HODNOTOU)



- > **Compare Type** – volba znaménka pro porovnání dvou hodnot.
-  **Threshold Value** – fixně zadaná hodnota porovnávaná s hodnotou zjištěnou čítačem otáček motoru.
- = **Compare Result** – výsledek porovnání dvou hodnot (true nebo false).
-  **Rotations** – port vrací počet otáček, o které se motor otočil.

## COMPARE – CURRENT POWER (POROVNÁNÍ ZJIŠTĚNÉ RYCHLOSTI MOTORU SE ZADANOU HODNOTOU)



- > **Compare Type** – volba znaménka pro porovnání dvou hodnot.
-  **Threshold Value** – fixně zadaná hodnota porovnávaná s hodnotou zjištěnou čítačem otáček motoru.
- = **Compare Result** – výsledek porovnání dvou hodnot (true nebo false).
-  **Current Power** – port vrací aktuální rychlost otáčení motoru.

## RESET (NULOVÁNÍ ČÍTAČE OTÁČEK MOTORU)



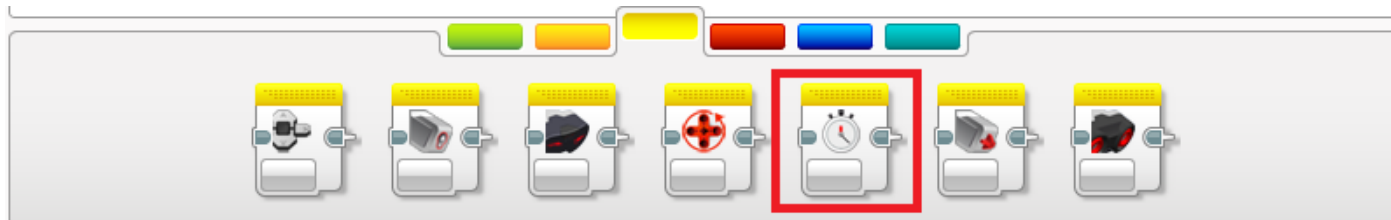
Režim sloužící k vynulování čítače otáček motoru.

# Timer (časovač)

Kde blok v učebnici využíváme?

- Kapitola 11. hra (měření času stisku tlačítka dotykového senzoru)

Umístění programového bloku





Režimy programového bloku

Programový blok **Timer** slouží k měření času ve vteřinách. Blok může pracovat v následujících režimech:




## MEASURE - TIME (MĚŘENÍ ČASU VE VTEŘINÁCH)



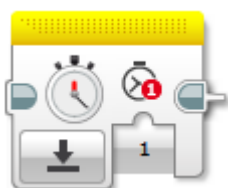
-  **Timer ID** – v programovacím prostředí můžeme vytvořit až 8 časovačů. Parametr označuje, který z nich aktuálně ovládáme.
-  **Elapsed Time** – výstupní parametr udávající naměřený čas.


## COMPARE – TIME (POROVNÁNÍ HODNOTY ČASOVAČ S FIXNĚ ZADANOU HODNOTOU)



-  **Timer ID** – v programovacím prostředí můžeme vytvořit až 8 časovačů. Parametr označuje, který z nich aktuálně ovládáme.
- > Compare Type** - volba znaménka pro porovnání dvou hodnot.
-  **Threshold Value** - fixně zadaná hodnota porovnávaná s hodnotou zjištěnou čítačem otáček motoru.
- = Compare Result** – výsledek porovnání dvou hodnot (true nebo false).
-  **Elapsed Time** – výstupní parametr udávající naměřený čas.

## RESET (NULOVÁNÍ ČÍTAČE)



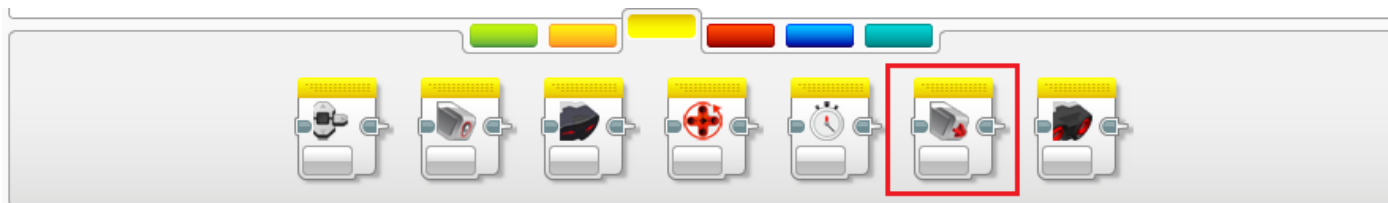
-  **Timer ID** – číselné označení čítače, která bude vynulována.

# Touch Sensor (dotykový sensor)

Kde blok v učebnici využíváme?

- Kapitola 5. mixér (ovládání mixéru s tlačítkem)
- Kapitola 6. závora (ovládání závory tlačítkem)
- Kapitola 11. hra (tlačítko ovladače pro hráče)

Umístění programového bloku



Režimy programového bloku

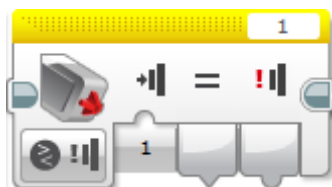
Programový blok slouží k ovládání dotykového senzoru, který umožňuje detekovat stisk tlačítka ve třech stavech (stisknuto, uvolněno, stisknuto a uvolněno). Blok může pracovat v následujících režimech:

## MEASURE STATE (ZJIŠTĚNÍ STAVU TLAČÍTKA)



**!|| State** – výstupní hodnotou tohoto portu je True v případě, že je tlačítko stisknuto, v opačném případě je návratová hodnota rovna False.

## COMPARE STATE (TESTOVÁNÍ STAVU SENZORU) – ZJIŠTĚNÍ PROVEDENÉHO ÚKONU



**+|| State** – vstupní port sloužící je zjištění, zda bylo tlačítko dotykového senzoru uvolněno (návratová hodnota 0), stisknuto (návratová hodnota 1) nebo stisknuto a uvolněno (návratová hodnota 2).

**= Compare Result** – výstupní port vracejí logickou návratovou hodnotu podle toho, zda byl testovaný stav detekován (True) nebo nikoliv (False).

**!|| Measured Value** – výstupní port vracejí návratovou hodnotu v závislosti na tom, jakého stavu senzoru bylo dosaženo (0 uvolněno, 1 stisknuto, 2 stisknuto a uvolněno).

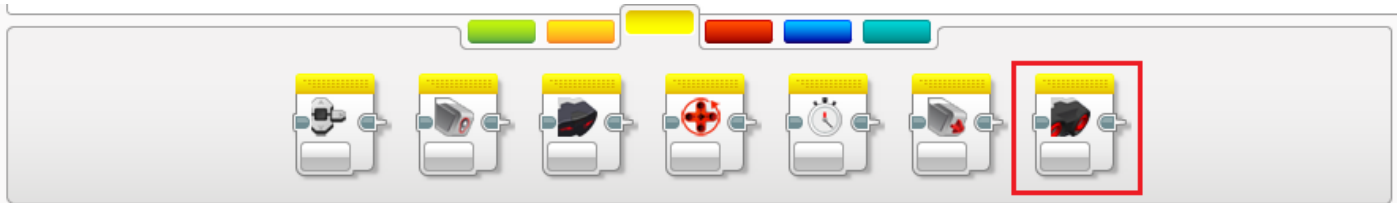


# Ultrasonic Sensor (ultrazvukový sensor)

Kde blok v učebnici využíváme?

- Kapitola 7. automatická závora (detekce vozidla před závorou)
- Kapitola 8. adaptivní tempomat (detekce objektu před vozidlem)
- Kapitola 11. hra (tlačítko ovladače pro hráče)

Umístění programového bloku



Režimy programového bloku

Programový blok slouží k ovládání ultrazvukového senzoru, který umožňuje detekovat předměty umístěné před senzorem a měřit vzdálenost mezi senzorem a překážkou. Blok může pracovat v následujících režimech:

## MEASURE - ADVANCED – CENTIMETRES (MĚŘENÍ VZDÁLENOSTI V CENTIMETRECH)



**Measuring Mode** (režim měření vzdálenosti):

- **Ping** – režim, ve kterém jsou zasílány jednotlivé ultrazvukové signály.
- **Continuous** – režim, ve kterém je zasílán konstantní, nepřerušovaný ultrazvukový signál.

**Distance in Centimeters** – výstupní port jehož návratovou hodnotou je naměřená vzdálenost v centimetrech.

## MEASURE - ADVANCED – INCHES (MĚŘENÍ VZDÁLENOSTI V PALCÍCH)



**Measuring Mode** (režim měření vzdálenosti):


- **Ping** - režim, ve kterém jsou zasílány jednotlivé ultrazvukové signály.
- **Continuous** - režim, ve kterém je zasílán konstantní, nepřerušovaný ultrazvukový signál.

**Distance in Centimetres** - výstupní port jehož návratovou hodnotou je naměřená vzdálenost v centimetrech.

**Poznámka:** Režimy **Measure – Distance – Centimetres** (měření vzdálenosti v centimetrech) a **Measure – Distance – Inches** (měření vzdálenosti v palcích) fungují naprosto stejným způsobem, jen neobsahují možnost změny režimu vysílání ultrazvukového signálu.


## MEASURE – PRESENCE (DETEKCE ULTRAZVUKOVÉHO SIGNÁLU)




 **Ultrasound Detected** – výstupní port, jehož návratová hodnota je logického datového typu. Pokud je signál detekován, je výsledkem hodnota True (pravda) v opačném případě False (nepravda).


## COMPARE - DISTANCE CENTIMETRES (POROVNÁNÍ NAMĚŘENÉ HODNOTY S FIXNĚ ZADANOU HODNOTOU V CENTIMETRECH)



 **Compare Type** – výběr způsobu porovnání zjištěné hodnoty s prahovou. Vybíráme, zda chceme zjistit, jestli je naměřená vzdálenost menší, větší nebo rovna prahové.


 **Threshold Value** – prahová hodnota, s kterou porovnááme naměřenou vzdálenost.


**= Compare Result** – výstupní port vracející logickou návratovou hodnotu v závislosti na tom, zda zjištěná hodnota byla v námi ověřované vzdálenosti (True) či nikoliv (False).

 **Distance in Centimetres** – návratovou hodnotou výstupního portu je naměřená vzdálenost v centimetrech.


## COMPARE - DISTANCE INCHES (POROVNÁNÍ NAMĚŘENÉ HODNOTY S FIXNĚ ZADANOU HODNOTOU V PALCÍCH)



 **Compare Type** – výběr způsobu porovnání zjištěné hodnoty s prahovou. Vybíráme, zda chceme zjistit, jestli je naměřená vzdálenost menší, větší nebo rovna prahové.

 **Threshold Value** – prahová hodnota, s kterou porovnááme naměřenou vzdálenost.

**= Compare Result** – výstupní port vracející logickou návratovou hodnotu v závislosti na tom, zda zjištěná hodnota byla v námi ověřované vzdálenosti (True) či nikoliv (False).

 **Distance in Inches** – návratovou hodnotou výstupního portu je naměřená vzdálenost v palcích.

## COMPARE - PRESENCE/LISTEN



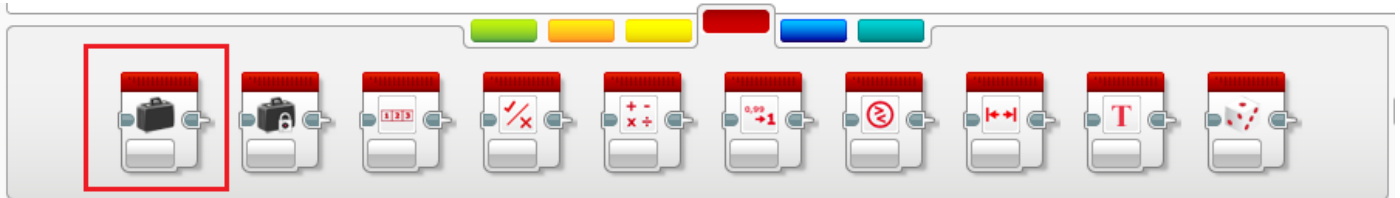
Funkce režimu je totožná s režimem **Measure – Presence**.

# Variable (proměnná)

Kde blok v učebnici využíváme?

- Kapitola 10. parkovací asistent (počítadlo, nejmenší a největší vzdálenost, detekce volného místa)
- Kapitola 11. hra (evidence nejlepšího času)

Umístění programového bloku

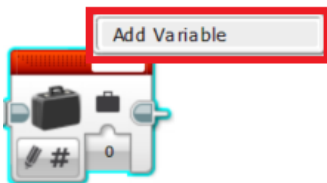


Režimy programového bloku

Programový blok **Variable** slouží k vytvoření proměnné, která může sloužit buďto ke čtení nebo pro zápis.

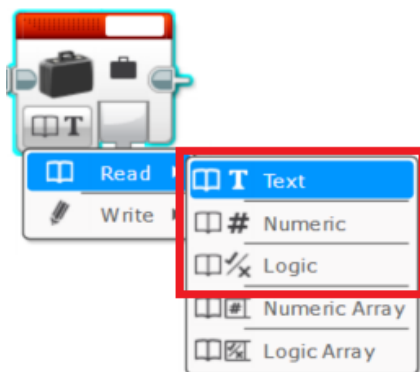
Vytvoření proměnné

Vytvoření proměnné (zadání jejího názvu) se provádí pomocí tlačítka v pravém horním rohu volbou „Add Variable“.



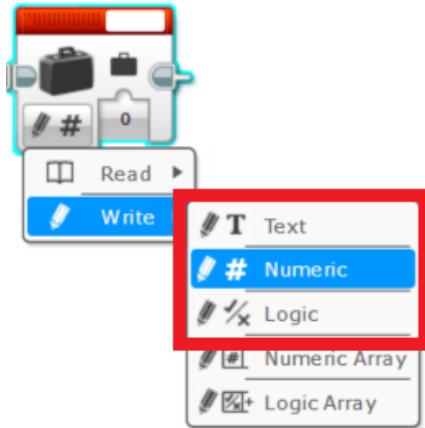
Blok **Variable** může pracovat v následujících režimech (vybíráme pouze základní):

**READ (ČTENÍ HODNOTY Z PROMĚNNÉ)**



- T Text** – čtení textového řetězce uloženého v proměnné.
- # Numeric** – čtení číselné hodnoty uložené v proměnné.
- Logic** – čtení logické hodnoty uložené v proměnné.

## WRITE (ZÁPIS HODNOTY DO PROMĚNNÉ)



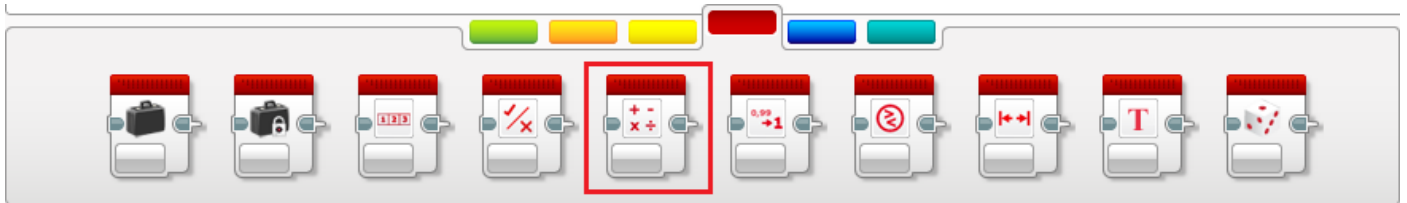
- T** **Text** – uložení textového řetězce do proměnné.
- #** **Numeric** – uložení číselné hodnoty do proměnné.
- ✓/x** **Logic** – uložení logické hodnoty do proměnné.

# Math (matematické operace)

Kde blok v učebnici využíváme?

- Kapitola 8. adaptivní tempomat (pomocný výpočet při stabilizaci pohybu)
- Kapitola 9. inteligentní robot (výpočet středové hodnoty pro pohyb po čáře)
- Kapitola 10. parkovací asistent (analýza programu s proměnnou)

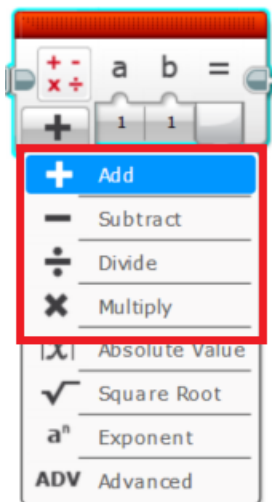
Umístění programového bloku



Režimy programového bloku

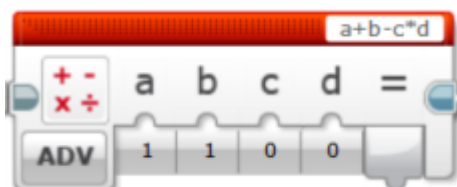
Programový blok **Math** slouží k provádění matematických operací. Na výběr jsou jak základní (sčítání, odčítání, násobení a dělení), tak i absolutní hodnota, odmocnina exponent nebo zápis matematických výrazů. Věnovat se budeme pouze některým, využitelným při řešení úloh obsažených v učebnici. Blok **Math** může pracovat v následujících režimech:

**ADD (SČÍTÁNÍ), SUBSTRACT (ODČÍTÁNÍ), DIVIDE (DĚLENÍ), MULTIPLY (NÁSOBENÍ)**



Provádění základních matematických operací (sčítání, odčítání, násobení a dělení) se zadanými hodnotami **a** a **b**.

**ADVANCED (ZÁPIS MATEMATICKÉHO VÝRAZU)**



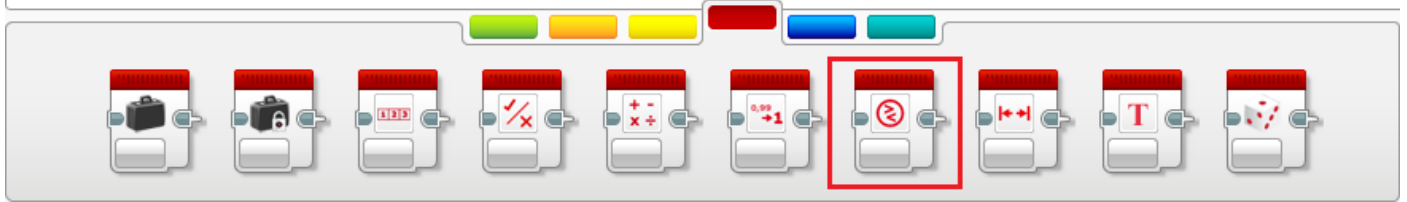
Režim umožňuje zápis matematického výrazu v okně vpravo nahoře s využitím až čtyř zadaných hodnot **a**, **b**, **c** a **d**.

# Compare (porovnání hodnot)

Kde blok v učebnici využíváme?

- Kapitola 10. parkovací asistent (analýza programu s proměnnou)

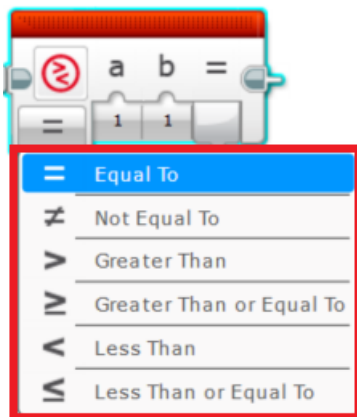
Umístění programového bloku



Režimy programového bloku

Programový blok **Compare** slouží k porovnání dvou zadaných hodnot. Blok může pracovat v následujících režimech:

## COMPARE (POROVNÁNÍ)



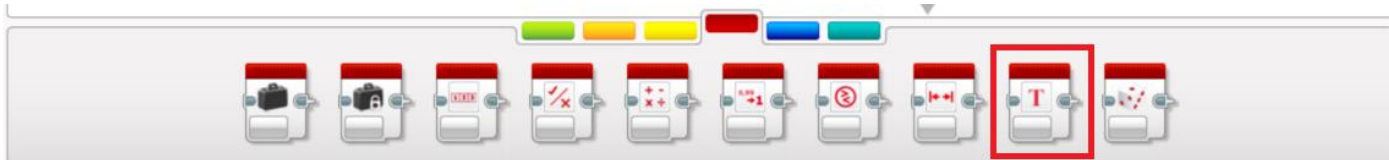
Programový blok obsahuje šest základních možností pro porovnání dvou zadaných hodnot **a** a **b**.

# Text (sloučení textových řetězců)

Kde blok v učebnici využíváme?

- Kapitola 8. adaptivní tempomat (výpis složitějšího řetězce textu na displej)

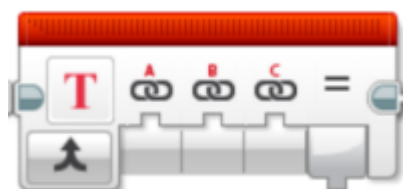
Umístění programového bloku



Režimy programového bloku

Programový blok **Text** slouží ke sloučení tří textových řetězců v jeden. Blok obsahuje pouze jediný režim:

**TEXT (SLOUČENÍ TEXTOVÝCH ŘETĚZCŮ)**



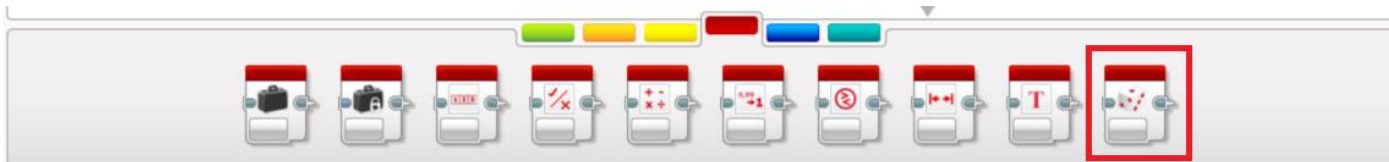
- A** – první řetězec textu.
- B** – druhý řetězec textu.
- C** – třetí řetězec textu.
- =** **Result** – výsledný řetězec textu po sloučení.

# Random (generování náhodného čísla)

Kde blok v učebnici využíváme?

- Kapitola 11. hra (generování náhodných čísel)

Umístění programového bloku






Režimy programového bloku

Programový blok **Random** slouží ke generování náhodného čísla ze zadaného rozsahu nebo logické hodnoty. Pro naše potřeby si vystačíme s režimem generování náhodného čísla. Blok v tomto režimu vypadá následovně:

## NUMERIC (GENEROVÁNÍ NÁHODNÝCH ČÍSEL)



-  **Lower Bound** – spodní hodnota rozsahu pro generování.
-  **Upper Bound** – horní hodnota rozsahu pro generování.
-  **Value** – výsledek generování náhodného čísla.



# Tvorba vlastních metod

Tvorba vlastních metod se využívá v situacích, kdy náš vytvořený programový kód dosahuje značných rozměrů a je tedy vhodné jej nějakým způsobem zjednodušit a zpřehlednit. To nám umožňuje průvodce pro vytváření vlastních bloků, díky kterému můžeme z určité části programu vytvořit nový programový blok.

## Kde tvorbu vlastních bloků v učebnici využíváme?

- Kapitola 10. parkovací asistent (zjednodušení rozsáhlejšího programu)

## Postup tvorby vlastního bloku

Funkci si představíme na příkladu, který používáme v učebnici. Jedná se o aktivitu 10.3.1 z kapitoly Parkovací asistent. Na obrázku máme hotový programový konstrukt, pomocí kterého robot detekuje začátek volného místa pro zaparkování a zjištění oznámí zvukovým signálem.



### 1) Výběr bloků

Jako první si musíme myší označit bloky, které chceme do nově vzniklého bloku zahrnout. Na obrázku vidíte správně označené bloky.

**Upozornění:** Do výběru nesmíte zahrnout úvodní blok *Start*, který nesmí být součástí vytvořeného bloku. Programovací prostředí vám výběr ani nepovolí a při výběru bloku *Start* zahlásí po spuštění tvůrce programových bloků chybu.



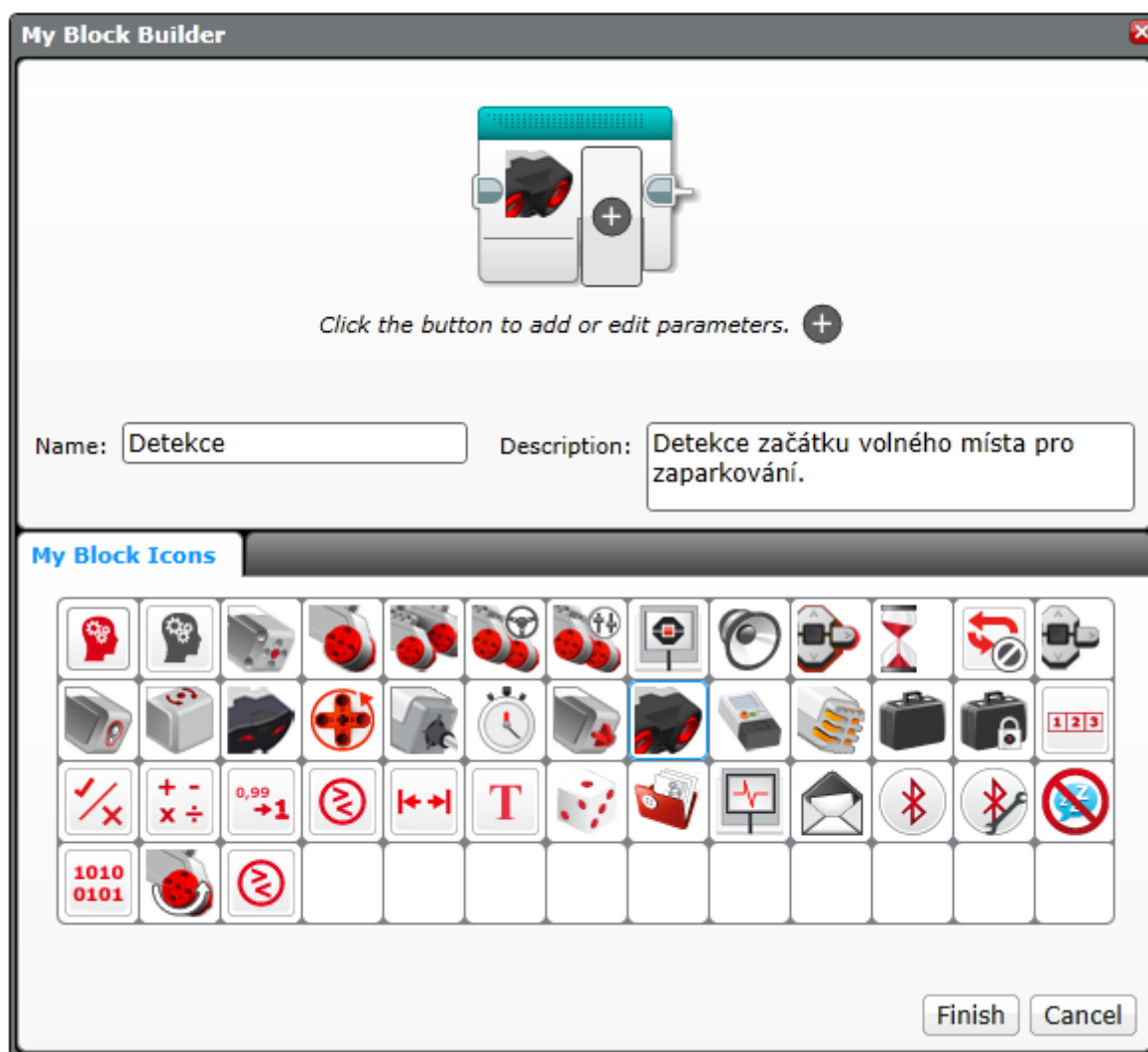
### 2) Spuštění průvodce pro tvorbu bloků

Funkce pro tvorbu vlastních bloků je umístěna v horním menu v záložce *Tools (nástroje)*, kde musíme vybrat funkci *My Block Builder (tvůrce bloků)*.



### 3) Návrh programového bloku

V okně, které se otevře, máme nyní možnost vytvořit podobu nového programového bloku. V horní části můžeme pomocí tlačítka + přidávat parametry programového bloku v podobě vstupních portů. Pole *Name* slouží k pojmenování bloku, které můžeme v poli *Description* doplnit vlastním volitelným popisem funkce nového bloku. Ve spodní části (*My Block Icons*) si můžeme zvolit ikonku, která bude náš blok reprezentovat. Nastavení poté stačí potvrdit tlačítkem *Finish*.



### 4) Nalezení programového bloku

Vytvořený programový blok se nám jednak zobrazí na programovací ploše, ale zároveň se nám přidá do poslední záložky s názvem *My Blocks*.



Obsah, který jsme do nového bloku vložili, si můžeme kdykoliv zobrazit. Stačí dvakrát kliknout na programový blok umístěný na ploše a otevře se nám nová záložka, ve které se zobrazí programový konstrukt, který jsme do vytvořeného bloku umístili.