

Programovací prostředí EV3 a RobotC robotické stavebnice LEGO Mindstorms EV3

Kapitola se věnuje programovacím prostředím EV3 a RobotC, která jsou určena k programování robotické stavebnice EV3. Popisuje samotná prostředí, vysvětluje princip vytváření programu v těchto dvou prostředích a popisuje také programovací jazyk k tomu určený. Seznámíte se s důležitými funkcemi EV3 a RobotC, které napomáhají k efektivnější práci s programem nebo jsou nezbytné pro vytváření programu či práci s řídicí jednotkou stavebnice. Kapitola si klade za cíl usnadnit Vám prvotní využívání některého z těchto dvou prostředí ve výuce.

Využité přístroje a software:

programovací prostředí EV3 a RobotC

robotická stavebnice LEGO Mindstorms EV3

Cílová skupina/náročnost: 1. až 4. ročník SŠ a odpovídající ročníky gymnázií

Všechny uvedené texty, obrázky a videa jsou vlastní, není-li uvedeno jinak. Autory Youtube embed videí lze nalézt při kliknutí na znak Youtube ve videu během přehrávání.

Autor:

Mgr. Jan Bařtko

K plnohodnotnému využití této studijní opory je nutný přístup k on-line zdrojům a materiálům.

Tento materiál vznikl z finanční podpory Evropského sociálního fondu a státního rozpočtu České republiky v rámci projektu „Popularizace vědy a badatelsky orientované výuky“, reg .č. CZ.1.07/2.3.00/45.0007.

Programovací prostředí EV3

1 Základní informace o projektu

Název

Programovací prostředí EV3

Anotace programu/zaměření/hlavní cíl

Programovací prostředí EV3 je ikonické programovací prostředí určené pro vytváření programů pro robotickou stavebnici LEGO Mindstorms EV3. Hlavním cílem této kapitoly je seznámit čtenáře nejprve se samotným programovacím prostředím a následně demonstrovat jeho funkce a možnosti pomocí jednoduchých aktivit přiřazených do některých kapitol.

Cílová skupina

1. až 4. ročník SŠ a odpovídající ročníky gymnázií

Organizační podmínky

Spolupráce studentů ve dvoučlenných, maximálně tříčlenných skupinách. Pro výuku je vhodné využít běžnou učebnu vybavenou počítači s nainstalovaným programovacím prostředím EV3.

Pomůcky

Robotická stavebnice LEGO Mindstorms EV3, programovací prostředí EV3.

Časová náročnost

Odhadovaná doba: zhruba 15 - 20 vyučovacích hodin.

Vazba na RVP

Rámcový vzdělávací program pro gymnázia.

Mezipředmětové vazby

Informatika (informační a komunikační technologie), elektrotechnika

2 Motivační rámec projektu



Text:

Všichni jistě znáte zkratku RUR (Rossumovi univerzální roboti). Zkratku, kterou ve svém stejnojmenném díle použil spisovatel Karel Čapek. V knize varoval před možnými negativními vlivy techniky na život člověka. My již dnes ale víme, že technika nám práci v mnohých případech hlavně usnadňuje. Navíc je nám také známo, že pokud se robot chová tak, že by ohrozil například naše zdraví, je to většinou chybou člověka. Špatnou manipulací, nedůslednou tvorbou nebo chybou ovládacího programu.

Při svém bádání s robotickou stavebnicí LEGO Mindstorms EV3 se ale žádného nebezpečí bát nemusíme. Veškeré chování robota máme pevně v rukou. V tomto modulu se dozvíte všechny podstatné informace týkající se robotického programovacího prostředí EV3. Jedná se o ikonické programovací prostředí umožňující vytváření robotických ovládacích programů. Při jeho používání nemusíte znát syntaxi žádného programovacího jazyka. Program se v něm vytváří pouze za pomoci logického uspořádávání a propojování programových bloků. Veškeré získané znalosti si můžete upevnit a ověřit splněním jednotlivých aktivit, které jsou obsaženy v každé kapitole.

Zdroj obrázku: commons.wikimedia.org, autor: D J Shin, BY-SA

Doporučený multimediální materiál

EV3 (home verze) ke stažení na oficiální stránkách LEGO: [stahujte ZDE](#) (odkaz viz. on-line kurz)

Ilustrační videa k demonstraci možností EV3: [více ZDE](#) (odkaz viz. on-line kurz)

Přehled možností EV3: [více ZDE](#) (odkaz viz. on-line kurz)

3 Poznámky k využití přístrojů

Pro tvorbu úvodu do práce s programovacím prostředím EV3 a sestavení úloh byla použita základní sada stavebnice EV3. Další informace o ní naleznete na následujících odkazech:

Informace na oficiálních internetových stránkách výrobce: [ZDE](#) (odkaz viz. on-line kurz)

Nabídka základních i doplňkových komponent na prodejním webu české společnosti EDUXE s.r.o.: [ZDE](#) (odkaz viz. on-line kurz)

Výrobce uvádí na svých stránkách také několik informací k programovacímu prostředí EV3. Naleznete je [ZDE](#). (odkaz viz. on-line kurz)

4 Projektový deník

Projektový deník slouží žákům k evidenci svého postupu v kurzu programování v EV3. K ověření znalosti každé části slouží příslušná aktivita, kterou by měl žák vyřešit. Při úspěšném vyřešení je proveden zápis do projektového deníku. Záznam by měl obsahovat poznámku o tom, kdy byla úloha zpracovávána, a poté krátký popis postupu a problémů, které bylo při tvorbě potřeba řešit. Po vyřešení úlohy vyučující zkontroluje funkčnost a správnost konstrukce nebo vytvořeného programu a zapiše do deníku hodnocení.

Projektový deník ke stažení v kurzu, zároveň jej najdete jako přílohu této tiskové opory.

5 Základní orientace v programovacím prostředí EV3

Programovací prostředí EV3 si můžeme rozdělit do několika částí, které si nyní popíšeme. Pro ilustraci je můžete najít vyznačené v následující animaci. (video viz. on-line kurz)

Popis jednotlivých částí EV3

Horní menu programu:

- obsahuje základní volby pro práci se souborem (vytvoření nového programu nebo projektu, uložení, editace),
- zahrnuje rozšiřující možnosti programovacího prostředí (editor zvuku a obrázků, tvůrce vlastních bloků, aktualizaci firmware, nastavení bezdrátového připojení, import nových programových bloků nebo správce paměti).

Programovací plocha:

- zabírá největší část programovacího prostředí,
- slouží k logickému uspořádávání a propojování programových bloků, začátek programu je zde vyznačen blokem Start.

Navigační tlačítka a orientace v programu:

- tlačítka umístěná vpravo nad programovací plochou,
- nalezneme zde výběrová tlačítka, vkládání komentářů, rychlou volbu pro ukládání, tlačítka pro krok zpět či vpřed nebo možnosti přiblížení a oddálení programovací plochy.

Knihovna programových bloků:

- umístěna pod programovací plochou, obsahuje všechny programovací bloky,
- rozdělena do šesti barevně odlišených kategorií (poslední obsahuje vlastní vytvořené bloky).

Správa připojených senzorů a modulů, informace o řídicí jednotce:

- okno umístěné v programovacím prostředí z pohledu uživatele vpravo dole,
- obsahuje informace o řídicí jednotce (verze firmware, zaplnění paměti), zařízeních připojených k řídicí jednotce, dostupné řídicí jednotky (připojené pomocí USB, Wi-fi nebo Bluetooth).

Tlačítka pro nahrání programu do řídicí jednotky:

- umístěná vpravo od okna pro správu řídicí jednotky,
- jedná se o volby buďto pouze pro stažení programu do řídicí jednotky, nebo i jeho okamžité spuštění,
- třetí tlačítko slouží k testování zvolené části programového kódu (po označení spustí zvolený úsek programového kódu).

5.1 Aktivita 1 - Příprava projektu

Téma	Příprava projektu v EV3	
Tematický celek	Programovací prostředí EV3	
Motivační rámec	Kdykoli vytváříte nějaký program, je dobré, abyste si jej vhodně pojmenovali, popsali, případně doplnili zdrojový kód o komentáře. Programovací prostředí EV3 umožňuje celý projekt doplnit o fotografie, videa či vlastní popis. Díky tomu i po delším čase, po kterém projekt otevřete, na první pohled zjistíte, jaká je funkce jednotlivých programů. V této aktivitě si přípravu nového projektu a jeho popis vyzkoušíte.	
Počet žáků	Skupina 8 - 10 studentů.	
Věk žáků	1. a 2. ročník SS a odpovídající ročníky gymnázií	
Pomůcky	Robotická stavebnice EV3 a počítače s nainstalovaným programovacím prostředím EV3, digitální fotoaparát nebo mobilní telefon.	
Stručný popis aktivity s využitím přístroje	Žáci si nejprve z robotické stavebnice sestaví vlastní model pojízdného robota pro úlohu zabývající se možnostmi pohybu pojízdného robota pomocí servomotorů. Pro tuto úlohu si vytvoří v programovacím prostředí EV3 nový projekt a příslušné programy, které si vhodně pojmenují. Následně pomocí digitálního fotoaparátu nebo mobilního telefonu pořídí fotografie svého modelu. Ty následně přidají do projektu v EV3, který si doplní vhodným popisem programu a dalšími komentáři.	
Vhodné místo	Běžná učebna vybavená počítači s nainstalovaným programovacím prostředím EV3.	
Cíle aktivity	Žáci budou schopni vytvořit nový projekt v programovacím prostředí EV3, doplnit ho o popisné informace, komentáře a multimediální prvky.	
Rozvíjené kompetence	Kompetence k učení, k řešení problémů.	
Předchozí znalosti	Nejsou potřeba žádné vstupní znalosti.	
Časový plán	Fáze činnosti s přístrojem	Metody a formy, motivace
15 minut	Tvorba modelu robota.	Spolupráce studentů ve skupinách, koordinace činnosti vyučujícím.
20 minut	Pořízení a úprava fotografií.	Spolupráce studentů ve skupinách, koordinace činnosti vyučujícím.
20 minut	Popis a tvorba projektu v programovacím prostředí EV3.	Spolupráce studentů ve skupinách, koordinace činnosti vyučujícím.
Hodnocení	Hodnocena bude úplnost a přehlednost vyplněných informací a kvalita doplňujících multimediálních prvků.	

Zadání

V této aktivitě se naučíte vytvořit a vhodným způsobem si připravit svůj první projekt v EV3. Příprava projektu je prvním krokem, který musíte absolvovat při vytváření nového programu. Zároveň jsou to také první úkony, díky kterým se zevrubně s programovacím prostředím seznámíte. Pro úspěšné absolvování tohoto úkolu musíte splnit toto zadání:

1. Sestavte z robotické stavebnice EV3 model pojízdného robota, který je poháněn dvěma servomotory.
2. Následně vytvořený model vhodně nafoťte pomocí digitálního fotoaparátu nebo mobilního telefonu. Nezapomeňte na to, že pořízené fotografie by měly vhodně představovat konstrukci robota případnému uživateli, který bude s programem v budoucnosti pracovat.
3. Založte si v programovacím prostředí nový projekt, uložte jej, vhodně pojmenujte jak projekt, tak dílčí program. Ve vlastnostech projektu stručně popište, k čemu projekt slouží, a doplňte jej o fotografie sestaveného modelu.

6 Základní výstupní moduly

Kapitola se věnuje základním výstupním modulům robotické stavebnice LEGO Mindstorms EV3. Výstupní zařízení slouží k různým možnostem signalizace či přenášení různé hnací síly navnek. V kapitole se věnujeme následujícím modulům:

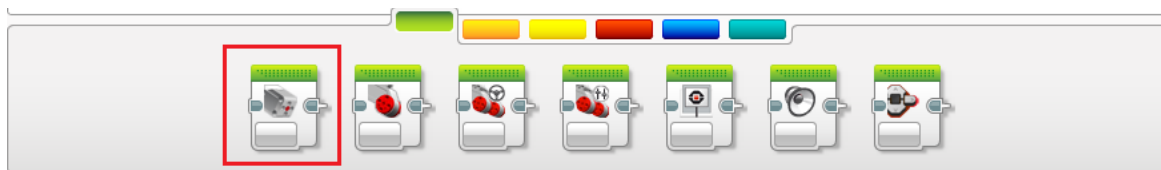
- Střední motor
- Velký servomotor
- Možnosti řízení více motorů
- Světelná signalizace řídicí jednotky
- Výstup na displej

6.1 Střední motor

Možnosti použití středního motoru

Umístění

Blok Medium Motor pro ovládání středního motoru je umístěn v zelené záložce Action (zobrazeno na obrázku). Použit jej můžeme přetažením na programovací plochu a umístěním na požadované místo v programu.



Režimy chodu středního motoru

Nastavení výstupního portu řídicí jednotky (A - D), ke kterému je motor připojen, provedeme v malém okně v pravém horním rohu programového bloku.

Vypnuto (Off)

Režim, který slouží k zablokování chodu motoru. Programový blok v tomto režimu obsahuje pouze jediný vstupní port, u kterého volíme, zda se má motor při pokynu k zastavení zastavit okamžitě (Break at End: True), nebo s pozvolným mírným dojezdem (Break at End: False).



Zapnuto (On)

Při použití programového bloku v tomto režimu dáváme motoru pokyn k neustálému otáčení zvolenou rychlostí (0 - 100 %), kterou nastavíme u jediného vstupního portu.



Otáčení po určitou dobu (On for Seconds)

Režim, který umožňuje nastavit otáčení motoru po určitou dobu stanovenou v sekundách. U prvního vstupního portu nastavíme požadovanou rychlost otáčení, u druhého zmíněný časový úsek a u třetího, zda se má motor na konci otáčení okamžitě zastavit (Break), nebo plynule dojet (Coast).



Otočení o určitý úhel (On for Degrees)

Režim, u kterého se míra natočení motoru stanovuje pomocí úhlových stupňů (0 - 360°). Zbývající dva vstupní porty jsou totožné jako u režimu pro otáčení po určitou dobu. Jedná se o nastavení rychlosti otáčení a způsob zastavení motoru (Break nebo Coast).



Otočení o určitý počet otáček (On for Rotations)

Režim je totožný se dvěma předchozími. Ovšem míra natočení se zde udává pomocí počtu nastavených otáček kolem své osy.



6.2 Aktivita 2 - Lopatková turbína

Téma	Lopatková turbína	
Tematický celek	Programovací prostředí EV3	
Motivační rámec	Trendem dnešní doby je využívání obnovitelných zdrojů energie. Zásoba přírodních zdrojů není nevyčerpatelná, a tak musíme stále hledat vhodné alternativy. Během slunečných dnů můžeme využívat například solární panely. Jak ovšem vyrábět elektřinu ve stejném objemu i v období, kdy není světla dostatek nebo v noci? V této úloze si sestrojíte jednoduchou turbínu fungující jako záložní způsob výroby energie v situacích, kdy je světla nedostatek.	
Počet žáků	Skupina 8 - 10 studentů.	
Věk žáků	1. až 4. ročník SS a odpovídající ročníky gymnázií	
Pomůcky	Robotická stavebnice EV3 a počítače s nainstalovaným programovacím prostředím EV3.	
Stručný popis aktivity s využitím přístroje	Žáci si nejprve vytvoří jednoduchý model lopatkové turbíny. Ta bude poháněna pomocí střední motoru. Její chod bude řízen barevným senzorem, který pomocí detekce okolního světla bude řídit rychlost otáčení.	
Vhodné místo	Běžná učebna vybavená počítači s nainstalovaným programovacím prostředím EV3.	
Cíle aktivity	Žáci se naučí využívat střední motor k pohonu rotačních součástí a řídit rychlost otáčení v závislosti na jiném zařízení.	
Rozvíjené kompetence	Kompetence k učení, k řešení problémů.	
Předchozí znalosti	Znalost práce s barevným senzorem.	
Mezipředmětové vztahy	Fyzika (světlo), informační a komunikační technologie (algoritmizace úloh).	
Casový plán	Fáze činnosti s přístrojem	Metody a formy, motivace
15 minut	Tvorba modelu robota.	Spolupráce studentů ve skupinách, koordinace činnosti vyučujícím.
20 minut	Tvorba programu a jeho průběžné testování.	Spolupráce studentů ve skupinách, koordinace činnosti vyučujícím.
20 minut	Testování funkčnosti modelu a programu.	Spolupráce studentů ve skupinách, koordinace činnosti vyučujícím.
Hodnocení	Hodnocena bude úplnost a přehlednost vyplněných informací a kvalita doplňujících multimediálních prvků.	

Zadání

Vytvořte funkční model turbíny jako záložního zdroje výroby elektrické energie. Požadavky na konstrukci modelu a program budou následující:

- Lopatky turbíny budou poháněny středním motorem.
- Řízení rychlosti otáčení bude prováděno díky barevnému senzoru.
 - a. Čím nižší bude intenzita okolního světla, tím se bude turbína otáčet rychleji.
- V programu vhodně nastavte prahovou hodnotu dostatečného slunečního svitu, při kterém se turbína zastaví.

Výsledný program ke stažení ve formátu .ev3

Program ke stažení v on-line kurzu

6.3 Velký servomotor

Možnosti použití velkého motoru

Umístění

Blok Large Motor pro ovládání velkého motoru je umístěn v zelené záložce Action (zobrazeno na obrázku). Použít jej můžeme přetažením na programovací plochu a umístěním na požadované místo v programu.



Režimy chodu velkého motoru

Nastavení výstupního portu řídicí jednotky (A - D), ke kterému je motor připojen, vybereme v malém okně v pravém horním rohu programového bloku.

Vypnuto (Off)

Režim, který slouží k zablokování chodu motoru. Programový blok v tomto režimu obsahuje pouze jediný vstupní port, u kterého volíme, zda se má motor při pokynu k zastavení zastavit okamžitě (Break at End: True), nebo s pozvolným mírným dojezdem (Break at End: False).



Zapnuto (On)

Při použití programového bloku v tomto režimu dáváme motoru pokyn k neustálému otáčení zvolenou rychlostí (0 - 100 %), kterou nastavíme u jediného vstupního portu.



Otáčení po určitou dobu (On for Seconds)

Režim otáčení motoru po určitou dobu zajišťuje otáčení po dobu, která se nastavuje na druhém vstupním portu v sekundách. Další volbou je u tohoto režimu rychlost otáčení udávaná v procentech (0 - 100 %) a způsob zastavení motoru. Pro okamžité zastavení volíme možnost Break, pro pozvolný dojezd motoru možnost Coast.



Otočení o určitý úhel (On for Degrees)

Režim On for Degrees funguje naprosto stejně jako předchozí režim On for Seconds. Místo doby otáčení nastavené v sekundách zde ovšem volíme míru natočení zadávanou ve stupních (0 - 360°). Všechny ostatní volby jsou totožné.



Otočení o určitý počet otáček (On for Rotations)

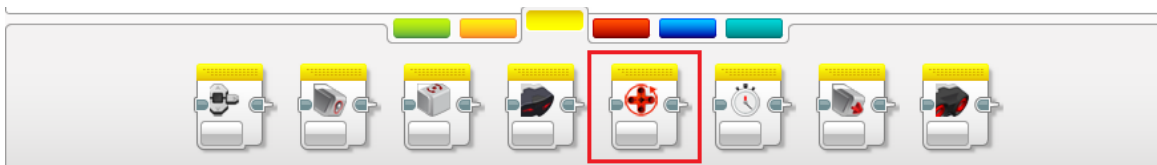
Režim natočení o určitý počet otáček je opět totožný jako předchozí. Míra natočení motoru se v tomto režimu udává jako počet provedených otáček kolem osy motoru.



Možnosti získávání dat z motorů

Umístění

Blok Motor Rotation pro získávání dat ze servomotoru je umístěn ve žluté záložce Sensor (zobrazeno na obrázku). Použít jej můžeme přetažením na programovací plochu a umístěním na požadované místo v programu.



Režimy měření dat

Nastavení výstupního portu řídicí jednotky (A - D), ke kterému je motor připojen, vybereme v malém okně v pravém horním rohu programového bloku.

Měření stupňů natočení (Measure Degrees)

Režim, ve kterém výstupní port bloku vrací změřené natočení motoru v úhlových stupních.



Měření otáček (Measure Rotations)

Režim, ve kterém výstupní port bloku vrací změřený počet otáček motoru.



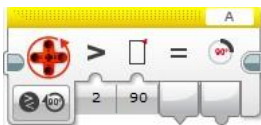
Měření rychlosti otáčení (Measure Current Power)

Režim, ve kterém výstupní port bloku vrací rychlost otáčení motoru od -100 do 100.



Porovnávání stupňů natočení (Compare Degrees)

Blok v tomto režimu nejen vrací počet stupňů natočení motoru, ale také umožňuje porovnávat zjištěnou hodnotu s hodnotou pevně zadanou (Threshold Value). Na výstupním portu Result navíc vrací hodnotu logického datového typu, který vyjadřuje, zda byla splněna podmínka nastavená pro porovnání (True), nebo nikoliv (False).



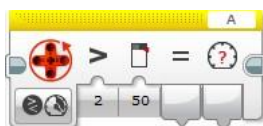
Porovnávání počtu otáček (Compare Rotations)

Blok v tomto režimu pracuje totožně jako v režimu pro porovnávání natočení v úhlových stupních. V tomto režimu ovšem pracujeme s počtem otáček.



Porovnávání rychlosti otáčení (Compare Current Power)

Režim bloku, který je totožný s oběma předchozími. Pouze pracuje s rychlostí otáčení motoru.



Reset (Nulování)

Režim, který nastavuje čítač všech tří měřených veličin (natočení ve stupních, otáčkách, rychlost) na nulu.



6.4 Aktivita 3 - Kuchyňská minutka

Téma	Kuchyňská minutka	
Tematický celek	Programovací prostředí EV3	
Motivační rámec	Pravděpodobně každý z nás se někdy v kuchyni setkal s kuchyňskou minutkou, malým zařízením sloužícím k měření času při přípravě pokrmů. Využívaly ji naše babičky i maminky. Doba pokročila a z klasických manuálních se vyvinuly i pokročilejší digitální, které jsou často opatřené displejem. Všechno potřebné k tomu, abychom podobnou minutku vytvořili z robotické stavebnice, ovšem máme k dispozici i my. Postačí nám k tomu displej řídicí jednotky a otočný mechanismus zajistí servomotor.	
Počet žáků	Skupina 8 - 10 studentů.	
Věk žáků	1. a 2. ročník SS a odpovídající ročníky gymnázií	
Pomůcky	Robotická stavebnice EV3 a počítače s nainstalovaným programovacím prostředím EV3.	
Stručný popis aktivity s využitím přístroje	Žáci si nejprve vytvoří vhodnou konstrukci kuchyňské minutky sestávající ze servomotoru opatřeného dobře ovladatelným otáčecím mechanismem pro zajištění pohodlného nastavení času měření. Odpočítávání začne ve chvíli, kdy bude stisknuto tlačítko připevněného dotykového senzoru.	
Vhodné místo	Běžná učebna vybavená počítači s nainstalovaným programovacím prostředím EV3.	
Cíle aktivity	Žáci se naučí využívat servomotor k ovládání pohyblivých součástí a pracovat s měřením a ovlivňováním vykonaného počtu otáček.	
Rozvíjené kompetence	Kompetence k učení, k řešení problémů.	
Předchozí znalosti	Znalost práce s dotykovým senzorem.	
Mezipředmětové vztahy	Informační a komunikační technologie (algoritmizace úloh).	
Časový plán	Fáze činnosti s přístrojem	Metody a formy, motivace
15 minut	Tvorba modelu robota.	Spolupráce studentů ve skupinách, koordinace činnosti vyučujícím.
30 minut	Tvorba programu a jeho průběžné testování.	Spolupráce studentů ve skupinách, koordinace činnosti vyučujícím.
10 minut	Testování funkčnosti modelu a programu.	Spolupráce studentů ve skupinách, koordinace činnosti vyučujícím.
Hodnocení	Hodnocena bude úplnost a přehlednost vyplněných informací a kvalita doplňujících multimediálních prvků.	

Zadání

Vytvořte funkční model kuchyňské minutky. Požadavky na konstrukci modelu a program budou následující:

1. Otočný mechanismus minutky bude realizován pomocí servomotoru.
2. Natočením mechanismu se díky detekci počtu otáček vyhodnotí, jak dlouhý časový úsek se má měřit.
3. Nastavovaný čas by se měl zobrazit na displeji.
4. Při měření času by se měla minutka otáčet a konec odpočítávání by měl oznámit zvukový signál.

Výsledný program ke stažení ve formátu .ev3

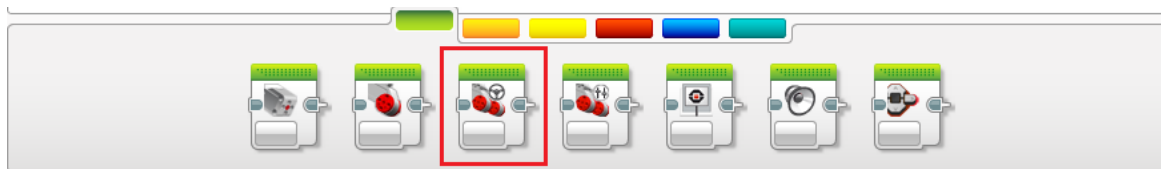
Program ke stažení v on-line kurzu

6.5 Řízení více motorů

Možnosti řízení robota pomocí dvou motorů

Umístění

Blok Move Steering pro řízení dvou servomotorů je umístěn v zelené záložce Action (zobrazeno na obrázku). Použít jej můžeme přetažením na programovací plochu a umístěním na požadované místo v programu.



Režimy bloku Move Steering

Nastavení výstupních portů řídicí jednotky (A - D), ke kterým jsou motory připojeny, provedeme v malém okně v pravém horním rohu programového bloku. Na rozdíl od řízení jediného motoru blokem Large Motor zde volíme dva výstupní porty.

Vypnuto (Off)

Jedná se o režim řízení, který zastaví motory, které byly do té doby v chodu. U vypnutí můžeme zvolit dva způsoby zastavení. Break, který motory okamžitě zastaví, a Coast, který umožní volné a setrvačné dojetí motorů.



Zapnuto (On)

Režim, který umožňuje souběžné otáčení motorů vpřed. Pomocí vstupního portu Steering je možné řídit směr, kterým se bude robot pohybovat (jiný než přímý). Druhý vstupní port (Power) slouží k nastavení rychlosti otáčení na stupnici od 0 do 100 %.



Otáčení po určité době (On for Seconds)

Režim, který je ve své podstatě totožný jako předchozí. Umožňuje ovšem řídit dobu otáčení motorů v sekundách pomocí vstupního portu Seconds. Druhou odlišností je možnost nastavení způsobu zastavení motorů (Break pro okamžité zastavení nebo Coast pro volný dojezd).



Otočení o určitý úhel (On for Degrees)

Režim s totožnými možnostmi jako režim On for Seconds. Místo nastavení doby pohybu ovšem využívá míru natočení motorů zadávanou ve stupních od 0 do 360°.



Otočení o určitý počet otáček (On for Rotations)

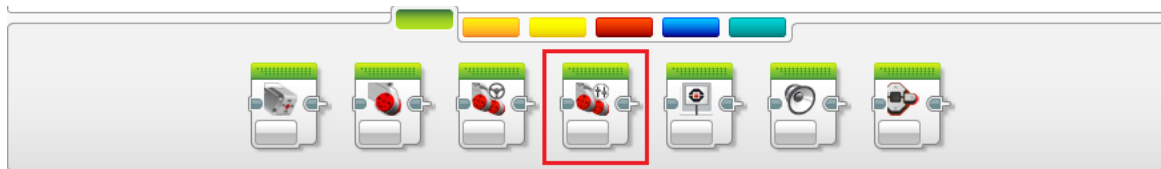
Třetí režim umožňující ovlivňovat dobu pohybu motorů. Oproti předchozím možnostem je zde nastavována pomocí počtu otáček motoru. Ostatní volby jsou totožné jako u předchozích režimů.



Možnosti řízení robota na bázi pásového vozidla

Umístění

Blok Move Tank pro řízení pásového vozidla je umístěn v zelené záložce Action (zobrazeno na obrázku). Použit jej můžeme přetažením na programovací plochu a umístěním na požadované místo v programu.



Vypnuto (Off)

Režimy bloku pro řízení pásového vozidla se v základu shodují s blokem pro řízení klasického robota řízeného dvěma motory (Move Steering). Režim pro zastavení (Off) zastaví oba připojené motory. Pomocí jediného vstupního portu můžeme volit způsob zastavení (Break pro okamžité zastavení nebo Coast pro volné dojetí).



Zapnuto (On)

Režim pro zapnutí a nekonečné otáčení motorů je již uzpůsoben řízení pásového vozidla. Obsahuje dva vstupní porty umožňující nastavit rychlost otáčení každému motoru zvlášť. Hodnota se pohybuje od -100 % do 100 %. Záporná hodnota vyjadřuje otáčení vzad, kladná poté otáčení vpřed.



Otáčení po určitou dobu (On for Seconds)

Rozšířením přechozího režimu On je režim On for Second. Ten umožňuje řídit dobu otáčení motoru v sekundách. Navíc umožňuje také zvolit způsob zastavení robota (Break nebo Coast).



Otočení o určitý úhel (On for Degrees)

Režim, který umožňuje řídit otáčení motorů nastavením míry jejich otočení v úhlových stupních (0 - 360°).



Otočení o určitý počet otáček (On for Rotations)

Režim On for Rotations slouží k řízení pohybu pásového vozidla nastavením počtu otáček, které mají motory vykonat. Ostatní nastavení je totožné jako u ostatních režimů pro ovlivnění doby otáčení motorů.



6.6 Aktivita 4 - Řízení pásového transportéru

Téma	Řízení pásového transportéru	
Tematický celek	Programovací prostředí EV3	
Motivační rámec	Při vytváření pojízdných modelů máme dvě možnosti, jak robota řídit. Prvním z typů je klasické kolové vozidlo a druhý model je ovládaný pomocí pásů. Pro každý z modelů využívá programovací prostředí EV3 jiný blok. Pokuste se tedy vymyslet, jak by mohl být ovládán pásový transportér nebo tank oproti klasickému kolovému vozidlu podobnému osobnímu automobilu.	
Počet žáků	Skupina 8 - 10 studentů.	
Věk žáků	1. až 4. ročník SS a odpovídající ročníky gymnázií	
Pomůcky	Robotická stavebnice EV3 a počítače s nainstalovaným programovacím prostředím EV3.	
Stručný popis aktivity s využitím přístroje	Žáci si neprve vytvoří model pojízdného robota opatřeného pásy. Robot se bude pomalu pohybovat vpřed a jeho jízda bude ovlivňována dvěma dotykovými senzory. Při stisknutí každého z nich se robot natočí na požadovanou stranu. Tlačítka tedy budou sloužit jako směrovače pro otáčení do stran.	
Vhodné místo	Běžná učebna vybavená počítači s nainstalovaným programovacím prostředím EV3.	
Cíle aktivity	Žáci se naučí využívat alternativní způsob pohonu servomotorů (pomocí pásů).	
Rozvíjené kompetence	Kompetence k učení, k řešení problémů.	
Předchozí znalosti	Znalost možností řízení motorů, práce s dotykovým senzorem a znalost práce s cykly a podmínkami.	
Mezipředmětové vztahy	Informační a komunikační technologie (algoritmizace úloh).	
Časový plán	Fáze činnosti s přístrojem	Metody a formy, motivace
20 minut	Tvorba modelu robota.	Spolupráce studentů ve skupinách, koordinace činnosti vyučujícím.
20 minut	Tvorba programu a jeho průběžné testování.	Spolupráce studentů ve skupinách, koordinace činnosti vyučujícím.
5 minut	Testování funkčnosti modelu a programu.	Spolupráce studentů ve skupinách, koordinace činnosti vyučujícím.
Hodnocení	Hodnocen bude sestavený model robota a úplnost a funkčnost vytvářeného programu.	

Zadání

Vytvořte funkční model robota poháněného pásy. Požadavky na konstrukci modelu a program budou následující: robot musí být dobře manévrovatelný a pásy na modelu dobře upevněny, natáčení robota do stran bude prováděno dvěma dotykovými senzory plnícími funkci joysticku, volte vhodnou rychlost otáčení motorů pro dobrou pohyblivost modelu.

Výsledný program ke stažení ve formátu .ev3

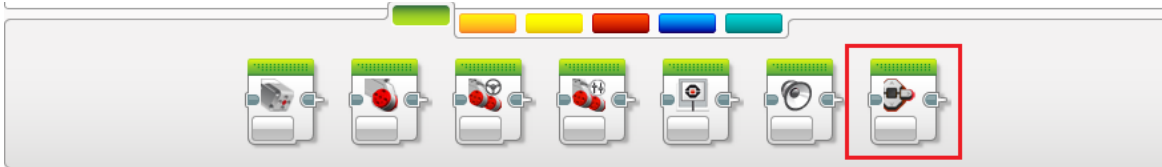
Program ke stažení v on-line kurzu.

6.7 Světelná a zvuková signalizace řídicí jednotky

Možnosti použití světelné signalizace řídicí jednotky

Umístění

Blok Brick Status Light pro řízení podsvícení tlačítek řídicí jednotky je umístěn v zelené záložce Action (zobrazeno na obrázku). Použít jej můžeme přetažením na programovací plochu a umístěním na požadované místo v programu.



Režimy bloku pro řízení světelné signalizace řídicí jednotky

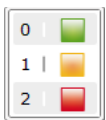
Vypnuto (Off)

Režim, který vypíná do té doby aktivní podsvícení řídicí jednotky.



Zapnuto (On)

Tímto režimem je možné zapnout světelnou signalizaci. Pomocí vstupního portu Color lze volit ze tří barevných provedení světelné signalizace, které můžete vidět na obrázku.



Vstupní port Pulse umožňuje spustit pulzování barevného podsvícení (True - spuštěno, False - vypnuto).



Základní nastavení (Reset)

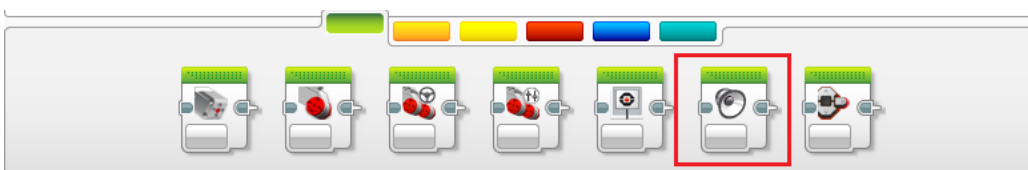
Režim Reset nastaví barevné podsvícení zpět do základního nastavení, což je zelené blikající podsvícení indikující spuštěný program.



Možnosti použití zvukové signalizace řídicí jednotky

Umístění

Blok Sound pro řízení podsvícení tlačítek řídicí jednotky je umístěn v zelené záložce Action (zobrazeno na obrázku). Použít jej můžeme přetažením na programovací plochu a umístěním na požadované místo v programu.



Zastavení přehrávání (Stop)

V tomto režimu blok zastaví přehrávání aktuálně přehrávaného zvuku prostřednictvím řídicí jednotky EV3.



Přehrání zvukového souboru (Play File)

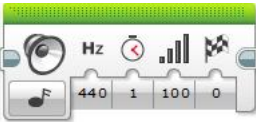
Režim umožňuje přehrát zvukový soubor uložený v řídicí jednotce EV3. Vstupní port Volume (hlasitost) slouží k nastavení hlasitosti přehrávaného zvuku od 0 do 100 % a vstupní port Play Type (způsob přehrávání) vyjadřuje, jak má být soubor přehrán. K dispozici máme následující možnosti:

- 0 - Čekání na dokončení (Wait for Completion) - čeká na dokončení přehrávání zvuku a poté pokračuje dále.
- 1 - Přehrání jedenkrát (Play Once) - zvuk je jednou přehrán a poté program pokračuje okamžitě dále.
- 2 - Opakování (Repeat) - zvuk je přehráván tak dlouho, dokud není přerušen blokem v režimu Stop.



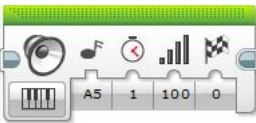
Přehrání tónu (Play Tone)

Režim pro přehrání tónu zvolené frekvence (Frequency) po určitou dobu trvání v sekundách (Duration) nastavenou hlasitostí. Možnosti přehrávání zvuku jsou totožné jako u předchozího režimu.



Přehrání noty (Play Note)

Režim pro přehrání zvolené hudební noty (Note) po určitou dobu v sekundách (Duration) zvolenou hlasitostí. Možnosti přehrávání zvuku jsou totožné jako u předchozího režimu.



6.8 Aktivita 5 - Železniční přejezd

Téma	Železniční přejezd	
Tematický celek	Programovací prostředí EV3	
Motivační rámec	Signalizace železničního přejezdu upozorňuje řidiče, že se k přejezdu blíží vlak, a je tudíž životu nebezpečné do něj vjíždět. Podoba železniční signalizace je různá. V této úloze si vytvoříte program, který bude simulovat funkci světelné signalizace na železničním přejezdu. Přidanou hodnotou bude zvuková signalizace upozorňující na blížící se nebezpečí.	
Počet žáků	Skupina 8 - 10 studentů.	
Věk žáků	1. a 2. ročník SS a odpovídající ročníky gymnázií	
Pomůcky	Robotická stavebnice EV3 a počítače s nainstalovaným programovacím prostředím EV3.	
Stručný popis aktivity s využitím přístroje	Úkolem žáků je využít zvukovou a světelnou signalizaci řídicí jednotky jako signalizátor blížícího se nebezpečí na železničním přejezdu. Detektorem blížícího se vlaku se stane ultrazvukový senzor, který pomocí naměřené vzdálenosti spustí světelnou a zvukovou signalizaci.	
Vhodné místo	Běžná učebna vybavená počítači s nainstalovaným programovacím prostředím EV3.	
Cíle aktivity	Žáci se naučí využívat světelnou a zvukovou signalizaci řídicí jednotky.	
Rozvíjené kompetence	Kompetence k učení, k řešení problémů.	
Předchozí znalosti	Znalost práce se světelnou a zvukovou signalizací.	
Mezipředmětové vztahy	Informační a komunikační technologie (algoritmizace úloh).	
Časový plán	Fáze činnosti s přístrojem	Metody a formy, motivace
40 minut	Tvorba programu a jeho průběžné testování.	Spolupráce studentů ve skupinách, koordinace činnosti vyučujícím.
Hodnocení	Hodnocena bude úplnost a přehlednost vyplněných informací a kvalita doplňujících multimediálních prvků.	

Zadání

Využijte řídicí jednotku k vytvoření programu, který bude simulovat signalizaci blížícího se vlaku na železničním přejezdu. Požadavky na tvorbu budou následující:

1. Použijte ultrazvukový senzor pro detekci blížícího se vlaku.
2. Při naměření kritické vzdálenosti se spustí přerušovaná světelná a také zvuková signalizace.
3. Jakmile se pomyslný vlak vzdálí opět za kritickou vzdálenost, signalizace se vypne.

Výsledný program ke stažení ve formátu .ev3

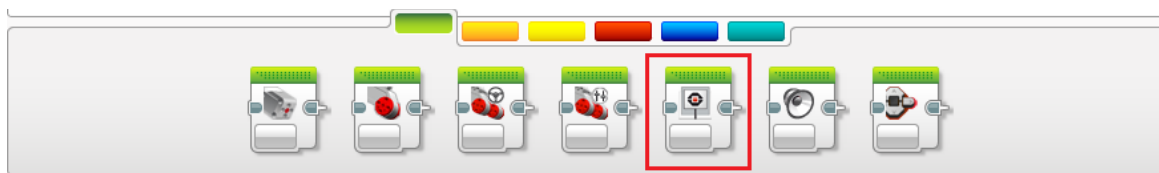
Program ke stažení v on-line kurzu.

6.9 Výstup na displej

Možnosti výstupu na displej řídicí jednotky

Umístění

Blok Display pro práci s displejem řídicí jednotky je umístěn v zelené záložce Action (zobrazeno na obrázku). Použít jej můžeme přetažením na programovací plochu a umístěním na požadované místo v programu.



Režimy programového bloku pro práci s displejem řídicí jednotky

Vypsání textu na pozici - v pixelech (Text - Pixels)

První ze dvou režimů umožňujících vypsání textu na displej umísťuje text na pozici konkrétního pixelu, který značí levý horní roh vypisovaného textového řetězce. Šířka displeje je 178 pixelů a výška 128 pixelů. Vstupní port X značí nastavení pozice v horizontální směru (-177 - +177). Vstupní port Y poté pozice ve vertikálním směru (-127 - +127). Volba Clear Screen umožňuje smazat displej před vypsáním textu. Volba Color umožňuje uživateli zvolit způsob vypsání textu. Pokud zvolíme Black, bude text vypsán černou barvou na bílé pozadí. Jestliže zvolíme White, bude vypsán bílým písmem na černé pozadí. U posledního vstupního portu Font můžeme nastavovat velikost vypisovaného textu (Normal, Bold nebo Large).

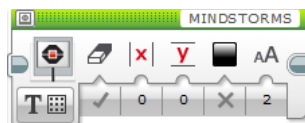
Požadovaný text k vypsání zapisujeme do pravého horního rohu bloku (na obrázku vzorově zapsán text MINDSTORMS). Tlačítkem Display Preview v pravém horním rohu bloku můžeme zobrazit náhled displeje a upravit tak případně pozici textu.

Po kliknutí na pole pro zápis textového řetězce je možné z menu vybrat volbu Wired. Díky tomu se na programovém bloku zobrazí vstupní port Text umožňující vypsát pomocí bloku Displej libovolný text přivedený na vstup pomocí vodiče.



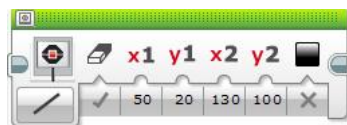
Vypsání textu na pozici - mřížka (Text - Grid)

Druhý režim pro výpis textu na displej obsahuje totožné volby jako režim Text - Pixels. Rozdíl je ovšem v pozici, na které je text umísťován. V tomto režimu není displej rozdělen na pixely, ale na mřížku tvořenou sloupci a řádky. Počátečním bodem výpisu je tedy buňka na zvolených souřadnicích. Displej je rozdělen na 22 sloupců po 8 pixelech (pozice X) a 12 řádků po 10 pixelech (pozice Y).



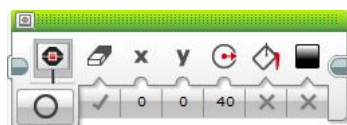
Vykreslení křivky (Shapes - Line)

Programový blok Display obsahuje čtyři režimy pro vykreslování obrazců. První z nich umožňuje vykreslovat křivky. Každá křivka je definována počátečním a koncovým bodem vykreslení. Souřadnice (x1, y1) určují počáteční pixel pro vykreslování a souřadnice (x2, y2) koncový pixel vykreslení. Stejně jako u vypsání textu umožňuje režim počáteční smazání displeje a volbu výpisu (černý text na bílém pozadí nebo bílý text na černém pozadí).



Vykreslení kruhu (Shapes - Circle)

Druhou možností pro vykreslení obrazců na displeji je zobrazení kruhu. Kruh se vykreslí na pozici, která je opět určena pixelem na souřadnicích X a Y. Pixel označuje střed kruhu. Druhým parametrem, který musíme nastavit, je poloměr kruhu (Radius). Ostatní možnosti jsou opět shodně s předchozími režimy.



Vykreslení čtverce nebo obdélníku (Shapes - Rectangle)

Režim pro vykreslení čtverce nebo na displej stejně jako předchozí pracuje s počátečními souřadnicemi pro vykreslení (x, y). Definují levý horní roh vykreslovaného obrazce. Následně musíme nastavit šířku (Width) a výšku (Height) obrazce udávanou v pixelech. Poslední charakteristickou volbou pro tento režim je možnost vykreslit obrazec s výplní (Fill).



Vykreslení bodu (Shapes - Point)

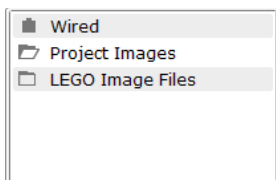
Posledním režimem pro vykreslení obrazců na displej je režim umožňující vykreslit jediný bod (pixel) displeje řídicí jednotky. Pixel je opět definován souřadnicemi (x, y). Následně již pouze záleží, zda chceme vykreslovat bod(y) černě na bílém pozadí, nebo bíle na černém pozadí.



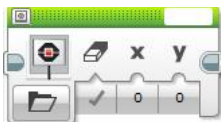
Vykreslení obrázku ze souboru (Image)

Na displej řídicí jednotky není možné vypisovat pouze text nebo obrazce pomocí propojených bodů (pixelů) displeje. Další možností je zobrazení obrázku uloženého v paměti řídicí jednotky. Obrázek vybereme po kliknutí na bílé pole v pravém horním rohu bloku.

Zobrazí se nám následující výběrové okno:



Zvolený obrázek následně vykreslíme na souřadnice (x, y). Pokud vybereme volbu Wired, zobrazí se na programovém bloku vstupní port File Name.



Smazání plochy displeje (Reset)

Režim, který umožňuje kompletně smazat plochu displeje řídicí jednotky.



6.10 Aktivita 6 - Kreslicí tabulka

Téma	Kreslicí tabulka	
Tematický celek	Programovací prostředí EV3	
Motivační rámec	Mnoho z nás si z dětství pamatuje ruční kreslicí tabulku s displejem a dvěma ovládacími kolečky pro kreslení. Jedno kolečko pro kreslení v horizontálním a druhé ve vertikálním směru. Jelikož tyto hračky nahradily jiné, mnohem propracovanější, setkáme se s takovou tabulkou jen zřídka. Není pro nás ovšem žádný problém si takovou tabulku pomocí robotické stavebnice vyrobit přímo pomocí řídicí jednotky. Jako ovládací prvky nám postačí její tlačítka.	
Počet žáků	Skupina 8 - 10 studentů.	
Věk žáků	1. a 2. ročník SS a odpovídající ročníky gymnázií	
Pomůcky	Robotická stavebnice EV3 a počítače s nainstalovaným programovacím prostředím EV3.	
Stručný popis aktivity s využitím přístroje	Úkolem žáků je vytvořit kreslicí tabulku pomocí displeje řídicí jednotky. Při stisku směrových tlačítek se bude vždy vykreslovat do požadovaného směru. Středové tlačítko slouží k mazání plochy displeje.	
Vhodné místo	Běžná učebna vybavená počítači s nainstalovaným programovacím prostředím EV3.	
Cíle aktivity	Žáci se naučí vykreslovat či vypisovat různé prvky na plochu displeje řídicí jednotky.	
Rozvíjené kompetence	Kompetence k učení, k řešení problémů.	
Předchozí znalosti	Znalost práce s tlačítky řídicí jednotky.	
Mezipředmětové vztahy	Informační a komunikační technologie (algoritmizace úloh).	
Casový plán	Fáze činnosti s přístrojem	Metody a formy, motivace
40 minut	Tvorba programu a jeho průběžné testování.	Spolupráce studentů ve skupinách, koordinace činnosti vyučujícím.
Hodnocení	Hodnocena bude úplnost a přehlednost vyplněných informací a kvalita doplňujících multimediálních prvků.	

Zadání

Vytvořte funkční model kreslicí tabulky. Požadavky na tvorbu budou následující:

1. Pro kreslení využijte plochu displeje řídicí jednotky.
2. Ovládání při vykreslování budou obstarávat tlačítka řídicí jednotky.
3. Prostřední tlačítko bude sloužit k mazání plochy displeje.

Výsledný program ke stažení ve formátu .ev3

Program ke stažení v on-line kurzu.

7 Základní vstupní moduly

Kapitola se věnuje základním vstupním modulům, které obsahuje základní sada stavebnice LEGO Mindstorms EV3. Vstupní zařízení snímají různé hodnoty z okolního prostředí nebo umožňují zasílat podněty zadávané uživatelem. Ty jsou poté na základě programu zpracovány.

V této kapitole naleznete informace o následujících vstupních zařízeních:

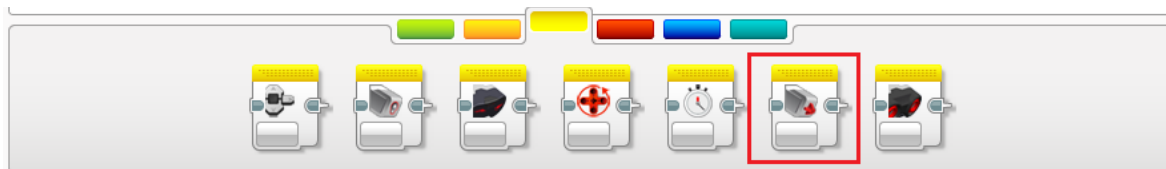
- Dotykový senzor
- Barevný senzor
- Gyroskopický senzor
- Ultrazvukový senzor
- Tlačítka řídicí jednotky

7.1 Dotykový senzor

Možnosti použití dotykového senzoru

Umístění

Blok Touch Sensor pro ovládání dotykového senzoru je umístěn ve žluté záložce Sensor (zobrazeno na obrázku). Použít jej můžeme přetažením na programovací plochu a umístěním na požadované místo v programu.



Režimy dotykového senzoru

Nastavení portu řídicí jednotky (1-4), ke kterému je senzor připojen, vybereme v malém okně v pravém horním rohu programového bloku **Měření stavu (Measure State)**

Pokud použijeme senzor v tomto režimu programového bloku, snímá pouze, zda bylo stisknuto tlačítko, či nikoliv. Blok proto obsahuje pouze jediný výstupní port, který nabývá hodnoty logického datového typu (výstupem může být True nebo False). Blok pracující v tomto režimu můžete vidět na obrázku.



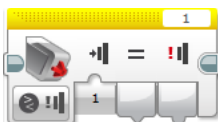
Porovnávání stavů (Compare State)

Při přepnutí programového bloku do režimu porovnávání stavů (Compare State) se nám zpřístupní jeden vstupní a dva výstupní porty bloku. Vstupní port status (State) umožňuje rozlišovat následující tři stavy tlačítka dotykového senzoru:

Released (0) - uvolnění tlačítka senzoru, Pressed (1) - stisk tlačítka senzoru, Bumped (2) - stisk a opětovné uvolnění tlačítka senzoru.

První z výstupních portů (Compare Result) vyjadřuje výsledek porovnání. Jeho hodnota je logického datového typu (True nebo False). Výsledek záleží na tom, zda byl detekován testovaný status tlačítka senzoru.

Druhý výstupní port (Measured Value) vrací číselné označení stavu tlačítka - uvolnění 0, stisk 1 a stisk a opětovné uvolnění 2. Vzhled programového bloku při práci v režimu porovnávání stavů můžete vidět na obrázku.



7.2 Aktivita 7 - Ruční mixer

Téma	Ruční mixér s použitím dotykového senzoru	
Tematický celek	Programovací prostředí EV3	
Motivační rámec	Dotykový senzor se hodí pro několik možností využití. Typicky se využívá jako tlačítko. Jeho velmi užitečnou vlastností je možnost rozeznávat tři stavy stisku. Tato aktivita se věnuje jeho použití jako spouštěče určité činnosti jiného zařízení.	
Počet žáků	Skupina 8 - 10 studentů.	
Věk žáků	1. a 2. ročník SS a odpovídající ročníky gymnázií	
Pomůcky	Robotická stavebnice EV3 a počítače s nainstalovaným programovacím prostředím EV3.	
Stručný popis aktivity s využitím přístroje	Úkolem této aktivity je seznámit studenty s možnostmi použití dotykového senzoru. Ten je v úkolu použit jako tlačítko ručního mixéru. Mixér je poháněn pomocí středního motoru. Při ovládní rozlišuje tři stavy. Po prvním stisku a uvolnění se roztočí pomaleji, po dalším stisku maximální rychlostí a při třetím stisku se vypne.	
Vhodné místo	Běžná učebna vybavená počítači s nainstalovaným programovacím prostředím EV3.	
Cíle aktivity	Studenti se naučí využívat dotykový senzor a rozeznávat jeho možné stavy.	
Rozvíjené kompetence	Kompetence k učení, k řešení problémů.	
Předchozí znalosti	Aktivita navazuje na kapitolu 5 Základní orientace v programovacím prostředí EV3.	
Mezipředmětové vztahy	Informační a komunikační technologie (algoritmizace).	
Casový plán	Fáze činnosti s přístrojem	Metody a formy, motivace
10 minut	Stavba modelu.	Spolupráce studentů ve skupinách, koordinace činnosti vyučujícím.
15 minut	Tvorba programu a průběžné testování.	Spolupráce studentů ve skupinách, koordinace činnosti vyučujícím.
5 minut	Závěrečné testování funkčnosti.	Spolupráce studentů ve skupinách, koordinace činnosti vyučujícím.
Hodnocení	Hodnocena bude průběžná práce na úloze, spolupráce studentů ve skupině a kvalita výsledného modelu a programu.	

Zadání

Aktivita si klade za cíl vás seznámit s funkcemi dotykového senzoru. V následující úloze se naučíte využít dotykový senzor jako spouštěč či prepínač určité reakce. Váš úkol je následující:

1. Postavte vhodný model ručního mixéru, který bude ovládaný pomocí tlačítka dotykového senzoru.
2. Po prvním stisknutí se mixér roztočí pomaleji, při druhém zrychlí na maximum a při třetím zastaví.

Výsledný program ke stažení ve formátu .ev3

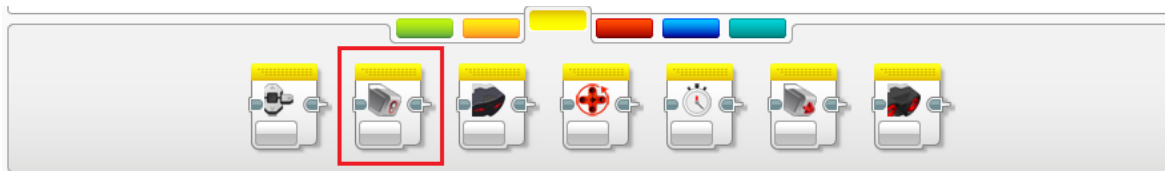
Program ke stažení v on-line kurz.

7.3 Barevný senzor

Možnosti použití dotykového senzoru

Umístění

Blok Color Sensor pro ovládání dotykového senzoru je umístěn ve žluté záložce Sensor (zobrazeno na obrázku). Použit jej můžeme přetažením na programovací plochu a umístěním na požadované místo v programu.



Režimy barevného senzoru

Nastavení portu řídicí jednotky (1-4), ke kterému je senzor připojen, vybereme v malém okně v pravém horním rohu programového bloku.

Rozlišení barev (Measure Color)



Režim rozlišení barev umožňuje zjišťovat barvu snímaného předmětu. Výstupem výstupního portu je číslo od 0 do 7 vyjadřující barvy tak, jak můžete vidět na obrázku.



Intenzita odraženého světla (Reflected Light Intensity)

V tomto režimu barevného senzoru můžeme měřit intenzitu odražené světla povrchů rozdílných barev. Každá barva či povrch odráží světlo jinak. Díky výstupnímu portu bloku v tomto režimu získáme intenzitu odraženého světla vyjádřenou číselně v rozmezí 0 - 100.



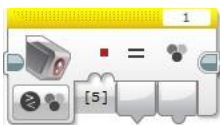
Intenzita okolního světla (Ambient Light Intensity)

Třetím režimem, ve kterém může barevný senzor pracovat, je režim měření intenzity okolního světla. Hodnotou, kterou v tomto režimu výstupní port vrací, je číselná hodnota v rozmezí 0 - 100.



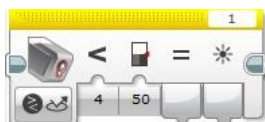
Porovnání barev (Compare Color)

V tomto režimu porovnáváme fixně zadané odstíny barev s barvou, kterou aktuálně senzor detekoval. Na vstupu Set of colors zvolíme barvy, které chceme detekovat. Následují dva výstupní porty. První z nich (Compare Result) je logického datového typu. Jeho výstup bude nabyvat hodnoty True v případě, že bude detekována některá ze zvolených barev. V opačném případě bude False. Druhý výstupní port vrací číselné označení (0-7) detekované barvy.



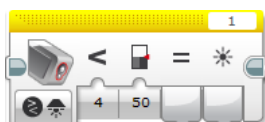
Porovnání intenzity odraženého světla (Compare - Reflected light intensity)

Režim pro porovnání hodnoty intenzity odraženého světla porovnává zjištěnou hodnotu s fixně zadanou prahovou hodnotou (Treshold value). Režim obsahuje také dva výstupní porty. První s názvem Compare Value vrací hodnotu logického datového typu. Nabývá hodnoty True v případě, že je splněna podmínka porovnání. V opačném případě vrací False. Druhý výstupní port vrací zjištěnou intenzitu světla na stupnici 0 - 100.



Porovnání intenzity okolního světla (Compare - Ambient light intensity)

Programový blok v tomto režimu pracuje naprosto stejně jako v režimu porovnání intenzity odraženého světla. Pouze pracuje s hodnotou intenzity světla v okolí.



Kalibrace - nastavení minima

Při práci s intenzitou odráženého světla je často nutné v programu určit, která hodnota má být vyhodnocována jako nejnižší a která jako nejvyšší. K tomu slouží u barevného senzoru kalibrace. V režimu minimum nastavuje hodnotu, která je dále v programu vyhodnocována jako minimální možná.



Kalibrace - nastavení maxima

Opakem k režimu Minimum je Maximum. V tomto režimu nastavujeme hodnotu, která bude dále v programu vyhodnocována jako maximální možná.



Kalibrace - nulování

Tento režim barevného senzoru nastaví znovu senzor do základního nastavení.



7.4 Aktivita 8 - Rozpoznávač barev

Téma	Rozpoznávač barev s využitím barevného senzoru	
Tematický celek	Programovací prostředí EV3	
Motivační rámec	Ne každý z nás má barevné citění a okamžitě rozpozná, o jakou barvu se jedná. Pociťujeme to hlavně při nákupu oblečení. Barev existuje tolik, že se v nich laik téměř nevyzná. Díky robotické stavebnici si ale dokážeme sestavit jednoduchý rozpoznávač barev, který nám alespoň základní barvy dokáže určit.	
Počet žáků	Skupina 8 - 10 studentů.	
Věk žáků	1. a 2. ročník SS a odpovídající ročníky gymnázií	
Pomůcky	Robotická stavebnice EV3 a počítače s nainstalovaným programovacím prostředím EV3.	
Stručný popis aktivity s využitím přístroje	Úkolem studentů je vytvořit jednoduchý model ručního rozpoznávače barev, který bude opatřen barevným senzorem. Ten bude zjišťovat barevný odstín snímaného materiálu a barvu následně vypíše na displej řídicí jednotky.	
Vhodné místo	Běžná učebna vybavená počítači s nainstalovaným programovacím prostředím EV3.	
Cíle aktivity	Studenti se naučí pracovat s barevným senzorem, seznámí se s jeho možnými režimy a s jeho využitím sestaví jednoduché zařízení k rozpoznávání barev.	
Rozvíjené kompetence	Kompetence k učení, k řešení problémů.	
Předchozí znalosti	Aktivita navazuje na kapitolu 6 Základní výstupní moduly.	
Mezipředmětové vztahy	Informační a komunikační technologie (algoritmizace), fyzika (světlo).	
Časový plán	Fáze činnosti s přístrojem	Metody a formy, motivace
10 minut	Stavba modelu.	Spolupráce studentů ve skupinách, koordinace činnosti vyučujícím.
15 minut	Tvorba programu a průběžné testování.	Spolupráce studentů ve skupinách, koordinace činnosti vyučujícím.
5 minut	Závěrečné testování a praktické ověření funkčnosti.	Spolupráce studentů ve skupinách, koordinace činnosti vyučujícím.
Hodnocení	Hodnocena bude průběžná práce na projektu a kvalita a funkčnost sestaveného modelu a vytvořeného programu.	

Zadání

V této aktivitě je vaším úkolem sestavit jednoduchý ruční rozpoznávač barev. Pro úspěšné splnění úlohy naplňte následující požadavky:

1. Sestavte model ručního rozpoznávače barev, který bude možné držet v jedné ruce (využijte barevný senzor).
2. Jakmile namíříte senzor na povrch, jehož barvu potřebujeme zjistit, informace o zjištěné barvě se vypíše na displej řídicí jednotky.

Výsledný program ke stažení ve formátu .ev3

Program ke stažení v on-line kurzu.

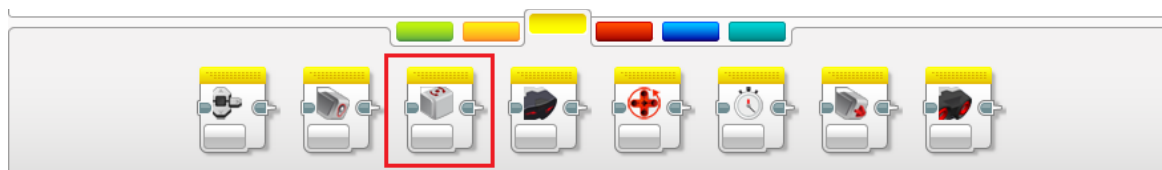
7.5 Gyroskopický senzor

Možnosti použití gyroskopického senzoru

Umístění

Blok Gyro Sensor pro ovládání ultrazvukového senzoru není součástí základní instalace Home verze programovacího prostředí EV3. Do programovacího prostředí je nutné jej doinstalovat. Blok naleznete ke stažení na stránkách společnosti LEGO (viz. on-line kurz)

Po instalaci se blok umístí do žluté záložky Sensor (zobrazeno na obrázku). Použit jej můžeme přetažením na programovací plochu a umístěním na požadované místo v programu.



Režimy gyroskopického senzoru

Nastavení portu řídicí jednotky (1-4), ke kterému je senzor připojen, vybereme v malém okně v pravém horním rohu programového bloku.

Měření úhlu natočení ve stupních (Measure Angle)

Režim měření úhlu umožňuje získávat údaje o míře natočení modelu robota, na kterém je gyroskopický senzor umístěn, v úhlových stupních. Jediný výstupní port vrací tento údaj v číselném vyjádření.



Rychlost rotace ve stupních za sekundu (Measure Rate)

Druhý režim měření s pomocí gyroskopického senzoru umožňuje získávat pomocí jediného výstupního portu hodnotu rychlosti rotace robota ve stupních za sekundu.



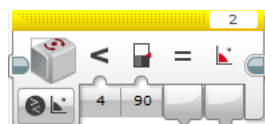
Měření úhlu natočení a rotace ve stupních za sekundu (Measure Angle and Rate)

Režim je kombinací obou předchozích.



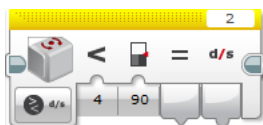
Porovnání úhlu natočení ve stupních (Compare Angle)

V tomto režimu porovnání můžeme poměřovat naměřenou hodnotu natočení s fixně zadanou hodnotou. Návratovou hodnotou je nám buďto výsledek splnění této podmínky (Compare Result) v podobě hodnoty True (při splnění), nebo False (při nesplnění). Druhou možností je vrácená hodnota udávaná ve stupních, kterou získáme pomocí druhého výstupního portu (Angle).



Porovnání naměřené rotace ve stupních za sekundu (Compare Rate)

Režim umožňuje totožné možnosti práce s hodnotou naměřenou gyroskopickým senzorem jako předchozí režim (Compare Angle). Místo úhlu natočení ve stupních ale pracuje s hodnotou naměřené rotace ve stupních za sekundu.



Nulování naměřeného natočení (Reset)

Režim Reset slouží k nulování posledního naměřeného natočení. Měření úhlu natočení se totiž určuje oproti poslednímu stavu robota. Je proto dobré vždy před měřením naměřenou hodnotu nulovat touto funkcí.



7.6 Aktivita 9 - Detektor pádu pevného disku

Téma	Detektor pádu disku	
Tematický celek	Programovací prostředí EV3	
Motivační rámec	Nenadálé problémy se spuštěním počítače mohou někdy pramenit z poruchy pevného disku počítače. Stát se tak může i v jiných případech. Například při pádu spuštěného notebooku na zem. Ať již je příčina jakákoliv, ve velké většině případů nastane nepříjemná ztráta dat. Pevné disky ovšem obsahují zařízení, které při detekci pádu dokáže bezpečně zaparkovat zápisovou hlavu tak, aby nedošlo k poškození disku. My si takové zařízení zkusíme vytvořit díky robotické stavebnici za pomoci gyroskopického senzoru.	
Počet žáků	Skupina 8 - 10 studentů.	
Věk žáků	1. a 2. ročník SS a odpovídající ročníky gymnázií	
Pomůcky	Robotická stavebnice EV3 a počítače s nainstalovaným programovacím prostředím EV3.	
Stručný popis aktivity s využitím přístroje	Žáci si nejprve sestaví jednoduchý model pevného disku. Uvnitř se pokusí vytvořit pohyblivý mechanismus, který bude simulovat jeho funkci (např. motor nebo pohyblivé rameno). Zařízení pro detekci pádu bude představovat gyroskopický senzor. Následně vytvoří program, který při prudké změně polohy zastaví chod disku.	
Vhodné místo	Běžná učebna vybavená počítači s nainstalovaným programovacím prostředím EV3.	
Cíle aktivity	Žáci se naučí využívat gyroskopický senzor.	
Rozvíjené kompetence	Kompetence k učení, k řešení problémů.	
Předchozí znalosti	Znalost práce s barevným senzorem .	
Mezipředmětové vztahy	Informační a komunikační technologie (algoritmizace úloh).	
Časový plán	Fáze činnosti s přístrojem	Metody a formy, motivace
15 minut	Tvorba modelu disku.	Spolupráce studentů ve skupinách, koordinace činnosti vyučujícím.
20 minut	Tvorba programu a jeho průběžné	Spolupráce studentů ve skupinách, koordinace činnosti vyučujícím.
15 minut	Testování funkčnosti modelu a	Spolupráce studentů ve skupinách, koordinace činnosti vyučujícím.
Hodnocení	Hodnocena bude úplnost a přehlednost vyplněných informací a kvalita doplňujících multimediálních prvků.	

Zadání

Vytvořte funkční model pevného disku se zařízením pro detekci pádu. Požadavky na konstrukci modelu a program budou následující:

1. Sestavte jednoduchý model pevného disku.
2. Uvnitř bude umístěn pohyblivý mechanismus poháněný například motorem.
3. Pro detekci pádu použijte gyroskopický senzor.
4. Jakmile bude detekována prudká změna polohy, chod disku se zastaví.

Výsledný program ke stažení ve formátu .ev3

Program ke stažení viz. on-line kurz

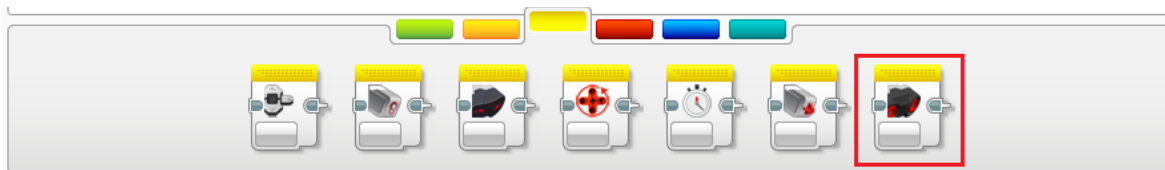
7.7 Ultrazvukový senzor

Možnosti použití ultrazvukového senzoru

Umístění

Blok Ultrasonic Sensor pro ovládání ultrazvukového senzoru není součástí základní instalace Home verze programovacího prostředí EV3. Do programovacího prostředí je nutné jej doinstalovat. Blok naleznete ke stažení na stránkách společnosti LEGO (viz. on-line kurz).

Po instalaci se blok umístí do žluté záložky Sensor (zobrazeno na obrázku). Použití jej můžeme přetažením na programovací plochu a umístěním na požadované místo v programu.



Režimy ultrazvukového senzoru

Nastavení portu řídicí jednotky (1-4), ke kterému je senzor připojen, vybereme v malém okně v pravém horním rohu programového bloku.

Měření vzdálosti - v centimetrech nebo palcích (Measure Distance - Centimeters, Inches)

První dva režimy, ve kterých může ultrazvukový senzor pracovat, slouží k měření vzdálenosti. První vyjadřuje naměřenou vzdálenost v centimetrech a druhý v palcích. V obou režimech se u programového bloku zobrazuje pouze jediný výstupní port, který vrací naměřenou hodnotu.



Detekce ultrazvukového signálu (Measure Presence)

Režim detekce ultrazvukového signálu pouze ověřuje, zda senzor zachytil nějaký ultrazvukový signál, či nikoliv. Blok v tomto režimu obsahuje pouze jediný výstupní port, který je logického datového typu. Hodnotu True vrací v případě, že signál zachytí. Ve všechno ostatních případech vrací False.



Rozšířené měření vzdálenosti (Measure Advanced - Centimeters, Inches)

Kromě jednoduchého režimu měření vzdálenosti obsahuje programový blok pro ultrazvukový senzor také rozšířený režim. Od původního režimu se liší možností nastavení způsobu vysílání ultrazvukového signálu. K dispozici máme tyto dvě možnosti:

- Ping (návratová hodnota 0) - vysílání signálu v pravidelných intervalech.
- Continuous (návratová hodnota 1) - souvislé (neustálé) vysílání signálu bez přerušení.



Porovnání vzdáleností v centimetrech nebo palcích (Compare Distance - Centimeters, Inches)

Režim slouží k porovnání vzdálenosti snímané senzorem s fixně zadanou hodnotou. Výstupem programového bloku v tomto režimu může být výsledek porovnání (výstupní port Compare Result), který vyjadřuje, zda byla splněna ověřovaná podmínka. Výstup je logického datového typu. Hodnotu True vrací v případě splnění podmínky, v opačném případě vrací False. Druhým výstupním portem je vzdálenost snímaná senzorem vyjádřená v centimetrech nebo v palcích.



Detekce ultrazvukového signálu (Compare Presence/Listen)

Režim, který je totožný s režimem Measure Presence.



7.8 Aktivita 10 - Turniket

Téma	Inteligentní turniket	
Tematický celek	Programovací prostředí EV3	
Motivační rámec	Na hudebních koncertech a různých sportovních akcích se zjišťuje různými způsoby aktuální počet návštěvníků. Většinou se tomu tak děje díky turniketům, které sčítají počet lidí, kteří jimi projdou. V následující úloze si takový turniket vytvoříme za pomoci ultrazvukového senzoru.	
Počet žáků	Skupina 8 - 10 studentů.	
Věk žáků	1. a 2. ročník SS a odpovídající ročníky gymnázií	
Pomůcky	Robotická stavebnice EV3 a počítače s nainstalovaným programovacím prostředím EV3.	
Stručný popis aktivity s využitím přístroje	V této úloze studenti vytvoří model jednoduchého turniketu, který sčítá počet návštěvníků kulturní akce. Ultrazvukový senzor bude přesně snímat prostor určený pro příchod do areálu. Jakmile jeho vysílaný signál protne některý z příchozích návštěvníků, bude započítán. Počet návštěvníků se bude postupně přičítat a vypisovat na displej.	
Vhodné místo	Běžná učebna vybavená počítači s nainstalovaným programovacím prostředím EV3.	
Cíle aktivity	Studenti se naučí pracovat s ultrazvukovým senzorem a naučí se zpracovávat jím zjištěná data.	
Rozvíjené kompetence	Kompetence k učení, k řešení problémů.	
Předchozí znalosti	Aktivita navazuje na kapitolu 5 Základní orientace v programovacím prostředí EV3.	
Mezipředmětové vztahy	Informační a komunikační technologie (algoritmizace).	
Casový plán	Fáze činnosti s přístrojem	Metody a formy, motivace
10 minut	Tvorba modelu turniketu.	Spolupráce studentů ve skupinách, koordinace činnosti vyučujícím.
25 minut	Tvorba programu a průběžné	Spolupráce studentů ve skupinách, koordinace činnosti vyučujícím.
5 minut	Ověření funkčnosti a závěrečné	Spolupráce studentů ve skupinách, koordinace činnosti vyučujícím.
Hodnocení	Hodnocena bude průběžná práce na projektu a kvalita sestaveného modelu a funkčnost vytvořeného programu.	

Zadání

V této aktivitě se seznámíte s funkcí ultrazvukového senzoru. Vaším úkolem je vytvořit inteligentní turniket pro kulturní akce, který sčítá počet návštěvníků. Požadavky na jeho funkčnost jsou následující:

1. Vytvořený model musí být jednoduchý a plně funkční (využijte ultrazvukový senzor).
2. Průchod turniketem musí být snímán ultrazvukovým senzorem, který bude detekovat každého příchozího návštěvníka.
3. Jakmile návštěvník turniketem projde, bude zaznamenán.
4. Aktuální počet návštěvníků se bude vypisovat na displeji řídicí jednotky.

Výsledný program ke stažení ve formátu .ev3

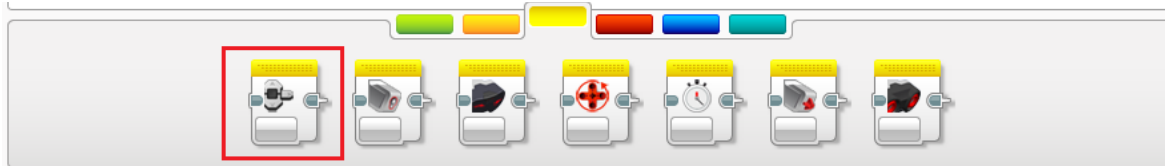
Program ke stažení viz. on-line kurz

7.9 Tlačítka řídicí jednotky

Možnosti použití tlačítek řídicí jednotky

Umístění

Blok Brick Buttons pro řízení reakcí na stisk tlačítek řídicí jednotky je umístěn ve žluté záložce Sensor (zobrazeno na obrázku). Použít jej můžeme přetažením na programovací plochu a umístěním na požadované místo v programu.

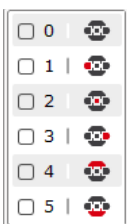


Režimy pro reakci na stisk tlačítek řídicí jednotky

Detekce stisku tlačítka (Measure Brick Buttons)

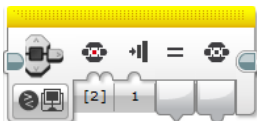


První režim, ve kterém můžeme pomocí bloku Brick Buttons pracovat se stiskem tlačítek řídicí jednotky, je režim pro detekci jejich stisků. Pomocí výstupního portu, který vrací číselné označení tlačítka (0 - 5), můžeme získat informaci o tom, které tlačítko řídicí jednotky bylo stisknuto. Schéma číslování tlačítek (návrátové hodnoty) můžete vidět na obrázku.



Compare Brick Buttons

Režim pro porovnání stisknutých tlačítek umožňuje zjišťovat, zda bylo stisknuto tlačítko, které požadujeme. Na prvním vstupním portu si zvolíme tlačítko, které chceme testovat. Pomocí druhého vstupního portu můžeme dokonce testovat, zda bylo stisknuto, uvolněno nebo stisknuto i uvolněno. První výstupní port Compare Result vrací návratovou hodnotu logického datového typu. Ta nabývá hodnoty True v případě, že je stisknuto testované tlačítko. V opačném případě vrací False. Druhý výstupní port vrací číselné označení stisknutého tlačítka (0 - 5).



7.10 Aktivita 11 - Bomba

Téma	Bomba
Tematický celek	Programovací prostředí EV3
Motivační rámec	Práce pyrotechnika není záviděníhodná. Stojí za ní roky zkušeností, znalostí elektrotechniky a hlavně pevné nervy. Zkušený pyrotechnik ovšem dokáže vyhodnotit povahu výbušniny a její funkci a zneškodnit ji. Vaším úkolem v této aktivitě ale bude vytvořit nepředvídatelnou bombu, jejíž chování nebude možné zjistit. Bude záviset pouze na štěstí, zda se jí povede zneškodnit, či nikoliv.
Počet žáků	Skupina 8 - 10 studentů.
Věk žáků	1. a 2. ročník SS a odpovídající ročníky gymnázií
Pomůcky	Robotická stavebnice EV3 a počítače s nainstalovaným programovacím prostředím EV3.
Stručný popis aktivity s využitím přístroje	Úkolem žáků je vytvořit program, který bude simulovat bombu. Potenciálnímu pyrotechnikovi se na displeji zobrazí výzva, aby bombu zneškodnil jedním ze tří nabízených tlačítek. U žádného z nich ovšem nebude zaručeno, že je správné. Význam tlačítek se totiž bude náhodně generovat. Program žáci doplní o vhodnou zvukovou signalizaci a výpis na displej.
Vhodné místo	Běžná učebna vybavená počítači s nainstalovaným programovacím prostředím EV3.
Cíle aktivity	Studenti se naučí pracovat s tlačítky řídicí jednotky a naučí se vyhodnocovat jejich stisk a reagovat na něj.
Rozvíjené kompetence	Kompetence k učení, k řešení problémů.
Předchozí znalosti	Znalost práce s tlačítky řídicí jednotky.
Mezipředmětové vztahy	Informační a komunikační technologie (algoritmizace).
Casový plán	Fáze činnosti s přístrojem Metody a formy, motivace
40 minut	Tvorba programu a jeho průběžné Spolupráce studentů ve skupinách, koordinace činnosti vyučujícím.
Hodnocení	Hodnocena bude průběžná práce na projektu a funkčnost vytvořeného programu.

Zadání

V této aktivitě se naučíte využívat tlačítka řídicí jednotky. Vaším úkolem je vytvořit z řídicí jednotky pomyslný model bomby. Co pro úspěšnou realizaci musíte zvládnout?

1. Zvolte si alespoň tři tlačítka řídicí jednotky, která budou představovat dráty, které musí pyrotechnik přestříhnout.
2. Programově zajistěte, aby žádné tlačítko nebylo určeno ke zneškodnění bomby, ale aby se jejich funkce náhodně měnila.
3. Program vhodně graficky a zvukově ošetřete.

Výsledný program ke stažení ve formátu .ev3

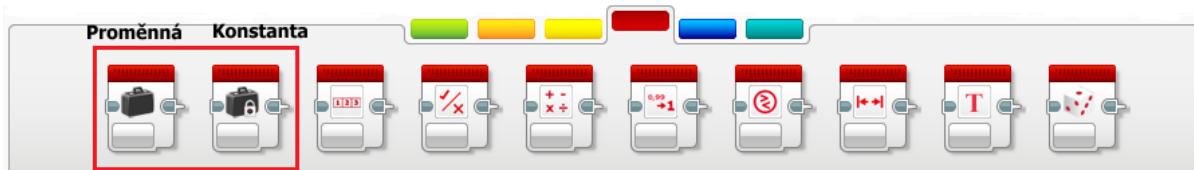
Program ke stažení viz. on-line kurz

8 Proměnné, konstanty, datové typy

Možnosti využití proměnné

Umístění

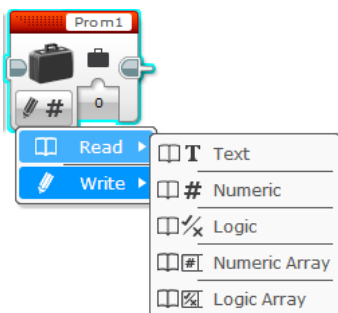
Blok Variable pro použití proměnné a blok Constant pro použití konstanty jsou umístěny v červené záložce Data Operations. Použít je můžeme přetažením na programovací plochu. Rozeznáme je lišícím se vzhledem programového bloku. Konstanta obsahuje malý symbol zámku značící fixní hodnotu, která je v ní uložena.



Možnosti čtení a zápisu

Konstantu je možné použít pouze pro čtení hodnoty v ní uložené. Proměnnou je možné použít jak pro čtení dat v ní uložených, tak pro zápis. U čtení i u zápisu musíme zvolit datový typ, kterého bude proměnná nabývat.

Při programování v EV3 můžeme využít následující datové typy: Text - pro textové a znakové řetězce, Numeric - pro čísla, Logic - pro hodnoty logického datového typu (True a False), Numeric Array - pro číselná pole, Logic Array - pro pole obsahující hodnoty logického datového typu.



Rozdíl mezi proměnnou pro čtení a pro zápis

Proměnná pro čtení umožňuje využít uloženou hodnotu. Pomocí datového vodiče ji proto můžeme z výstupního portu bloku přivést na vstupní port jiného programového bloku. U proměnné pro zápis můžeme naopak hodnotu do bloku Variable přivést, a tedy uložit. Rozdíl mezi blokem nastaveným pro čtení a pro zápis můžete vidět na obrázku.



9 Programové řízení

V programování využíváme často konstrukce, které se v programu vykonávají opakovaně. Abychom je nemuseli v kódu zapisovat několikrát, což by mohlo být v některých případech nereálné, využíváme cykly. V případech, kdy potřebujeme rozhodnout o naplnění nějakého stavu, hodnoty nebo adekvátně reagovat na nastalé podmínky, využijeme podmíněné výrazy. Tato kapitola se věnuje možnostem využití cyklů a podmínek v programovacím prostředí EV3.

Podkapitoly:

- Cykly
- Podmínky

9.1 Cykly

Možnosti použití cyklu v EV3

Umístění

Blok Loop, který v EV3 reprezentuje cyklus, je umístěn v oranžové záložce Flow Control (zobrazeno na obrázku). Použít jej můžeme přetažením na programovací plochu a umístěním na požadované místo v programu.



Popis bloku cyklu

Název cyklu - v horní části cyklu je možné volit jeho název (v políčku, ve kterém je defaultně nastaveno číslování od 01 dále). Volba režimu - slouží ke zvolení způsobu řízení cyklu (například počtem průchodů, časem, senzorem atd.).

Výstupní port - vyjadřuje počet opakování cyklu.

Vstupní port - závisí na zvoleném režimu (ovlivňuje například počet průchodů cyklem - u režimu Count). Prostor pro umístění programových bloků - slouží k umístění bloků programu, které se budou vykonávat.

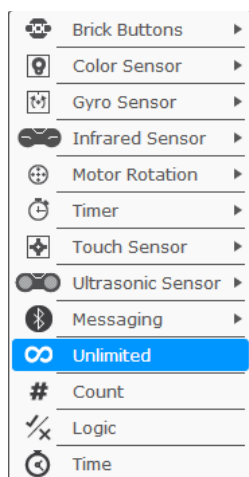


Možnosti řízení cyklu

Programovací prostředí EV3 neumožňuje vytvářet cykly s podmínkou na začátku, ale pouze s podmínkou na konci a s pevným počtem průchodů. Na obrázku můžete vidět možnosti řízení cyklu:

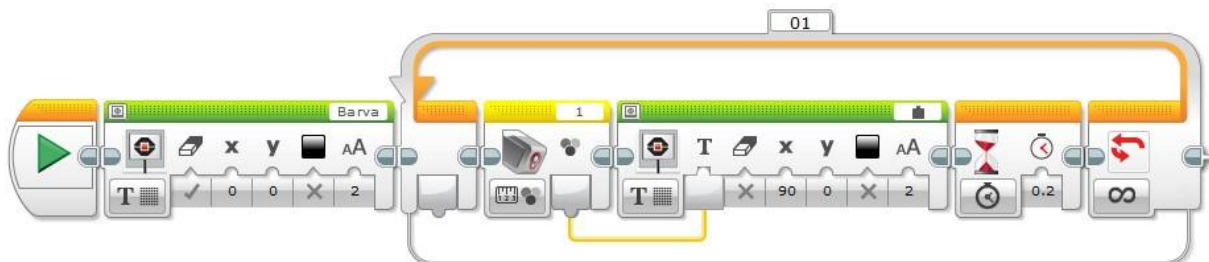
Senzorem - počet možností řízení senzorem se odvíjí od počtu nainstalovaných programových bloků pro ovládání senzorů. Nekonečné provádění cyklu (Unlimited) - možnost neustálého provádění (nekonečný cyklus).
Cyklus s pevným počtem průchodů (Count) - cyklus je vykonáván na základě nastaveného počtu průchodů.

Cyklus řízený logickou hodnotou (Logic) - cyklus vykonávaný, dokud je naplněna logická hodnota True nebo False. Cyklus řízený časem (Time) - cyklus je vykonáván po určitou dobu uváděnou v sekundách.



Příklad použití cyklu

Na obrázku můžete vidět příklad použití cyklu. Jednoduchý programový konstrukt díky nekonečnému provádění cyklu neustále zjišťuje, jaké barvy je snímáný povrch pomocí barevného senzoru. Hodnotu poté vypisuje na displej.

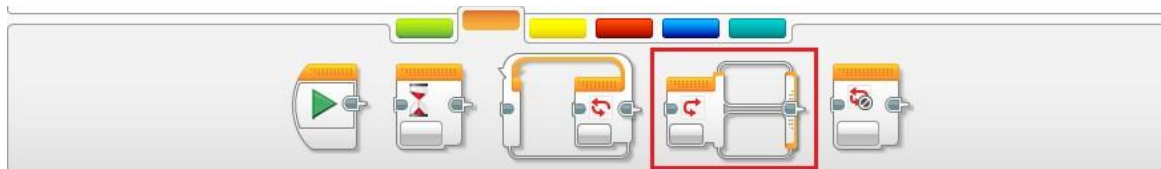


9.2 Podmínky

Možnosti použití podmínek v EV3

Umístění

Blok Switch, který v EV3 reprezentuje podmínku, je umístěn v oranžové záložce Flow Control (zobrazeno na obrázku). Použit jej můžeme přetažením na programovací plochu a umístěním na požadované místo v programu.

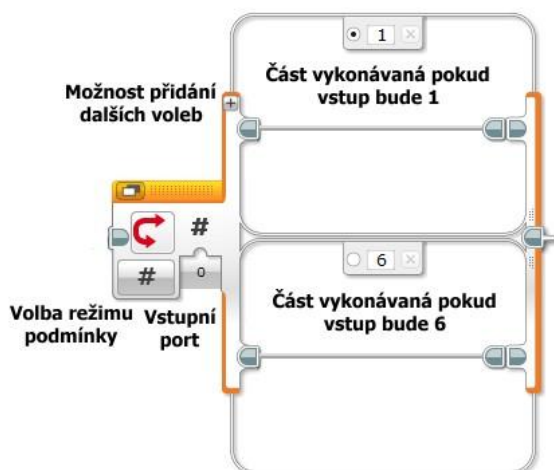


Popis bloku cyklu

Volba režimu - slouží ke zvolení způsobu řízení podmínky (například vstupní číselnou hodnotou, logickou hodnotou, senzorem atd.). Vstupní port - slouží k testování porovnávané hodnoty s hodnotou fixně zadanou, testování specifických hodnot snímaných senzorem atd.).

Prostor pro umístění programových bloků - v základu máme k dispozici část, do které umísťujeme sekvenci programových bloků, která se vykoná v případě, že je podmínka splněna, a druhou část, která se vykoná, když podmínka splněna není. Blok switch umožňuje vytvořit také přepínač case. Slouží k tomu malá ikonka plus v horní části bloku, díky které můžeme přidat další možnosti. Na základě toho můžeme například u barevného senzoru rozlišovat, která barva byla detekována, a na základě toho programově reagovat (podobu bloku Switch při práci s barevným senzorem vidíte na obrázku).

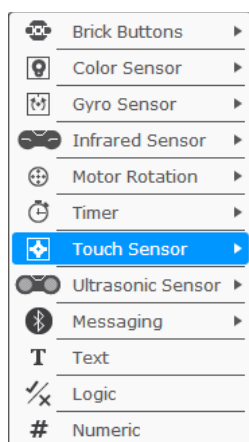
Volba portu - zobrazuje se (v levé části bloku) pouze v případě, že je podmínka řízena senzorem.



Možnosti řízení podmínky

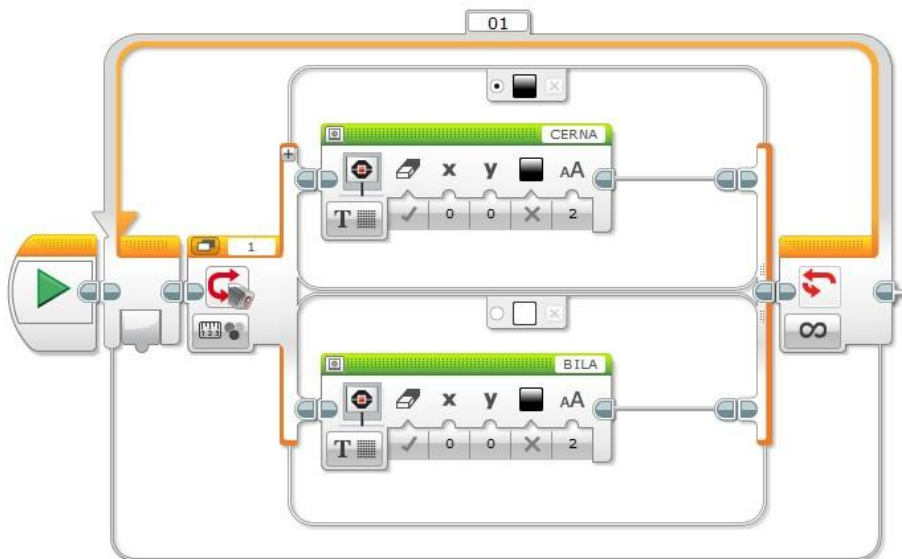
Programovací prostředí EV3 neumožňuje řídit podmínky několika způsoby, které můžete vidět na obrázku: Senzorem - vykonávání podmínky je řízeno vstupní hodnotou zjištěnou senzorem.

Řetězcem znaků - podmínka je vykonávána na základě ověřování textového řetězce přivedeného na vstup. Logic - vykonávání podmínky na základě ověřování logické hodnoty (True nebo False) na jejím vstupu. Numeric - podmínka vykonávaná na základě porovnání číselné hodnoty na vstupu.



Příklad použití cyklu

Na obrázku můžete vidět příklad použití kombinace podmínky a cyklu. Program ověřuje, zda byla detakována černá nebo bílá barva pomocí barevného senzoru, a výsledek vypisuje na displej.



9.3 Aktivita 12 - Detektor světla

Téma	Detektor světla	
Tematický celek	Programovací prostředí EV3	
Motivační rámec	Zařízení detekující úroveň světla v okolí nalezneme v mnoha mechanismech. Jedním z nich může být například automatické osvětlení, které na základě detekce světelných podmínek spouští pouliční osvětlení, či nikoliv. My se v této aktivitě pokusíme takový detektor sestavit. Využijeme k tomu barevný senzor, který režim detekce světla v okolí obsahuje.	
Počet žáků	Skupina 8 - 10 studentů.	
Věk žáků	1. až 4. ročník SS a odpovídající ročníky gymnázií	
Pomůcky	Robotická stavebnice EV3 a počítače s nainstalovaným programovacím prostředím EV3.	
Stručný popis aktivity s využitím přístroje	Úkolem žáků je sestavit a naprogramovat jednoduchý detektor světla, který rozlišuje světelné podmínky v okolí. Na displeji následně zobrazuje, zda je světla v okolí dostatek, či nikoliv.	
Vhodné místo	Běžná učebna vybavená počítači s nainstalovaným programovacím prostředím EV3.	
Cíle aktivity	Žáci se naučí využívat cykly a podmínky.	
Rozvíjené kompetence	Kompetence k učení, k řešení problémů.	
Předchozí znalosti	Znalost práce s cykly a podmínky. Znalost režimů barevného senzoru.	
Mezipředmětové vztahy	Informační a komunikační technologie (algoritmizace).	
Časový plán	Fáze činnosti s přístrojem	Metody a formy, motivace
45 minut	Tvorba programu a jeho průběžné	Spolupráce studentů ve skupinách, koordinace činnosti vyučujícím.
Hodnocení	Hodnocena bude průběžná práce na projektu a funkčnost vytvořeného programu.	

Zadání

V této aktivitě se naučíte využívat tlačítka řídicí jednotky. Vaším úkolem je vytvořit z řídicí jednotky pomyslný model bomby. Co pro úspěšnou realizaci musíte zvládnout?

1. Zvolte si alespoň tři tlačítka řídicí jednotky, která budou představovat dráty, které musí pyrotechnik přestříhnout.
2. Programově zajistěte, aby žádné tlačítko nebylo určeno ke zneškodnění bomby, ale aby se jejich funkce náhodně měnila.
3. Program vhodně graficky a zvukově ošetřete.

Výsledný program ke stažení ve formátu .ev3

Program ke stažení viz. on-line kurz

10 Matematické operace

Provádění matematických operací je nezbytnou nutností v programování v jakémkoliv jazyce. V této kapitole se dozvíte o následujících matematických operacích:

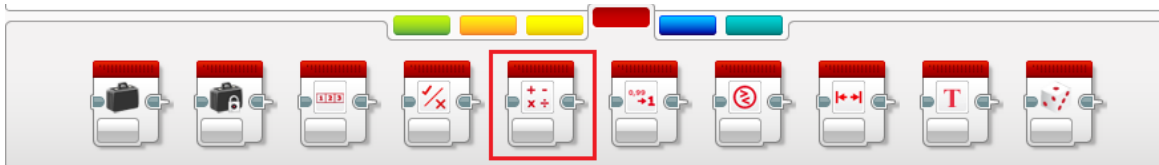
- Základní matematické operace
- Logické operace
- Pole
- Generování náhodných čísel
- Rozšiřující matematické operace

10.1 Základní matematické operace

Možnosti využití základních matematických operací

Umístění

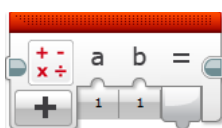
Blok Math pro provádění základních matematických operací je umístěn v červené záložce Data Operations (zobrazeno na obrázku). Použít jej můžeme přetažením na programovací plochu a umístěním na požadované místo v programu.



Režimy pro provádění základních matematických operací

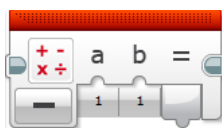
Sčítání (Add)

Režim pro sčítání (Add) umožňuje sečíst dvě vstupní hodnoty. Ty můžeme na portech A a B buďto ručně zadat, nebo je přivést pomocí datového vodiče. Výsledek poté získáme pomocí výstupního portu Result.



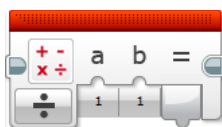
Odčítání (Subtract)

Režim pro odčítání (Subtract) umožňuje odečíst dvě vstupní hodnoty. Můžeme je buďto ručně zadat na portech A a B, nebo přivést pomocí datového vodiče. Výsledek opět získáme na výstupním portu Result.



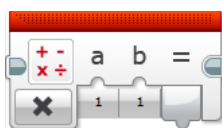
Dělení (Divide)

Režim pro dělení funguje principiálně stejně jako režimy pro sčítání a odčítání. Na vstupních portech A a B zadáme vstupní hodnoty a na výstupu Result následně získáme výsledek.



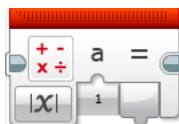
Násobení (Multiply)

Režim pro násobení je opět totožný s předchozími. Na vstupních portech A a B zadáme vstupní hodnoty a na výstupu Result poté získáme výsledek.



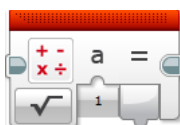
Absolutní hodnota (Absolut Value)

Režim Absolut Value vrací na výstupním portu Result absolutní hodnotu čísla zadaného na vstupním portu A.



Odmocnina (Square Root)

Režim Square Root vrací na výstupním portu Result odmocninu čísla zadaného na vstupním portu A.



Mocnina (Exponent)

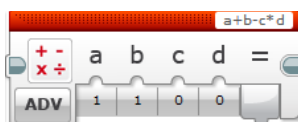
Režim Exponent umožňuje pracovat s odmocninami. Na vstupním portu A zadáme základ mocniny (číslo) a na vstupním portu N požadovaný exponent. Výsledek operace následně vrací výstupní port Result.



Advanced (Pokročilé funkce)

V režimu Advanced je možné počítat složitější matematické výrazy či rovnice. K dispozici máme 4 vstupní porty (A, B, C, D) pro zadání číselných hodnot. Výsledek následně vrací výstupní port Result.

Zápis rovnice či výrazu se provádí v okně, které se zobrazí po kliknutí do bílého pole v pravém horním rohu bloku. K zápisu lze využít množství operací a funkcí (základní matematické operace, goniometrické funkce, mocniny, odmocniny a další).

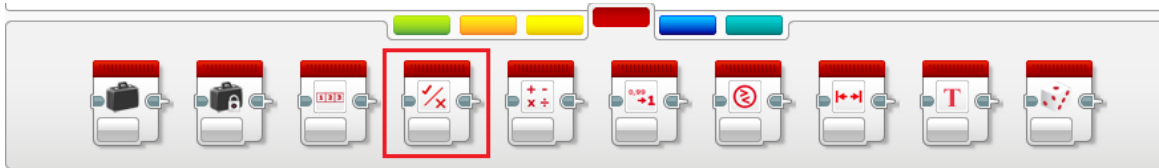


10.2 Logické operace

Možnosti využití logických matematických operací

Umístění

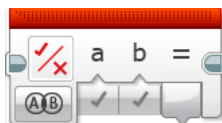
Blok Logic Operations pro provádění logických matematických operací je umístěn v červené záložce Data Operations (zobrazeno na obrázku). Použít jej můžeme přetažením na programovací plochu a umístěním na požadované místo v programu.



Režimy pro provádění logických matematických operací

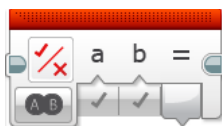
Logický součin (And)

Režim pro logický součin obsahuje dva vstupní porty logického datového typu (mohou nabývat hodnot True a False). Výsledek operace je True v případě, že hodnoty obou vstupních portů nabývají hodnotu True. Výsledek vrací výstupní port Result.



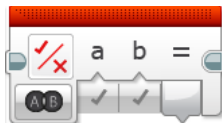
Logický součet (Or)

Režim pro logický součet obsahuje dva vstupní porty A a B, které jsou logického datového typu (nabývají hodnot True nebo False). Výsledkem operace, který vrací výstupní port Result, je True v případě, že hodnota alespoň jednoho vstupního portu je True.



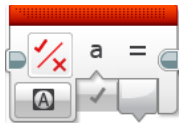
Exkluziv Or (XOR)

Režim exkluziv Or obsahuje, stejně jako oba předchozí režimy, dva vstupní porty (A a B) logického datového typu. Výsledek vrací výstupní port Result. Výsledkem operace je True v případě, že pouze jeden ze vstupních portů nabývá hodnoty True.



Negace (Not)

Režim Not umožňuje negaci vstupní hodnoty logického datového typu na portu A. Výsledkem je tedy obrácená hodnota.

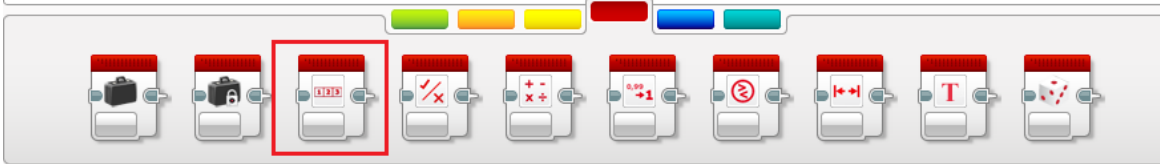


10.3 Pole

Možnosti práce s polem

Umístění

Blok Array Operations pro práci s polem je umístěn v červené záložce Data Operations (zobrazeno na obrázku). Použít jej můžeme přetažením na programovací plochu a umístěním na požadované místo v programu.



Režimy pro práci s polem

Připsání číselné hodnoty (Append- Numeric)

Režim Append s volbou Numeric slouží k přidání číselné položky na konec pole. Pomocí tohoto režimu je také možné vytvořit pole nové. Na vstupním portu Array In (vstup) se zadávají položky nově vytvářeného pole. Pokud uživatel nezadá nic, do pole se zapíše pouze nově zadávaná položka. Ta se zadává navstupním portu Value. Výstupní port Array Out (výstup pole) vrací výsledek celé operace (pole s přidanou položkou).



Připsání logické hodnoty (Append - Logic)

Princip fungování režimu je naprosto totožný s režimem Append - Numeric s tím rozdílem, že se nejedná o číselné pole, ale o pole logických hodnot.



Přečtení prvku pole (Read at Index - Numeric)

V tomto režimu vrací programový blok na výstupním portu Value hodnotu zvoleného prvku pole. Na vstupním portu Array In si zvolíme pole, ze kterého chceme číst, a na portu Index pořadové číslo (index) požadovaného prvku (první prvek pole má index 0).

Příklad: Pokud zadáme čtení indexu 2 pro pole [1,2,3], vráceným výsledkem na výstupním portu Value bude hodnota 3.



Přečtení prvku pole (Read at Index - Logic)

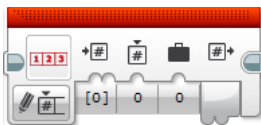
Režim pro čtení prvku pole logických hodnot je totožný s režimem pro čtení prvku číselného pole. Výslednou hodnotou je tedy hodnota logického datového typu.



Zápis prvku pole na pozici (Write at Index - Numeric)

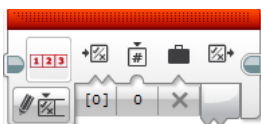
Režim umožňuje zapsat na zvolenou pozici číselného pole libovolnou číselnou hodnotu. Na vstupním portu Array In zvolíme pole, do kterého se má zapisovat. Na vstupním portu Index pozici, na kterou se má zvolená hodnota, kterou zadáme na vstupním portu Value, zapsat. Výsledkem operace vráceným na výstupním portu Array Out je pole s námi zapsaným prvkem.

Příklad: Máme definované pole hodnot [1,2,3,4,5]. Na pozici číslo 3 chceme zapsat číslo 9. Výsledkem operace bude pole [1,2,3,9,5].



Zápis prvku pole na pozici (Write at Index - Logic)

Režim pro zápis prvku pole logických hodnot obsahuje totožné volby jako předchozí režim pro číselná pole. Na zvolenou pozici se ovšem zapisuje hodnota logického datového typu.



Délka pole (Length - Numeric)

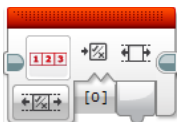
Režim Length vrací délku (počet prvků) zvoleného pole. Na vstupním portu Array In si zvolíme číselné pole. Výstupní port Length následně vrací délku zvoleného pole.

Příklad: Pokud chceme pomocí programového bloku pro práci s polem zjistit délku pole [1,2,3,4,5], výsledkem operace bude číslo 5 (pole obsahuje 5 prvků).



Délka pole (Length - Logic)

Režim Length pro pole logických hodnot umožňuje stejné možnosti jako režim pro zjišťování délky číselného pole.

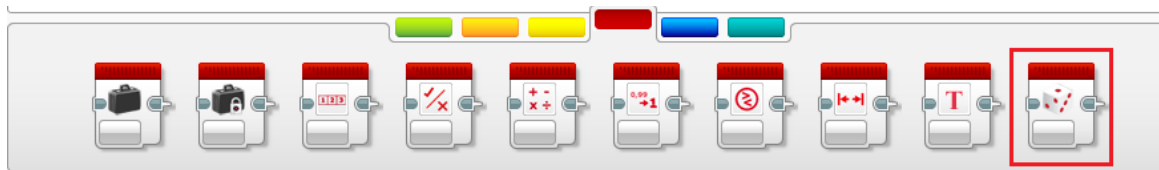


10.4 Generování náhodných čísel

Možnosti generování náhodných čísel v EV3

Umístění

Blok Random pro generování náhodných čísel je umístěn v červené záložce Data Operations (zobrazeno na obrázku). Použít jej můžeme přetažením na programovací plochu a umístěním na požadované místo v programu.



Režimy pro generování náhodných čísel

Generování čísel (Numeric)

Režim Numeric umožňuje generovat náhodné číslo ze zvoleného rozpětí hodnot. Na prvním vstupním portu Lower Bound (spodní hranice) zadáváme spodní hranici rozpětí hodnot, ze kterého se má číslo generovat. Na druhém vstupním portu Upper Bound (horní hranice) poté zadáváme horní hranici číselného rozpětí. Výstupní port Value následně vrací náhodně vygenerované číslo ze zvoleného rozpětí.



Generování logických hodnot (Logic)

Režim pro generování náhodných hodnot obsahuje pouze jeden vstupní port Probability of True (pravděpodobnost vygenerování hodnoty True). Ten nabývá hodnoty od 0 do 100 % a můžeme s jeho pomocí ovlivnit, jaká bude procentuálně vyjádřená pravděpodobnost, že náhodně vygenerovaná logická hodnota bude True. Výsledek generování poté vrací výstupní port Value.

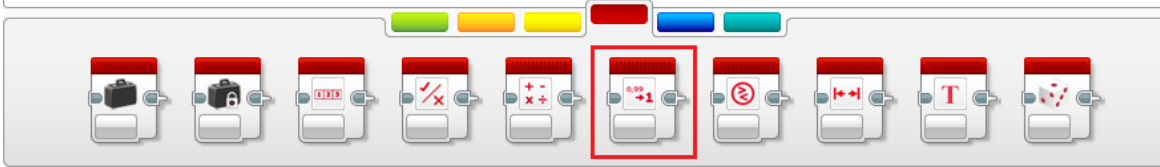


10.5 Další datové operace

Možnosti zaokrouhlování číselných hodnot

Umístění

Blok Round pro zaokrouhlování číselných hodnot je umístěn v červené záložce Data Operations (zobrazeno na obrázku). Použít jej můžeme přetažením na programovací plochu a umístěním na požadované místo v programu.

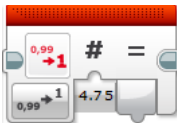


Režimy bloku pro zaokrouhlování

K nejbližšímu (To Nearest)

Režim umožňuje zaokrouhlovat číselnou hodnotu zadanou na vstupním portu Index k nejbližšímu celému číslu. Zaokrouhledenou hodnotu vrací výstupní port Result.

Příklad: Číslo 4,75 bude zaokrouhleno na 5.



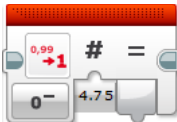
Nahoru (Round Up)

Režim umožňující zaokrouhlování číselných hodnot zadaných na vstupním portu Index směrem nahoru. Zaokrouhledenou hodnotu vrací výstupní port Result. *Příklad:* Číslo 4,75 bude zaokrouhleno na 5.



Dolů (Round Down)

Režim umožňující zaokrouhlovat číselnou hodnotou zadanou na vstupním portu Index směrem dolů. Zaokrouhledenou hodnotu vrací výstupní port Result. *Příklad:* Číslo 4,75 bude zaokrouhleno na 4.



Odříznutí desetinné části (Truncate)

Režim Truncate umožňuje oříznutí čísla na zvolený počet desetinných míst. Do pole Input zadáme desetinné číslo a u možnosti Number of Decimals zvolíme požadovaný počet desetinných míst. Výsledek operace vrací výstupní port Result.

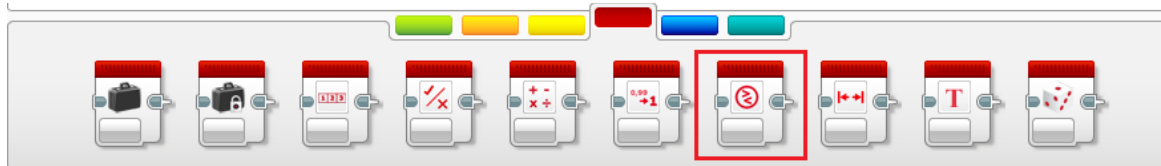
Příklad: Pokud zadáme číslo 4,75 a budeme požadovat jeho oříznutí na 1 desetinné místo, výsledkem bude číslo 4,7.



Možnosti porovnávání číselných hodnot

Umístění

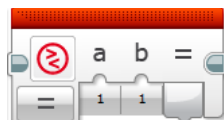
Blok Compare pro porovnávání číselných hodnot je umístěn v červené záložce Data Operations (zobrazeno na obrázku). Použit jej můžeme přetažením na programovací plochu a umístěním na požadované místo v programu.



Režimy bloku pro porovnávání číselných hodnot

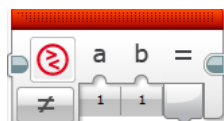
Rovná se (Equal To)

Blok v tomto režimu ověřuje, zda se čísla zadaná na vstupních portech A a B rovnají. Pokud se čísla rovnají, výstupní port Result vrací hodnotu True, v opačném případě False.



Nerovná se (Not Equal To)

V tomto režimu blok ověřuje, jestli jsou čísla zadaná na vstupních portech A a B rozdílná. Pokud ano, vrací výstupní port Result True, v opačném případě False.



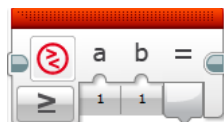
Větší než (Greater Than)

Blok v tomto režimu ověřuje, zda je číslo zadané na vstupním portu A větší než číslo na portu B. Pokud ano, vrací výstupní port Result hodnotu True, v opačném případě False.



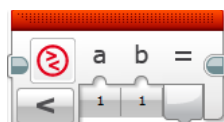
Větší než nebo rovno (Greater Than or Equal To)

Blok v tomto režimu porovnává, zda je číslo zadané na vstupním portu A větší nebo rovno číslu zadanému na portu B. V případě, že ano, vrací výstupní port logickou hodnotu True, v opačném případě False.



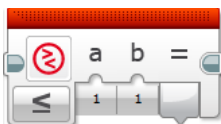
Menší než (Less Than)

Režim, ve kterém programový blok ověřuje, zda je hodnota zadaná na vstupním portu A menší než hodnota na vstupním portu B. Pokud ano, výstupní port Result vrací hodnotu True, v opačném případě False.



Menší než nebo rovno (Less Than or Equal To)

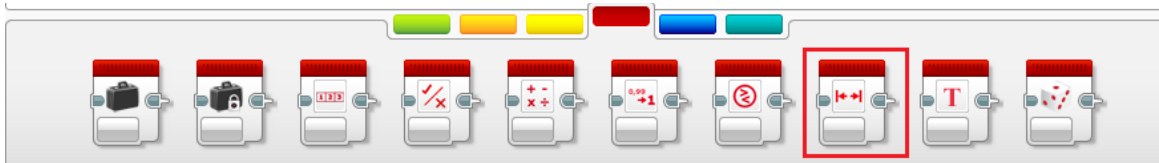
Režim, ve kterém programový blok ověřuje, zda je hodnota zadaná na vstupním portu A menší nebo rovna hodnotě B. Pokud ano, výstupní port Result vrací hodnotu True, v opačném případě False.



Možnosti určování rozmezí čísel

Umístění

Blok Range pro určování příslušnosti zadaného čísla do rozmezí hodnot je umístěn v červené záložce Data Operations (zobrazeno na obrázku). Použít jej můžeme přetažením na programovací plochu a umístěním na požadované místo v programu.



Režimy bloku pro určování příslušnosti zadaného čísla do rozmezí hodnot

Uvnitř rozmezí (Inside)

Blok v tomto režimu ověřuje, zda je hodnota zadaná na vstupním portu Test Value (testovaná hodnota) uvnitř zadaného rozsahu. Ten se určuje na zbylých dvou vstupních portech. Na portu Lower Bound (spodní hranice) zadáváme spodní hranici rozmezí a na vstupním portu Upper Bound (horní hranice) zase horní hranici rozmezí. Výsledek vrácený výstupním portem Result je logického datového typu. Výsledkem je True v případě, že testovaná hodnota je uvnitř zadaného rozsahu, v opačném případě je výsledek False.



Vně rozmezí (Outside)

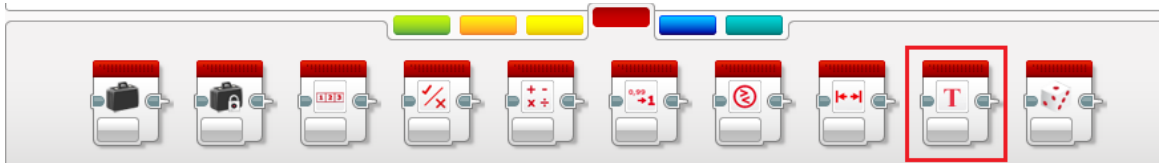
Funkce bloku v tomto režimu je totožná s předchozím režimem. Pouze ověřuje, zda je zadaná hodnota vně určeného rozmezí.



Možnosti spojování textových řetězců

Umístění

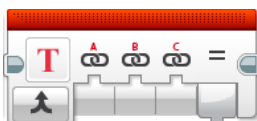
Blok Text pro slučování textových řetězců je umístěn v červené záložce Data Operations (zobrazeno na obrázku). Použít jej můžeme přetažením na programovací plochu a umístěním na požadované místo v programu.



Režimy bloku pro slučování textových řetězců

Spojení (Merge)

Programový blok slouží ke spojení až tří textových řetězců (A, B, a C) v jeden. Výsledkem vráceným výstupním portem Result jsou tyto zadané řetězce spojené v jeden.



10.6 Aktivita 13 - Stopky

Téma	Stopky	
Tematický celek	Programovací prostředí EV3	
Motivační rámec	Každý správný trenér potřebuje stopky. Využívají je trenéři vrcholových sportovců, závodníků, ale také například učitelé tělesné výchovy. Pomocí nich kontrolují výkony svých svěřenců a ověřují případné zlepšení v jejich sportovních výkonech. Jednoduché stopky si můžeme vytvořit také pomocí robotické stavebnice. Ty naše ovšem budou vhodné spíše pro motoristické sporty či měření běžeckých výkonů.	
Počet žáků	Skupina 8 - 10 studentů.	
Věk žáků	1. až 4. ročník SS a odpovídající ročníky gymnázií	
Pomůcky	Robotická stavebnice EV3 a počítače s nainstalovaným programovacím prostředím EV3.	
Stručný popis aktivity s využitím přístroje	Úkolem žáků je sestavit a naprogramovat jednoduchý model digitálních stopek. Ty budou měřit kolo závodníka, dokud nebude například stisknuto tlačítko dotykového senzoru. Následně uloží časový údaj do pole. Z uložených údajů bude následně vypočítáván průměrný čas na kolo, který bude vypisován na displej.	
Vhodné místo	Běžná učebna vybavená počítači s nainstalovaným programovacím prostředím EV3.	
Cíle aktivity	Žáci se naučí využívat pole.	
Rozvíjené kompetence	Kompetence k učení, k řešení problémů.	
Předchozí znalosti	Teoretická znalost matematických operací a práce s polem.	
Mezipředmětové vztahy	Informační a komunikační technologie (algoritmizace).	
Časový plán	Fáze činnosti s přístrojem	Metody a formy, motivace
45 minut	Tvorba programu a jeho průběžné	Spolupráce studentů ve skupinách, koordinace činnosti vyučujícím.
Hodnocení	Hodnocena bude průběžná práce na projektu a funkčnost vytvořeného programu.	

Zadání

Aktivita slouží k tomu, abyste se naučili využívat v programovacím prostředí EV3 pole. Za tímto účelem naprogramujete model stopek, které budou splňovat následující požadavky:

1. Zvolte si, čím budou stopky ovládány (např. tlačítko řídicí jednotky nebo tlačítko dotykového senzoru).
2. Stopky se spustí po spuštění programu a po každém stisknutí tlačítka bude zaznamenán čas aktuálně měřeného kola.
3. Ze zaznamenaných časů jednotlivých kol bude vypočítáván průměr, který bude vypisován na displej.

Výsledný program ke stažení ve formátu .C

Program ke stažení viz. on-line kurz

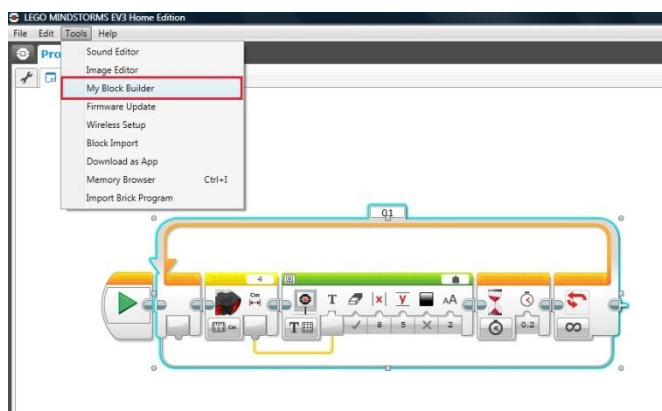
11 Tipy a triky

Vytváření vlastních metod

Vytváření vlastních metod je určeno pro situace, kdy se nám v programu určitý úsek kódu několikrát opakuje, nebo pro případy, kdy si chceme usnadnit práci a zdrojový kód zpřehlednit. Jeho princip spočívá ve vytvoření jediného programového bloku, který obsahuje sekvenci zdrojového kódu (několik bloků).

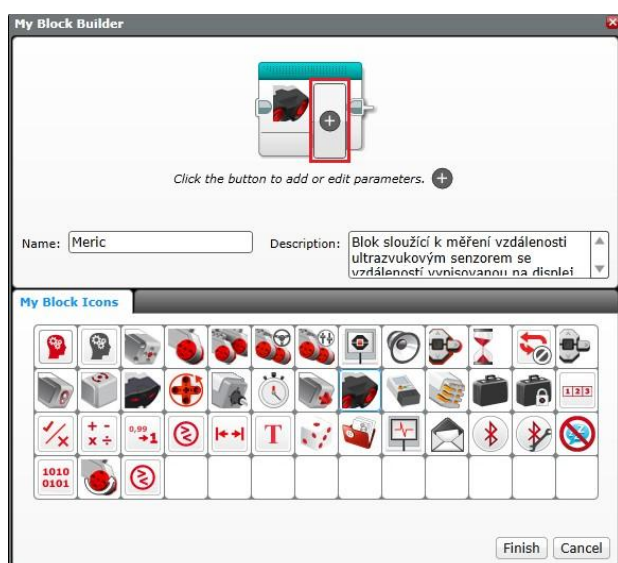
Umístění volby pro vytváření vlastních metod

Funkci pro vytvoření vlastních metod nalezneme v horním menu v záložce Tools (Nástroje) pod volbou My Block Builder (Tvůrce mých vlastních bloků). Dříve než začneme vlastní blok vytvářet, si musíme označit část zdrojového kódu, která bude v novém bloku obsažena.



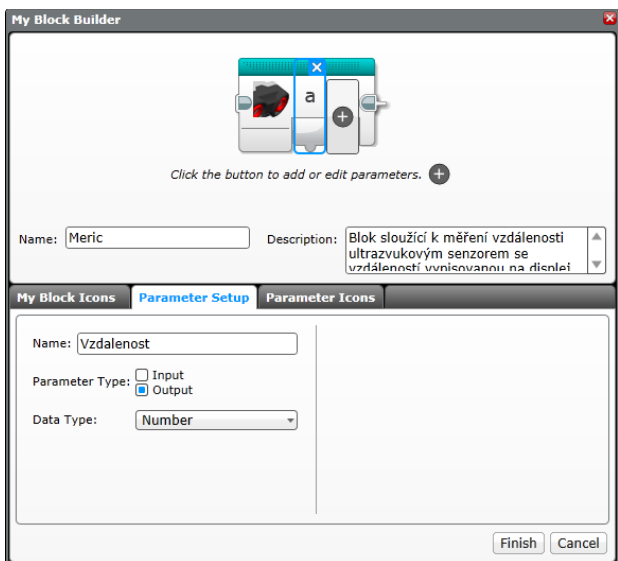
Úprava vzhledu bloku

V průvodci je nejprve nutné zadat unikátní název bloku a jeho popis, který nám v budoucnu usnadní orientaci v tom, co nový blok vykonává. Zvolit si můžeme také ikonu bloku z připravené palety obrázků. Následně můžeme přejít k nastavení vstupních a výstupních portů (pokud je chceme u bloku využít). Přidáme je pomocí tlačítka +, umístěného na náhledu bloku.

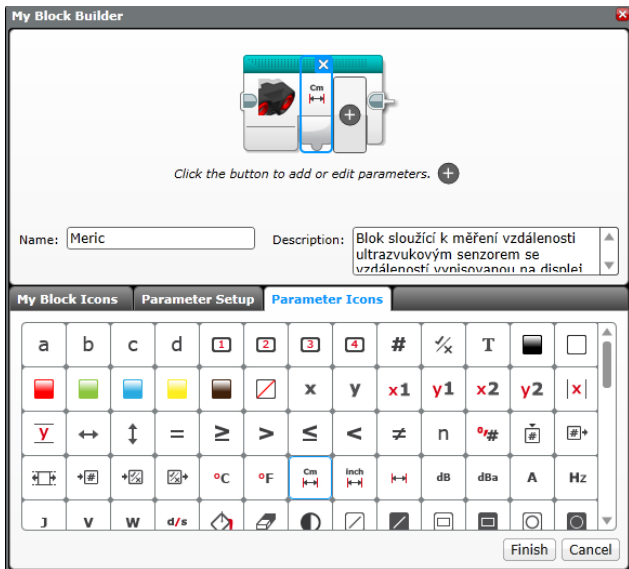


Nastavení vstupních a výstupních portů bloku

Po přidání libovolného počtu portů můžeme přejít k jejich nastavení. Zvolíme si název portu, jeho typ (vstupní nebo výstupní) a datový typ, kterého bude nabývat.

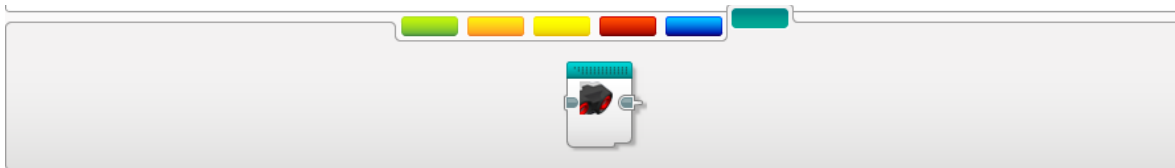


Na druhé záložce následně volíme z galerie ikonku, která se bude na portu zobrazovat.



Umístění nového bloku

Po kliknutí na tlačítko Finish se nový blok vytvoří a umístí se do záložky My Blocks (Mé bloky).



12 Závěrečné tipy

Doporučený multimediální materiál

Uživatelský manuál pro EV3 – odkaz viz. on-line kurz

Programovací prostředí RobotC

1 Základní informace o projektu

Název

Programovací prostředí RobotC

Anotace programu/zaměření/hlavní cíl

Programovací prostředí RobotC je programovací prostředí založené na syntaxi programovacího jazyka C. Je určeno pro vytváření programů pro ovládání širokého spektra robotických stavebnic. Jedná se například o VEX, RCX nebo Arduino. V této kapitole se seznámíte se samotným programovacím prostředím, jeho funkcemi a ovládním a základní funkci si procvičíte díky množství připravených aktivit.

Cílová skupina

1. až 4. ročník SŠ a odpovídající ročníky gymnázií

Organizační podmínky

Spolupráce studentů ve dvoučlenných, maximálně tříčlenných skupinách.

Pomůcky

Robotická stavebnice EV3, programovací prostředí RobotC.

Časová náročnost (popř. jak je možné program rozložit – jedná-li se o celoroční program)

Odhadovaná doba - zhruba 15 - 20 vyučovacích hodin.

Vazba na RVP

Rámcový vzdělávací program pro gymnázia.

Mezipředmětové vazby

Informatika (informační a komunikační technologie), elektrotechnika.

2 Motivační rámec projektu

Text:

Grafická programovací prostředí typu EV3 umožňují nezkušeným uživatelům nahlédnout do tajů programování s pomocí robotické stavebnice. Logickým uspořádáním a propojováním programových bloků zjednodušují první kroky a uvedení do problematiky pro nezkušené programátory.

Programovací prostředí RobotC od společnosti Robomatter, jehož syntaxe je založena na principu programovacího prostředí C, můžeme považovat za další krok na cestě k pochopení složitějších principů programování. Díky širším možnostem, které oproti grafickým programovacím prostředím nabízí, je možné vytvářet složitější programové konstrukty. Podobnost se syntaxí programovacího jazyka C může být také jedním z prvních kroků k pochopení tohoto jazyka. Prostředí RobotC není vyvíjeno pouze pro platformu LEGO Mindstorms EV3, ale i jiné (např. VEX, RCX nebo Arduino).

Doporučený multimediální materiál

Video viz. on-line kurz

Oficiální stránky výrobce programovacího prostředí: [vice zde](#) (odkaz viz. on-line kurz)

On-line dostupná uživatelská příručka pro RobotC: [vice zde](#) (odkaz viz. on-line kurz)

3 Poznámky k využití přístrojů

Pro tvorbu úvodu do práce s programovacím prostředím RobotC a sestavení úloh byla použita základní sada stavebnice EV3. Další informace o ní naleznete na následujících odkazech:

Informace na oficiálních internetových stránkách výrobce ([odkaz viz. on-line kurz](#))

On-line uživatelský manuál pro programovací prostředí RobotC: ([odkaz viz. on-line kurz](#))

4 Projektový deník

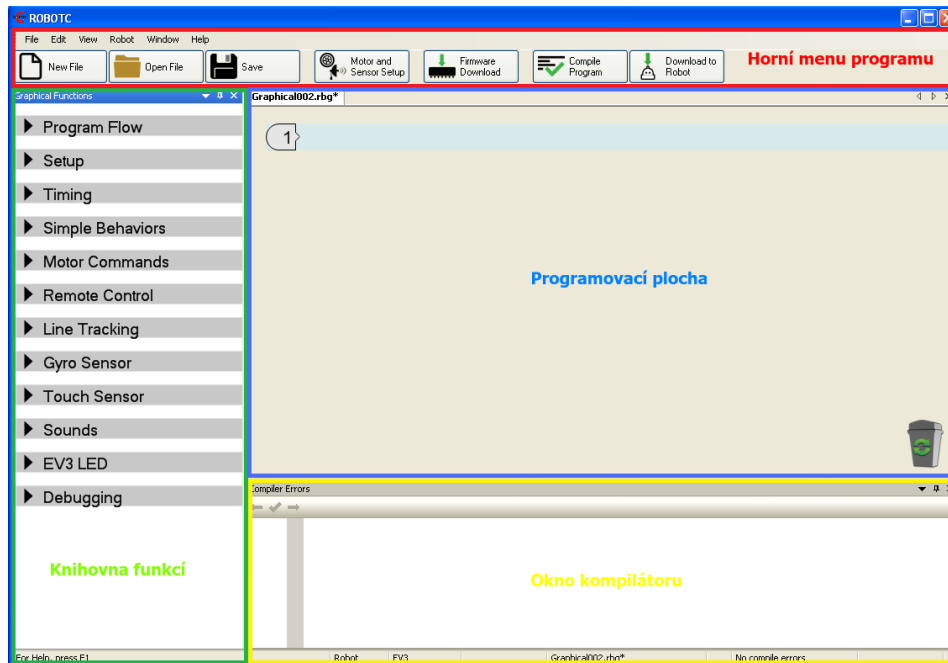
Projektový deník slouží k evidenci postupu žáků ve studiu kapitoly. Tabulky jsou uzpůsobeny tak, aby si žák měl možnost poznamenat problémy, které musel při řešení programové úlohy řešit, a co mu činilo u příkladu největší obtíže. Po kontrole zápisu a příslušného programu, případně modelu, vyučujícím je do tabulky zapsáno hodnocení splnil, nebo nesplnil. V případě potřeby může být doplněna také známka.

Projektový deník ke stažení (ve formátu PDF) v on-line kurzu. Také je přílohou této tiskové opory.

5 Grafické programovací prostředí RobotC

Aktuální verze programovacího prostředí RobotC (4.X) neumožňuje pouze vytvářet programový kód zápisem programovacího jazyka založeného na jazyce C, jako tomu bylo dříve, aktuálně obsahuje také grafický režim zápisu programového kódu.

Popis prostředí



Na ilustračním obrázku můžete vidět náhled grafické verze programovacího prostředí RobotC. Můžeme si jej rozdělit do čtyř základních částí:

Horní menu programu

Horní část programovacího prostředí (označena červeně) obsahuje standardní menu pro práci se souborem, editace a různé nastavení prostředí. Ve spodní části menu je umístěno několik pohotovostních tlačítek pro usnadnění vytváření programu. Jedná se o často využívané funkce při tvorbě programu.

New file - vytvoření nového programu.

Open file - otevření existujícího souboru.

Save - uložení programu.

Motor and Sensor Setup - deklarace motorů a senzorů.

Firmware Download - nahrání firmware.

Compile Program - kompilace programu.

Download to Robot - nahrání programu do řídicí jednotky EV3.

Knihovna funkcí

V levé části prostředí (označeno zeleně) se nachází knihovna programových funkcí, které je možné drag and drop způsobem přetáhnout na programovací plochu.

Programovací plocha

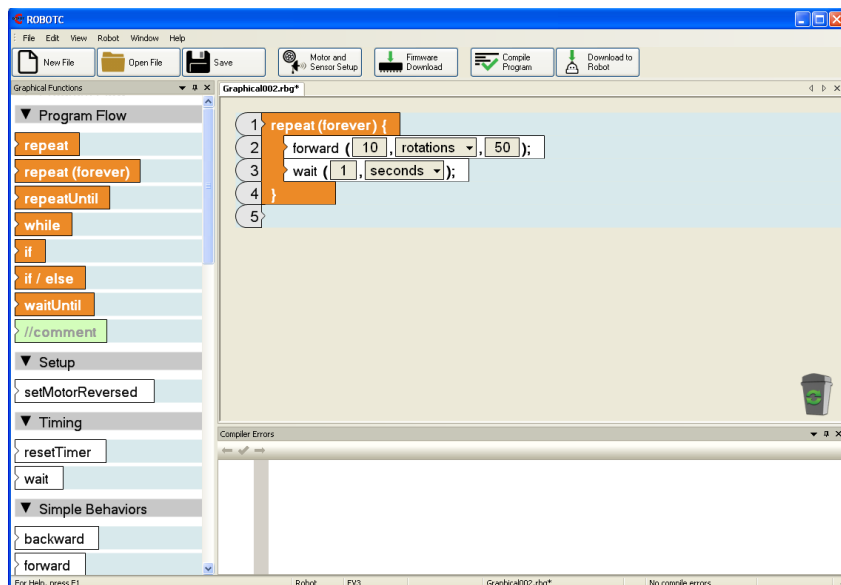
Největší část okna programovacího prostředí určená pro tvorbu programu pomocí skládání a vnořování jednotlivých grafických příkazů a funkcí.

Okno kompilátoru

Okno, ve kterém se zobrazují chyby a upozornění po kompilaci programu.

6 Základy tvorby programu

Grafická verze programovacího prostředí RobotC obsahuje v levé části knihovnu příkazů a funkcí, které je možné přetáhnout a umístit na programovací plochu. Na ilustračním obrázku můžete vidět příklad zápisu jednoduchého nekonečného cyklu. Jednotlivé funkce obsahují rozbalovací nabídky, ze kterých je možné volit parametry nebo měnit vstupní hodnoty.



Nevýhody grafické verze programovacího prostředí RobotC

Grafická verze programovacího prostředí RobotC neobsahuje všechny funkce, které obsahuje verze s textovým zápisem programového kódu. Zcela například chybí možnost jakéhokoliv výpisu na displej. Dále je také citelná absence vytváření proměnných a konstant.

Využitelnost prostředí

Programovací prostředí se dá využít pro základní výuku algoritmizace. Žáci se s jeho pomocí mohou naučit vytvářet jednoduché postupy pro řešení jednoduchých problémů. V následujících kapitolách naleznete náměty na aktivity pro grafickou verzi programovacího prostředí. Jedná se o vybrané aktivity, se kterými se setkáte v části věnované programovacímu prostředí EV3. Můžete tak porovnat jejich zápis v jiném programovacím prostředí.

6.1 Aktivita 1 - Ruční mixér

Zadání

Téma	Ruční mixér s použitím dotykového senzoru	
Tematický celek	Programovací prostředí RobotC (grafická verze prostředí)	
Motivační rámec	Dotykový senzor se hodí pro několik možností využití. Typicky se využívá jako tlačítko. Jeho velmi užitečnou vlastností je možnost rozeznávat tři stavy stisku. Tato aktivita se věnuje jeho použití jako spouštěče určité činnosti jiného zařízení.	
Počet žáků	Skupina 8 - 10 studentů.	
Věk žáků	1. až 4. ročník SŠ a odpovídající ročníky gymnázií	
Pomůcky	Robotická stavebnice EV3 a počítače s nainstalovaným programovacím prostředím RobotC.	
Stručný popis aktivity s využitím přístroje	Úkolem této aktivity je seznámit studenty s možnostmi použití dotykového senzoru. Ten je v úkolu použit jako tlačítko ručního mixéru. Mixér je poháněn pomocí středního motoru. Při ovládní rozlišuje tři stavy. Po prvním stisku a uvolnění se roztočí pomaleji, po dalším stisku maximální rychlostí a při třetím stisku se vypne.	
Vhodné místo	Běžná učebna vybavená počítači s nainstalovaným programovacím prostředím RobotC.	
Cíle aktivity	Studenti se naučí využívat dotykový senzor a rozeznávat jeho možné stavy.	
Rozvíjené kompetence	Kompetence k učení, k řešení problémů.	
Předchozí znalosti	Základní ovládání programovacího prostředí RobotC, základy algoritmizace.	
Mezipředmětové vztahy	Informační a komunikační technologie (algoritmizace).	
Časový plán	Fáze činnosti s přístrojem	Metody a formy, motivace
10 minut	Stavba modelu.	Spolupráce studentů ve skupinách, koordinace činnosti
15 minut	Tvorba programu a průběžné	Spolupráce studentů ve skupinách, koordinace činnosti
5 minut	Závěrečné testování funkčnosti.	Spolupráce studentů ve skupinách, koordinace činnosti
Hodnocení	Hodnocena bude průběžná práce na úloze, spolupráce studentů ve skupině a kvalita výsledného modelu a programu.	

Aktivita si klade za cíl vás seznámit s funkcemi dotykového senzoru. V následující úloze se naučíte využít dotykový senzor jako spouštěč či přepínač určité reakce. Váš úkol je následující:

1. Postavte vhodný model ručního mixéru, který bude ovládán pomocí tlačítka dotykového senzoru.
2. Po prvním stisknutí se mixér roztočí pomaleji, při druhém zrychlí na maximum a při třetím zastaví.

Řešení pro grafickou verzi programovacího prostředí RobotC

Řešení v grafické verzi RobotC se liší od textového zápisu. Nesmíte zapomenout na správnou deklaraci motorů a senzoru tak, aby byla zajištěna funkčnost programu.

Neustálé provádění programu zaručíme cyklem repeat (forever). Následně v programu ověřujeme počet stisknutí tlačítek. Na základě zjištěné hodnoty je ovlivňována rychlost otáčení motoru, který pohání ruční mixér. V úloze si procvičíte práci s cykly, podmínkami, vyhodnocování hodnoty zjištěné senzorem a také nastavení rychlosti otáčení motoru.

```

1 resetBumpedValue (touchSensor);
2 repeat (forever) {
3   if (getBumpedValue(touchSensor) == 1) {
4     setMotor (motorA, 50);
5   }
6   if (getBumpedValue(touchSensor) == 2) {
7     setMotor (motorA, 100);
8   }
9   if (getBumpedValue(touchSensor) == 3) {
10    resetBumpedValue (touchSensor);
11    setMotor (motorA, 0);
12  }
13 }
14

```

Vysvětlivky ke zdrojovému kódu

resetBumpedValue () - nulování čítače stisků tlačítka dotykového senzoru

repeat (forever) - cyklus s nekonečným počtem vykonávání

if () - podmínka

getBumpedValue() - detekce počtu stisknutí tlačítka dotykového senzoru

setMotor () - ovládání rychlosti otáčení motoru

Výsledný program ke stažení ve formátu .RBG

Program ke stažení viz. on-line kurz

6.2 Aktivita 2 - Výstražný semafor se zvukovou signalizací

Zadání

Téma	Výstražný semafor se zvukovou signalizací	
Tematický celek	Programovací prostředí RobotC	
Motivační rámec	Signalizace železničního přejezdu upozorňuje řidiče, že se k přejezdu blíží vlak, a je tudíž životu nebezpečné do něj vjíždět. Podoba železniční signalizace je různá. V této úloze si vytvoříte program, který bude simulovat funkci světelné signalizace na železničním či jiném přejezdu. Přidanou hodnotou bude zvuková signalizace upozorňující na blížící se nebezpečí.	
Počet žáků	Skupina 8 - 10 studentů.	
Věk žáků	1. až 4. ročník SŠ a odpovídající ročníky gymnázií	
Pomůcky	Robotická stavebnice EV3 a počítače s nainstalovaným programovacím prostředím EV3.	
Stručný popis aktivity s využitím přístroje	Úkolem žáků je využít zvukovou a světelnou signalizaci řídicí jednotky jako signalizátor blížícího se nebezpečí na železničním přejezdu. Detektorem blížícího se vlaku se stane ultrazvukový senzor, který pomocí naměřené vzdálenosti spustí světelnou a zvukovou signalizaci.	
Vhodné místo	Běžná učebna vybavená počítači s nainstalovaným programovacím prostředím RobotC.	
Cíle aktivity	Žáci se naučí využívat světelnou a zvukovou signalizaci řídicí jednotky.	
Rozvíjené kompetence	Kompetence k učení, k řešení problémů.	
Předchozí znalosti	Znalost práce se světelnou a zvukovou signalizací.	
Mezipředmětové vztahy	Informační a komunikační technologie (algoritmizace úloh).	
Časový plán	Fáze činnosti s přístrojem	Metody a formy, motivace
40 minut	Tvorba programu a jeho průběžné	Spolupráce studentů ve skupinách, koordinace činnosti
Hodnocení	Hodnocena bude úplnost a přehlednost vyplněných informací a kvalita doplňujících multimediálních prvků.	

Zadání

Využijte řídicí jednotku k vytvoření programu, který bude simulovat signalizaci blížícího se vlaku na železničním přejezdu. Požadavky na tvorbu budou následující:

1. Použijte ultrazvukový senzor pro detekci blížícího se vlaku.
2. Při naměření kritické vzdálenosti se spustí přerušovaná světelná a také zvuková signalizace.
3. Jakmile se pomyslný vlak vzdálí opět za kritickou vzdálenost, signalizace se vypne.

Řešení pro grafickou verzi programovacího prostředí RobotC

Na obrázku můžete vidět zápis programového kódu úlohy v grafickém zápisu programovacího prostředí RobotC. Celý program se vykonává neustále dokola, proto je použit cyklus repeat(forever). Dále je v úloze použita výstražná zvuková signalizace a také podsvícení tlačítek řídicí jednotky. Chování výstrahy je řízeno pomocí vzdálenosti snímané ultrazvukovým senzorem.

```

1 repeat (forever) {
2   wait ( 1 , seconds );
3   while ( getUSDistance(sonarSensor) <= 20 ) {
4     playSound ( soundBeepBeep );
5     setLEDColor ( ledRedPulse );
6     wait ( 1 , seconds );
7   }
8   setLEDColor ( ledOrange );
9 }
10

```

Vysvětlivky ke zdrojovému kódu

repeat (forever) - cyklus s nekonečným počtem vykonávání

while () - cyklus s podmínkou na začátku

wait () - oddálení vykonávání příkazu

getUSDistance () - funkce pro získání hodnoty snímané ultrazvukovým senzorem

playSound () - příkaz pro přehrání zvuku

setLEDColor () - nastavení barvy, kterou bude svítit čelní panel řídicí jednotky

Výsledný program ke stažení ve formátu .RBG

Program ke stažení viz. on-line kurz

6.3 Aktivita 3 - Detektor pádu pevného disku

Téma	Detektor pádu disku	
Tematický celek	Programovací prostředí RobotC	
Motivační rámec	Nenadálé problémy se spuštěním počítače mohou někdy pramenit z poruchy pevného disku počítače. Stát se tak může i v jiných případech. Například při pádu spuštěného notebooku na zem. Ať již je příčina jakákoliv, ve velké většině případů nastane nepříjemná ztráta dat. Pevné disky ovšem obsahují zařízení, které při detekci pádu dokáže bezpečně zaparkovat zápisovou hlavu tak, aby nedošlo k poškození disku. My si takové zařízení zkusíme vytvořit díky robotické stavebnici za pomoci gyroskopického senzoru.	
Počet žáků	Skupina 8 - 10 studentů.	
Věk žáků	1. až 4. ročník SŠ a odpovídající ročníky gymnázií	
Pomůcky	Robotická stavebnice EV3 a počítače s nainstalovaným programovacím prostředím RobotC.	
Stručný popis aktivity s využitím přístroje	Žáci si nejprve sestaví jednoduchý model pevného disku. Uvnitř se pokusí vytvořit pohyblivý mechanismus, který bude simulovat jeho funkci (např. motor nebo pohyblivé rameno). Zařízení pro detekci pádu bude představovat gyroskopický senzor. Následně vytvoří program, který při prudké změně polohy zastaví chod disku.	
Vhodné místo	Běžná učebna vybavená počítači s nainstalovaným programovacím prostředím RobotC.	
Cíle aktivity	Žáci se naučí využívat gyroskopický senzor.	
Rozvíjené kompetence	Kompetence k učení, k řešení problémů.	
Předchozí znalosti	Znalost práce s barevným senzorem.	
Mezipředmětové vztahy	Informační a komunikační technologie (algoritmizace úloh).	
Časový plán	Fáze činnosti s přístrojem	Metody a formy, motivace
15 minut	Tvorba modelu disku.	Spolupráce studentů ve skupinách, koordinace činnosti vyučujícím.
20 minut	Tvorba programu a jeho průběžné testování.	Spolupráce studentů ve skupinách, koordinace činnosti vyučujícím.
15 minut	Testování funkčnosti modelu a programu.	Spolupráce studentů ve skupinách, koordinace činnosti vyučujícím.
Hodnocení	Hodnocena bude úplnost a přehlednost vyplněných informací a kvalita doplňujících multimediálních prvků.	

Zadání

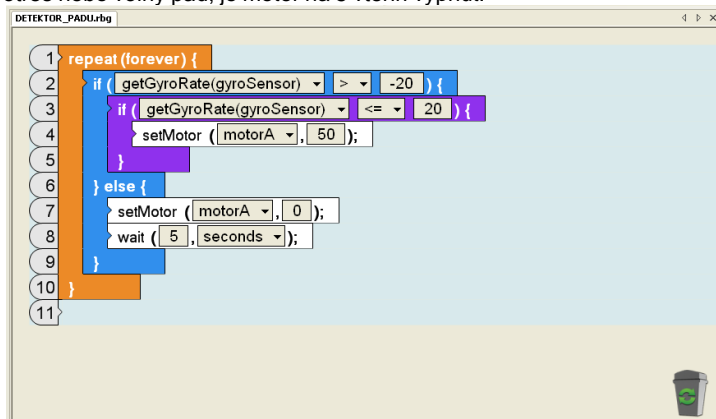
Vytvořte funkční model pevného disku se zařízením pro detekci pádu. Požadavky na konstrukci modelu a program budou následující:

1. Sestavte jednoduchý model pevného disku.
2. Uvnitř bude umístěn pohyblivý mechanismus poháněný například motorem.
3. Pro detekci pádu použijte gyroskopický senzor.
4. Jakmile bude detekována prudká změna polohy, chod disku se zastaví.

Řešení pro grafickou verzi programovacího prostředí RobotC

Obrázek ilustruje zápis grafického programového kódu programu pro ovládání detektoru pádu disku.

Pro nekonečné provádění programu je použit nekonečný cyklus repeat(forever). Následně je detekována poloha disku. Při pouze drobném chvění gyroskopického senzoru pokračuje disk v chodu. Pokud je detekován prudký otřes nebo volný pád, je motor na 5 vteřin vypnut.



Vysvětlivky ke zdrojovému kódu

repeat (forever) - cyklus s nekonečným počtem vykonávání

if () - podmínka

getGyroRate () - zjištění hodnoty snímané senzorem

setMotor () - nastavení rychlosti otáčení motoru

wait () - oddálení vykonávání příkazu

Výsledný program ke stažení ve formátu .RBG

[Program ke stažení](#) viz. on-line kurz

7 Textově orientované programovací prostředí

Charakteristika prostředí:

- program vytvářen textovým zápisem programovacího jazyka,
- syntaxe jazyka založena na bázi programovacího jazyka C,
- tvůrce prostředí - Robomatter, Inc.

video viz. on-line kurz

Popis jednotlivých částí RobotC

Horní menu programu:

- nalezneme jej v horní části programovacího prostředí,
- obsahuje základní volby pro práci se souborem (vytvoření nového programu, otevření nebo uložení),
- zobrazení různých plovoucích panelů nástrojů,
- nastavení využívané platformy, aktualizace firmware, konfigurace motorů a senzorů.

Akční ikony:

New File - vytvoření nového souboru,
Open File - otevření existujícího programu, *Save* - uložení vytvářeného programu,
Fix Formatting - naformátuje ve zvoleném souboru veškeré odsazení a závorky,
Motor and Sensor Setup - volba sloužící k nastavení motorů a senzorů (přiřazení na porty, pojmenování, volba režimu), *Firmware Download* - nahrání firmware do řídicí jednotky,
Compile Program - kompilace programu (bez nahrání programu do řídicí jednotky), *Download to Robot* - kompilace programu s následným nahráním do paměti řídicí jednotky,
Start - spustí provádění programu (zobrazí se až po nahrání programu do paměti řídicí jednotky),
Stop - zastaví provádění spuštěného programu (zobrazí se až po nahrání programu do paměti řídicí jednotky).

Knihovna funkcí:

- umístěna vlevo,
- obsahuje stromovou strukturu všech kategorií, kde jsou umístěny využitelné funkce a příkazy.

Programovací plocha:

- prostor pro zápis programu,
- pro snazší orientaci v programu je vlevo umístěno číslování jednotlivých řádků programu.

Okno kompilátoru:

- na tomto místě se zobrazují chyby v programu zjištěné při kompilaci.

Poznámka

V následujících podkapitolách naleznete návrhy aktivit na procvičení práce s moduly stavebnice LEGO Mindstorms EV3 v programovacím prostředí RobotC. Úlohy jsou ve většině totožné s úlohami obsaženými v kapitole o programovacím prostředí EV3. Díky tomu si můžete procvičit tvorbu paralelně jak v grafickém, tak i textovém programovacím prostředí.

7.1 Aktivita 1 - Lopatková turbína

Téma	Lopatková turbína	
Tematický celek	Programovací prostředí RobotC	
Motivační rámec	Trendem dnešní doby je využívání obnovitelných zdrojů energie. Zásoba přírodních zdrojů není nevyčerpatelná, a tak musíme stále hledat vhodné alternativy. Během slunečných dnů můžeme využívat například solární panely. Jak ovšem vyrábět elektřinu ve stejném objemu i v období, kdy není světla dostatek nebo v noci? V této úloze si sestrojíte jednoduchou turbínu fungující jako záložní způsob výroby energie v situacích, kdy je světla nedostatek.	
Počet žáků	Skupina 8 - 10 studentů.	
Věk žáků	1. až 4. ročník SŠ a odpovídající ročníky gymnázií	
Pomůcky	Robotická stavebnice EV3 a počítače s nainstalovaným programovacím prostředím RobotC.	
Stručný popis aktivity s využitím přístroje	Žáci si nejprve vytvoří jednoduchý model lopatkové turbíny. Ta bude poháněna pomocí střední motoru. Její chod bude řízen barevným senzorem, který pomocí detekce okolního světla bude řídit rychlost otáčení.	
Vhodné místo	Běžná učebna vybavená počítači s nainstalovaným programovacím prostředím RobotC.	
Cíle aktivity	Žáci se naučí využívat střední motoru k pohonu rotačních součástí a řídit rychlost otáčení v závislosti na jiném zařízení.	
Rozvíjené kompetence	Kompetence k učení, k řešení problémů.	
Předchozí znalosti	Znalost práce s barevným senzorem.	
Mezipředmětové vztahy	Fyzika (světlo), Informační a komunikační technologie (algoritmizace úloh).	
Časový plán	Fáze činnosti s přístrojem	Metody a formy, motivace
15 minut	Tvorba modelu robota.	Spolupráce studentů ve skupinách, koordinace činnosti vyučujícím.
20 minut	Tvorba programu a jeho průběžné testování.	Spolupráce studentů ve skupinách, koordinace činnosti vyučujícím.
20 minut	Testování funkčnosti modelu a programu.	Spolupráce studentů ve skupinách, koordinace činnosti vyučujícím.
Hodnocení	Hodnocena bude úplnost a přehlednost vyplněných informací a kvalita doplňujících multimediálních prvků.	

Zadání

Vytvořte funkční model turbíny jako záložního zdroje výroby elektrické energie. Požadavky na konstrukci modelu a program budou následující:

- Lopatky turbíny budou poháněny středním motorem.
- Řízení rychlosti otáčení bude prováděno díky barevnému senzoru.
 - Čím nižší bude intenzita okolního světla, tím se bude turbína otáčet rychleji.
- V programu vhodně nastavte prahovou hodnotu dostatečného slunečního svitu, při kterém se turbína zastaví.

Výsledný program ke stažení ve formátu .C

Program ke stažení viz. on-line kurz

7.2 Aktivita 2 - Kuchyňská minutka

Téma	Kuchyňská minutka	
Tematický celek	Programovací prostředí RobotC	
Motivační rámec	Pravděpodobně každý z nás se někdy v kuchyni setkal s kuchyňskou minutkou. Malým zařízením sloužícím k měření času při přípravě pokrmů. Využívaly ji naše babičky i maminky. Doba pokročila a z klasických manuálních se vyvinuly i pokročilejší digitální, které jsou často opatřené displejem. Vše potřebné k tomu, abychom podobnou minutku vytvořili z robotické stavebnice, ovšem máme k dispozici i my. Postačí nám k tomu displej řídicí jednotky a otočný mechanismus zajistí servomotor.	
Počet žáků	Skupina 8 - 10 studentů.	
Věk žáků	1. až 4. ročník SŠ a odpovídající ročníky gymnázií	
Pomůcky	Robotická stavebnice EV3 a počítače s nainstalovaným programovacím prostředím RobotC.	
Stručný popis aktivity s využitím přístroje	Žáci si nejprve vytvoří vhodnou konstrukci kuchyňské minutky sestávající ze servomotoru opatřeného dobře ovladatelným otáčecím mechanismem pro zajištění pohodlného nastavení času měření. Odpočítávání začne ve chvíli, kdy bude stisknuto tlačítko připevněného dotykového senzoru.	
Vhodné místo	Běžná učebna vybavená počítači s nainstalovaným programovacím prostředím RobotC.	
Cíle aktivity	Žáci se naučí využívat servomotor k ovládání pohyblivých součástí a pracovat s měřením a ovlivňováním vykonaného počtu otáček.	
Rozvíjené kompetence	Kompetence k učení, k řešení problémů.	
Předchozí znalosti	Znalost práce s dotykovým senzorem.	
Mezipředmětové vztahy	Informační a komunikační technologie (algoritmizace úloh).	
Časový plán	Fáze činnosti s přístrojem	Metody a formy, motivace
15 minut	Tvorba modelu robota.	Spolupráce studentů ve skupinách, koordinace činnosti vyučujícím.
30 minut	Tvorba programu a jeho průběžné testování.	Spolupráce studentů ve skupinách, koordinace činnosti vyučujícím.
10 minut	Testování funkčnosti modelu a programu.	Spolupráce studentů ve skupinách, koordinace činnosti vyučujícím.
Hodnocení	Hodnocena bude úplnost a přehlednost vyplněných informací a kvalita doplňujících multimediálních prvků.	

Zadání

Vytvořte funkční model kuchyňské minutky. Požadavky na konstrukci modelu a program budou následující:

1. Otočný mechanismus minutky bude realizován pomocí servomotoru.
2. Natočením mechanismu se díky detekci počtu otáček vyhodnotí, jak dlouhý časový úsek se má měřit.
3. Nastavovaný čas by se měl zobrazit na displeji.
4. Při měření času by se měla minutka otáčet a konec odpočítávání by měl oznámit zvukový signál.

Výsledný program ke stažení ve formátu .C

Program ke stažení viz. on-line kurz

7.3 Aktivita 3 - Řízení pásového transportéru

Téma	Řízení pásového transportéru	
Tematický celek	Programovací prostředí RobotC	
Motivační rámec	V těžko dostupných terénech na nestabilním podkladu často kolová vozidla nezvládnou bez problému průchod terénem. Je potřeba proto využít vozidla pásová. Využívají se v armádě, ale i mezi civilními občany. Typickým příkladem je sněžná rolba nebo pásový transportér. V této aktivitě si podobné vozidlo sestavíme a hlavně se je naučíme ovládat.	
Počet žáků	Skupina 8 - 10 studentů.	
Věk žáků	1. až 4. ročník SŠ a odpovídající ročníky gymnázií	
Pomůcky	Robotická stavebnice EV3 a počítače s nainstalovaným programovacím prostředím RobotC.	
Stručný popis aktivity s využitím přístroje	Žáci si nejprve vytvoří model pojízdného robota opatřeného pásy. Robot bude opatřen dvěma dotykovými senzory. V případě, že budou stisknuta tlačítka obou senzorů současně, robot se bude pohybovat vpřed. Při stisku pouze jediného z tlačítek se bude robot pohybovat na zvolenou stranu.	
Vhodné místo	Běžná učebna vybavená počítači s nainstalovaným programovacím prostředím RobotC.	
Cíle aktivity	Žáci se naučí využívat alternativní způsob pohonu servomotorů (pomocí pásů).	
Rozvíjené kompetence	Kompetence k učení, k řešení problémů.	
Předchozí znalosti	Znalost možností řízení motorů, práce s dotykovým senzorem a znalost práce s cykly a podmínkami.	
Mezipředmětové vztahy	Informační a komunikační technologie (algoritmizace úloh).	
Časový plán	Fáze činnosti s přístrojem	Metody a formy, motivace
20 minut	Tvorba modelu robota.	Spolupráce studentů ve skupinách, koordinace činnosti vyučujícím.
20 minut	Tvorba programu a jeho průběžné testování.	Spolupráce studentů ve skupinách, koordinace činnosti vyučujícím.
5 minut	Testování funkčnosti modelu a programu.	Spolupráce studentů ve skupinách, koordinace činnosti vyučujícím.
Hodnocení	Hodnocen bude sestavený model robota a úplnost a funkčnost vytvářeného programu.	

Zadání

Vytvořte funkční model robota poháněného pásy. Požadavky na konstrukci modelu a program budou následující: robot musí být dobře

manévrovatelný a pásy na modelu dobře upevněny, natáčení robota do stran bude prováděno dvěma dotykovými senzory plnicími funkcí joysticku, volte vhodnou rychlost otáčení motorů pro dobrou pohyblivost modelu.

Výsledný program ke stažení ve formátu .C

Program ke stažení viz. on-line kurz

7.4 Aktivita 4 - Železniční přejezd

Téma	Železniční přejezd	
Tematický celek	Programovací prostředí RobotC	
Motivační rámec	Signalizace železničního přejezdu upozorňuje řidiče, že se k přejezdu blíží vlak, a je tudíž životu nebezpečné do něj vjíždět. Podoba železniční signalizace je různá. V této úloze si vytvoříte program, který bude simulovat funkci světelné signalizace na železničním přejezdu. Přidanou hodnotou bude zvuková signalizace upozorňující na blížící se nebezpečí.	
Počet žáků	Skupina 8 - 10 studentů.	
Věk žáků	1. až 4. ročník SŠ a odpovídající ročníky gymnázií	
Pomůcky	Robotická stavebnice EV3 a počítače s nainstalovaným programovacím prostředím RobotC (textová	
Stručný popis aktivity s využitím přístroje	Úkolem žáků je využít zvukovou a světelnou signalizaci řídicí jednotky jako signalizátor blížícího se nebezpečí na železničním přejezdu. Detektorem blížícího se vlaku se stane ultrazvukový senzor, který pomocí naměřené vzdálenosti spustí světelnou a zvukovou signalizaci.	
Vhodné místo	Běžná učebna vybavená počítači s nainstalovaným programovacím prostředím RobotC.	
Cíle aktivity	Žáci se naučí využívat světelnou a zvukovou signalizaci řídicí jednotky.	
Rozvíjené kompetence	Kompetence k učení, k řešení problémů.	
Předchozí znalosti	Znalost práce se světelnou a zvukovou signalizací.	
Mezipředmětové vztahy	Informační a komunikační technologie (algoritmizace úloh).	
Časový plán	Fáze činnosti s přístrojem	Metody a formy, motivace
40 minut	Tvorba programu a jeho průběžné testování.	Spolupráce studentů ve skupinách, koordinace činnosti vyučujícím.
Hodnocení	Hodnocena bude úplnost a přehlednost vyplněných informací a kvalita doplňujících multimediálních prvků.	

Zadání

Využijte řídicí jednotku k vytvoření programu, který bude simulovat signalizaci blížícího se vlaku na železničním přejezdu. Požadavky na tvorbu budou následující:

1. Použijte ultrazvukový senzor pro detekci blížícího se vlaku.
2. Při naměření kritické vzdálenosti se spustí přerušovaná světelná a také zvuková signalizace.
3. Jakmile se pomyslný vlak vzdálí opět za kritickou vzdálenost, signalizace se vypne.

Výsledný program ke stažení ve formátu .C

Program ke stažení viz. on-line kurz

7.5 Aktivita 5 - Kreslicí tabulka

Téma	Kreslicí tabulka	
Tematický celek	Programovací prostředí RobotC	
Motivační rámec	Mnoho z nás si z dětství pamatuje ruční kreslicí tabulku s displejem a dvěma ovládacími kolečky pro kreslení. Jedno kolečko pro kreslení v horizontálním a druhé ve vertikálním směru. Jelikož tyto hračky nahradily jiné, mnohem propracovanější, setkáme se s takovou tabulkou jen zřídka. Není pro nás ovšem žádný problém si takovou tabulku pomocí robotické stavebnice vyrobit přímo pomocí řídicí jednotky. Jako ovládací prvky nám postačí její tlačítka.	
Počet žáků	Skupina 8 - 10 studentů.	
Věk žáků	1. až 4. ročník SŠ a odpovídající ročníky gymnázií	
Pomůcky	Robotická stavebnice EV3 a počítače s nainstalovaným programovacím prostředím RobotC.	
Stručný popis aktivity s využitím přístroje	Úkolem žáků je vytvořit kreslicí tabulku pomocí displeje řídicí jednotky. Při stisku směrových tlačítek se bude vždy vykreslovat do požadovaného směru. Středové tlačítko slouží k mazání plochy displeje.	
Vhodné místo	Běžná učebna vybavená počítači s nainstalovaným programovacím prostředím RobotC.	
Cíle aktivity	Žáci se naučí vykreslovat či vypisovat různé prvky na plochu displeje řídicí jednotky.	
Rozvíjené kompetence	Kompetence k učení, k řešení problémů.	
Předchozí znalosti	Znalost práce s tlačítky řídicí jednotky.	
Mezipředmětové vztahy	Informační a komunikační technologie (algoritmizace úloh).	
Časový plán	Fáze činnosti s přístrojem	Metody a formy, motivace
40 minut	Tvorba programu a jeho průběžné testování.	Spolupráce studentů ve skupinách, koordinace činnosti vyučujícím.
Hodnocení	Hodnocena bude úplnost a přehlednost vyplněných informací a kvalita doplňujících multimediálních prvků.	

Zadání

Vytvořte funkční model kreslicí tabulky. Požadavky na tvorbu budou následující:

1. Pro kreslení využijte plochu displeje řídicí jednotky.
2. Ovládání při vykreslování budou obstarávat tlačítka řídicí jednotky.
3. Prostřední tlačítko bude sloužit k mazání plochy displeje.

Výsledný program ke stažení ve formátu .C

Program ke stažení viz. on-line kurz

7.6 Aktivita 6 - Ruční mixér

Téma	Ruční mixér s použitím dotykového senzoru	
Tematický celek	Programovací prostředí RobotC	
Motivační rámec	Dotykový senzor se hodí pro několik možností využití. Typicky se využívá jako tlačítko. Jeho velmi užitečnou vlastností je možnost rozeznávat tři stavy stisku. Tato aktivita se věnuje jeho použití jako spouštěče určité činnosti jiného zařízení.	
Počet žáků	Skupina 8 - 10 studentů.	
Věk žáků	1. až 4. ročník SŠ a odpovídající ročníky gymnázií	
Pomůcky	Robotická stavebnice EV3 a počítače s nainstalovaným programovacím prostředím RobotC.	
Stručný popis aktivity s využitím přístroje	Úkolem této aktivity je seznámit studenty s možnostmi použití dotykového senzoru. Ten je v úkolu použit jako tlačítko ručního mixéru. Mixér je poháněn pomocí středního motoru. Při ovládání rozlišuje tři stavy. Po prvním stisku a uvolnění se roztočí pomaleji, po dalším stisku maximální rychlostí a při třetím stisku se vypne.	
Vhodné místo	Běžná učebna vybavená počítači s nainstalovaným programovacím prostředím RobotC.	
Cíle aktivity	Studenti se naučí využívat dotykový senzor a rozeznávat jeho možné stavy.	
Rozvíjené kompetence	Kompetence k učení, k řešení problémů.	
Předchozí znalosti	Znalost práce s dotykovým senzorem.	
Mezipředmětové vztahy	Informační a komunikační technologie (algoritmizace).	
Časový plán	Fáze činnosti s přístrojem	Metody a formy, motivace
10 minut	Stavba modelu.	Spolupráce studentů ve skupinách, koordinace činnosti vyučujícím.
15 minut	Tvorba programu a průběžné testování.	Spolupráce studentů ve skupinách, koordinace činnosti vyučujícím.
5 minut	Závěrečné testování funkčnosti.	Spolupráce studentů ve skupinách, koordinace činnosti vyučujícím.
Hodnocení	Hodnocena bude průběžná práce na úloze, spolupráce studentů ve skupině a kvalita výsledného modelu a programu.	

Zadání

Aktivita si klade za cíl vás seznámit s funkcemi dotykového senzoru. V následující úloze se naučíte využít dotykový senzor jako spouštěč či přepínač určité reakce. Váš úkol je následující:

1. Postavte vhodný model ručního mixéru, který bude ovládaný pomocí tlačítka dotykového senzoru.
2. Po prvním stisknutí se mixér roztočí pomaleji, při druhém zrychlí na maximum a při třetím zastaví.

Výsledný program ke stažení ve formátu .C

Program ke stažení viz. on-line kurz

7.7 Aktivita 7 - Rozpoznávač barev

Téma	Rozpoznávač barev s využitím barevného senzoru	
Tematický celek	Programovací prostředí RobotC	
Motivační rámec	Ne každý z nás má barevné citění a okamžitě rozpozná, o jakou barvu se jedná. Pociťujeme to hlavně při nákupu oblečení. Barev existuje tolik, že se v nich laik téměř nevyzná. Díky robotické stavebnici si ale dokážeme sestavit jednoduchý rozpoznávač barev, který nám alespoň základní barvy dokáže určit.	
Počet žáků	Skupina 8 - 10 studentů.	
Věk žáků	1. až 4. ročník SŠ a odpovídající ročníky gymnázií	
Pomůcky	Robotická stavebnice EV3 a počítače s nainstalovaným programovacím prostředím RobotC (textová verze).	
Stručný popis aktivity s využitím přístroje	Úkolem studentů je vytvořit jednoduchý model ručního rozpoznávače barev, který bude opatřený barevným senzorem. Ten bude zjišťovat barevný odstín snímaného materiálu a barvu následně vypíše na displej řídicí jednotky.	
Vhodné místo	Běžná učebna vybavená počítači s nainstalovaným programovacím prostředím RobotC.	
Cíle aktivity	Studenti se naučí pracovat s barevným senzorem, seznámí se s jeho možnými režimy a s jeho využitím sestaví jednoduché zařízení k rozpoznávání barev.	
Rozvíjené kompetence	Kompetence k učení, k řešení problémů.	
Předchozí znalosti	Znalost práce s barevným senzorem.	
Mezipředmětové vztahy	Informační a komunikační technologie (algoritmizace), fyzika (světlo).	
Časový plán	Fáze činnosti s přístrojem	Metody a formy, motivace
10 minut	Stavba modelu.	Spolupráce studentů ve skupinách, koordinace činnosti vyučujícím.
15 minut	Tvorba programu a průběžné testování.	Spolupráce studentů ve skupinách, koordinace činnosti vyučujícím.
5 minut	Závěrečné testování a praktické ověření funkčnosti.	Spolupráce studentů ve skupinách, koordinace činnosti vyučujícím.
Hodnocení	Hodnocena bude průběžná práce na projektu a kvalita a funkčnost sestaveného modelu a vytvořeného programu.	

Zadání

V této aktivitě je vaším úkolem sestavit jednoduchý ruční rozpoznávač barev. Pro úspěšné splnění úlohy naplňte následující požadavky:

1. Sestavte model ručního rozpoznávače barev, který bude možné držet v jedné ruce (využijte barevný senzor).
2. Jakmile namíříte senzor na povrch, jehož barvu potřebujeme zjistit, informace o zjištěné barvě se vypíše na displej řídicí jednotky.

Výsledný program ke stažení ve formátu .C

Program ke stažení viz. on-line kurz

7.8 Aktivita 8 - Detektor pádu pevného disku

Téma	Detektor pádu disku	
Tematický celek	Programovací prostředí RobotC	
Motivační rámec	Nenadálé problémy se spuštěním počítače mohou někdy pramenit z poruchy pevného disku počítače. Stát se tak může i v jiných případech. Například při pádu spuštěného notebooku na zem. Ať již je příčina jakákoliv, ve velké většině případů nastane nepříjemná ztráta dat. Pevné disky ovšem obsahují zařízení, které při detekci pádu dokáže bezpečně zaparkovat zápisovou hlavu tak, aby nedošlo k poškození disku. My si takové zařízení zkusíme vytvořit díky robotické stavebnici za pomoci gyroskopického senzoru.	
Počet žáků	Skupina 8 - 10 studentů.	
Věk žáků	1. až 4. ročník SŠ a odpovídající ročníky gymnázií	
Pomůcky	Robotická stavebnice EV3 a počítače s nainstalovaným programovacím prostředím RobotC.	
Stručný popis aktivity s využitím přístroje	Žáci si nejprve sestaví jednoduchý model pevného disku. Uvnitř se pokusí vytvořit pohyblivý mechanismus, který bude simulovat jeho funkci (např. motor nebo pohyblivé rameno). Zařízení pro detekci pádu bude představovat gyroskopický senzor. Následně vytvoří program, který při prudké změně polohy zastaví chod disku.	
Vhodné místo	Běžná učebna vybavená počítači s nainstalovaným programovacím prostředím RobotC.	
Cíle aktivity	Žáci se naučí využívat gyroskopický senzor.	
Rozvíjené kompetence	Kompetence k učení, k řešení problémů.	
Předchozí znalosti	Znalost funkce a režimů gyroskopického senzoru.	
Mezipředmětové vztahy	Informační a komunikační technologie (algoritmizace úloh).	
Časový plán	Fáze činnosti s	Metody a formy, motivace
15 minut	Tvorba modelu disku.	Spolupráce studentů ve skupinách, koordinace činnosti vyučujícím.
20 minut	Tvorba programu a jeho průběžné testování.	Spolupráce studentů ve skupinách, koordinace činnosti vyučujícím.
15 minut	Testování funkčnosti modelu a programu.	Spolupráce studentů ve skupinách, koordinace činnosti vyučujícím.
Hodnocení	Hodnocena bude úplnost a přehlednost vyplněných informací a kvalita doplňkových multimediálních prvků.	

Zadání

Vytvořte funkční model pevného disku se zařízením pro detekci pádu. Požadavky na konstrukci modelu a program budou následující:

1. Sestavte jednoduchý model pevného disku.
2. Uvnitř bude umístěn pohyblivý mechanismus poháněný například motorem.
3. Pro detekci pádu použijte gyroskopický senzor.
4. Jakmile bude detekována prudká změna polohy, chod disku se zastaví.

Výsledný program ke stažení ve formátu .C

Program ke stažení viz. on-line kurz

7.9 Aktivita 9 - Turniket

Téma	Inteligentní turniket	
Tematický celek	Programovací prostředí RobotC	
Motivační rámec	Na hudebních koncertech a různých sportovních akcích se zjišťuje různými způsoby aktuální počet návštěvníků. Většinou se tomu tak děje díky turniketům, které sčítají počet lidí, kteří jimi projdou. V následující úloze si takový turniket vytvoříme za pomoci ultrazvukového senzoru.	
Počet žáků	Skupina 8 - 10 studentů.	
Věk žáků	1. až 4. ročník SŠ a odpovídající ročníky gymnázií	
Pomůcky	Robotická stavebnice EV3 a počítače s nainstalovaným programovacím prostředím RobotC (textová)	
Stručný popis aktivity s využitím přístroje	V této úloze studenti vytvoří model jednoduchého turniket, který sčítá počet návštěvníků kulturní akce. Ultrazvukový senzor bude přesně snímat prostor určený pro příchod do areálu. Jakmile jeho vysílaný signál protne některý z příchozích návštěvníků, bude započítán. Počet návštěvníků se bude postupně přičítat a vypisovat na displej.	
Vhodné místo	Běžná učebna vybavená počítači s nainstalovaným programovacím prostředím RobotC.	
Cíle aktivity	Studenti se naučí pracovat s ultrazvukovým senzorem a naučí se zpracovávat jím zjištěná data.	
Rozvíjené kompetence	Kompetence k učení, k řešení problémů.	
Předchozí znalosti	Znalost funkce a režimů ultrazvukového senzoru.	
Mezipředmětové vztahy	Informační a komunikační technologie (algoritmizace).	
Časový plán	Fáze činnosti s přístrojem	Metody a formy, motivace
10 minut	Tvorba modelu turniket.	Spolupráce studentů ve skupinách, koordinace činnosti vyučujícím.
15 minut	Tvorba programu a průběžné testování.	Spolupráce studentů ve skupinách, koordinace činnosti vyučujícím.
5 minut	Ověření funkčnosti a závěrečné testování.	Spolupráce studentů ve skupinách, koordinace činnosti vyučujícím.
Hodnocení	Hodnocena bude průběžná práce na projektu a kvalita sestaveného modelu a funkčnost vytvořeného programu.	

Zadání

V této aktivitě se seznámíte s funkcí ultrazvukového senzoru. Vaším úkolem je vytvořit inteligentní turniket pro kulturní akce, který sčítá počet návštěvníků. Požadavky na jeho funkčnost jsou následující:

1. Vytvořený model musí být jednoduchý a plně funkční (využijte ultrazvukový senzor).
2. Průchod turniketem musí být snímán ultrazvukovým senzorem, který bude detekovat každého příchozího návštěvníka.
3. Jakmile návštěvník turniketem projde, bude zaznamenán.
4. Aktuální počet návštěvníků se bude vypisovat na displeji řídicí jednotky.

Výsledný program ke stažení ve formátu .C

Program ke stažení viz. on-line kurz

7.10 Aktivita 10 - Bomba

Téma	Bomba	
Tematický celek	Programovací prostředí RobotC	
Motivační rámec	Práce pyrotechnika není záviděníhodná. Stojí za ní roky zkušeností, znalostí elektrotechniky a hlavně pevné nervy. Zkušený pyrotechnik ovšem dokáže vyhodnotit povahu výbušniny a její funkci a zneškodnit ji. Vaším úkolem v této aktivitě ale bude vytvořit nepředvídatelnou bombu, jejíž chování nebude možné zjistit. Bude záviset pouze na štěstí, zda se jí povede zneškodnit, či nikoliv.	
Počet žáků	Skupina 8 - 10 studentů.	
Věk žáků	1. až 4. ročník SŠ a odpovídající ročníky gymnázií	
Pomůcky	Robotická stavebnice EV3 a počítače s nainstalovaným programovacím prostředím RobotC.	
Stručný popis aktivity s využitím přístroje	Úkolem žáků je vytvořit program, který bude simulovat bombu. Potenciálnímu pyrotechnikovi se na displeji zobrazí výzva, aby bombu zneškodnil jedním ze tří nabízených tlačítek. U žádného z nich ovšem nebude zaručeno, že je správné. Význam tlačítek se totiž bude náhodně generovat. Program žáci doplní o vhodnou zvukovou signalizaci a výpis na displej.	
Vhodné místo	Běžná učebna vybavená počítači s nainstalovaným programovacím prostředím RobotC.	
Cíle aktivity	Studenti se naučí pracovat s tlačítky řídicí jednotky a naučí se vyhodnocovat jejich stisk a reagovat na něj.	
Rozvíjené kompetence	Kompetence k učení, k řešení problémů.	
Předchozí znalosti	Znalost práce s tlačítky řídicí jednotky.	
Mezipředmětové vztahy	Informační a komunikační technologie (algoritmizace).	
Časový plán	Fáze činnosti s přístrojem	Metody a formy, motivace
45 minut	Tvorba programu a jeho průběžné testování.	Spolupráce studentů ve skupinách, koordinace činnosti vyučujícím.
Hodnocení	Hodnocena bude průběžná práce na projektu a funkčnost vytvořeného programu.	

Zadání

V této aktivitě se naučíte využívat tlačítka řídicí jednotky. Vaším úkolem je vytvořit z řídicí jednotky pomyslný model bomby. Co pro úspěšnou realizaci musíte zvládnout?

1. Zvolte si alespoň tři tlačítka řídicí jednotky, která budou představovat dráty, které musí pyrotechnik přestříhnout.
2. Programově zajistěte, aby žádné tlačítko nebylo určeno ke zneškodnění bomby, ale aby se jejich funkce náhodně měnila.
3. Program vhodně graficky a zvukově ošetřete.

Výsledný program ke stažení ve formátu .C

Program ke stažení viz. on-line kurz

7.11 Aktivita 11 - Detektor světla

Téma	Detektor světla	
Tematický celek	Programovací prostředí RobotC	
Motivační rámec	Zařízení detekující úroveň světla v okolí nalezneme v mnoha mechanismech. Jedním z nich může být například automatické osvětlení, které na základě detekce světelných podmínek spouští pouliční osvětlení či nikoliv. My se v této aktivitě pokusíme takový detektor sestavit. Využijeme k tomu barevný senzor, který režim detekce světla v okolí obsahuje.	
Počet žáků	Skupina 8 - 10 studentů.	
Věk žáků	1. až 4. ročník SŠ a odpovídající ročníky gymnázií	
Pomůcky	Robotická stavebnice EV3 a počítače s nainstalovaným programovacím prostředím RobotC.	
Stručný popis aktivity s využitím přístroje	Úkolem žáků je sestavit a naprogramovat jednoduchý detektor světla, který rozlišuje světelné podmínky v okolí. Na displeji následně zobrazuje, zda je světla v okolí dostatek, či nikoliv.	
Vhodné místo	Běžná učebna vybavená počítači s nainstalovaným programovacím prostředím RobotC.	
Cíle aktivity	Žáci se naučí využívat cykly a podmínky.	
Rozvíjené kompetence	Kompetence k učení, k řešení problémů.	
Předchozí znalosti	Znalost práce s cykly a podmínky. Znalost režimů barevného senzoru.	
Mezipředmětové vztahy	Informační a komunikační technologie (algoritmizace).	
Časový plán	Fáze činnosti s přístrojem	Metody a formy, motivace
45 minut	Tvorba programu a jeho průběžné testování.	Spolupráce studentů ve skupinách, koordinace činnosti vyučujícím.
Hodnocení	Hodnocena bude průběžná práce na projektu a funkčnost vytvořeného programu.	

Zadání

V této aktivitě se naučíte využívat tlačítka řídicí jednotky. Vaším úkolem je vytvořit z řídicí jednotky pomyslný model bomby. Co pro úspěšnou realizaci musíte zvládnout?

1. Zvolte si alespoň tři tlačítka řídicí jednotky, která budou představovat dráty, které musí pyrotechnik přestříhnout.
2. Programově zajistěte, aby žádné tlačítko nebylo určeno ke zneškodnění bomby, ale aby se jejich funkce náhodně měnila.
3. Program vhodně graficky a zvukově ošetřete.

Výsledný program ke stažení ve formátu .C

Program ke stažení viz. on-line kurz

7.12 Aktivita 12 - Stopky

Téma	Stopky	
Tematický celek	Programovací prostředí RobotC	
Motivační rámec	Každý správný trenér potřebuje stopky. Využívají je trenéři vrcholových sportovců, závodníků, ale také například učitelé tělesné výchovy. Pomocí nich kontrolují výkony svých svěřenců a ověřují případné zlepšení v jejich sportovních výkonech. Jednoduché stopky si můžeme vytvořit také pomocí robotické stavebnice.	
Počet žáků	Skupina 8 - 10 studentů.	
Věk žáků	1. až 4. ročník SŠ a odpovídající ročníky gymnázií	
Pomůcky	Robotická stavebnice EV3 a počítače s nainstalovaným programovacím prostředím RobotC.	
Stručný popis aktivity s využitím přístroje	Úkolem žáků je sestavit a naprogramovat jednoduchý model digitálních stopek. Ty budou měřit postupně tři kola, která závodník urazil, a na závěr vypíší průměrný čas na kolo ze změřených tří kol. Jako měřicí tlačítko stopek může posloužit tlačítko dotykového senzoru nebo některé z tlačítek řídicí jednotky.	
Vhodné místo	Běžná učebna vybavená počítači s nainstalovaným programovacím prostředím RobotC.	
Cíle aktivity	Žáci se naučí využívat pole.	
Rozvíjené kompetence	Kompetence k učení, k řešení problémů.	
Předchozí znalosti	Znalost práce s polem.	
Mezipředmětové vztahy	Informační a komunikační technologie (algoritmizace).	
Časový plán	Fáze činnosti s přístrojem	Metody a formy, motivace
45 minut	Tvorba programu a jeho průběžné testování.	Spolupráce studentů ve skupinách, koordinace činnosti vyučujícím.
Hodnocení	Hodnocena bude průběžná práce na projektu a funkčnost vytvořeného programu.	

Zadání

Aktivita slouží k tomu, abyste se naučili využívat v programovacím prostředí RobotC pole. Za tímto účelem naprogramujete model stopek, které budou splňovat následující požadavky:

1. Zvolte si, čím budou stopky ovládány (např. tlačítko řídicí jednotky nebo tlačítko dotykového senzoru).
2. Stopky se spustí po spuštění programu a po každém stisknutí tlačítka bude zaznamenán čas aktuálně měřeného kola.
3. Celkem budou zaznamenána tři kola a následně bude spočten průměrný čas na kolo (všechny časy budou vypsány na displej řídicí jednotky).

Výsledný program ke stažení ve formátu .C

Program ke stažení viz. on-line kurz

8 Tipy a triky

Vytváření vlastních metod

Při vytváření programu v RobotC nemusíme využívat pouze funkce obsažené v knihovně funkcí. Je možné si vytvářet také funkce vlastní, které lze následně použít na libovolných místech v programu. Díky tomu, že nebudeme funkci zapisovat opakovaně, zkrátíme a zpřehledníme svůj program. Zároveň si také usnadníme jeho vytváření.

Funkce bez návratové hodnoty

Prvním typem vlastní vytvářené funkce je taková funkce, která při svém zavolání nevrací žádnou hodnotu, pouze se provede určitá sekvence příkazů. Funkce se definuje klíčovým slovem **void** značícím funkci bez návratové hodnoty, za kterým je uveden její název, případně parametr uvedený v závorce. V těle funkce se poté uvádí kód, který se při jejím zavolání vykoná.

Na obrázku níže můžete vidět příklad funkce bez návratové hodnoty včetně jejího zavolání v hlavní části programu.

```
void vzdalenost (int x)
{
    if (SensorValue (ultrazvuk) > x)
    {
        motor[motorA] = 0;
        motor[motorB] = 0;
    }
}

task main()
{
    vzdalenost(20);
}
```

S návratovou hodnotu

Druhá možnost realizace vlastní funkce je taková, od které očekáváme získání určitého výsledku. Z toho důvodu je nutné v jejím zápisu definovat, jakého datového typu bude vrácený výsledek nabývat. Klíčovým slovem **return** na závěr určíme, jakou hodnotu chceme po zavolání funkce získat.

Na obrázku níže můžete vidět jednoduchý příklad deklarace funkce s návratovou hodnotou i s jejím následným zavoláním a použitím v hlavní části programu.

```
int vzdalenost (int x)
{
    x = (getUSDistance (ultrazvuk)) * 22;
    return x;
}

task main()
{
    while (true)
    {
        int x = 0;
        int vypocet = vzdalenost (x);
    }
}
```

Dotaz task

Rozšiřující možnosti řízení umožňuje vlastní vytvořený dotaz task. Při jeho deklaraci v programu je definován klíčovým slovem **task** následovaným názvem dotazu a jeho parametry. Jeho zavolání v hlavní části programu je poté provedeno příkazem **startTask**, jehož parametrem v závorce je název dotazu. Od klasické funkce se dotaz liší možnostmi jeho řízení. Jedná se například o řízení výpočetního výkonu přiřazeného pro vykonávání funkce nebo priority jejího vykonávání.

Jednoduchý příklad deklarace a následného zavolání dotazu můžete vidět na následujícím obrázku.


```

task detekce ()
{
  while(true)
  {
    if(getGyroRate(gyro) > -20 & getGyroRate(gyro) < 20)
    {
      motor[motorA] = 20;
    }
    else
    {
      motor[motorA] = 0;
      waitMsec(5000);
    }
  }
}

task main()
{
  startTask(detekce);
}

```

9 Závěrečné tipy

Doporučený multimediální materiál

Nabídka základních i doplňkových komponent na prodejním webu české společnosti EDUXE s.r.o. (odkaz viz. on-line kurz)

Oficiální stránky společnosti Robomatter - tvůrce programovacího prostředí RobotC (odkaz viz. on-line kurz)

Projektový deník - Programovací prostředí EV3

Jméno:

Třída:

<i>Programovací prostředí EV3</i>	<i>Stručný postup (problémy řešené při práci, způsob řešení)</i>	<i>Hodnocení vyučujícího (splněno/nesplněno)</i>
Aktivita 1 - Příprava projektu Datum:		
Aktivita 2 - Lopatková turbína Datum:		
Aktivita 3 - Kuchyňská minutka Datum:		
Aktivita 4 - Řízení pásového transportéru Datum:		
Aktivita 5 - Železniční přejezd Datum:		
Aktivita 6 - Kreslící tabulka Datum:		

<i>Programovací prostředí EV3</i>	<i>Stručný postup (problémy řešené při práci, způsob řešení)</i>	<i>Hodnocení vyučujícího (splněno/nesplněno)</i>
Aktivita 7 - Ruční mixér Datum:		
Aktivita 8 - Rozpoznávač barev Datum:		
Aktivita 9 - Detektor pádu pevného disku Datum:		
Aktivita 10 - Turniket Datum:		
Aktivita 11 - Bomba Datum:		
Aktivita 12 - Detektor světla Datum:		
Aktivita 13 - Stopky Datum:		

Projektový deník - Grafické programovací prostředí RobotC

Jméno:

Třída:

<i>Grafické programovací prostředí RobotC</i>	<i>Stručný postup (problémy řešené při práci, způsob řešení)</i>	<i>Hodnocení vyučujícího (splněno/nesplněno)</i>
Aktivita 1 - Ruční mixér Datum:		
Aktivita 2 - Výstražný semafor se zvukovou signalizací Datum:		
Aktivita 3 - Detektor pádu pevného disku Datum:		

Projektový deník - Textově orientované programovací prostředí RobotC

Jméno:

Třída:

<i>Programovací prostředí RobotC</i>	<i>Stručný postup (problémy řešené při práci, způsob řešení)</i>	<i>Hodnocení vyučujícího (splněno/nesplněno)</i>
Aktivita 1 - Lopatková turbína Datum:		
Aktivita 2 - Kuchyňská minutka Datum:		
Aktivita 3 - Řízení pásového transportéru Datum:		
Aktivita 4 - Železniční přejezd Datum:		
Aktivita 5 - Kreslící tabulka Datum:		

<i>Programovací prostředí EV3</i>	<i>Stručný postup (problémy řešené při práci, způsob řešení)</i>	<i>Hodnocení vyučujícího (splněno/nesplněno)</i>
Aktivita 6 - Ruční mixér Datum:		
Aktivita 7 - Rozpoznávač barev Datum:		
Aktivita 8 - Detektor pádu pevného disku Datum:		
Aktivita 9 - Turniket Datum:		
Aktivita 10 - Bomba Datum:		
Aktivita 11 - Detektor světla Datum:		
Aktivita 12 - Stopky Datum:		