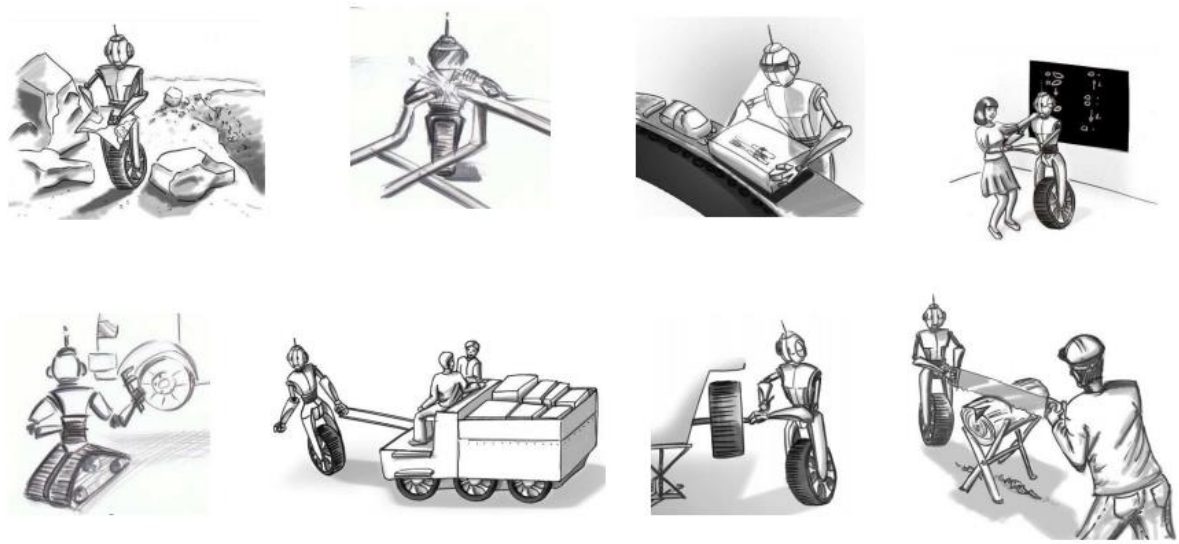
ZÁPADOČESKÁ UNIVERZITA V PLZNI
FAKULTA STROJNÍ
KATEDRA PRŮMYSLOVÉHO INŽENÝRSTVÍ A MANAGEMENTU



Základy řízení robotů pro strojní inženýrství

Autoři:
Ing. Bc. Miroslav Malaga
Doc. Ing. Zdeněk Ulrych, Ph.D.

Základy řízení robotů pro strojní inženýrství

Autoři:
Ing. Bc. Miroslav Malaga
Doc. Ing. Zdeněk Ulrych, Ph.D.

Vydala:
Západočeská univerzita v Plzni
Univerzitní 8, 301 00 Plzeň

První vydání, 145 stran
Plzeň 2020

ISBN 978-80-261-0486-5

Obsah

1 Úvod

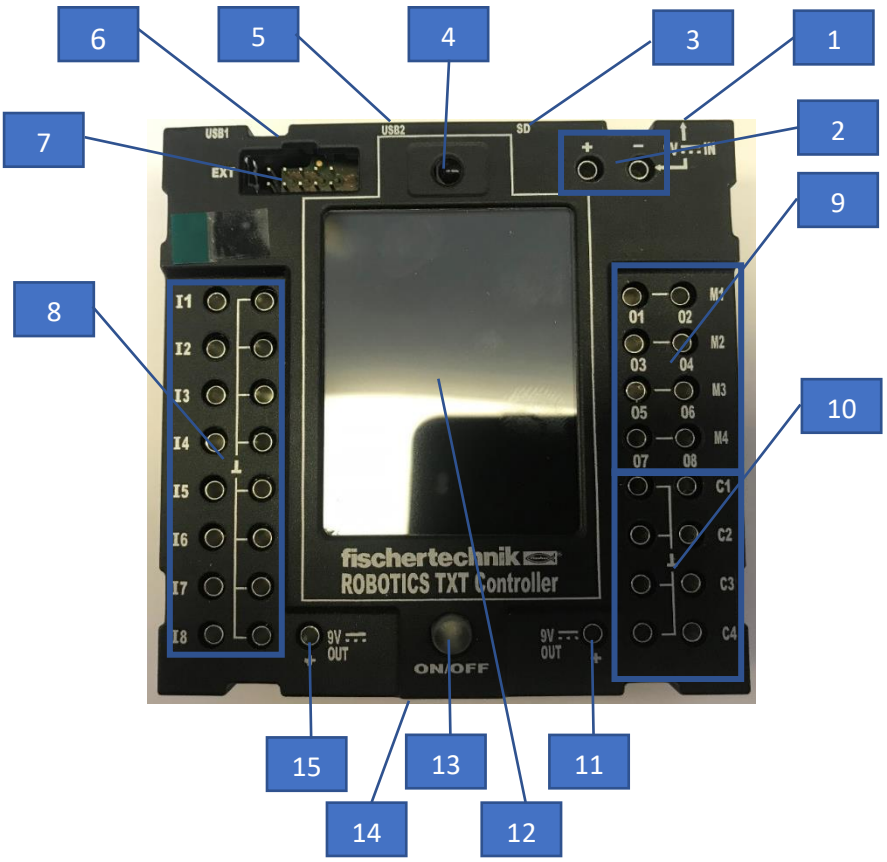
V dnešní době je čím dál tím důležitější využívat různé prvky automatizace. Je to dáno požadavkem na zvyšování produktivity práce, zvyšujícími se nároky zákazníků na kvalitu produktů, na cenu produktů, ale také zvyšujícími se požadavky zákazníků na možnost individualizace finálních produktů podle jejich přání a potřeb. Flexibilní automatizace je jednou z možných cest, jak tyto protikladné požadavky řešit. Mezi progresivní používané prvky automatizace můžeme zařadit různé softwarově programovatelné zařízení (např. stroje, roboty, …). Na budoucí strojní inženýry, kteří by měli buď navrhovat různá softwarově programovatelná zařízení, nebo by je měli nastavovat, popřípadě je jen efektivně využívat, jsou kladeny čím dál tím větší nároky na jejich znalosti z různých oborů. Cílem tohoto textu není naučit čtenáře programovat skutečná průmyslová zařízení, ale algoritmicky zvládnout naprogramovat různá zařízení (např. roboty), které jsou realizovaná pomocí modelů postavených ze stavebnic.

V rámci cvičení předmětu Počítačová podpora ve strojírenství se využívá stavebnice Fischertechnik 519341 STEM ENGINEERING, díky které se studenti seznámí s tematickými moderními technologiemi. V případě předmětu KPV/PPVS jsou to hlavně základy kybernetiky, řídicích systémů, senzorové techniky, motorizace, automatizace, robotiky, digitální komunikace a programování. Okrajově pak s mobilní robotikou, konstruováním, mechanickými systémy a návrhem systémů.

2 Základní přehled stavebnice pro potřeby KPV/PPVS

Řídící jednotka slouží k ovládání postavených modelů. Program se vytvoří pomocí vývojového prostředí ROBO PRO a do řídící jednotky se nahraje buď přes USB kabel, nebo bezdrátově (WIFI, Bluetooth).

2.1 ROBOTICS TXT Controller (řídící jednotka)



Obrázek 1: ROBOTICS TXT Controller (řídící jednotka)

1. Vstup pro napájení síťovým zdrojem
2. Vstupy pro napájení bateriovým setem (9 V). Dodržujeme pravidlo, že PLUS je červený drát, MINUS/ZEM je zelený drát.
3. Vstup pro Micro SD kartu. Lze využít pro rozšíření paměti.
4. IR přijímací dioda (infra dioda jako např. u TV ovladačů). Umožňuje např. dálkové ovládání projektu pomocí infraportu.
5. Mini USB vstup. Slouží pro propojení řídící jednotky s počítačem a následně zajišťuje např. přenos zkompilovaných programů do řídící jednotky. Přes USB kabel není řídící jednotka napájena. Vždy je potřeba použít baterii, nebo napájecí zdroj.
6. USB A vstup, v případě uvedené stavebnice pro USB kameru.
7. Rozšiřující rozhraní – umožňuje propojení řídicích jednotek, a navíc přidává další vstupy a výstupy.
8. Univerzální vstupy (I1 – I8). Lze je využít jako digitální, snímající digitální stav 0/1, nebo jako analogové, které čtou buď napětí, nebo proud. V levém sloupečku je PLUS, v pravém ZEM.
9. Výstupy (M1 – M4, O1 – O8). Jsou to výkonové výstupy, které dodávají 9V elektřiny jakékoliv součásti, která je k nim připojena. Mělo by se jednat pouze o motory, lampy, bzučáky nebo elektromagnety. Připojení M1 – M4 jsou diferenciální výstupy, což umožňuje chod motorů

v obou směrech. 01 – O8 se používá se zemí a může běžet 8 různých výstupu pouze v jednom směru.

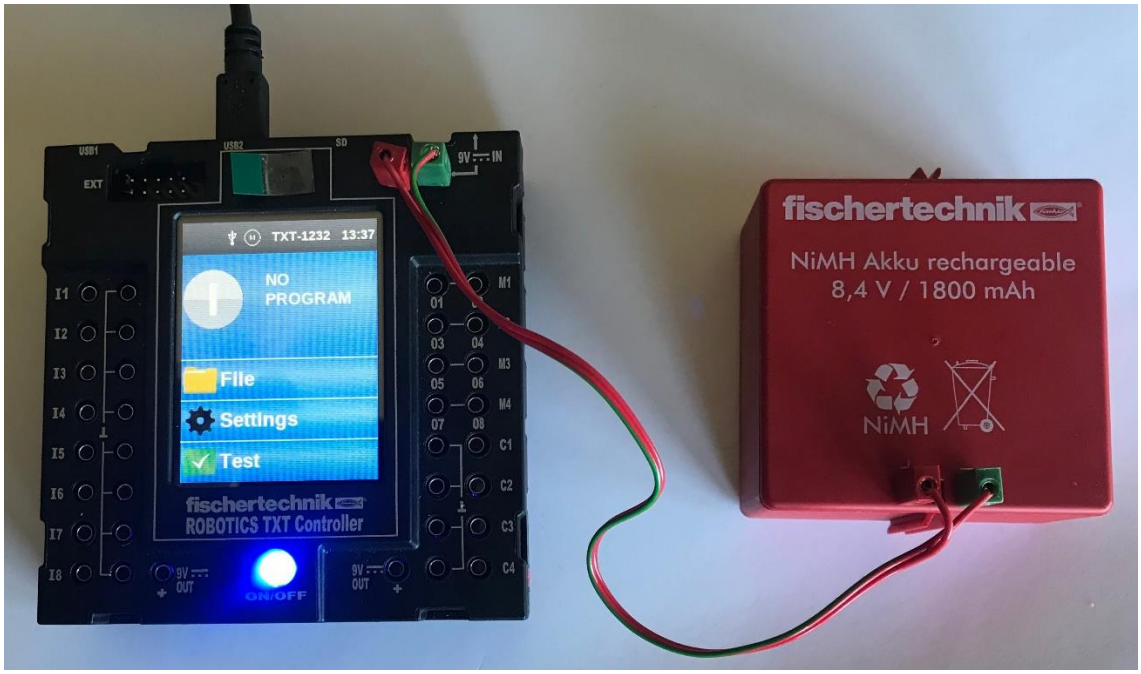
10. Vstupy (C1 – C4). Snímají digitální stav 1/0. Typické komponenty jsou přepínače, nebo fototranzistor. Využívají se také pro encondéry na motorech – tedy pro krokový motor, pokud chceme řídit otáčky, nebo jej synchronizovat s jiným. V pravém sloupečku je PLUS, v levém ZEM.
11. Napájení 9 V s neustálým proudem – neřídí se programem.
12. Displej s dotykovým ovládáním.
13. Zapnutí/vypnutí řídící jednotky.
14. Reproduktor.
15. Neregulované napájení 9 V– neřídí se programem.

Řídící jednotku je možné napájet pomocí:

- Baterie
- Ze sítě napájecím zdrojem a pomocí Power Controller

2.1.1 Napájení baterií

Řídící jednotku lze napájet baterií. Tato možnost je vhodná hlavně pro samostatně pohyblivé projekty typu autíčko, vozík na materiál apod., nebo při práci se stavebnicí mimo dosah el. zásuvek. Zapojení při využívání napájení z baterie viz Obrázek 2. Je nutné zachovat správnost propojení, tedy PLUS baterie na PLUS řídící jednotky a MINUS baterie na MINUS řídící jednotky.



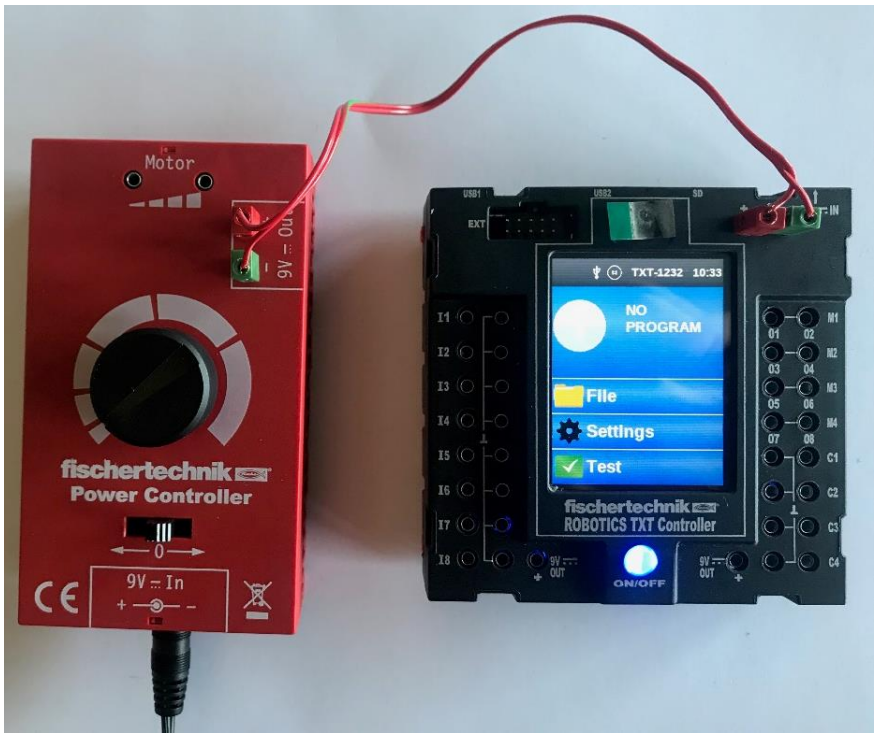
Obrázek 2: Napájení řídící jednotky baterií

2.1.2 Napájení ze sítě a Power Controller

Pro nejjednodušší napájení ze sítě stačí zapojit napájecí zdroj napřímo do řídící jednotky. Druhá možnost je navíc využít Power Controller, který kromě napájení umožňuje i např. řídit výkon motoru pomocí otočného knoflíku.

Obrázek 3: Power Controller

1. Neregulovaný výstup 9 V, vhodný pro napájení řídící jednotky.
2. Regulovaný výstup 0 – 9 V řízený otočným knoflíkem.
3. Otočný knoflík pro řízení regulovaného výstupu.
4. Přepínač pro výstup – lze výstup vypnout, nebo změnit +/-. U motoru to prakticky znamená změnit směr otáček.
5. Vstup pro napájení kabelem (je součástí Power Controlleru, který lze využít i napřímo pro napájení řídící jednotky, nabíjení baterie, nebo napájení komponent, u kterých vyžadujeme neustálé zapnutí.



Obrázek 4: Napájení řídící jednotky přes Power Controller

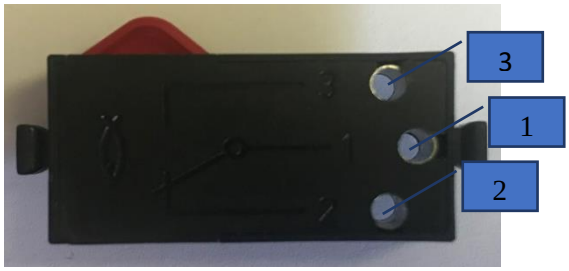
2.2 Základní prvky pro potřeby KVP/PPVS

V následujících podkapitolách jsou popsány používané prvky používané při tvorbě modelů. U jednotlivých prvků je ukázáno, jak je možné je zapojit k řídící jednotce.

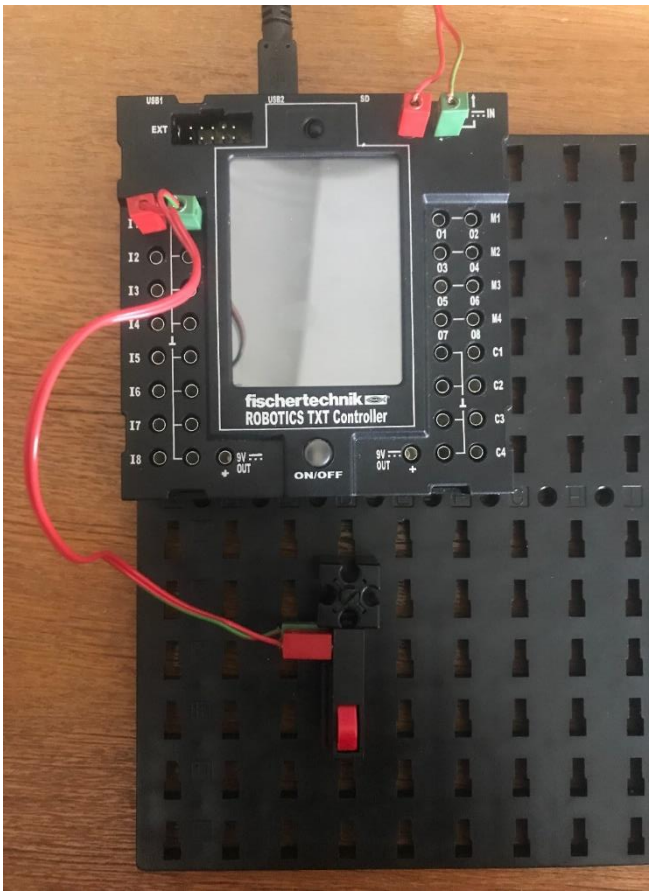
2.2.1 Spínač

V tomto případě se jedná přesně řečeno o koncový spínač. Je to zařízení, které v elektrotechnice slouží ke spínání a rozepínání elektrických obvodů v koncových polohách. Velmi jednoduše se dají v tomto případě využít i jako tlačítka, která např. spouští některou z naprogramovaných procedur. Využívají se jako vstupy informací, zapojujeme je tedy do řídící jednotky do pinů I(1 - 8).

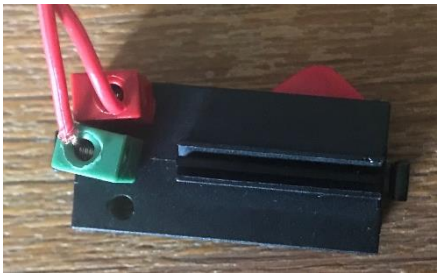
- Při zapojení 1 – 3 je obvod rozepnutý a stiskem tlačítka se uzavře.
- Při zapojení 1 – 2 je obvod zapnutý a stiskem tlačítka se rozepne.
- Pokud používáme tlačítko pro spouštění nějaké události, využíváme na tlačítku piny 1 a 3.



Obrázek 5: Spínač (Koncový spínač)



Obrázek 6: Spínač - ukázka možného zapojení do řídící jednotky na vstup I1



Obrázek 7: Spínač - detail zapojení do spínače

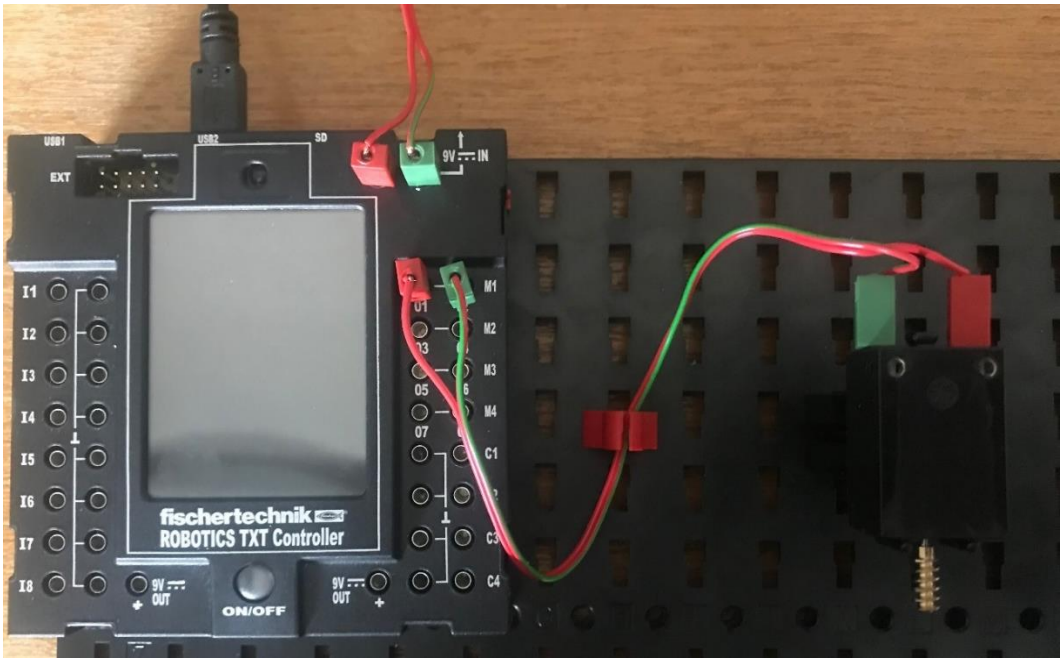
2.2.2 Mini motor a XS motor

Jedná se o jednoduché elektromotory, u kterých lze bez dalších přidaných prvků pouze řídit rychlost otáček a směr otáček. Motory mini a XS se liší pouze velikostí a možnostmi mechanického přichycení ke konstrukci. Zapojujeme je na řídící jednotce do pinů M(1 - 4).

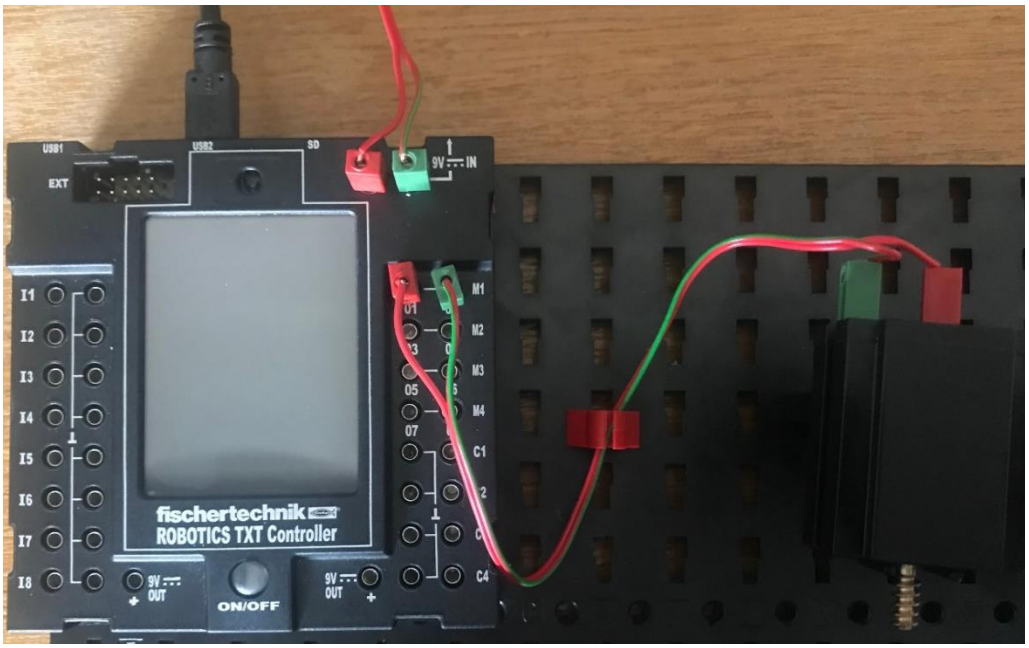
Princip elektromotoru je založený na využití silových účinků magnetického pole. Zjednodušeně lze říci že využíváme vzájemné přitahování a odpuzování dvou elektromagnetů. Sílu a polaritu těchto elektromagnetů můžeme řídit velikostí protékajícího elektrického proudu (regulace rychlosti otáček).

Obrázek 8: Mini motor

Obrázek 9: XS motor



Obrázek 10: Mini motor - ukázka možné zapojení na výstup M1



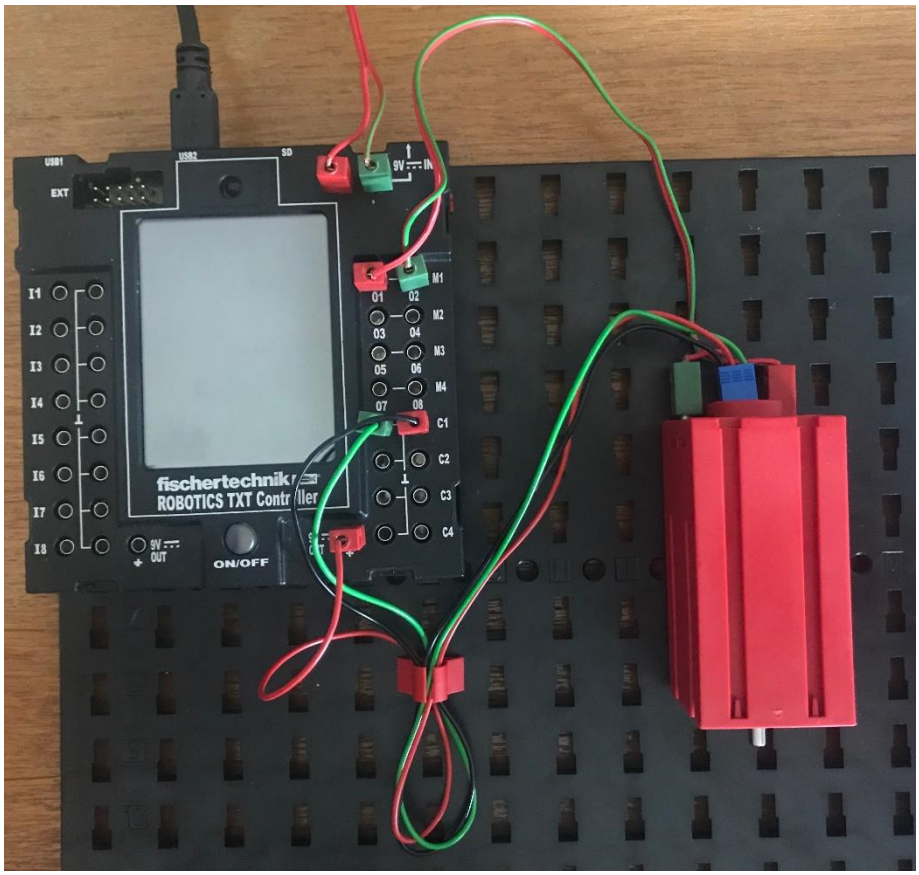
Obrázek 11: XS motor - ukázka možného zapojení na výstup M1

2.2.3 Krokový motor

Jedná se o elektromotor, který je navíc vybaven převodovou hlavou, která snižuje otáčky na výstupu a zvyšuje točivý moment. Navíc je vybaven obvodem, který slouží k počítání otáček motoru, díky čemuž je bez dalších přidaných součástí možné přesně řídit chod motoru, nebo jej lze např. synchronizovat s druhým zapojeným motorem (stejného typu). Pro řízení otáček, nebo synchronizaci motoru je potřeba použít k normálnímu zapojení motoru ještě 3žílový kabel, kde:

- Červený kabel se zapojuje na +9V
- Zelený kabel je zem
- Černý kabel slouží pro přenos signálu a na řídící jednotce se zapojuje na C(1 - 4)

Obrázek 12: Krokový motor

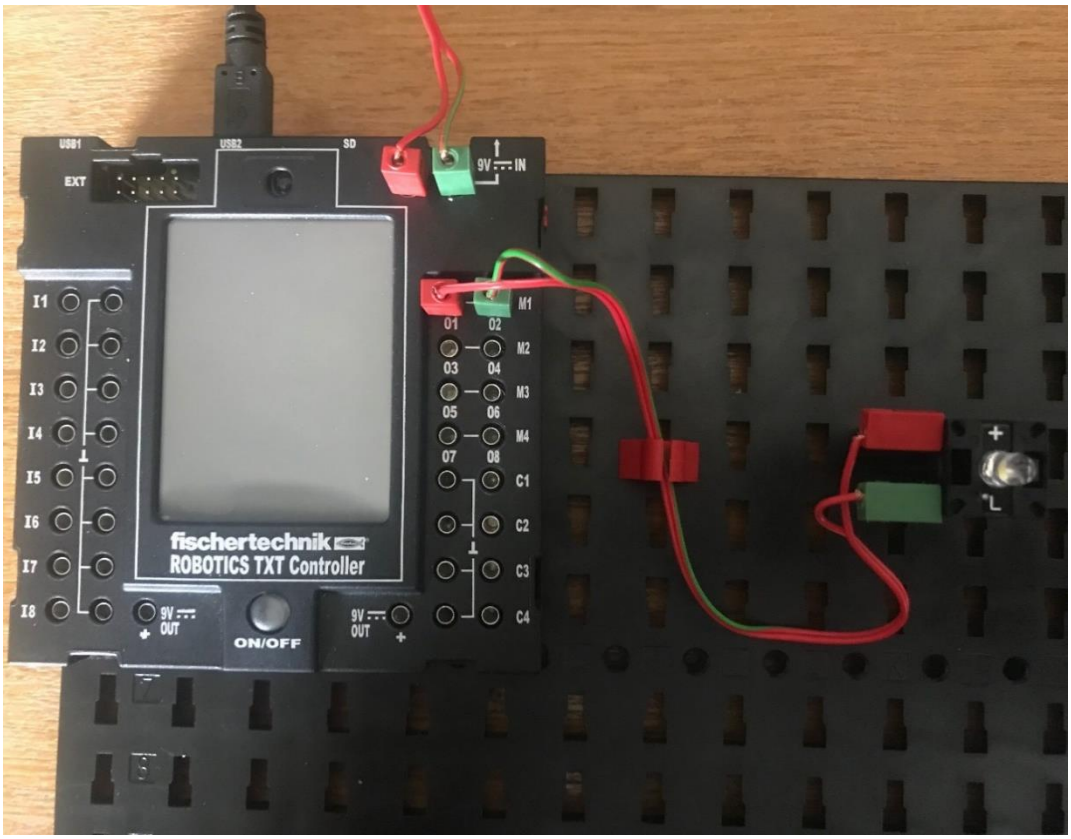


Obrázek 13: Krokový motor - ukázka zapojení na výstupy M1(motor) a C1 + 9 V (řízení motoru)

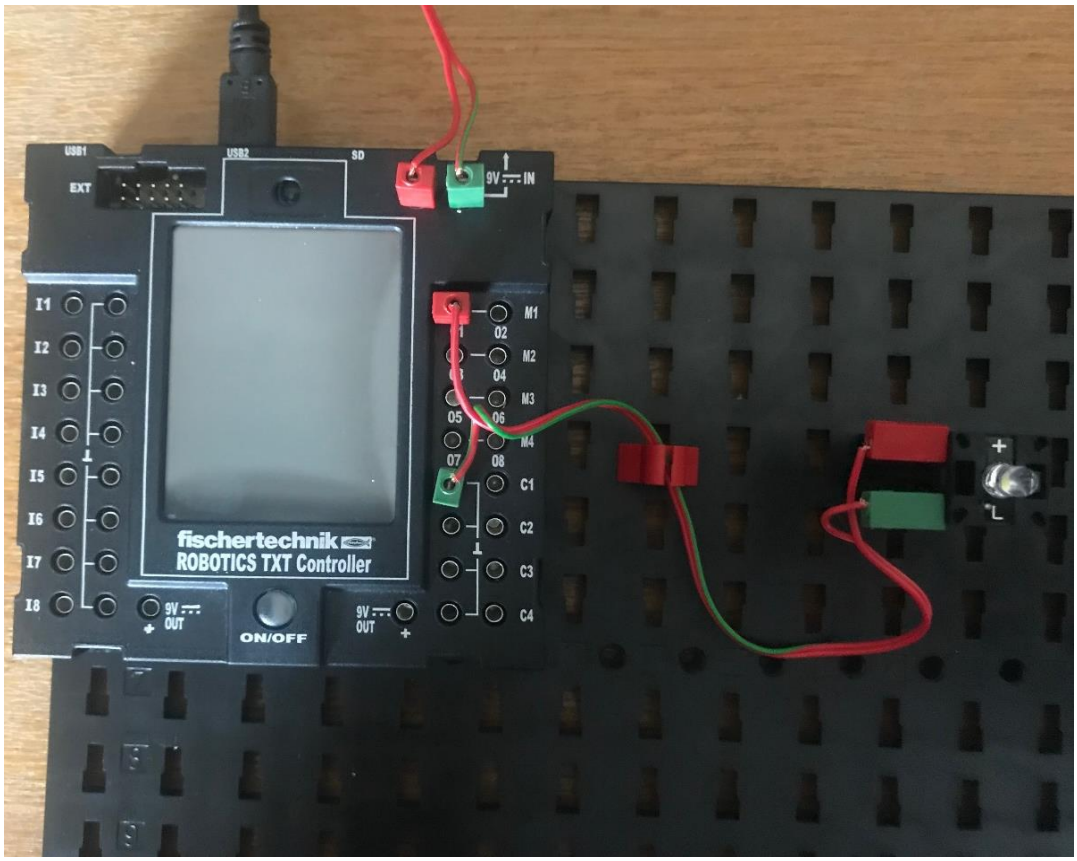
2.2.4 LED

Light-Emitting Diode, česky elektroluminiscenční dioda, popř. světelná dioda. V elektrotechnice se jedná o diodu, která emituje světlo, případně infračervené, nebo ultrafialové záření. LED mohou vydávat jak teplé, tak studené světlo (záleží na typu LED). V tomto konkrétním případě emituje světlo. Zapojení na řídící jednotce je na piny M(1 – 4, nebo O(1-8) a zem. Je potřeba dbát na to, aby v případě LED bylo zapojeno PLUS řídící jednotky na PLUS diody!

Obrázek 14: LED komponenta



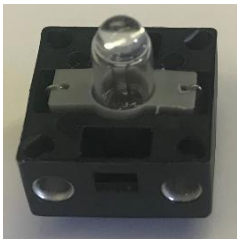
Obrázek 15: LED - ukázka zapojení na výstup M1



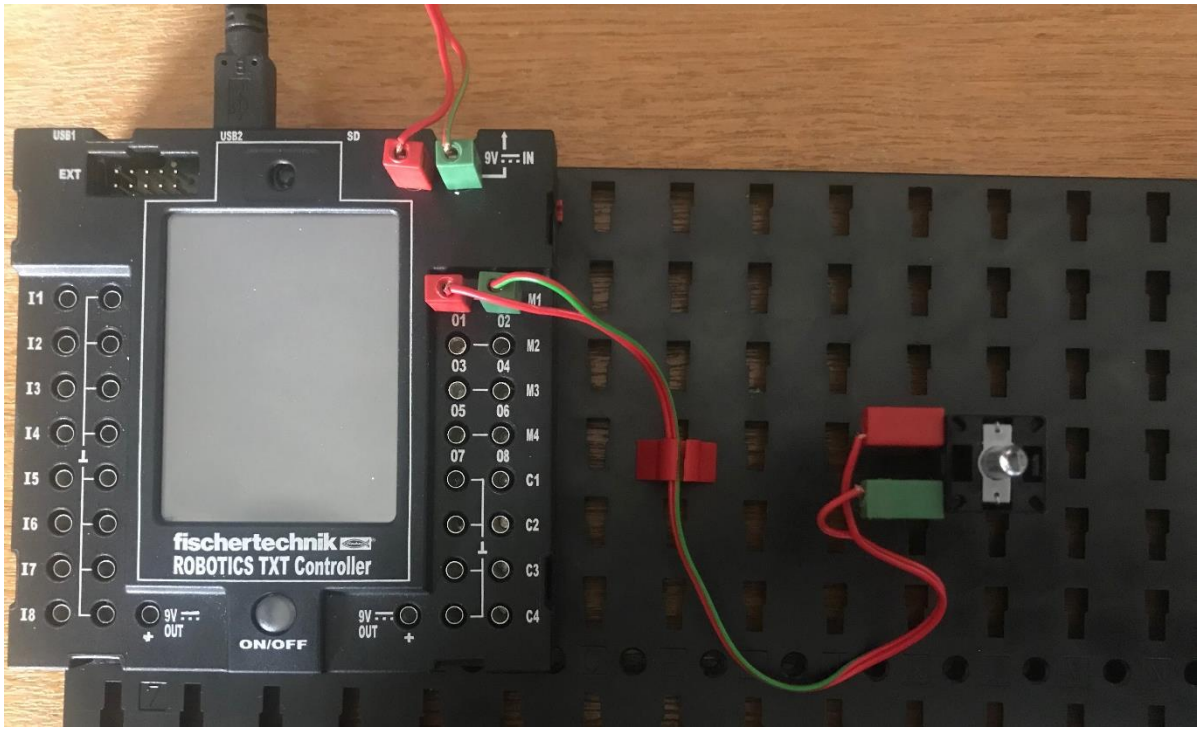
Obrázek 16: LED - ukázka zapojení na výstup O1 a zem

2.2.5 Žárovka

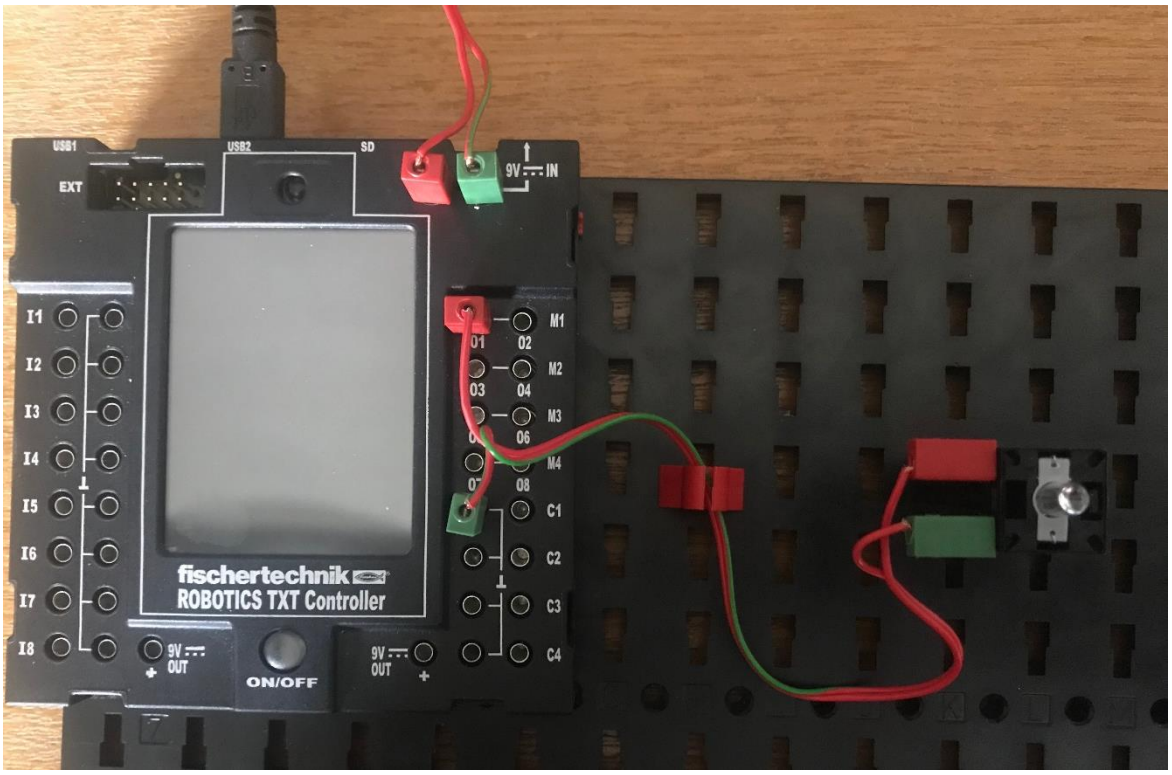
V tomto případě se jedná opravdu o obyčejnou žárovku, tedy nejjednodušší zařízení k přeměně elektrické energie na světlo. Funguje na principu zahřívání tenkého vodiče elektrickým proudem. Žárovka vždy vydává teplé světlo. Na řídící jednotku ji napojujeme na piny M(1 - 4), nebo O(1-8) a zem.



Obrázek 17: Žárovka



Obrázek 18: Žárovka - ukázka zapojení na výstup M1



Obrázek 19: Žárovka - ukázka zapojení na výstup O1 a zem

2.2.6 Fototranzistor

Tato součástka slouží k vyhodnocení, jestli na ni dopadá světlo a bodle toho má stav 1, nebo 0. Lze ji např. tedy využít pro automaticky spouštěný větráček – svítí-li na fototranzistor světlo, bude automaticky spuštěn větráček. Při zapojení je potřeba dbát na to, aby PLUS bylo přivedeno na červeně označený vstup fototranzistoru! Protože slouží jako vstupní zařízení – dává nám informaci, jestli světlo dopadá/nedopadá (a to následně vyhodnocujeme), je potřeba jej zapojovat na řídící jednotce na piny I(1 - 8).

Obrázek 20: Fototransistor

Obrázek 21: Fototranzistor - ukázka zapojení na vstup I1

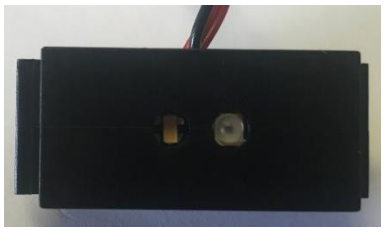
2.2.7 Optický barevný senzor

Optický barevný senzor měří vlastnosti povrchových barev, nebo světelných zdrojů. Hodnoty, získané senzorem se budou lišit v závislosti na osvětlení místnosti, vzdálenosti od snímaného povrchu a tvaru snímaného povrchu. Senzor se skládá z jasné LED, která prostor před senzorem osvětluje a receptoru, který zachycuje odražené světlo (princip je postavený na tom, že materiály absorbují různá množství světla na základě barvy, typu materiálu a textury povrchu). Z výše uvedeného vyplývá, že je potřeba jej vždy zkalibrovat podle okolních podmínek, tedy ve vlastním kódu ošetřit, že naměřená (vrácená) hodnota optickým senzorem za daných podmínek odpovídá např. červené barvě. Optický senzor nám totiž nedá barvu, ale číselnou hodnotu, které my musíme barvu přiřadit. Neměla by to být jedna hodnota = barva, ale „rozumný interval hodnot“ = barva. Je to dáno tím, že při snímání barvy senzorem nám senzor může pokaždé vrátit lehce jinou hodnotu. Protože se jedná o zařízení, které hodnoty čte a předává je řídí jednotce, tak jej zapojujeme na piny I(1 - 8), nebo C(1-4). Toto zařízení vyžaduje zapojení na 9 V, proto může vypadat zapojení např. takto[1]:

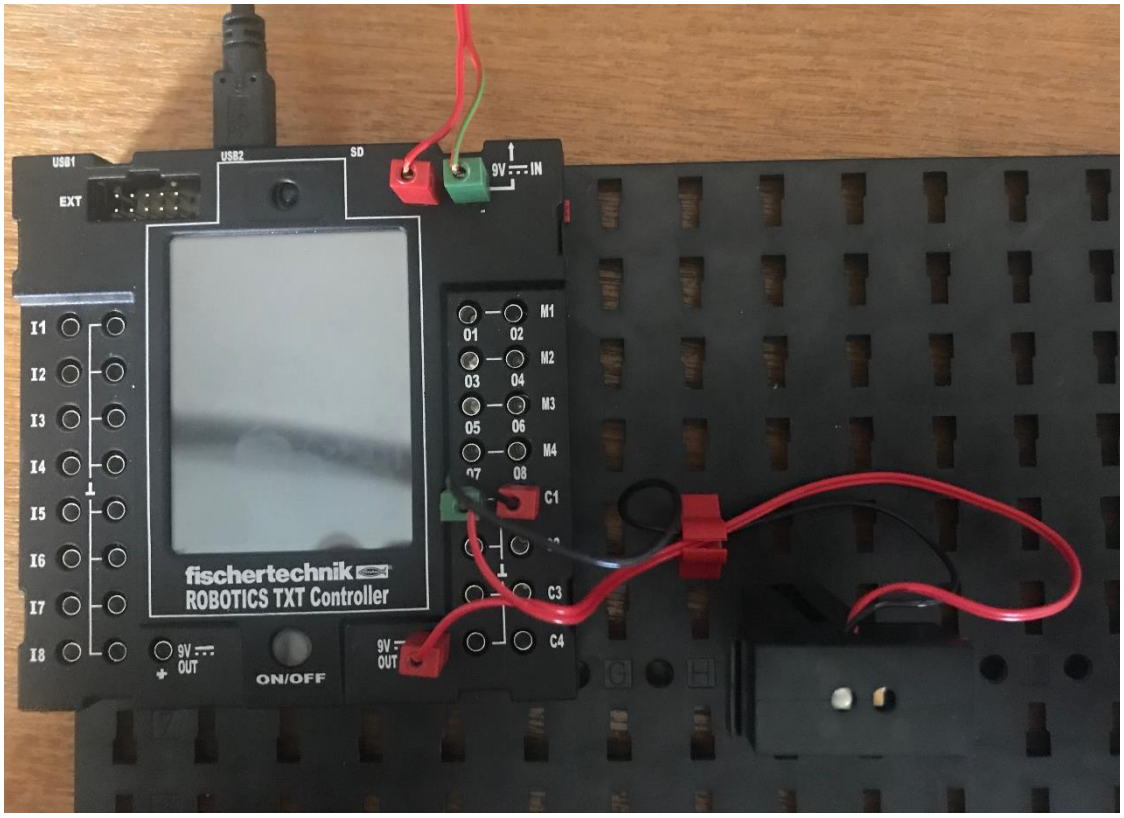
- Černý kabel (základní napájení) +I1

[1] Bude ukázáno na konkrétním praktickém příkladu

- Zelený kabel (zem) – zem I8
- Červený kabel (+) – napájení +9 V



Obrázek 22: Optický barevný senzor



Obrázek 23: Optický barevný senzor - ukázka zapojení na vstup C1 a +9V

2.2.8 NTC rezistor

Česky také Termistor. Jedná se o součástku, jejíž elektrický odpor je závislý na teplotě – proto je možné jej využít pro její měření. Abychom hodnoty získané z termistoru byli schopni převést např. na °C, musíme znát VA charakteristiku termistoru, která však není lineární. Tzn. musíme mít k dispozici např. převodní tabulku od výrobce termistoru, nelze použít pro převod trojčlenku. NTC rezistor (nebo také negastor) je termistor s negativním teplotním koeficientem, tedy se zahřátím součástky odpor klesá.

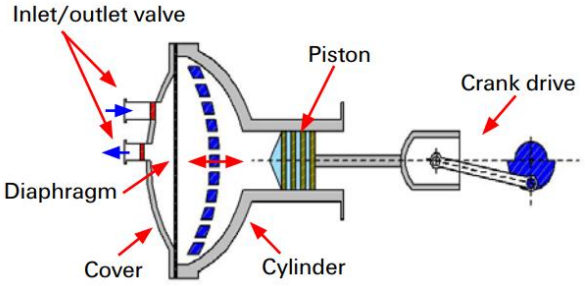
Obrázek 24: NTC rezistor

Obrázek 25: NTC rezistor - ukázka zapojení na vstup I1

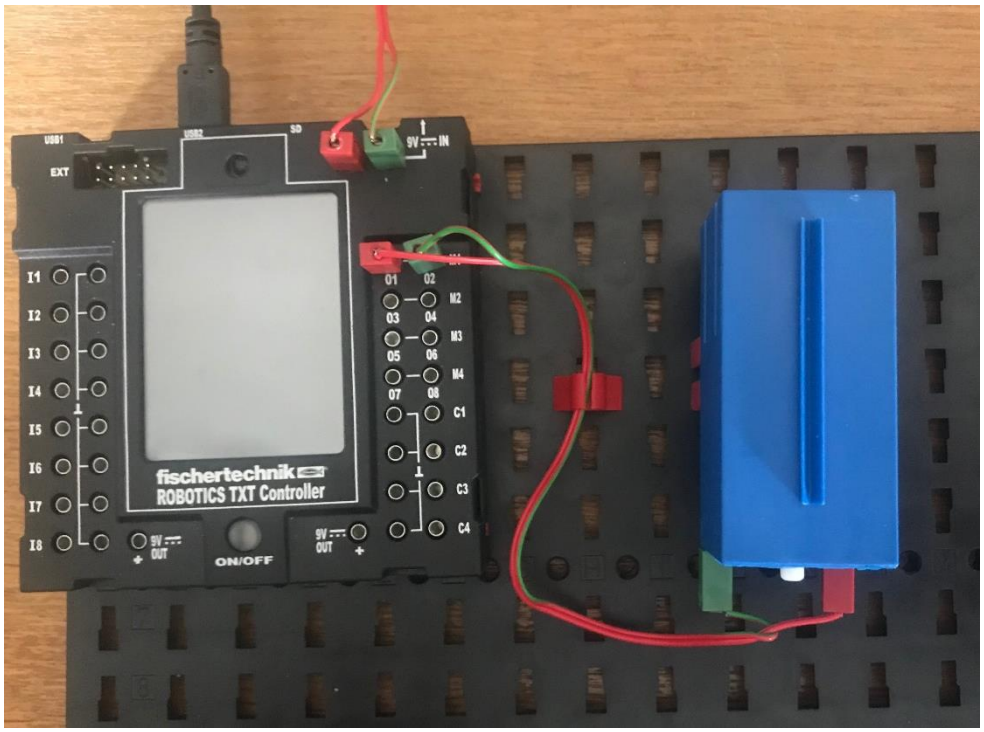
2.2.9 Vzduchový kompresor

Vzduchový kompresor je jednoduše řečeno motorová pumpa vzduchu. Větší, popř. složitější pneumatické systémy doplňují vzduchový kompresor vzduchovou nádrží, která slouží jako rezervoár pro udržení tlaku vzduchu např. i při vypnutí kompresoru. Kompresor je možné zapojit jak na 9V+ pin a zem z pinu I8 (pak poběží kompresor neustále), tak i na piny M(1 - 4), kdy můžeme řídit jeho zapnutí/vypnutí jen na potřebnou dobu.

Obrázek 26: Kompresor



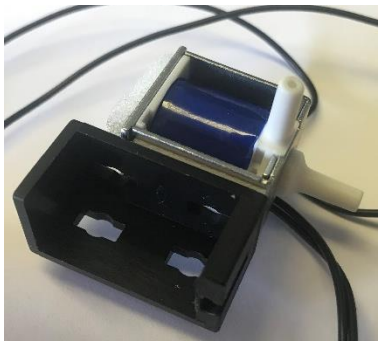
Obrázek 27: Princip membránového kompresoru [1]



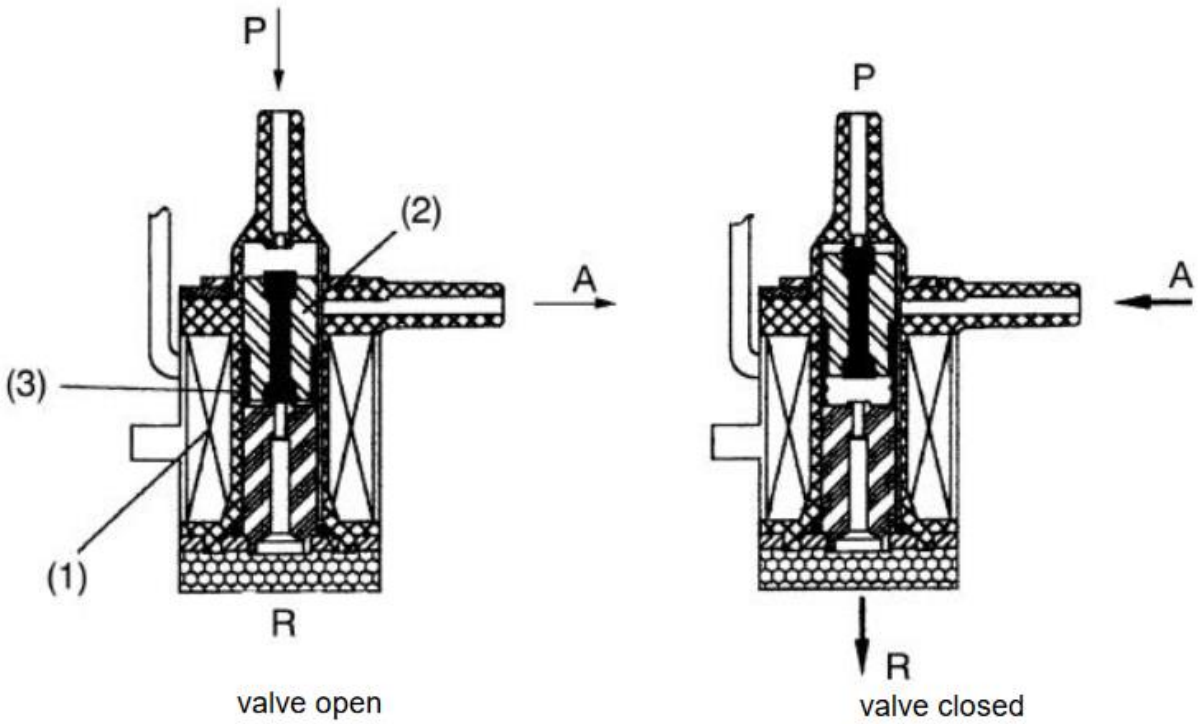
Obrázek 28: Vzduchový kompresor – ukázka zapojení na výstup M1

2.2.10 Solenoidový ventil

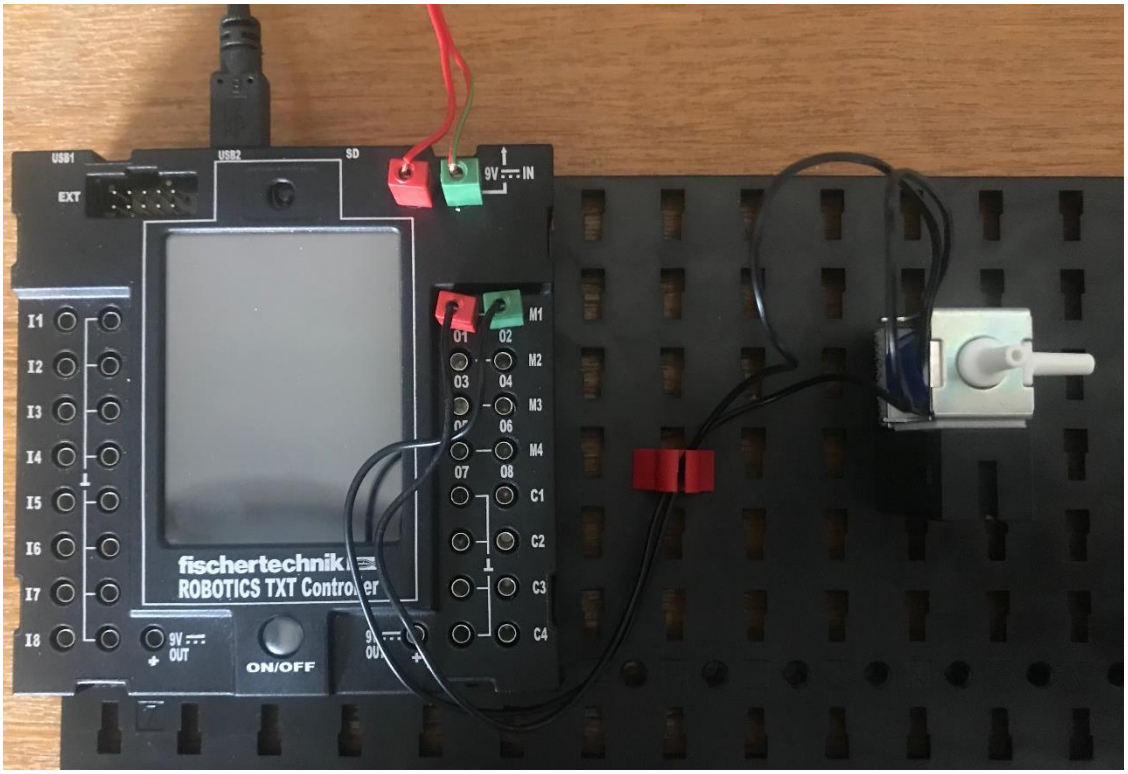
Solenoidový ventil je elektromagneticky ovládaný ventil. V tomto případě se jedná o 3/2 cestný ventil. Tzn. že v jedné poloze dochází k průtoku tekutiny a ve druhé poloze k vyčerpání tekutiny – tedy jsou 2 cesty kapaliny. Trojka v označení znamená, že jsou 3 vstupy/výstupy. První dva jsou na obrázku na první pohled patrné, třetí je zezadu ventilu překrytý molitanem, který slouží jako tlumič. Ventil se zapojuje stejně jako např. motor do pinů M(1 - 4) na řídící jednotce.



Obrázek 29: Solenoidový ventil



Obrázek 30: Princip fungování solenoidového ventilu [1]



Obrázek 31: Solenoidový ventil - ukázka zapojení na výstup M1

2.2.11 Pneumatický válec

Pneumatický válec je mechanické zařízení sloužící k převodu síly stlačeného vzduchu na mechanický pohyb. V tomto případě se jedná o jednočinný válec, tedy při přivedení vzduchu se pístnice vysune a její vratný pohyb obstará pružina uvnitř válce.

Obrázek 32: Pneumatický válec (jednocestný)

2.2.12 Vakuové sací zařízení

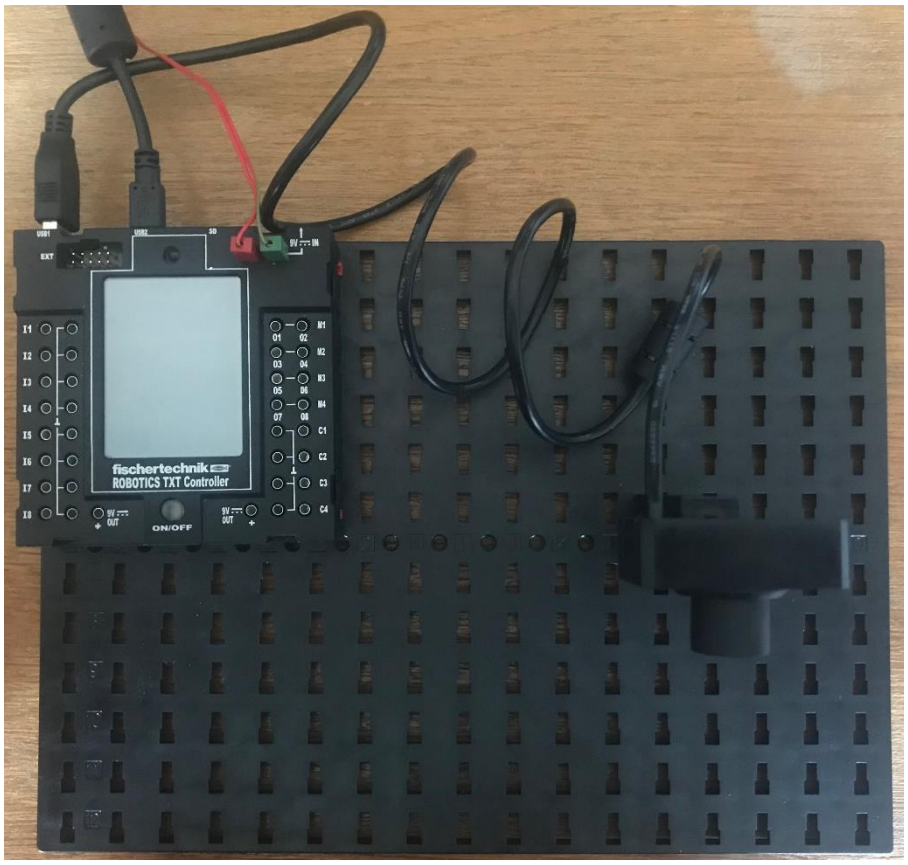
V tomto případě se jedná o speciální koncovku z velmi přizpůsobivé gumy. Při přiložení koncovky a vytvoření podtlaku se předmět „přisaje“ a je možné jej např. přemístit. Nutným předpokladem je, že uchopovaný předmět má vhodnou povrchovou strukturu.

Obrázek 33: Vakuové sací zařízení

2.2.13 USB kamera

Jedná se o klasickou webovou kameru s připojením do USB. Kamera disponuje manuálním zaostřením. Lze ji využít např. pro vyhodnocování předmětů, hledání/následování cesty apod.

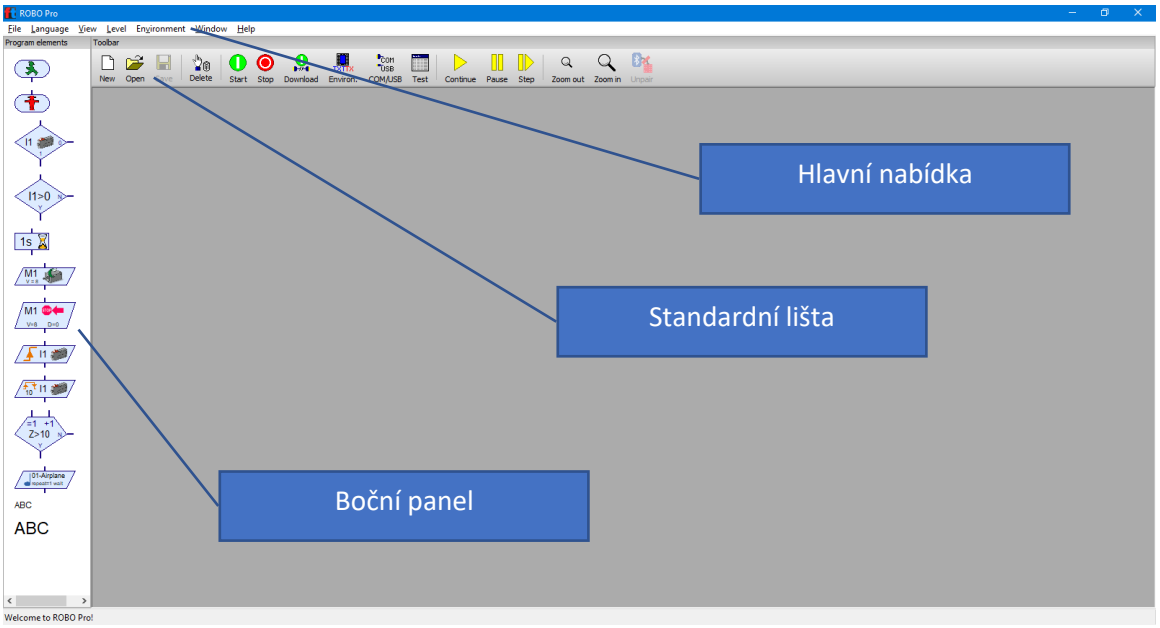
Obrázek 34: USB kamera



Obrázek 35: USB Kamera - ukázka zapojení

2.3 Software ROBO Pro

Software ROBO Pro, který slouží k vytváření programů a komunikaci s řídící jednotkou je přehledný a uživateli zvyklému na práci v MS Windows by měla stačit poměrně krátká doba k zvládnutí jeho ovládání. Lze jej rozdělit na hlavní nabídku, boční panel a horní lištu ikon.



Obrázek 36: Popis GUI programu - rozdělení okna

2.3.1 Hlavní nabídka

File	Pro uživatele MS Windows standardní menu. Umožňuje založit nový program, otevřít existující program, uložit program a tisknout.
Language	Umožňuje přepnout aplikaci do jiného jazyka.
Draw	Obsahuje funkce pro editování vložených symbolů vývojového diagramu, jejich mazání a editování. Všechny tyto možnosti jdou dělat rychleji za pomoci myši a klávesnice (mazání Delete, uchopení a přesun symbolu – levé tlačítko myši a její pohyb, …).
View	Obsahuje pouze možnost zobrazení/skrytí popisků ikon ve standardní liště.
Level	Obsahuje 5 úrovní. Volbou úrovně se ovlivňuje nabídka bočního panelu. Úroveň 1 zpřístupňuje pouze základní elementy pro tvorbu programu. Úroveň 5 zpřístupňuje všechny elementy, které v ROBO Pro existují. Nechce-li nad tím uživatel přemýšlet, je vhodné rovnou si zapínat úroveň 5.
Environment	Slouží pro nastavení správné řídící jednotky. My využíváme ROBO TXT/TXT Controller.
Bluetooth	Umožňuje připojení k řídící jednotce přes Bluetooth.
Window	Umožňuje přepínat rozložení oken otevřených programů.
Help	Přístup do nápovědy, odkaz na www stránky výrobce a odkaz pro stahování aktualizací.

2.3.2 Standardní lišta

Zde je potřeba myslet na to, že ikony, které se zobrazují, závisí na úrovni zvolené v hlavní nabídce v menu Level.

Sekce pro práci se soubory

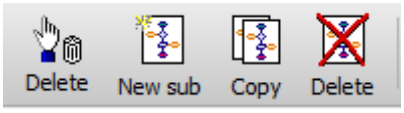
Obrázek 37: Standardní lišta - Práce se soubory

New Založení nového programu.

Open Otevření existujícího programu, resp. načtení ze souboru.

Save Uložení programu.

Sekce pro práci s elementy a podprogramy



Obrázek 38: Standardní lišta - Práce s elementy a podprogramy

Delete	Smazání označeného elementu, nebo jiného objektu v programu. Rychlejší je použití klávesy DELETE na klávesnici.
New sub	Založení podprogramu.
Copy	Vytvoření kopie právě editovaného podprogramu.
Delete	Smazání právě editovaného/nahlíženého podprogramu.

Sekce pro kompilaci a spuštění programu, propojení s řídící jednotkou a její otestování

Obrázek 39: Standardní lišta - Kompilace a spuštění programu, propojení s řídící jednotkou a její otestování

Start	Nahrání programu do řídící jednotky a jeho spuštění.
Stop	Zastavení programu právě běžícího v řídící jednotce.
Download	Stažení programu přímo do řídící jednotky. Při nahrávání programu lze vybrat, jestli se nahraje trvale, nebo do RAM řídící jednotky a zda se má po nahrání rovnou spustit, nebo jen načíst ovládací panel a zobrazit jej na displeji řídící jednotky.
Environment	Slouží pro nastavení správné řídící jednotky. My využíváme ROBO TXT/TXT Controller. Stejné jako v hlavní nabídce.
Bluetooth	Umožňuje připojení k řídící jednotce přes Bluetooth. Stejné jako v hlavní nabídce.
COM/USB	Umožňuje navolit, k jakému portu a jakým způsobem má PC s řídící jednotkou komunikovat a jaký typ řídící jednotky je využíván. V našem případě je řídící jednotkou ROBO TXT/TXT Controller.

TEST Umožňuje otestovat propojení a jednotlivé zapojené komponenty. Může se tak vyzkoušet před spuštěním programu, že komponenty jsou zapojeny správně a funkční.

Sekce debugeru

Obrázek 40: Standardní lišta - Debuger

Continue Spuštění programu v debugeru.

Pause Zastavení programu na právě vykonávaném elementu.

Step Přesun na další element v rámci diagramu.

Sekce ostatní

Obrázek 41: Standardní lišta - Ostatní

Zoom out Přiblížení/zazoomování v rámci programu.

Zoom in Oddálení/odzoomování v rámci programu.

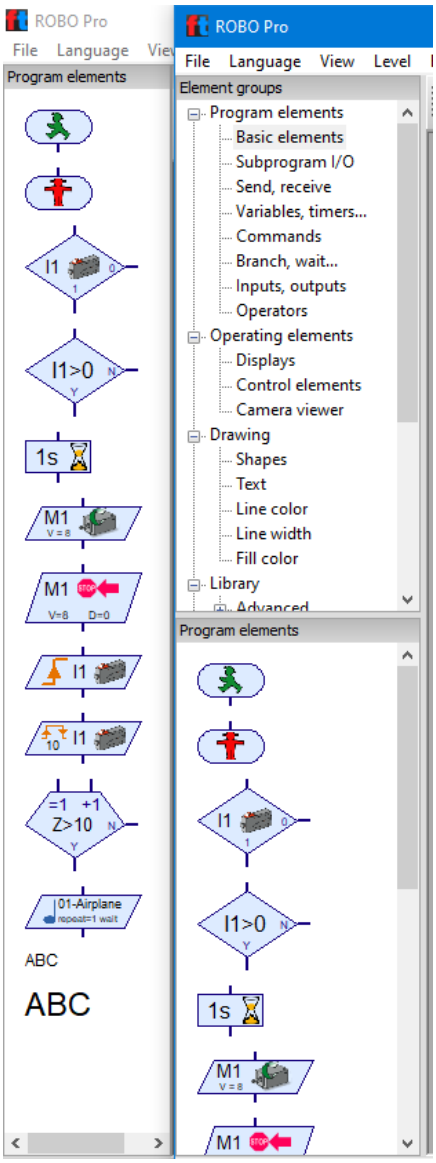
Unpair Odpárování řídící jednotky připojené přes Bluetooth.

2.3.3 Boční panel

V bočním panelu se zobrazuje nabídka elementů a objektů pro tvorbu programu, které se následně myší přetahují do okna programu. Zde je potřeba myslet na to, že ikony, které se zobrazují, závisí na úrovni zvolené v hlavní nabídce v menu Level. Úrovně jsou:

- Level 1: Beginners
- Level 2: Subprograms
- Level 3: Variables
- Level 4: Custom commands
- Level 5: Objects

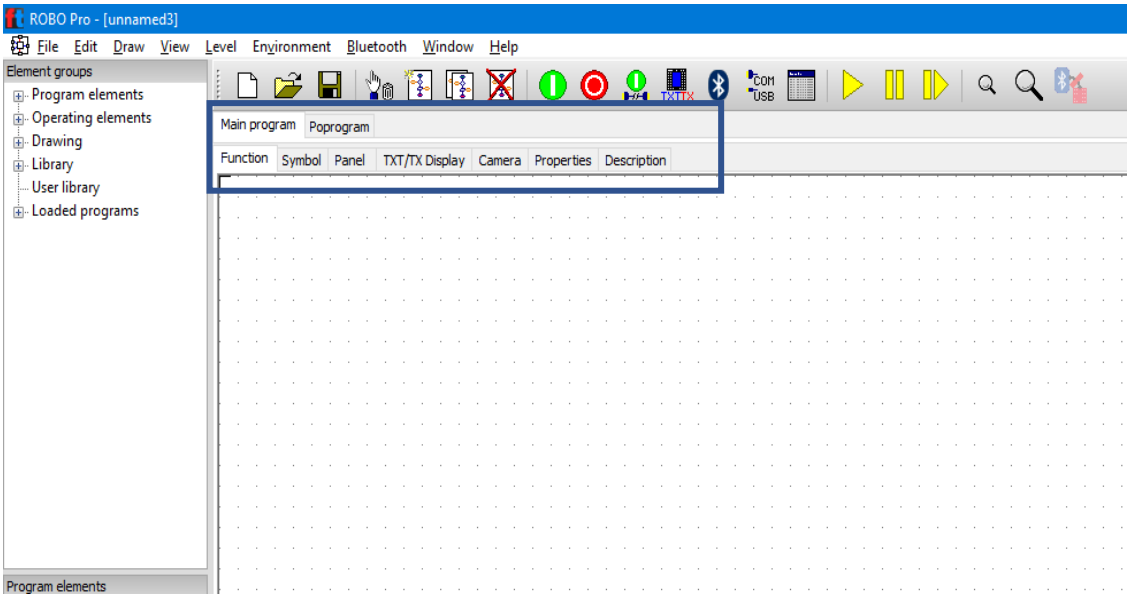
Platí, že vždy následující Level obsahuje nabídku všech předchozích Levelů. Pokud se tímto programátor nechce zatěžovat, může si vždy zvolit rovnou Level 5 a tím má přístup k veškeré funkcionalitě. Smysl Levelů je v tom, že začátečník, který prochází tutoriály od FischerTechnik, si zvolí potřebný Level a není matený nabízenou funkcionalitou, která je v daném případě nepotřebná a třeba i nad aktuální znalosti začátečníka.



Obrázek 42: Porovnání nabídky bočního panelu pro Level 1:Beginners a Level5: Objects

2.3.4 Okno nového programu

Po založení nového programu, nebo načtení souboru s programem se otevře okno programu, které se skládá ze dvou řad záložek, kde první řada záložek slouží k zobrazení programu/podprogramu a druhá řada záložek slouží jako pracovní prostory pro práci s programem/podprogramem.



Obrázek 43: Detal programu s jedním založeným podprogramem

Function — Slouží k programování programu, nebo podprogramu formou tvorby vývojových diagramů.

Symbol — Záložka umožňuje programátorovi nadefinovat symbol pro podprogram.

Panel — Vytváření ovládacího, nebo informačního panelu.

TXT/XT Display — Návrhu a tvorba ovládacího GUI, který se při nahrání do řídící jednotky zobrazí na dotykovém displeji řídící jednotky.

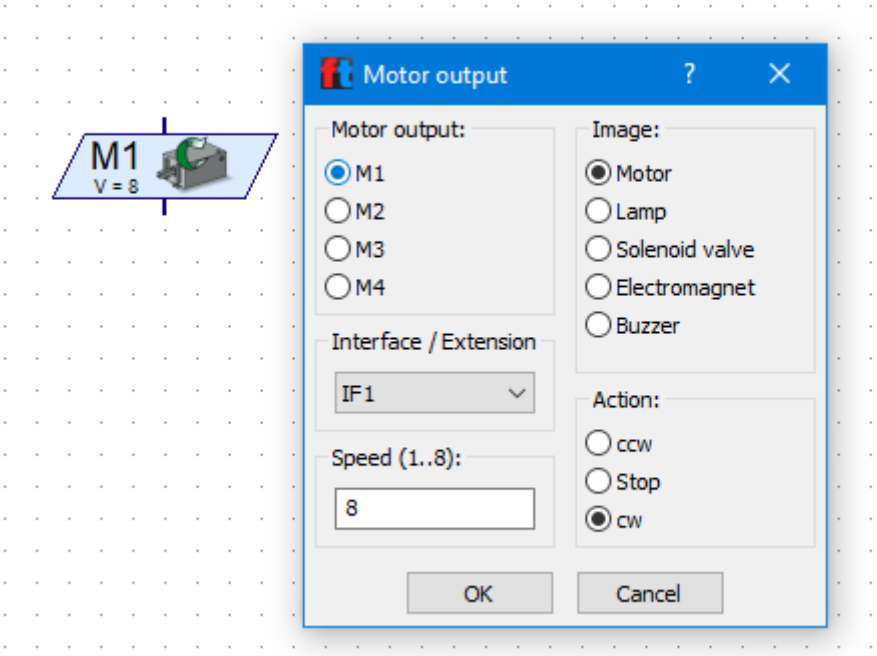
Camera — Zobrazení dat z USB kamery a jejím nastavení (počet snímků, nastavení vertikálního/horizontálního obrazu apod.)

Properties — Pokročilejší správa programu/podprogramu. Lze definovat název programu, počet procesů, přidělení paměti procesům apod.

Description — slouží jako poznámkový blok. Např. k přesnému popisu programu, částí programu, a jiným poznámkám…

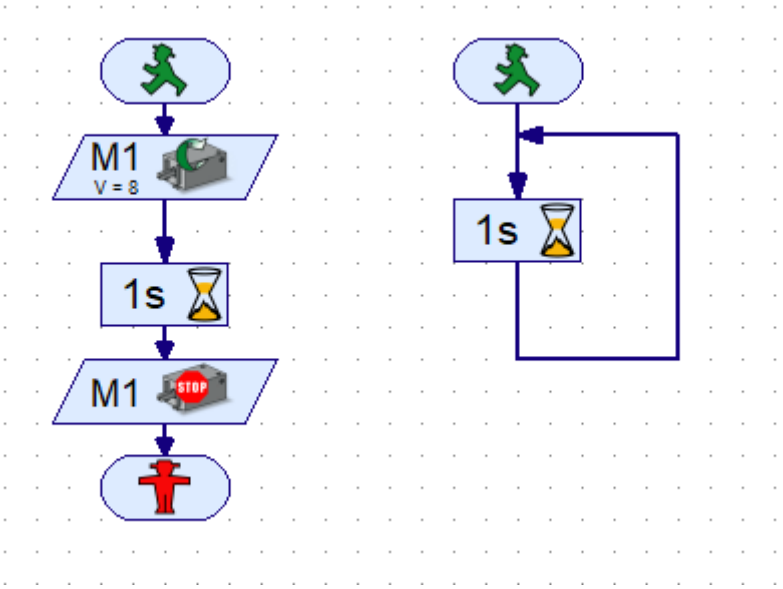
2.3.5 Tvorba programu v ROBO Pro

Programování v ROBO Pro probíhá vytvářením vývojových diagramů. Program se tedy skládá z komponent dostupných v bočním panelu. Přes pravé tlačítko myši lze vloženým elementům měnit vlastnosti. Prakticky např. při vložení elementu M1 lze přes pravé tlačítko myši nastavit, jak se bude chovat (motor, světlo, ventil, …), na jakém výstupu dochází k jeho napájení a řízení, např. v případě motoru i to, jak se bude chovat. Pro motor typu Mini, nebo XS lze nastavit rychlost otáček v rozsahu 1 – 8 a směr otáček, popř. zastavení motoru.



Obrázek 44: Nastavení vlastností komponenty vyvolané pravým tlačítkem myši nad komponentou

Takto vložené elementy se pak propojují v program. Pokud elementy přiblížíme myší k sobě, tak je program propojí automaticky, pokud ne, tak je lze propojit manuálně chycením levým tlačítkem myši na vyvedeném pinu a tažením myši na potřebné místo. Takto lze propojit vyvedený pin komponenty kamkoliv, včetně jiného propojení.



Obrázek 45: Jednoduchý program – vložené a propojené elementy (spuštění motoru na 1 sekundu), nekonečný program s cyklem který na základě 1 sekundy generuje věčnost.

Jakýkoliv element, nebo blok elementů, včetně propojení lze myší označit a přesunout/smazat popř. zkopírovat buď klávesovou zkratkou CTRL-C/CTRL-V, nebo jen držením CTRL a tažením myší. Při kopírování se elementy kopírují včetně nastavených vlastností.

3 Základní modely

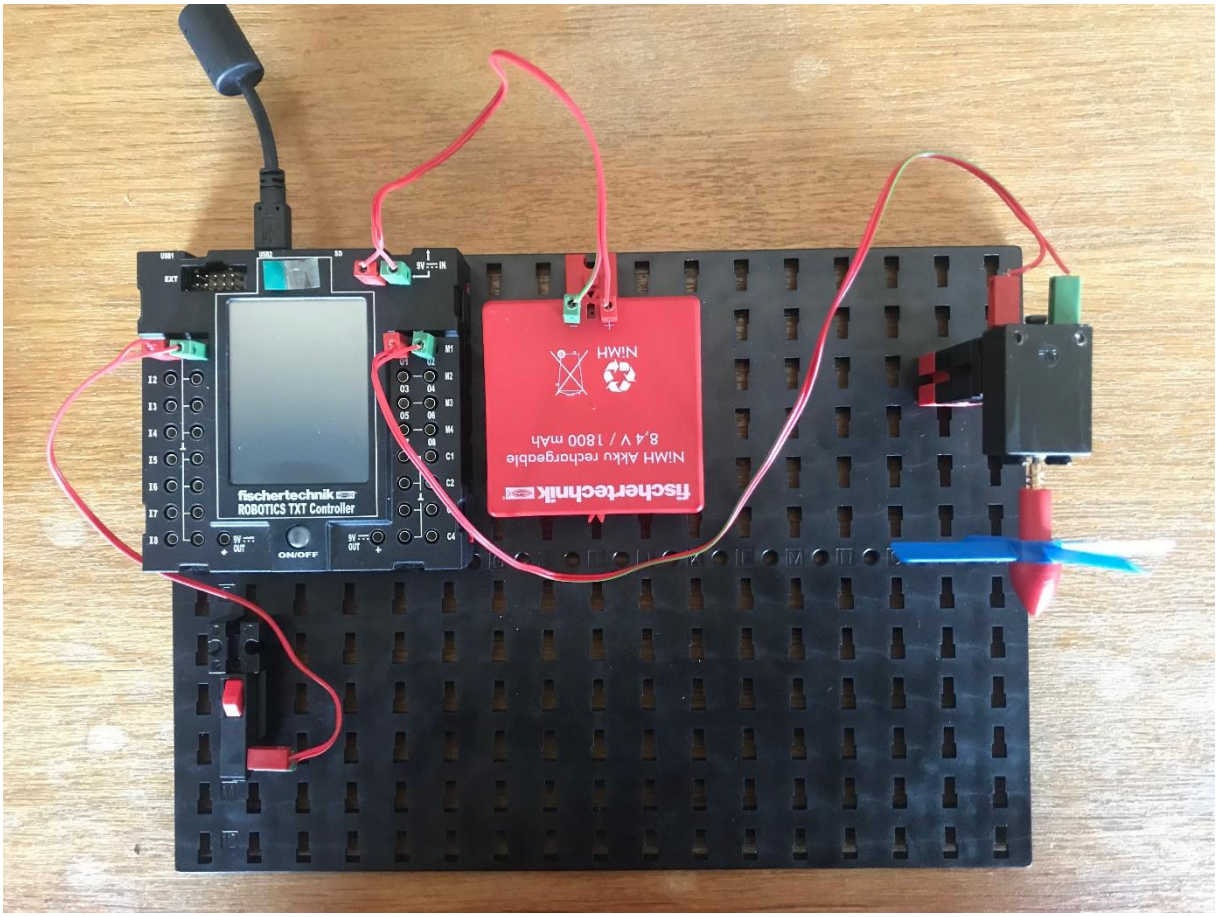
V následujících podkapitolách je na jednoduchých příkladech vysvětleno zapojení modelu a nastavení řídícího programu v ROBO Pro.

3.1 Větráček – základní podrobné cvičení

Větráček je velmi jednoduché zařízení s jedním spínačem a motorem, který otáčí vrtulí. Pro sestavní tedy potřebujeme právě tyto dvě komponenty a řídící jednotku. Kromě základního zapojení si ukážeme i nastavení komunikace mezi počítačem a řídící jednotkou, otestování komponent (bez potřeby software), vytvoření jednoduchého programu a jeho spuštění na modelu větráčku.

3.1.1 Větráček – zapojení komponent

Zapojení je v tomto případě triviální, na vstup I1 je přiveden spínač, na výstup M1 je přivedený motor větráčku, viz následující obrázek.



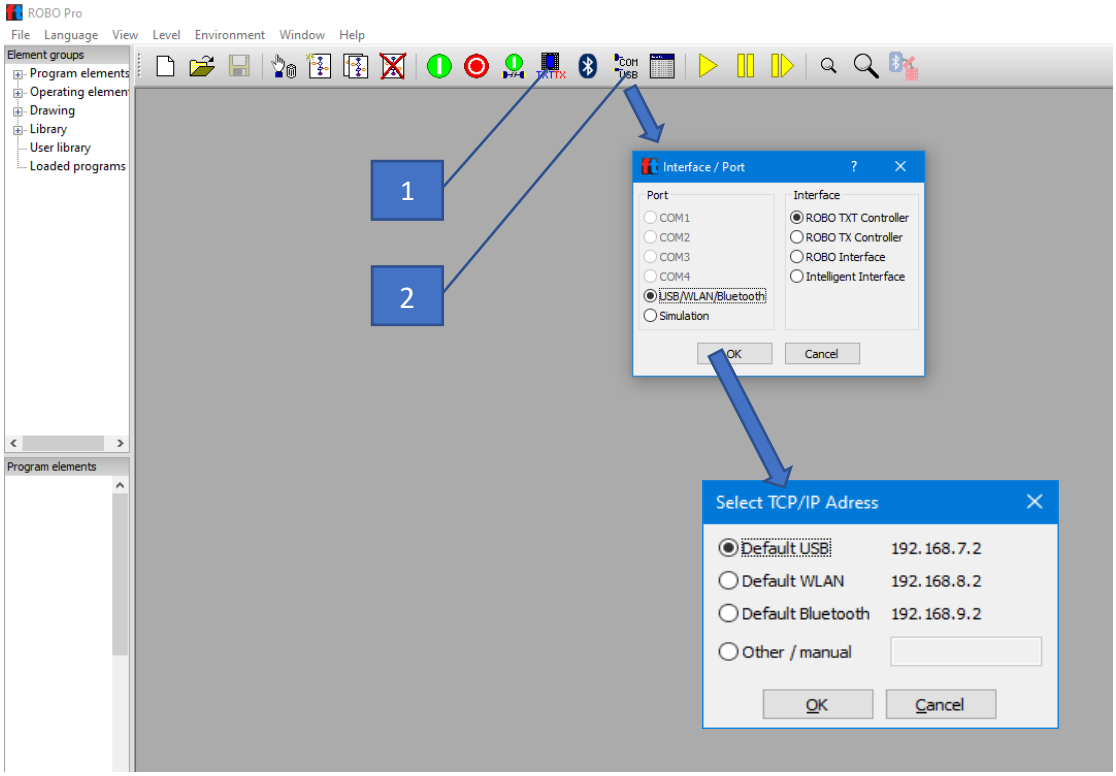
Obrázek 46: Větráček - zapojení komponent

3.1.2 Nastavení způsobu komunikace mezi počítačem a řídící jednotkou

Při propojení řídící jednotky s počítačem a jejím zapnutí je vhodné zkontrolovat, že je nastaven správný způsob komunikace:

1. V horní liště musí být ikona TXTTX (vždy tam bude ikona TXTTX, nebo ROBOIF). Pokud je ROBOIF, je potřeba kliknutím a něj se přepnout na TXTTX. Toto je nastavení správné řídící jednotky, kterou využíváme my.
2. Následně přes ikonou COM/USB se rozklikne seznam způsobů propojení – Zde musíme mít zvolený požadovaný způsob (v našem případě USB/WLAN/Bluetooth a Interface ROBO TXT

Controller (opět námi používaná řídící jednotka)). Po potvrzení je následně potřeba ještě zvolit, zda komunikace bude probíhat přes USB, WLAN, nebo Bluetooth. V našem případě používáme USB.

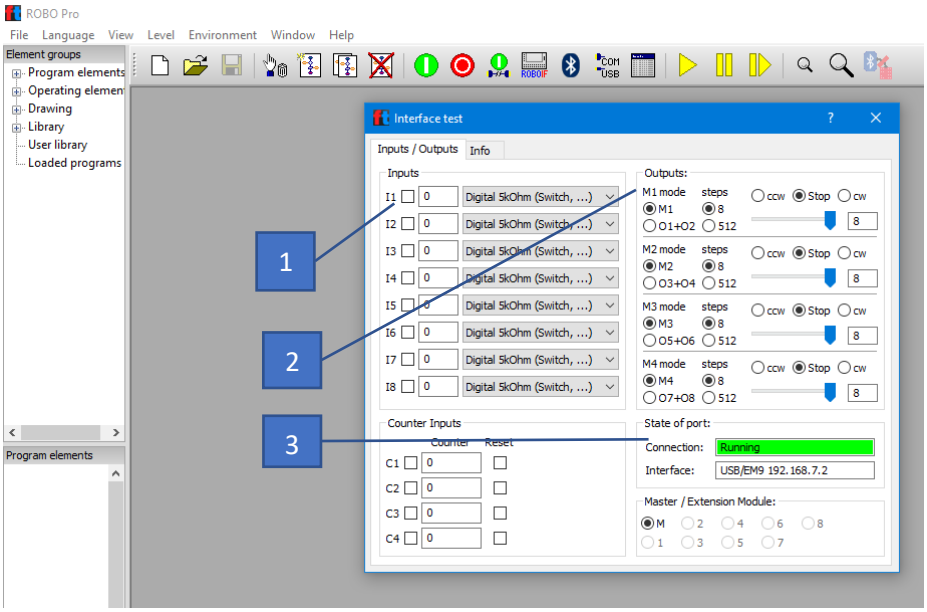


Obrázek 47: Nastavení způsobu komunikace mezi počítačem a řídící jednotkou

3.1.3 Otestování funkčnosti komunikace a komponent

Na tomto případě si ukážeme, jak lze otestovat, že propojení PC – řídící jednotka je funkční a zda komponenty jsou funkční a zapojeny takovým způsobem, aby mohly fungovat. Pro otestování není potřeba vytvářet jakýkoliv program. Okno kontroly je přístupné přes ikonu Test interface v horní liště, hned vedle ikony COM/USB. V otevřeném okně je možné si otestovat jakýkoliv vstup/výstup. Protože my používáme vstup I1 pro tlačítko a výstup M1 pro motor, tak test vypadá následovně:

1. Pokud je vše v pořádku, tak při stisknutí tlačítka na modelu se na vstupu I1 v okně testu interface přepíná hodnota z 0 na 1.
2. Pokud je vše v pořádku, tak u výstupu M1 si můžeme manuálně zapnout motor modelu v libovolném směru a libovolné intenzitě.
3. Vizuálně si lze zkontrolovat, jestli zvolený způsob komunikace je aktivní.

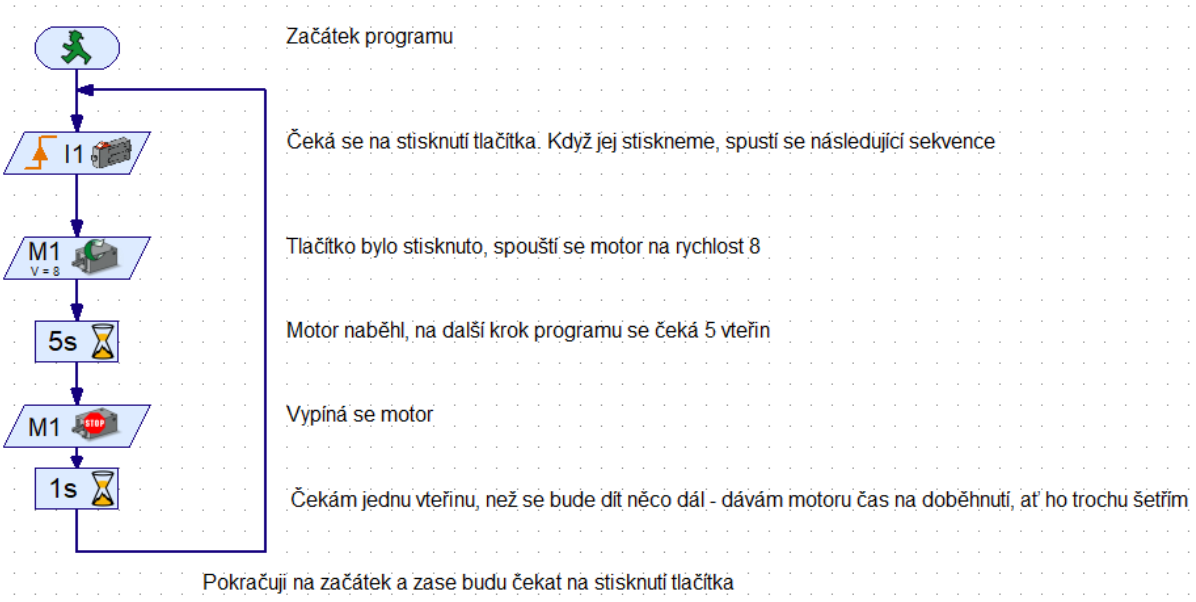


Obrázek 48: Otestování komunikace s modelem a funkčnosti komponent

Stejným způsobem by se testovaly i další zapojené komponenty a lze tak otestovat vše, co je k řídí jednotce připojeno.

3.1.4 Program pro řízení větráčku

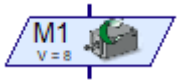
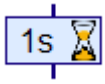
Cílem cvičení je oživit model takovým způsobem, aby po stisknutí tlačítka se větráček roztočil na maximální možnou rychlost, běžel 5 vteřin a po této době se vypl a čekal na další stisknutí tlačítka. Program tedy může vypadat např. následovně:



Obrázek 49: Větráček – řídící program

Přehled použitých elementů je v následující tabulce.

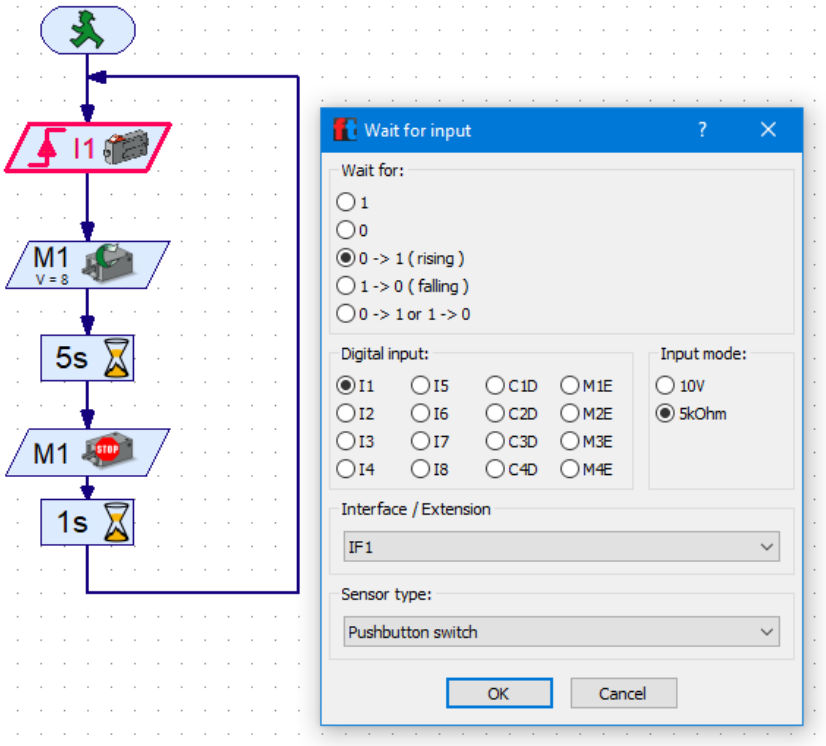
Tabulka 1: Větráček - přehled použitých elementů

Použitý element	Základní zobrazení elementu	Počet	Umístění elementu v nabídce	Připojení na vstup (Input), výstup (Output/Motor)	Poznámka
Start		1x	Program elements – Basic elements	-	-
Wait for input		1x		I1	-
Motor output		2x		M1	1x start, 1x stop
Time delay		2x		-	-

Dále si popíšeme nastavení jednotlivých elementů.

3.1.4.1 Wait for input

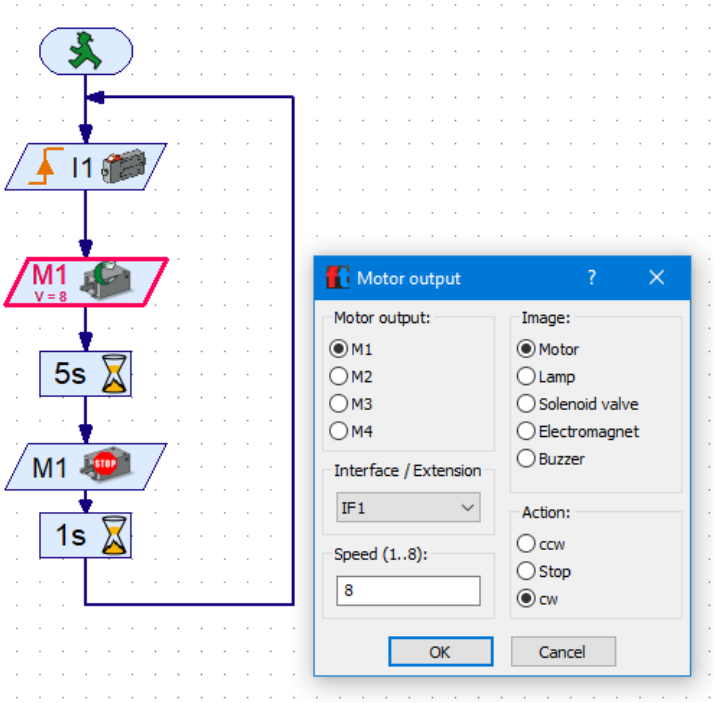
Jedná se o element ze sekce Basic elements. Tímto elementem říkáme, že čekáme na stisknutí tlačítka přivedeného na I1. Po jeho stisknutí proběhne následující sekvence.



Obrázek 50: Větráček - nastavení elementu Wait for input

3.1.4.2 Motor output

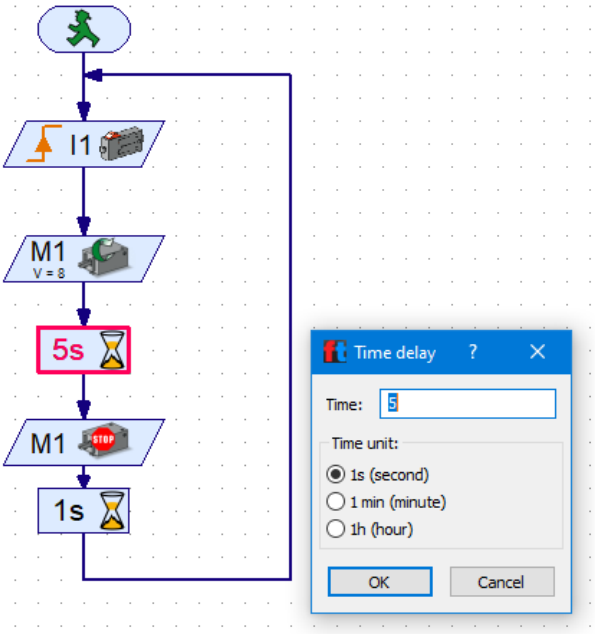
Jedná se o element ze sekce Basic elements. Tímto elementem říkáme, že spustíme motor na výstupu M1 na nejvyšší rychlost 8.



Obrázek 51: Větráček - nastavení elementu Motor output

3.1.4.3 Time delay

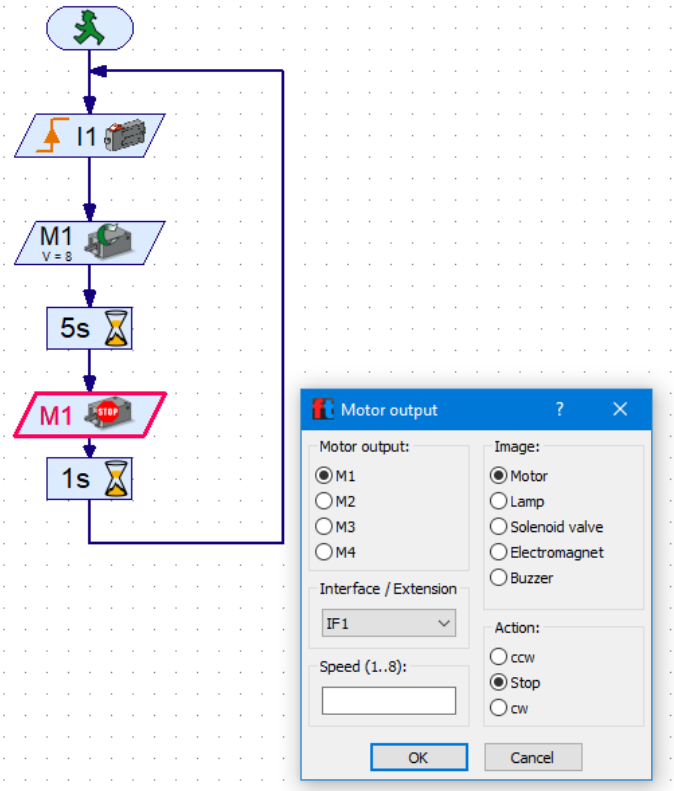
Jedná se o element ze sekce Basic elements, ale najdete jej i v sekci Branch, wait… - je to jeden a ten samý element. Tímto elementem říkáme, že následující element provede svou práci až po uběhnutí nastaveného času. V tomto případě motor poběží 5s a až poté se provede jeho zastavení vhodným elementem.



Obrázek 52: Větráček - nastavení elementu Time delay

3.1.4.4 Motor output II

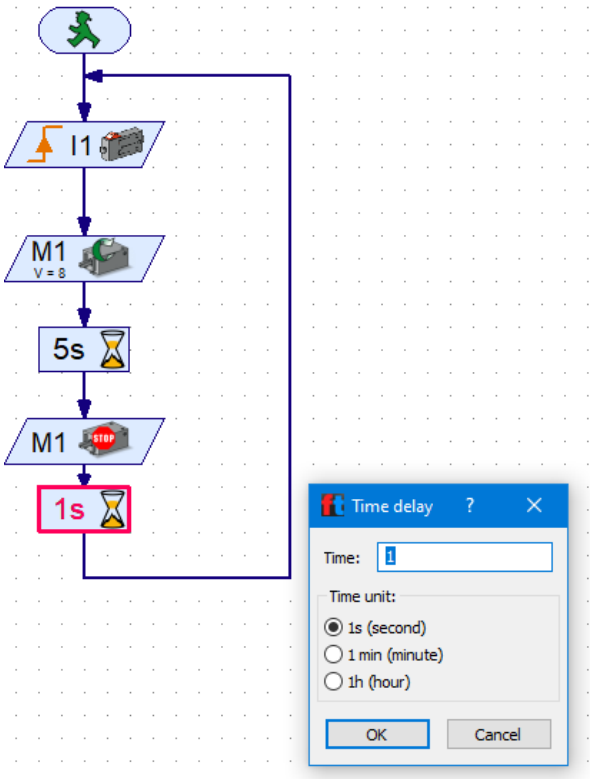
Jedná se o element ze sekce Basic elements. Tímto elementem říkáme, že zastavujeme motor.



Obrázek 53: Větráček - nastavení elementu Motor output II

3.1.4.5 Time delay II

Jedná se o element ze sekce Basic elements, ale najdete jej i v sekci Branch, wait… - je to jeden a ten samý element. Tímto elementem říkáme, že následující element provede svou práci až po uběhnutí nastaveného času. V tomto případě motor necháváme motoru čas 1s na doběhnutí. Tedy 1s bude trvat, než program dovolí zaregistrovat další stisknutí tlačítka a znovuspuštění celého cyklu. Je to jen proto, abychom zajistili, že motor bude mít dost času na doběhnutí a uklidnění.



Obrázek 54: Větráček - nastavení elementu Time delay II

3.1.5 Zkompilování programu a nahrání do řídící jednotky

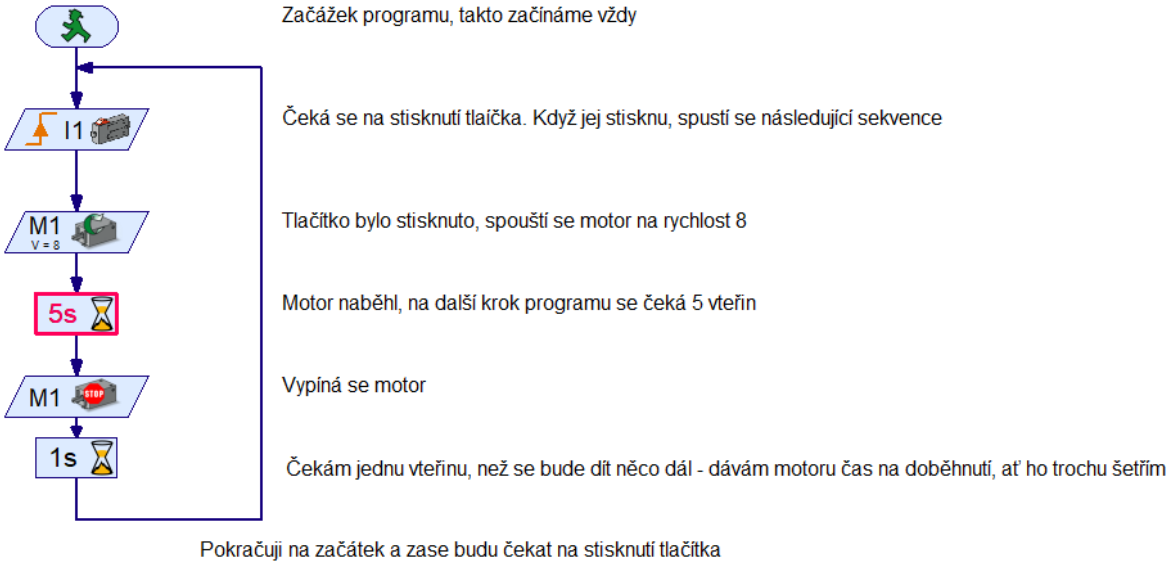
Když je hotový program, aktivní připojení k řídící jednotce a připojený model, stačí přes zelené tlačítko (Start program in online mode) nechat program zkompilovat a nahrát do řídící jednotky. Zastavení programu se provede červeným tlačítkem (Stop all running programs). Toto platí pro online komunikaci. Pokud je chtěno nahrát program do řídící jednotky, aby byl použitelný i po odpojení PC od řídící jednotky, je potřeba použít zelené tlačítko Download.

Když se program zkompiluje a nahraje (v tomto případě v online modu), zobrazí se na displeji řídící jednotky informace, že je nahraný online program, tedy nápis ONLINE a program lze využít, tedy stisknout tlačítko a nechat se chladit.

Obrázek 55: Obrazovka řídící jednotky - program je zkompilován a nahrán do řídící jednotky

Pokud je program spuštěný v online modu, je možné sledovat na monitoru počítače, v které fázi programu se zrovna model nachází. Právě probíhající část programu je vždy zvýrazněna červeně, viz

následující obrázek. Také je dobré si všimnout, že když je program aktivní, zmizí rastr pracovní plochy a je to tedy na první pohled poznat.



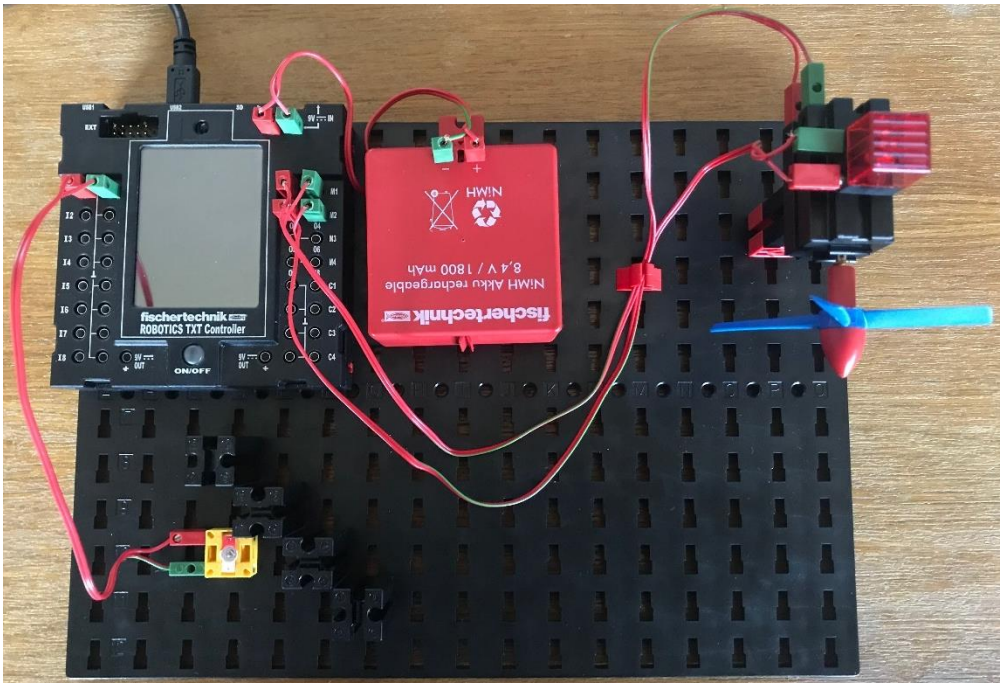
Obrázek 56: Puštěný program na modelu v online modu - zrovna probíhá 5s běhu větráčku

3.2 Předělání větráčku na model větrné elektrárny

V tomto cvičení předěláme větráček na model větrné elektrárny. Ukážeme si, jak se při tom pracuje s optickými moduly, tedy žárovkou/led a světelným senzorem. V programování si ukážeme jak se program větví. Pro sestavení modelu je potřeba motor, světelný senzor a žárovka. Velkoryse přejdeme, že elektromotor otáčí vrtulí, a ne vrtule elektromotorem – je to jen model.

3.2.1 Větrná elektrárna – zapojení komponent

Zapojení komponent je následující, viz obrázek. Na vstup I1 je přivedený světelný senzor, na výstup M1 je přivedený motor s vrtulí a na výstup M2 je přivedené světlo (s červenou krytkou).



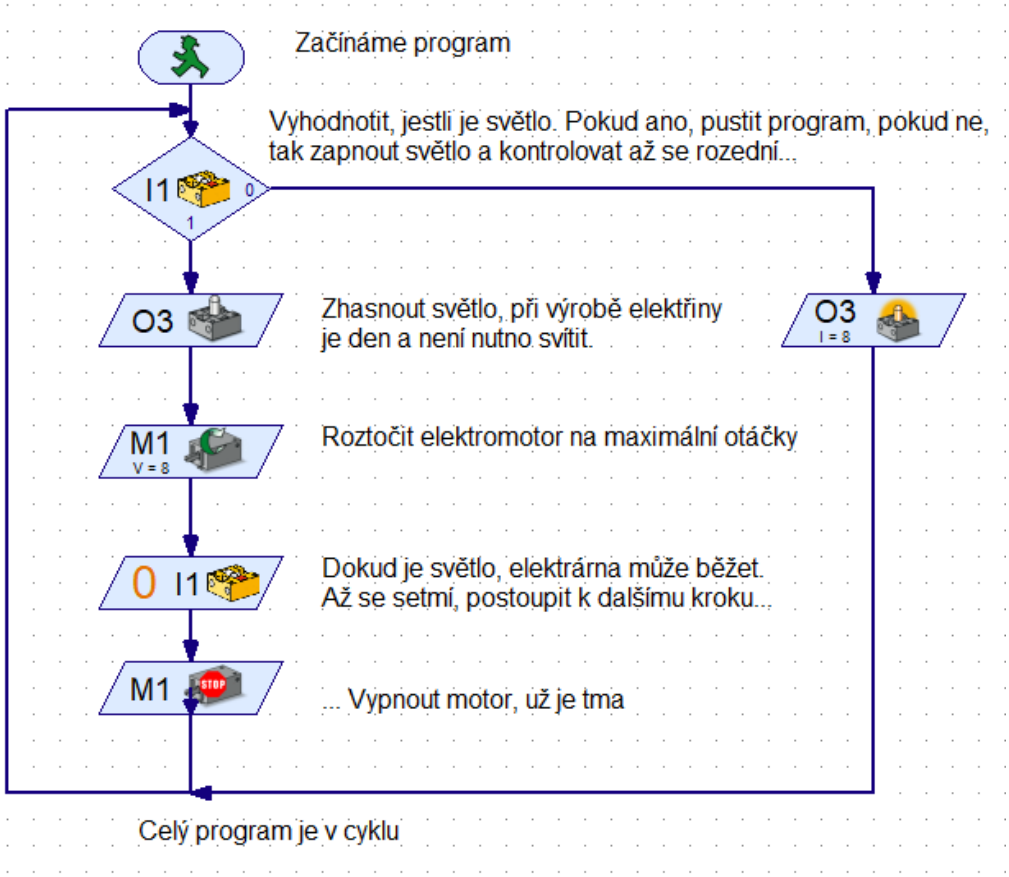
Obrázek 57: Větrná elektrárna – zapojení modelu

3.2.2 Větrná elektrárna - řídící program

Jak chceme, aby se větrná elektrárna chovala…

1. V noci bude svítit červené světlo, aby nám do ní nenarážela letadla. Vrtule bude v klidu, nechceme budit blízko bydlící občany.
2. Ve dne budeme šetřit světlo, letadla uvidí větrnou elektrárnu i bez něj, a vrtule se může točit.
3. Toto chování chceme mít uzavřené v cyklu, aby se vše spouštělo a vypínalo automaticky a my se o větrnou elektrárnu nemuseli denně starat.

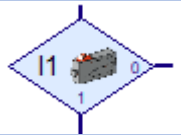
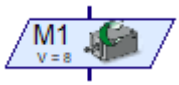
Možnost, jak může vypadat program je na následujícím obrázku. Ze stejných součástek by šel vytvořit např. i sušák rukou, kdy by světelný senzor vyhodnocoval, jestli vidí světlo ze žárovky, nebo nevidí – tedy jsou mezi ním a žárovkou vloženy ruce a má se spustit jejich sušení.



Obrázek 58: Větrná elektrárna - řídící program

Přehled použitých elementů je v následující tabulce.

Tabulka 2: Větrná elektrárna - přehled použitých elementů

Použitý element	Základní zobrazení elementu	Počet	Umístění elementu v nabídce	Připojení na vstup (Input), výstup (Output/Motor)	Poznámka
Start		1x	Program elements – Basic elements	-	-
Wait for input		1x		I1	Nastaveno pro fototranzistor
Digital Branch		1x		I1	Nastaveno pro fototranzistor
Lamp output		2x		O3	1x on, 1x off
Motor output		2x		M1	1x start, 1x stop

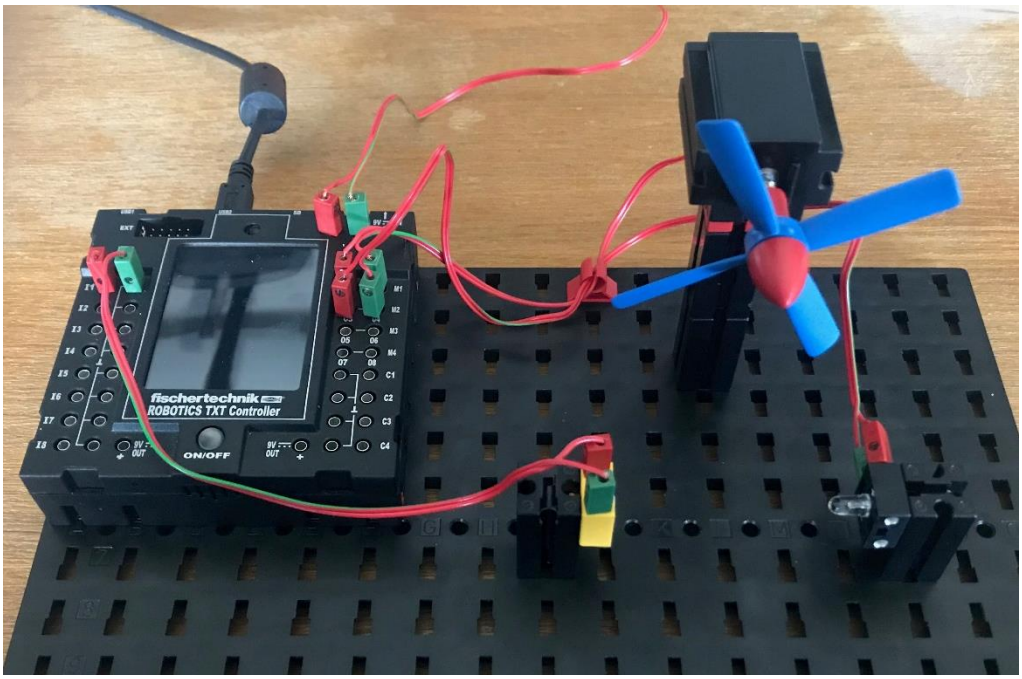
3.3 Vysoušeč rukou – princip světelné brány

V tomto cvičení využijeme části předchozích cvičení. Úkolem je vytvořit vysoušeč rukou, tedy zařízení, kde je světelná brána rozhodující o spuštění/nespuštění ventilátoru. Na cvičení se v tomto případě pouze lehce modifikuje model využitý v předchozím cvičení.

Světelná brána je využití komponenty emitující světlo (v tomto případě např. žárovka, nebo LED) na jedné straně a fototranzistor, který vyhodnocuje, zda na něj emitované světlo ze světelné komponenty dopadá, nebo ne, na straně druhé. Pokud světlo na fototranzistor dopadá, je větrák v klidu. Pokud ne, tedy mezi světelnou komponentu a fototranzistor jsou vloženy ruce, větrák se na určitou dobu sepne.

3.3.1 Vysoušeč rukou – zapojení komponent

Zapojení komponent je následující, viz obrázek. Na vstup I1 je přivedený fototranzistor (světelný senzor), na výstup M1 je přivedený motor s vrtulí a na výstup O3/O4 je přivedené světlo, v tomto případě LED.

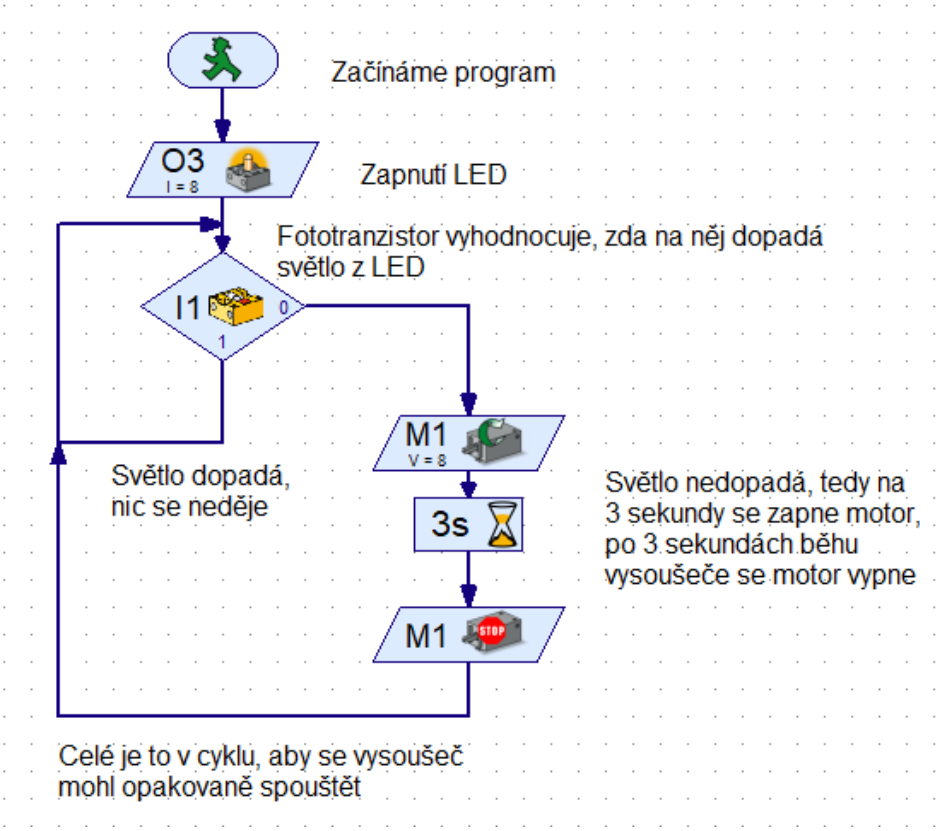


Obrázek 59: Vysoušeč rukou - zapojení komponent

3.3.2 Vysoušeč rukou – řídící program

Jak chceme, aby se vysoušeč choval…

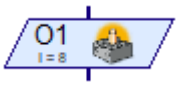
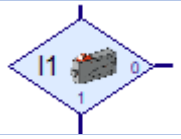
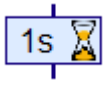
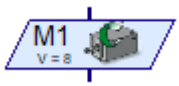
1. Pokud není mezi LED komponentou a fototranzistorem překážka, tak je vysoušeč v klidu
2. Pokud se mezi LED komponentu a fototranzistor vloží překážka (ruce), vysoušeč se na 3 sekundy spustí a bude vysoušet ruce.
3. Program bude v cyklu, tedy kdykoliv se vloží překážka mezi LED a fototranzistor, tak se vysoušeč opětovně zapne.



Obrázek 60: Vysoušeč rukou - řídící program se světelnou bránou

Přehled použitých elementů je v následující tabulce.

Tabulka 3: Vysoušeč rukou - přehled použitých elementů

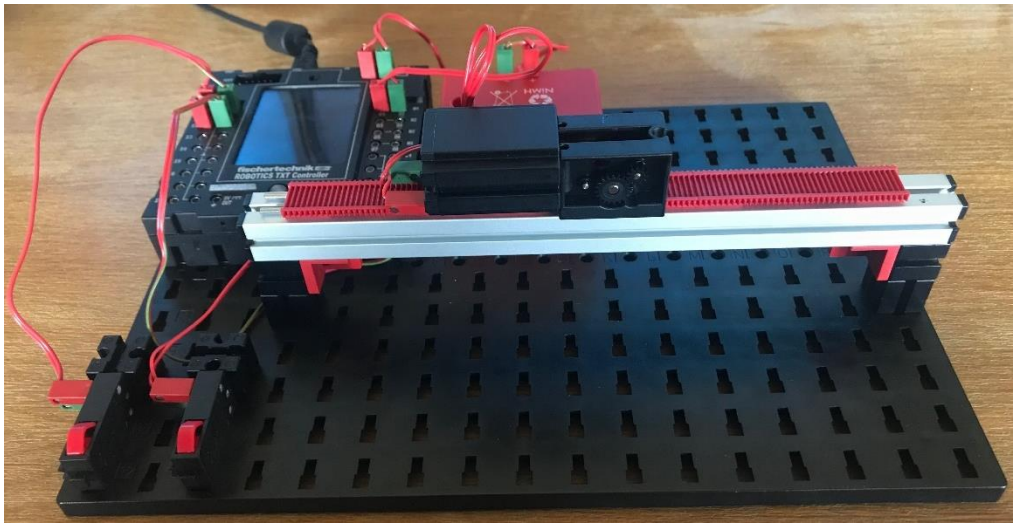
Použitý element	Základní zobrazení elementu	Počet	Umístění elementu v nabídce	Připojení na vstup (Input), výstup (Output/Motor)	Poznámka
Start		1x	Program elements – Basic elements	-	-
Lamp output		1x		O3	-
Digital Branch		1x		I1	Nastaveno na fototranzistor
Time delay		1x		-	-
Motor output		2x		M1	-

3.4 Manuálně ovládaný pojezd

V tomto cvičení si zopakujeme vyhodnocování podmínek (větvení programu). Cílem je mít manuálně ovládaný pojezd. Model se skládá ze dvou tlačítek a motoru. Motor je spojen s převodovkou, která přenáší otáčky na výstupu elektromotoru na ozubené kolečko, které je v kontaktu s ozubeným pásem pevně spojeným s nosníkem.

3.4.1 Manuálně ovládaný pojezd – zapojení komponent

Zapojení komponent je následující, viz obrázek. Na vstupy I1 a I2 jsou přivedeny spínače, na výstup M1 je přivedený motor zajišťující posuv pojezdu.



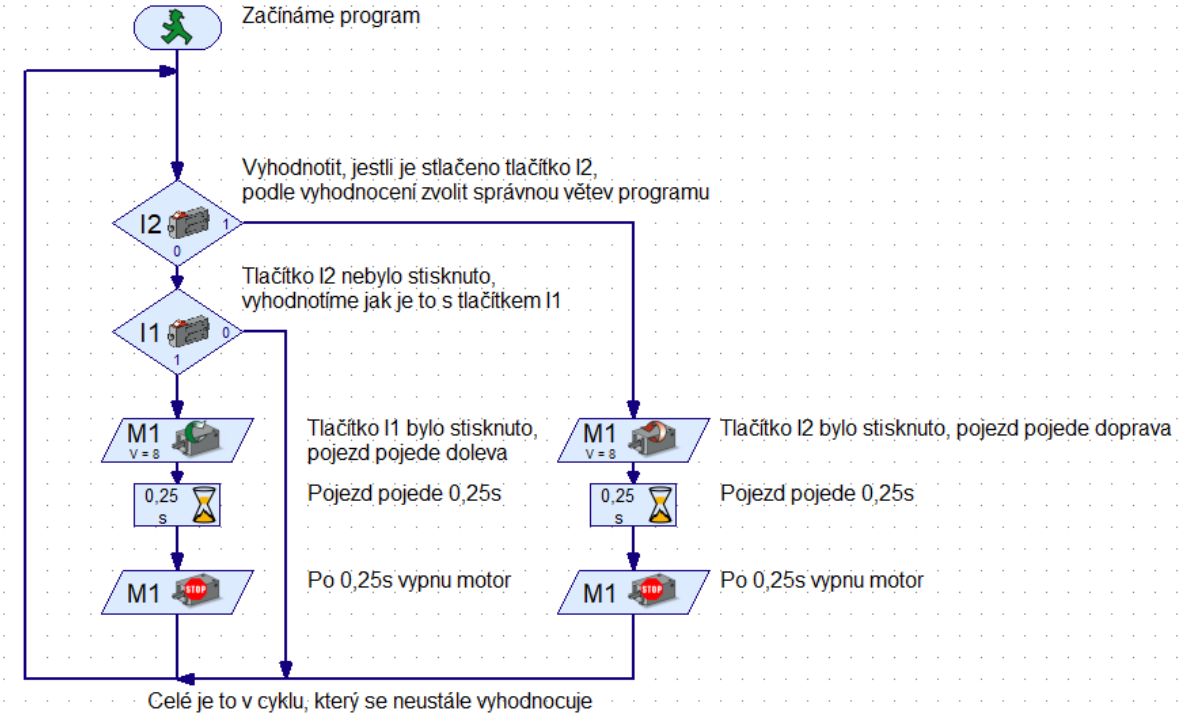
Obrázek 61: Manuálně ovládaný pojezd – zapojení modelu

3.4.2 Manuálně ovládaný pojezd - ovládací program

Jak chceme, aby se pojezd choval…

1. Při stisknutí levého tlačítka se pojezd bude pohybovat doleva.
2. Při stisknutí pravého tlačítka se pojezd bude pohybovat doprava.
3. Při držení tlačítka se pojezd bude pohybovat bez přerušení po dobu držení tlačítka.
4. Program bude v cyklu, tedy kdykoliv stisknu jakékoliv tlačítko, tak se pojezd bude pohybovat odpovídajícím směrem.

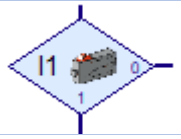
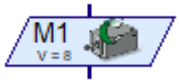
Zde je dobré upozornit na drobnou „vychytralost“. Pokud se má pojezd pohybovat po celou dobu držení tlačítka, pomůžeme si nastavením malé hodnoty na dobu běhu motoru. Čím menší doba běhu motoru bude, tím rychleji bude pojezd reagovat na stisknutí/nestisknutí tlačítka. Možnost, jak může vypadat program je na následujícím obrázku.



Obrázek 62: Manuálně ovládaný pojezd – řídící program

Přehled použitých elementů je v následující tabulce.

Tabulka 4: Manuálně ovládaný pojezd - přehled použitých elementů

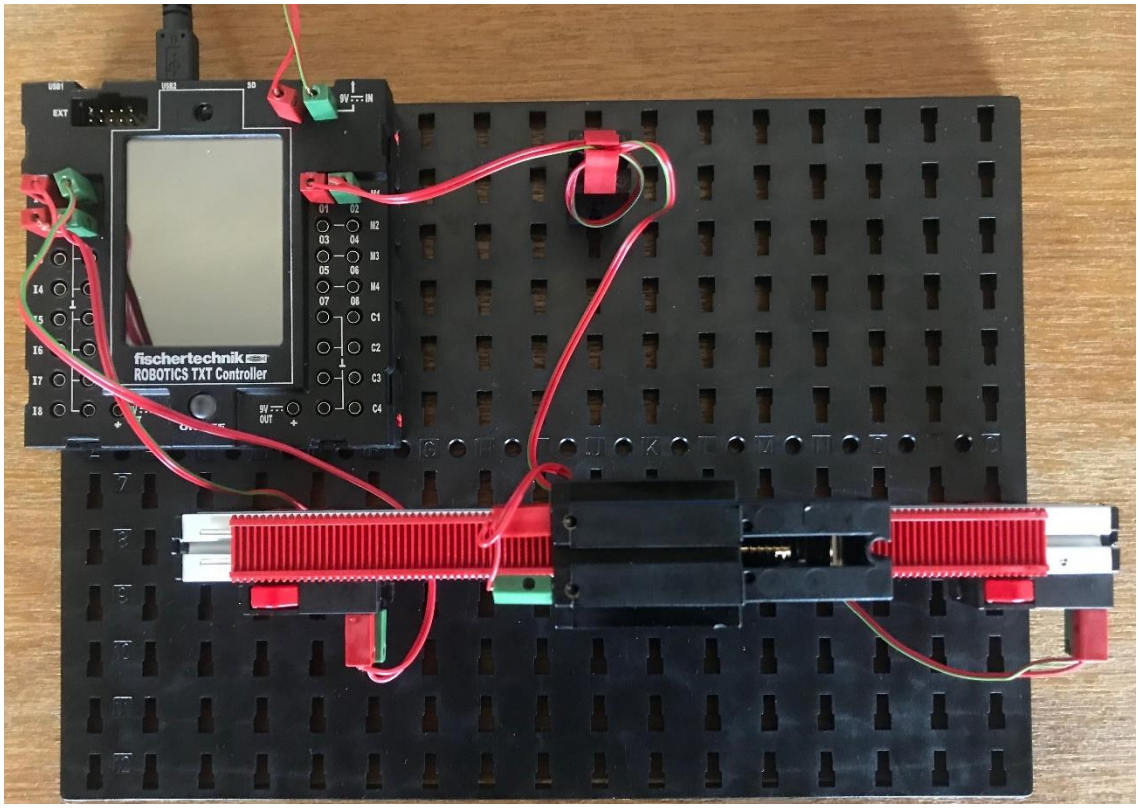
Použitý element	**Základní zobrazení elementu**	**Počet**	**Umístění elementu v nabídce**	**Připojení na vstup (Input), výstup (Output/Motor)**	**Poznámka**
Start		1x	Program elements – Basic elements	-	-
Digital Branch		2x		I1, I2	Pozor, 1x jsou v programu prohozeny výstupy
Time delay		2x		-	-
Motor output		4x		M1	-

3.5 Pojezd mezi pevně danými body

V tomto cvičení jednoduše modifikujeme předchozí model. Zde se naučíme vytvářet pro program a model ovládací panel, vytvářet a využívat podprogramy[2], pracovat s elementy typu Command, Panel displayem, rozhodováním na základě podmínky a zopakujeme si vyhodnocování podmínek (větvení programu), využijeme spínače jako skutečné koncové spínače. Cílem je mít pojezd, který po stisknutí tlačítka na ovládacím panelu (softwarové tlačítko) přesune pojezd na požadovanou pozici. Model se skládá ze dvou tlačítek a motoru. Motor je spojen s převodovkou, která přenáší otáčky na výstupu elektromotoru na ozubené kolečko, které je v kontaktu s ozubeným pásem pevně spojeným s nosníkem.

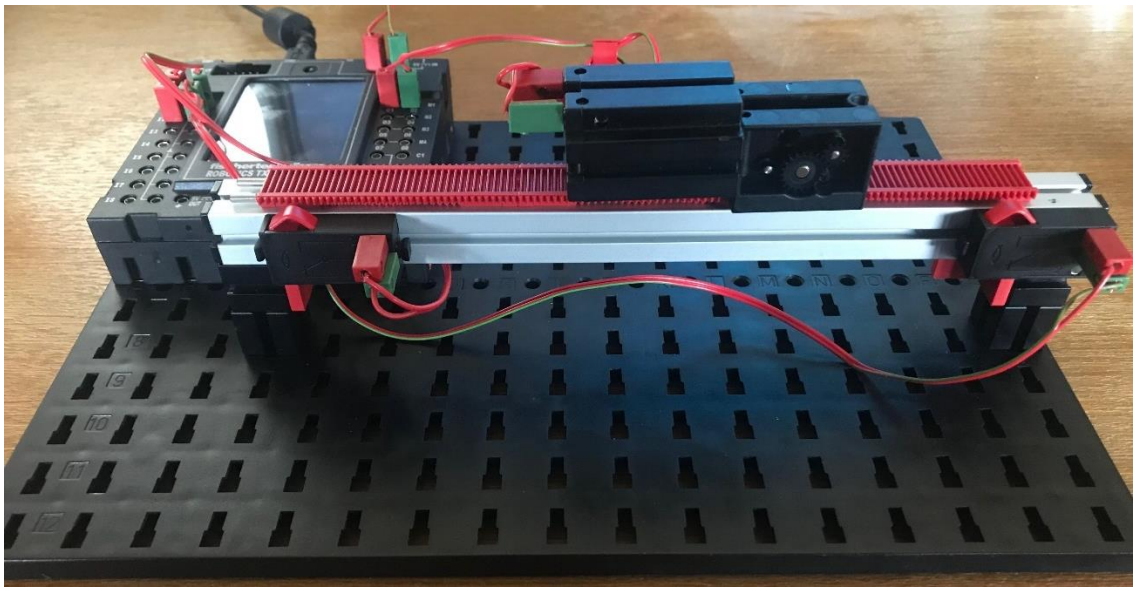
3.5.1 Pojezd mezi pevně danými body – zapojení komponent

Zapojení komponent je následující, viz obrázek. Na vstupy I1 a I2 jsou přivedeny koncové spínače (I1 spínač vlevo, I2 spínač vpravo), na výstup M1 je přivedený motor zajišťující posuv pojezdu.



Obrázek 63: Pojezd mezi pevně danými body - zapojení komponent, pohled shora

[2] „Když neodskočíš z cyklu programu, nikdy nebudeš v podprogramu.“ Kryton, Červený trpaslík, Spravedlnost



Obrázek 64Pojezd mezi pevně danými body - zapojení komponent, pohled zepředu

3.5.2 Pojezd mezi pevně danými body – řídící program

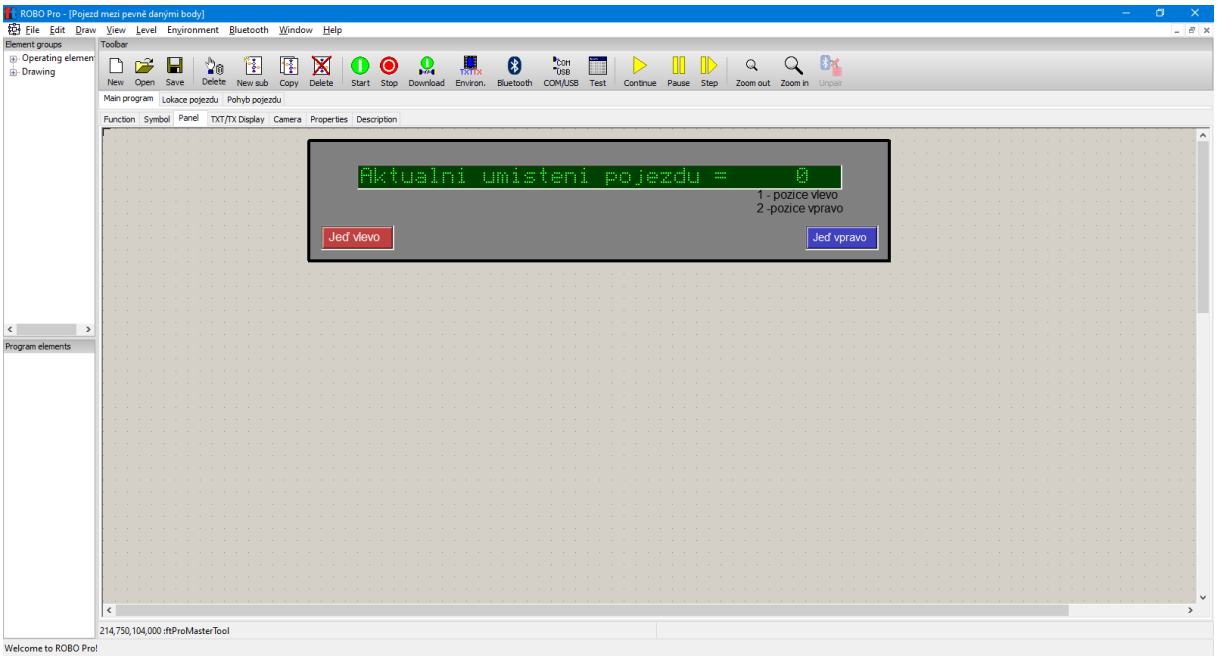
Jak chceme, aby se pojezd choval…

1. K ovládání chceme používat ovládací panel.
2. Při stisknutí vhodného softwarového tlačítka na ovládacím panelu se pojezd přesune ke koncovému spínači doleva.
3. Při stisknutí vhodného softwarového tlačítka na ovládacím panelu se pojezd přesune ke koncovému spínači doprava.
4. Program bude v cyklu, tedy kdykoliv stisknu jakékoliv softwarové tlačítko, tak se pojezd bude pohybovat odpovídajícím směrem a přesune se na požadované místo.
5. Na ovládacím panelu bude informace o tom, v které z koncových poloh se zrovna pojezd nachází.

V tomto případě už je program trochu komplikovanější. Bude potřeba vytvořit 2 podprogramy. Jeden pro pohybování vozíku, druhý pro hlídání lokace. Hlavní program bude využívat oba podprogramy a bude mít naprogramovaný ovládací panel. Možnost, jak může program a podprogramy vypadat jsou na následujících obrázcích.

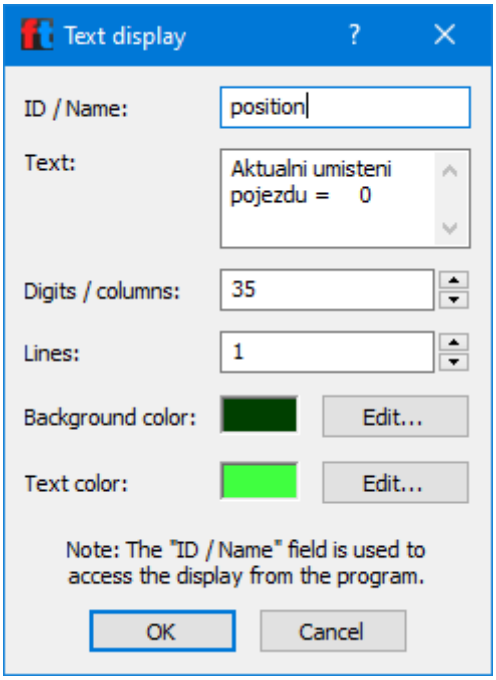
3.5.2.1 Ovládací panel

V hlavním programu na kartě Panel je potřeba vytvořit GUI ovládacího panelu. Přehled použitých elementů pro GUI je součástí přehledu použitých elementů pro hlavní program.



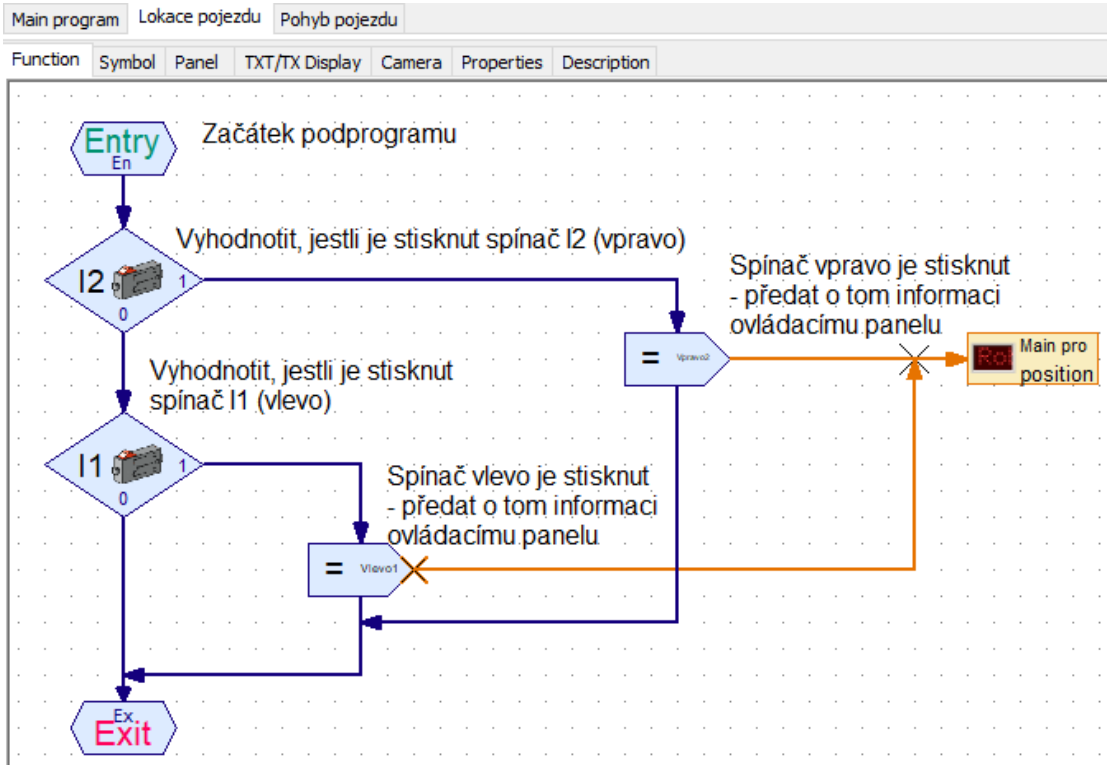
Obrázek 65: Ovládací panel - takto může vypadat

Text display – vyplníme ID displeje na „position“ (abychom se v programu vyznali) a nastavíme mu rozumný popisek (např. „Aktualni umisteni pojezdu“).



Obrázek 66: Nastavení Text displaye

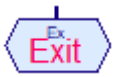
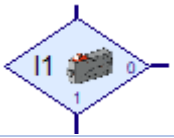
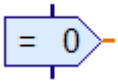
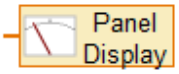
3.5.2.2 Podprogram lokace pojezdu

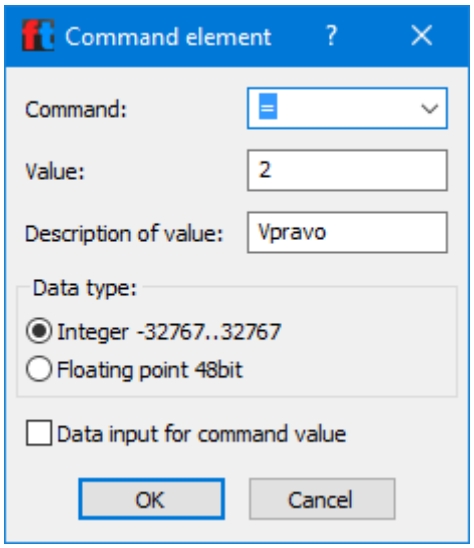


Obrázek 67: Podprogram lokace pojezdu

Přehled použitých elementů je v následující tabulce.

Tabulka 5: Pojezd mezi pevně danými body - podprogram lokace pojezdu - přehled použitých elementů

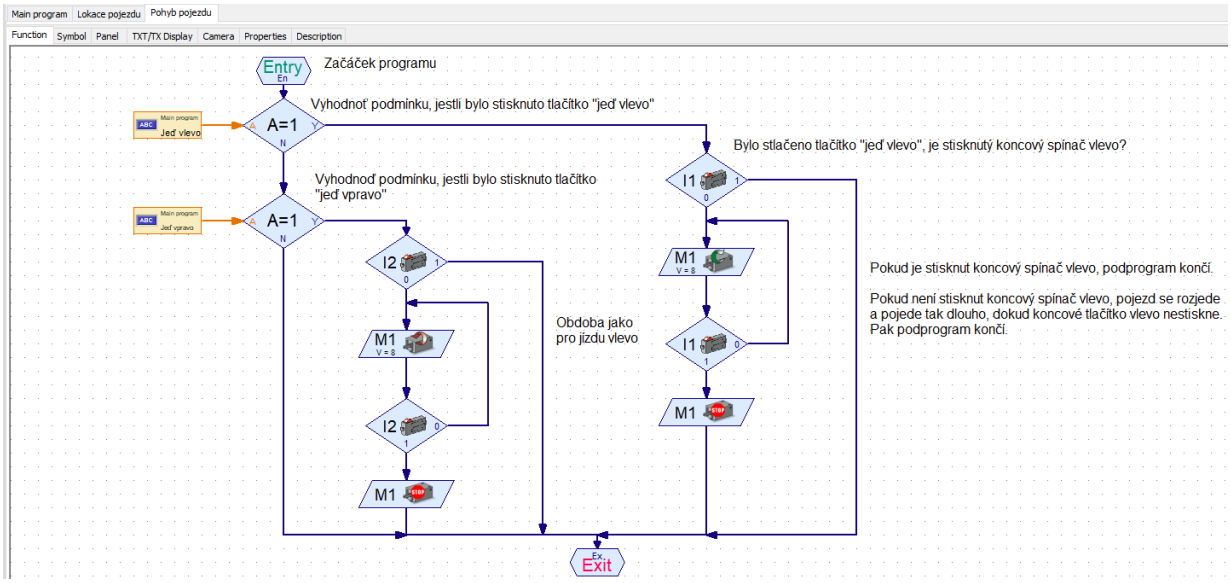
Použitý element	Základní zobrazení elementu	Počet	Umístění elementu v nabídce	Připojení na vstup (Input), výstup (Output/Motor)	Poznámka
Subprogram Entry		1x	Program elements – Subprogram I/O	-	-
Subprogram Exit		1x		-	-
Digital Branch		2x	Program elements – Basics elements	I1, I2	-
Assignment		2x	Program elements - Commands	-	-
Panel display		1x	Program elements – Inputs, outputs	-	-



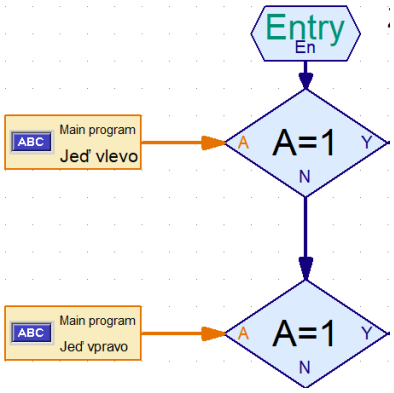
Obrázek 68: Ukázka nastavení stavu - stisknutí spínače

3.5.2.3 Podprogram pohybu pojezdu

Na následujícím obrázku je zobrazen podprogram pohybu pojezdu, kde je řešeno, co se stane, když zmáčkneme patřičné tlačítko pro pojezd.



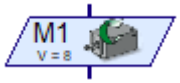
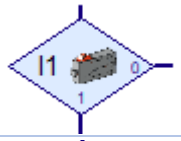
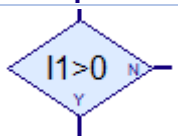
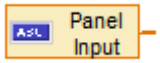
Obrázek 69: Podprogram pro pohyb pojezdu



Obrázek 70: Podprogram pro pohyb pojezdu – detail vstupu parametrů z hlavního programu do podprogramu

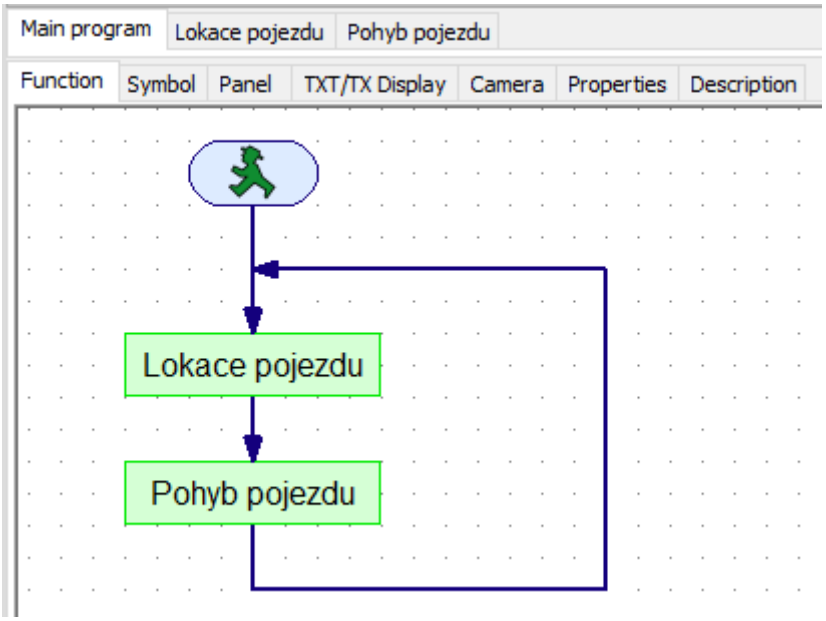
Přehled použitých elementů je v následující tabulce.

Tabulka 6: Pojezd mezi pevně danými body - podprogram pohybu pojezdu - přehled použitých elementů

Použitý element	Základní zobrazení elementu	Počet	Umístění elementu v nabídce	Připojení na vstup (Input), výstup (Output/Motor)	Poznámka
Subprogram Entry		1x	Program elements – Subprogram I/O	-	-
Motor output		4x	Program elements – Basic elements	M1	-
Digital Branch		4x		I1, I2	-
Analog branch		2x		-	Input value nastaveno na = 1 (čekáme na stisknutí)
Panel Input		2x	Program elements – Inputs, Outputs	-	-

3.5.2.4 Vytvoření těla hlavního programu

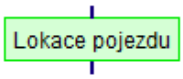
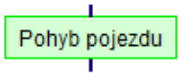
V tuto chvíli máme vytvořený ovládací panel pro ovládání (v hlavním programu) a 2 podprogramy. Poslední krok je podprogramy použít v těle hlavního programu, viz obr.:



Obrázek 71: Hlavní program s využitím podprogramů

Přehled použitých elementů je v následující tabulce.

Tabulka 7: Pojezd mezi pevně danými body - program - přehled použitých elementů

Použitý element	Základní zobrazení elementu	Počet	Umístění elementu v nabídce	Připojení na vstup (Input), výstup (Output/Motor)	Poznámka
Start		1x	Program elements – Basics elements	-	-
Subprogram Lokace pojezdu		1x	Loaded Programs – Pojezd mezi pevně danými body	-	Podprogramy se nabízí v Loaded programs až po jejich vytvoření!
Subprogram Pohyb pojezdu		1x		-	
Text display		1x	Operating elements - Displays	-	Zobrazení polohy pojezdu. Zobrazení je na panelu
Button		2x	Operating elements – Control elements	-	Pro vyslání pojezdu do požadované polohy, tlačítka jsou na panelu

3.5.3 Samostatné cvičení

Upravte úlohu řešící pojezd mezi pevně danými body následovně:

- Přidejte na ovládací panel 3. tlačítko, které zastaví pojezd v pozici, ve které se zrovna nachází motor.

3.6 Systém pro průběžné měření teploty se záznamem dat do souboru

V tomto cvičení se naučíme odečítat hodnoty z NTC odporu, který je využitelný pro měření teplot, omezit počet cyklů na požadovanou hodnotu, použít proměnou a pracovat se soubory, konkrétně zapisovat hodnoty do CSV souboru.

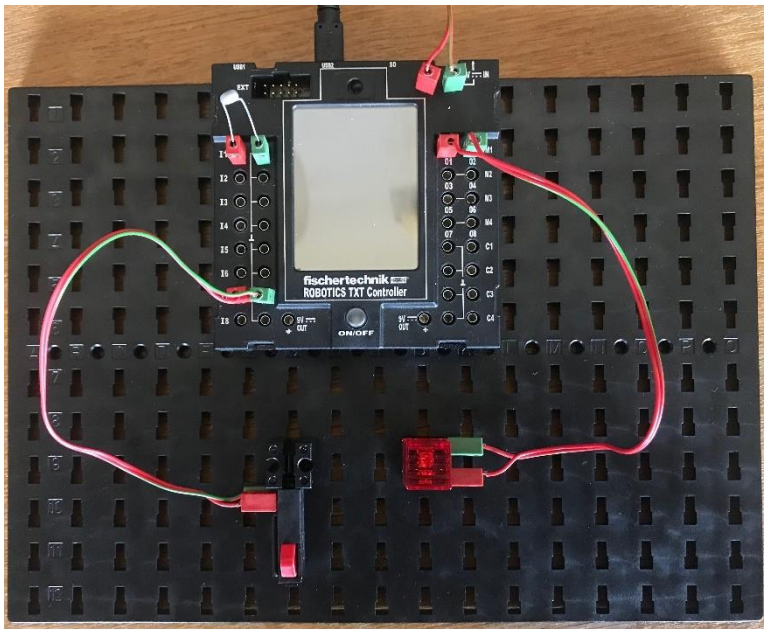
CSV soubor

Prakticky se jedná o textový dokument, kde jsou uložena tabulková data do řádků a sloupců (obdoba excelu). Jednotlivé záznamy jsou odděleny speciálním znakem. V naší oblasti se nejčastěji používá znak středníku, ale existují systémy, které používají jako oddělující znak např. tabulátor, čárku, uvozovky apod. Tento soubor se využívá jak pro ukládání, tak načítání dat. Díky jasné struktuře se poměrně jednoduše data programově zpracovávají. S CSV souborem lze pracovat v jakémkoliv textovém editoru (Notepad, Notepad++, PSPad, …), tabulkových procesorech (MS Excel, LO/OO Calc, …), ale i specializovaných programech např. pro skladování, apod.

3.6.1 Systém pro průběžné měření teploty – zapojení komponent

Pro odečítání hodnot potřebujeme NTC rezistor (princip viz kapitola NTC rezistor), dále využijeme některou ze světelných komponent (je jedno, zda žárovku, nebo LED) a spínač. Více není potřeba. NTC rezistor lze zapojit přímo na řídící jednotku, nebo na vyvedené dráty mimo ni.

Zapojení realizujeme takto: na vstup I1 přivedeme NTC rezistor, na výstup O1/O2 přivedeme světelnou komponentu a na vstup I7 přivedeme spínač viz následující obrázek.



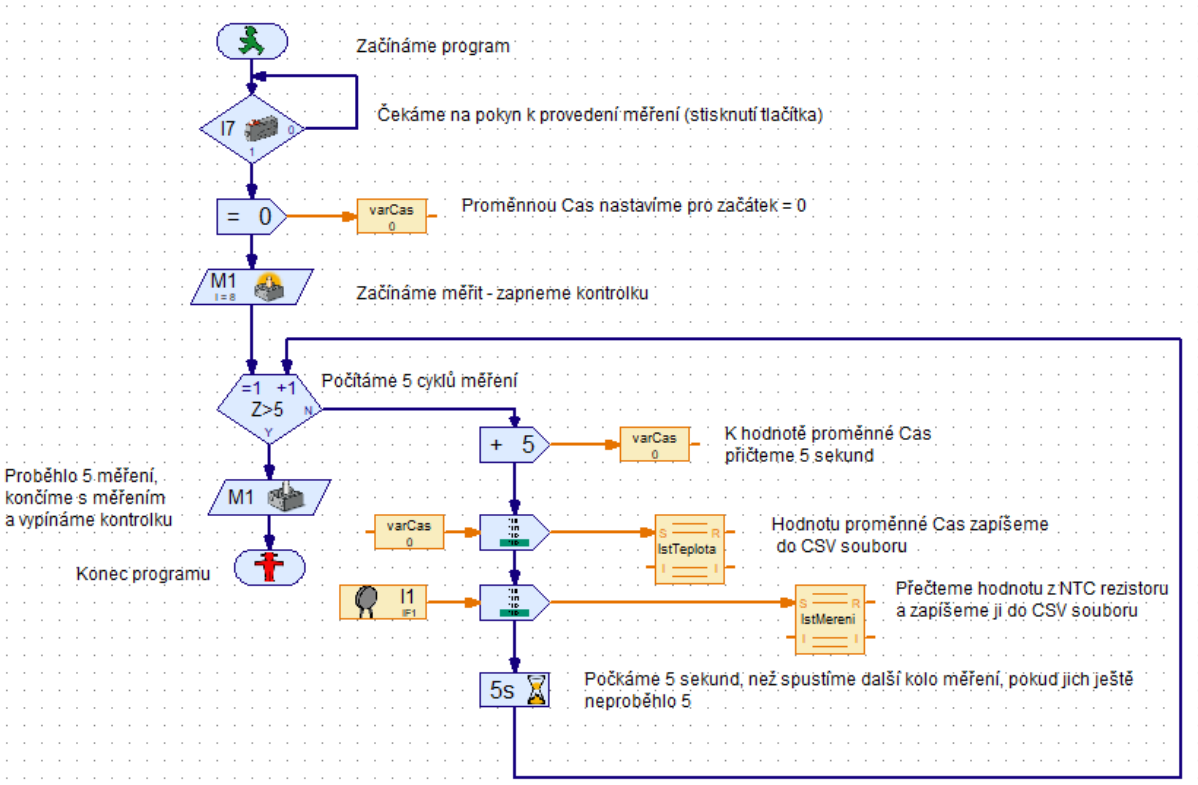
Obrázek 72: Systém pro měření teploty - zapojení komponent

3.6.2 Systém pro měření teploty – řídící program

Jak chceme, aby se náš systém choval…

1. Z NTC odporu budeme každých 5 sekund odečítat hodnotu, v rámci programu 5x.
2. Hodnotu budeme zapisovat do CSV souboru včetně pořadí odečtu.
3. Cyklus 5ti odečtení teploty a vytvoření CSV souboru s odečtenými údaji se provede po stisknutí tlačítka do souboru na ploše PC.
4. To, že program běží bude indikováno vhodnou svítící komponentou.

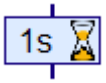
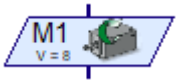
Program může vypadat např. takto, viz obr. Při měření můžete přiložit prsty na odpor a tím jej zahřát – v měření se to projeví jako pokles odečítané hodnoty (viz princip viz kapitola NTC rezistor).

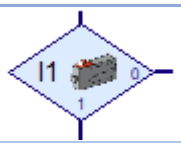
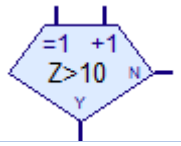
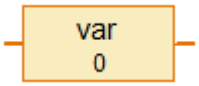
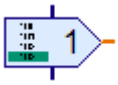
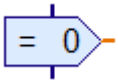


Obrázek 73: Systém pro měření teploty - řídící program

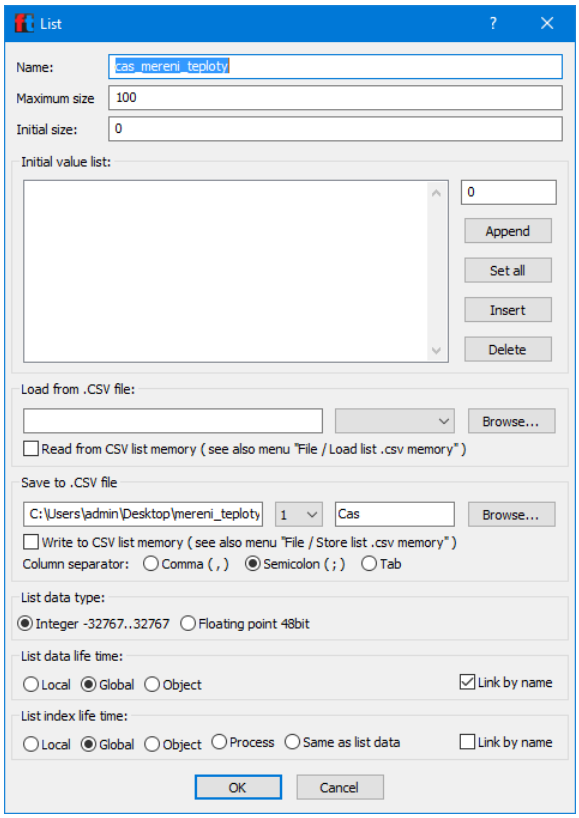
Přehled použitých elementů je v následující tabulce.

Tabulka 8: Systém pro měření teploty - přehled použitých elementů

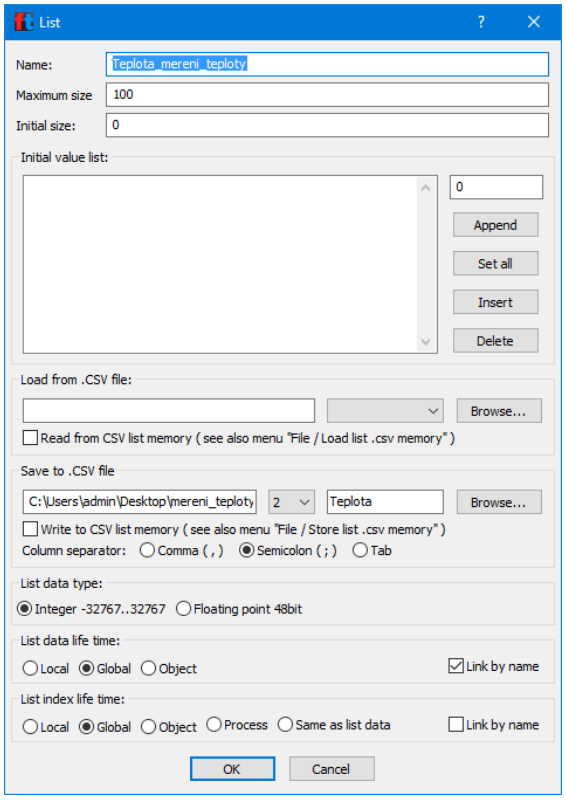
Použitý element	Základní zobrazení elementu	Počet	Umístění elementu v nabídce	Připojení na vstup (Input), výstup (Output/Motor)	Poznámka
Start		1x	Program elements – Basics elements	-	Paralelně běží 2 funkce
End		1x		-	-
Time delay		1x		-	Měření každých 5 sekund
Motor output		2x		M1	V nastavení přepnuto na světelný element. Kontrolka signalizuje probíhající měření

Použitý element	Základní zobrazení elementu	Počet	Umístění elementu v nabídce	Připojení na vstup (Input), výstup (Output/Motor)	Poznámka
Digital Branch		1x		I7	Měření se spustí tlačítkem na vstupu I7
Counter Loop		1x		-	Počítání měření – provádíme jich 5 s rozestupy 5 sekund
Variable		3x	Program elements – Variables, timers…	-	Do proměnné varCas si ukládáme vždy proběhlých 5 sekund
List		2x		-	Vytvoření .CSV souboru se sloupečky Cas a Teplota
Append value		2x	Program elements - Commands	-	Přidávání hodnot do listu. V našem případě do .CSV souboru
Assignment		2x		-	Na začátku nastavujeme proměnné varCas hodnotu 0, poté vždy přičítáme +5 sekund
Universal Input		1x	Program elements – Inputs, Outputs	I1	Odečítání hodnot z NTC rezistoru. Lze vložit přímo předdefinovaný Universal input pro NTC rezistor, nebo vložit jakýkoliv Universal input a ve vlastnostech přepnout na NTC rezistor

Nastavení zápisu dat do CSV souboru (element list) provedeme tak, aby se hodnoty zapisovaly do soboru nazvaného „mereni_teploty.csv“, který bude vytvořený na ploše. Tento soubor neexistuje, řídící program si jej založí a naměřenými hodnotami jej naplní. Do sloupečku **A** budeme zapisovat čas měření, do sloupečku **B** naměřenou hodnotu. Nastavení zápisu dat do elementu **List** viz následující obrázky.



Obrázek 74: Nastavení elementu List pro zápis proměnné Cas do CSV souboru



Obrázek 75: Nastavení elementu List pro zápis teploty do CSV souboru

3.6.3 Samostatné cvičení

Doplňte do programu informační panel, který bude informovat o tom, kolikáté měření právě probíhá.

3.7 Počítání otáček obyčejného motoru

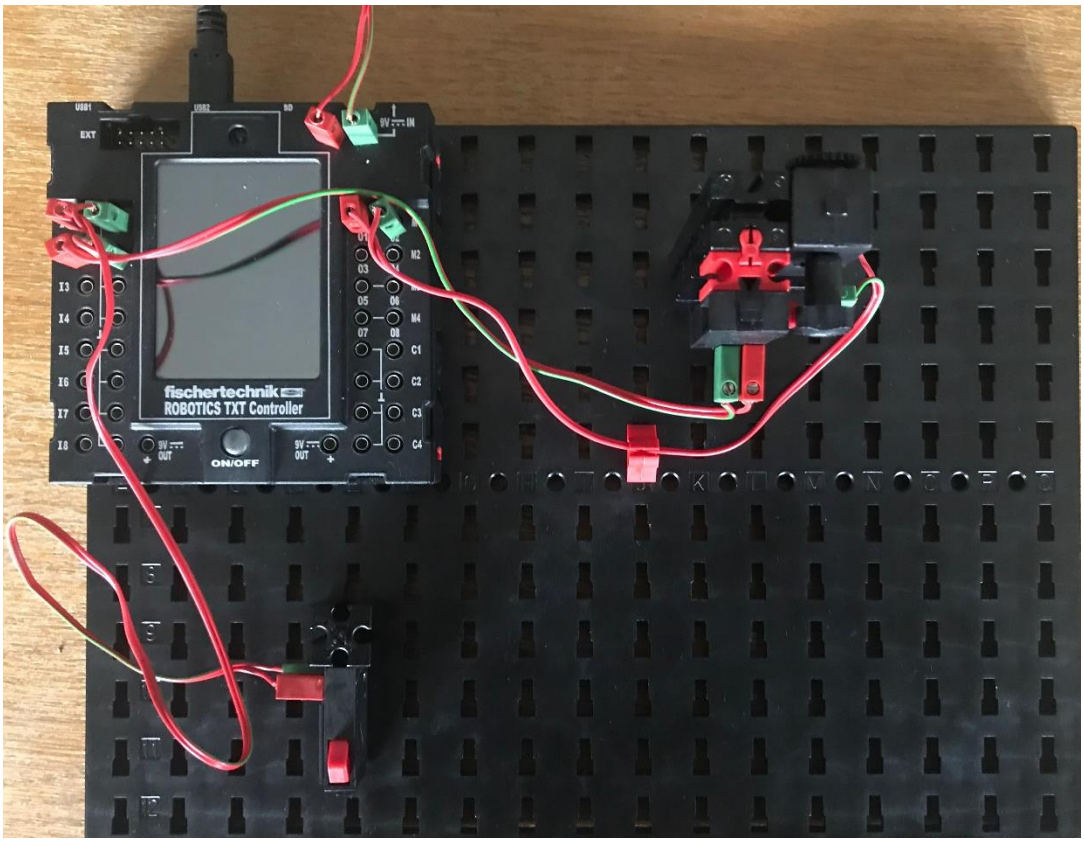
V tomto cvičení si ukážeme, jak počítat otáčky u nekrokových motorů, v případě naší stavebnice tedy motorů Mini a XS pomocí koncového spínače. V nástroji pro programování musíme tedy najít takový element, který dokáže „počítat“ pulzy (spojení/rozpojení) koncového spínače.

3.7.1 Počítání otáček obyčejného motoru – zapojení komponent

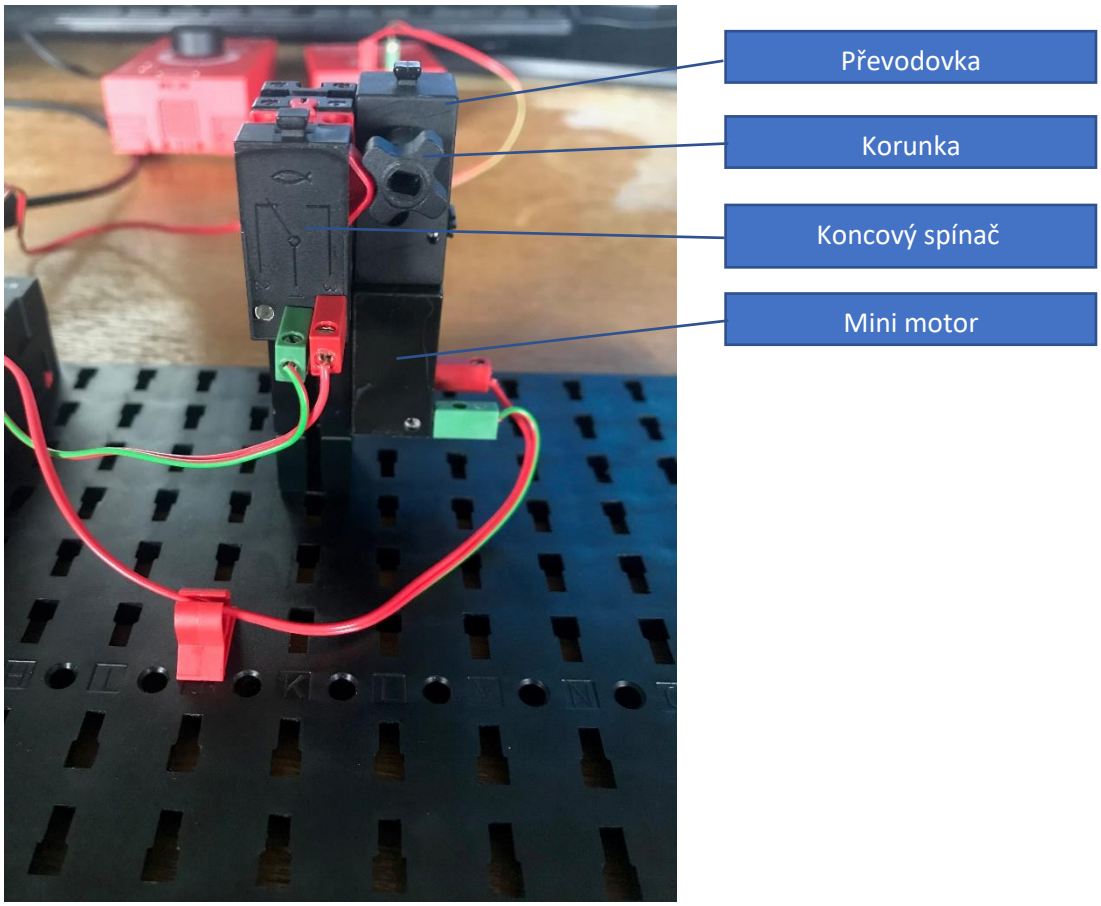
Zapojení komponent je následující, viz obrázky. Na vstupy I1 a I2 jsou přivedeny koncové spínače. I1 spínač pro spuštění programu, I2 spínač počítající pulzy – tedy otáčky motoru. Na výstup M1 je přivedený motor napojený na malou převodovku, u které chceme počítat otáčky na výstupu. Na výstupu je pak nasazena korunka, která spíná/rozpíná koncový spínač, ze kterého pulzy odečítáme.

Korunka má 4 „zuby“, při jejím otočení o 360° to je tedy celkem 8 pulzů. 4 pulzy jsou sepnutí, 4 pulzy rozepnutí. Otočení výstupu převodovky o 360° se tedy sestává z těchto pulzů:

Sepnutí-Rozepnutí-Sepnutí-Rozepnutí-Sepnutí-Rozepnutí-Sepnutí-Rozepnutí.



Obrázek 76: Počítání otáček obyčejného motoru - zapojení komponent

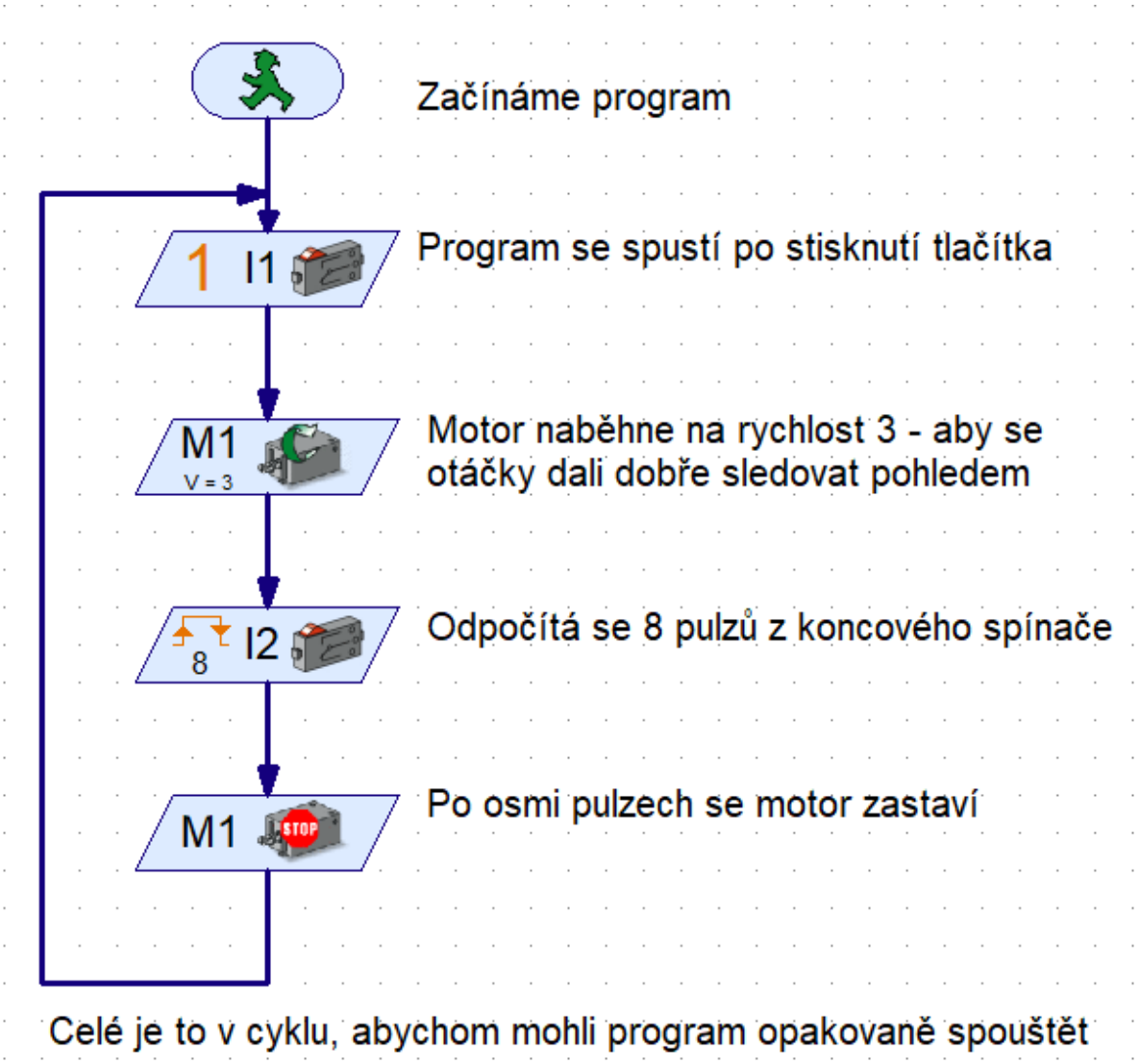


Obrázek 77: Počítání otáček obyčejného motoru - detail zapojení komponent

3.7.2 Počítání otáček obyčejného motoru – řídící program

Jak chceme, aby se motor s převodovkou choval…

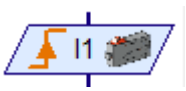
1. Při stisknutí spouštěcího tlačítka se výstup převodovky otočí o 360°
2. Program bude v cyklu, tedy kdykoliv stisknu spouštěcí tlačítko, tak se provede otočení výstupu převodovky o 360°

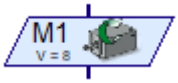


Obrázek 78: Počítání otáček obyčejného motoru - řídící program

Přehled použitých elementů je v následující tabulce.

Tabulka 9: Počítání otáček obyčejného motoru - přehled použitých komponent

Použitý element	**Základní zobrazení elementu**	**Počet**	**Umístění elementu v nabídce**	**Připojení na vstup (Input), výstup (Output/Motor)**	**Poznámka**
Start		1x	Program elements – Basic elements	-	-
Wait for input		1x		I1	Nastaveno čekání na hodnotu „1“
Pulse counter		1x		I2	-

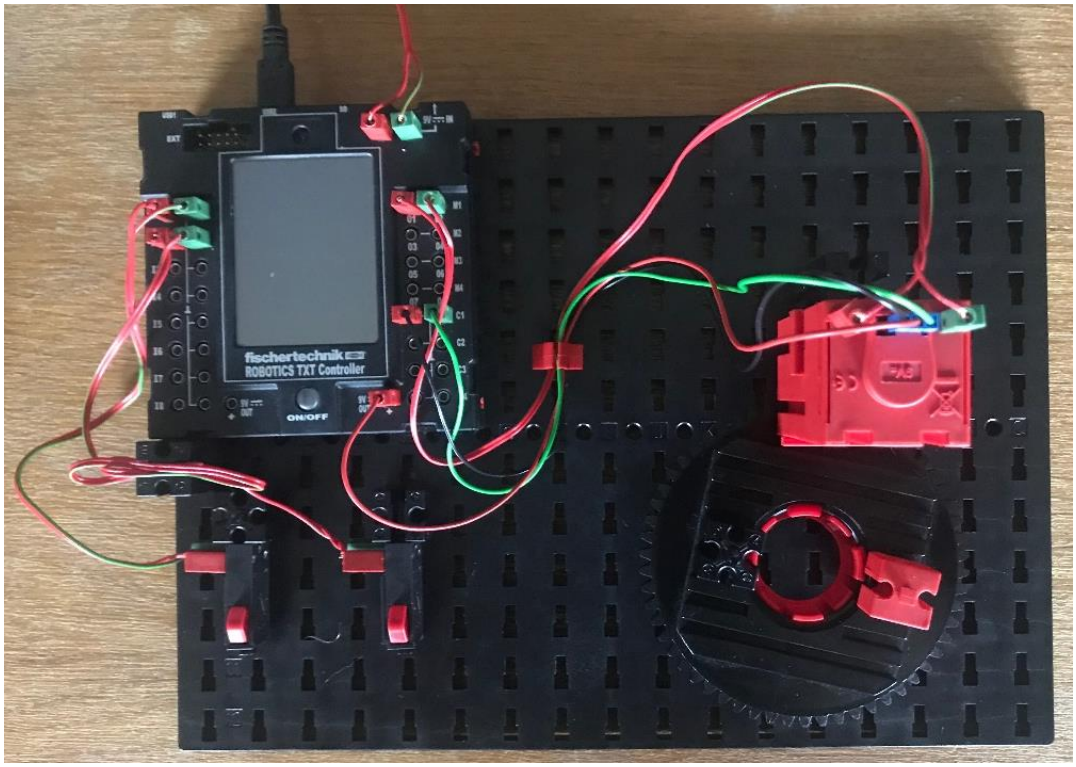
Motor output		2x		M1	-

3.8 Jednoduchý obráběcí stůl s pevným natáčením

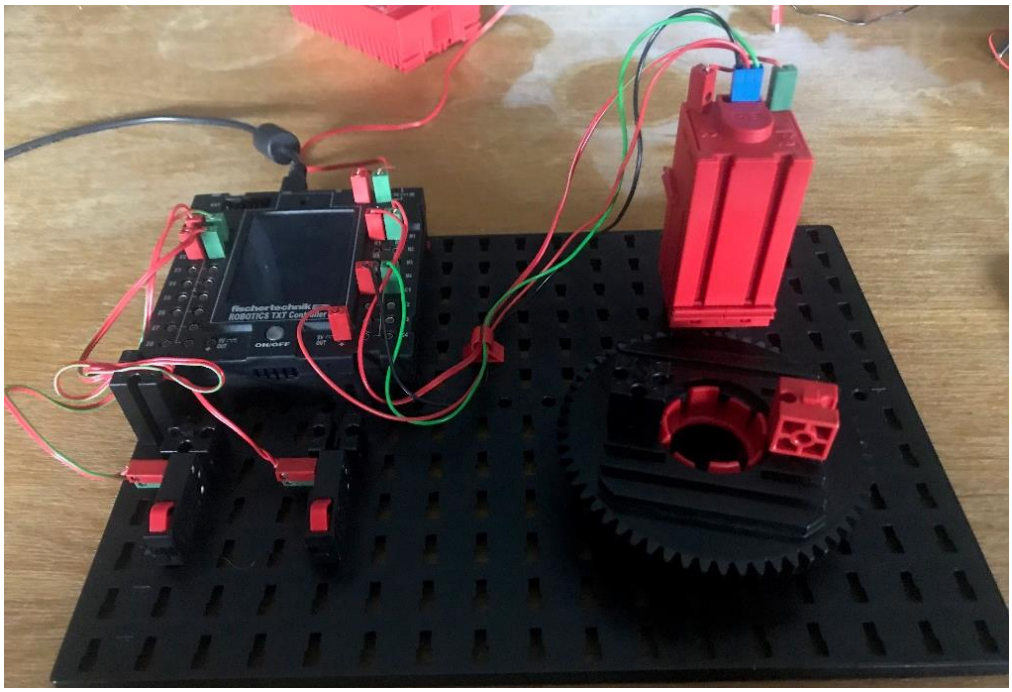
V tomto cvičení se naučíme řídit krokový motor, který lze využít všude, kde je potřeba buď přesně krokovat pohyb, nebo pohyb více motorů přesně synchronizovat.

3.8.1 Jednoduchý obráběcí stůl – zapojení komponent

Zapojení komponent je následující, viz obrázek. Na vstupy I1 a I2 jsou přivedeny koncové spínače. I1 spínač pro spuštění otáčení stolu ve směru hodinových ručiček, I2 spínač pro spuštění otáčení stolu proti směru hodinových ručiček. Na výstup M1 je přivedený krokový motor a na C1 a 9 V+ pak krokovací mechanismus motoru, který jej umožňuje přesně řídit.



Obrázek 79: Jednoduchý obráběcí stůl - spojení komponent, pohled shora

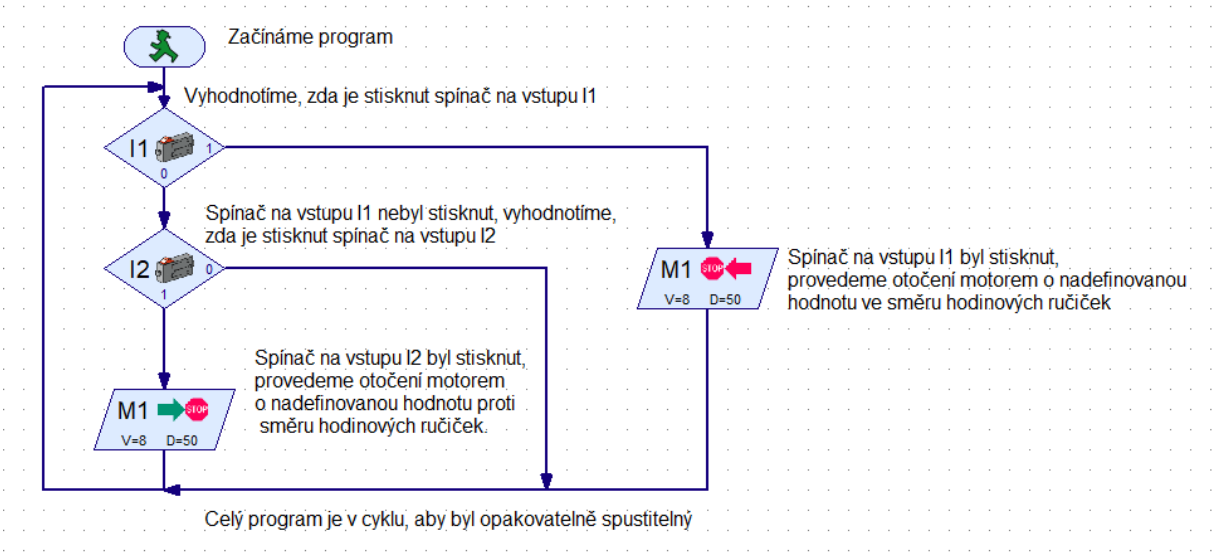


Obrázek 80: Jednoduchý obráběcí stůl - spojení komponent, pohled zpředu

3.8.2 Jednoduchý obráběcí stůl – řídící program

Jak chceme, aby se obráběcí stůl choval…

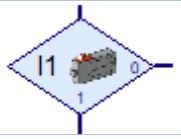
1. Při stisknutí spouštěcího tlačítka I1 se stůl otočí o nadefinovaný údaj ve směru hodinových ručiček.
2. Při stisknutí spouštěcího tlačítka I2 se stůl otočí o nadefinovaný údaj proti směru hodinových ručiček.



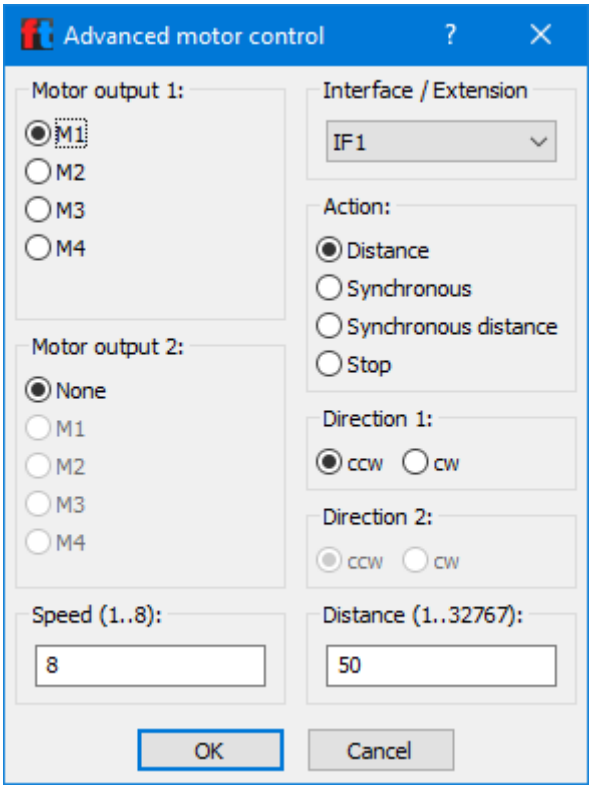
Obrázek 81: Jednoduchý obráběcí stůl - řídící program

Přehled použitých elementů je v následující tabulce.

Tabulka 10: Jednoduchý obráběcí stůl - přehled použitých komponent

Použitý element	Základní zobrazení elementu	Počet	Umístění elementu v nabídce	Připojení na vstup (Input), výstup (Output/Motor)	Poznámka
Start		1x	Program elements – Basic elements	-	-
Digital Branch		2x		I1, I2	-
Encoder motor		2x		M1	1x otáčí ve směru a 1x proti směru hodinových ručiček

V detailu elementu pro krokový motor, viz následující obrázek, lze vyčíst, co vše nám krokový motor umožňuje. Nejdůležitější informace pro nás je směr otáček a rychlost otáček (stejně jako u mini motoru a XS motoru), ale navíc můžeme navolit údaje **Distance** a zadat „vzdálenost“, kterou má motor uběhnout, nebo jej synchronizovat s jiným krokovým motorem (to ukáže další příklad). U položky **Distance** je potřeba si uvědomit, že se musí vždy zjistit, jaký údaj odpovídá našim požadavkům. Tedy chceme-li otočku pracovního stolu např. o 360°, musíme si zjistit, jaký údaj tomu odpovídá, tedy musíme si motor s připojenou periferií kalibrovat.



Obrázek 82: Krokový motor a jeho řízení - nastavení elementu

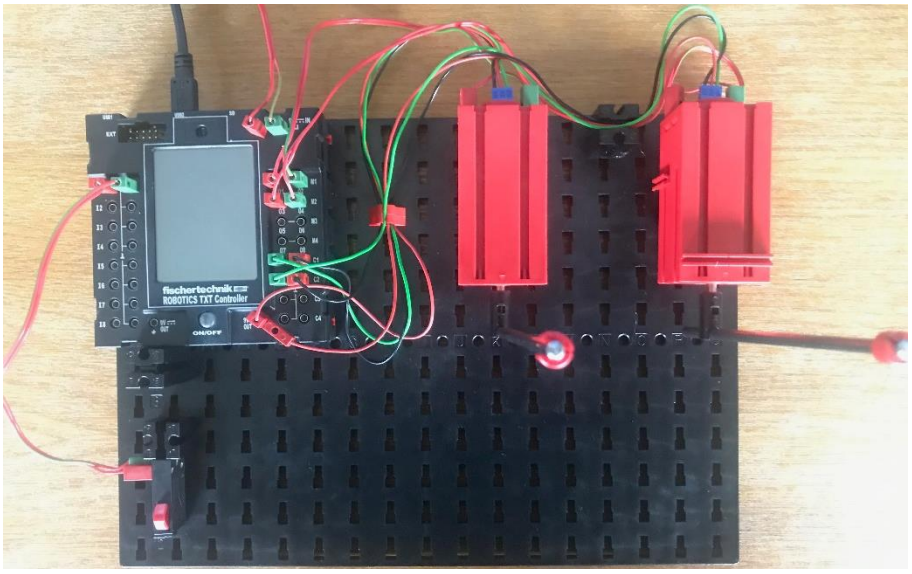
V našem případě jen nastavíme hodnotu např. 50.

3.9 Synchronizovaní ukazatelé

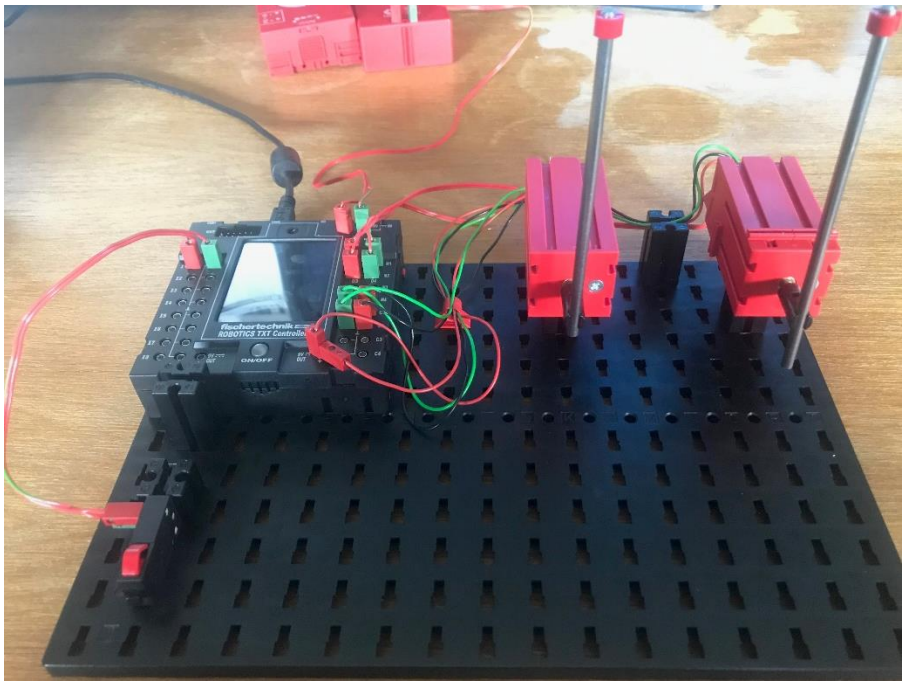
V tomto cvičení se naučíme synchronizovat 2 krokové motory na příkladu 2 ukazatelů řízených synchronně.

3.9.1 Synchronizovaní ukazatelé – zapojení komponent

Zapojení komponent je následující, viz obrázek. Na vstup I1 je přivedený spínač, který spouští motory synchronně. Na výstupy M1 a M2 jsou přivedeny krokové motor a na C1, C2 a 9 V+ pak krokovací mechanismy motorů, které slouží k synchronizaci motorů.



Obrázek 83: Synchronizovaní ukazatelé - zapojení komponent, pohled shora

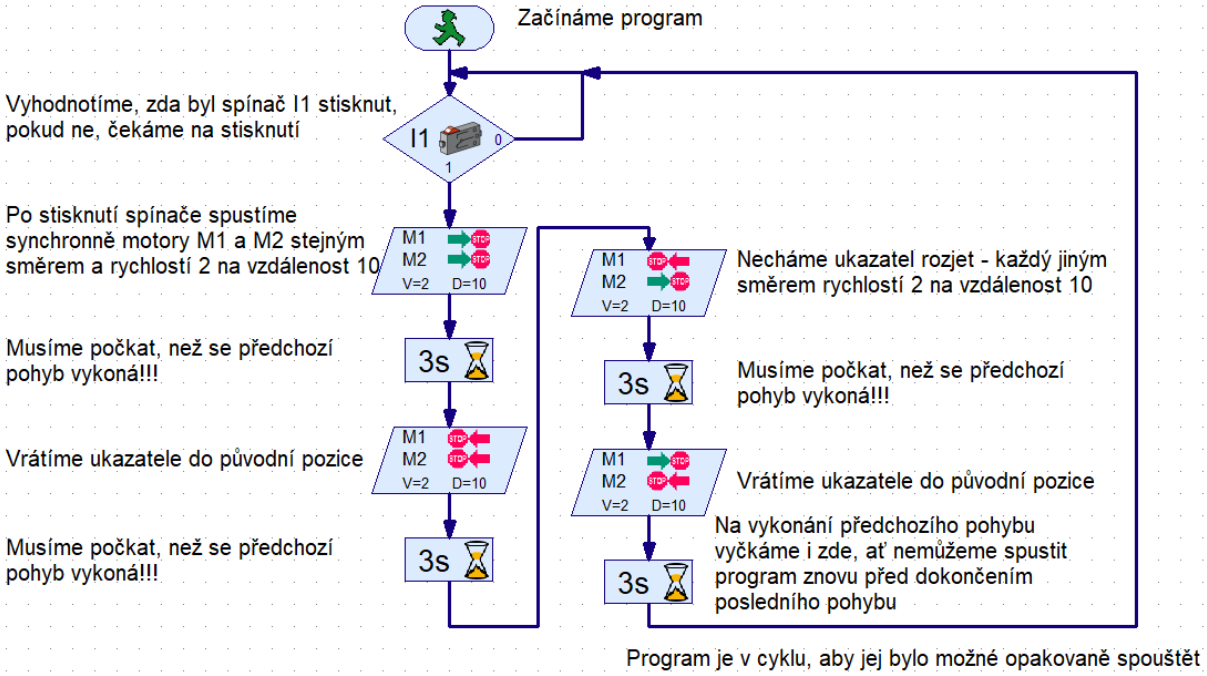


Obrázek 84 : Synchronizovaní ukazatelé - zapojení komponent, pohled zpředu

3.9.2 Synchronizovaní ukazatelé – řídící program

Jak chceme, aby se synchronizovaní ukazatelé chovali…

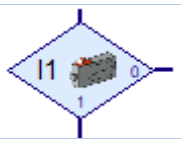
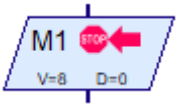
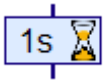
1. Při stisknutí spouštěcího tlačítka I1 se ukazatelé synchronizovaně natočí na jednu stranu a poté se vrátí do původní polohy.
2. Následně se pohnou každý na jinou stranu, vytvoří tak písmeno „V“ a poté se vrátí do původní polohy.
3. Program bude v cyklu, aby jej šlo opakovaně spouštět spouštěcím tlačítkem.



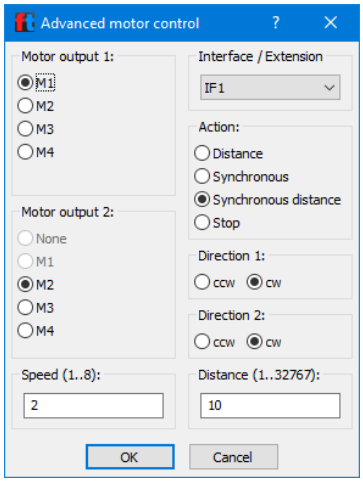
Obrázek 85: Synchronizovaní ukazatelé - řídící program

Přehled použitých elementů je v následující tabulce.

Tabulka 11: Synchronizovaní uživatelé - přehled použitých komponent

Použitý element	Základní zobrazení elementu	Počet	Umístění elementu v nabídce	Připojení na vstup (Input), výstup (Output/Motor)	Poznámka
Start		1x	Program elements – Basic elements	-	-
Digital Branch		1x		I1	-
Encoder motor		4x		M1, M2	Pozor, element nutno nastavit pro synchronizovaný pohyb dvou motorů
Time delay		4x		-	-

Ukázka nastavení elementu krokového motoru viz obrázek. Více v předchozím příkladu.



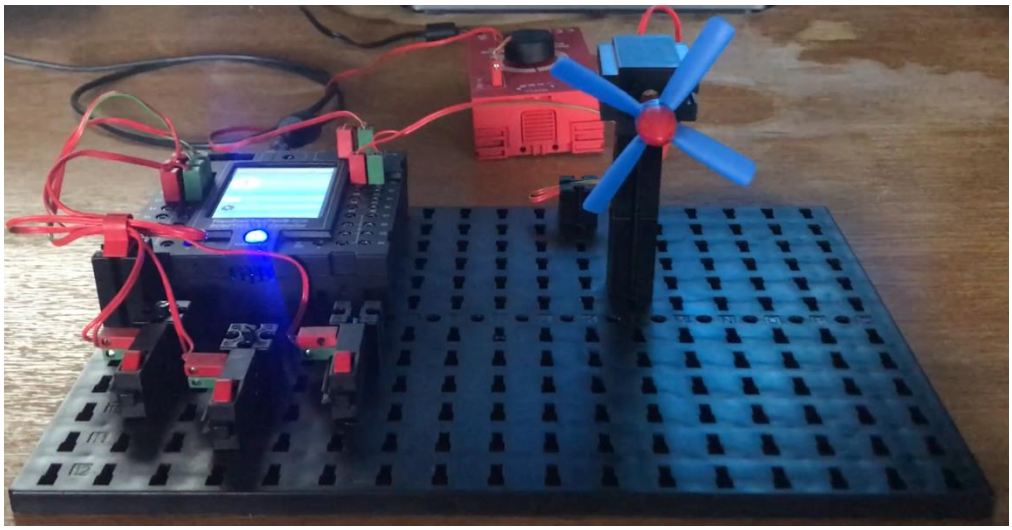
Obrázek 86: Synchronizovaní ukazatelé - detail nastavení synchronizace krokových motorů

3.10 Logický motor

V tomto cvičení se na chvíli vrátíme k větráku a naučíme se využívat logické operátory a matematické operace, zopakujeme si podprogramy a naučíme se předávat potřebné hodnoty do podprogramu a naopak z podprogramu je vracet do programu.

3.10.1 Logický motor – zapojení komponent

Zapojení komponent je následující, viz obrázek. Na vstupy I1, I2 a I3 jsou přivedeny spínače. Na výstup M1 je přivedený motor.



Obrázek 87: Logický motor - zapojení komponent

3.10.2 Logický motor – řídící program

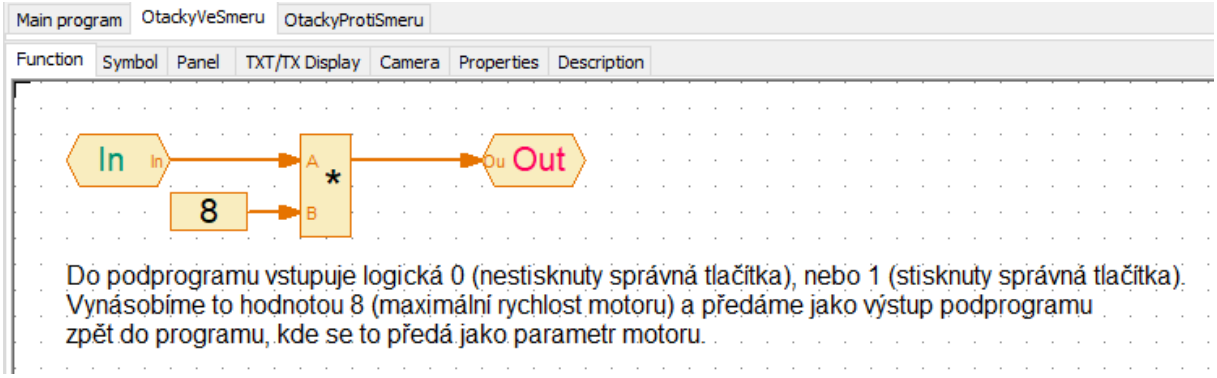
Jak chceme, aby se logický motor choval…

1. Při stisknutí pouze jednoho, jakéhokoliv tlačítka se nic neděje.
2. Tlačítko I1 je bezpečnostní
3. Tlačítko I2 spouští otáčky ve směru hodinových ručiček. Podmínka je, že k němu musí být stisknuto i tlačítko I1.

4. Tlačítko I3 spouští otáčky v protisměru hodinových ručiček. Podmínka je, že k němu musí být stisknuto i tlačítko I1.

Stejného výsledku lze dosáhnout propojením elementů spínačů s vhodně nadefinovanými podmínkami. Úkolem tohoto cvičení je si ukázat, že to lze i jinak a že existují logické operace a jak fungují. Navíc lze motor (a i další komponenty) ovládat předáním parametru, a ne pouze elementem pro ovládání motoru.

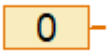
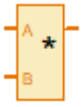
Praktické využití je např. bezpečné spouštění stroje, kdy chceme mít jistotu, že obsluha má obě ruce mimo a nehrozí tak její poranění.

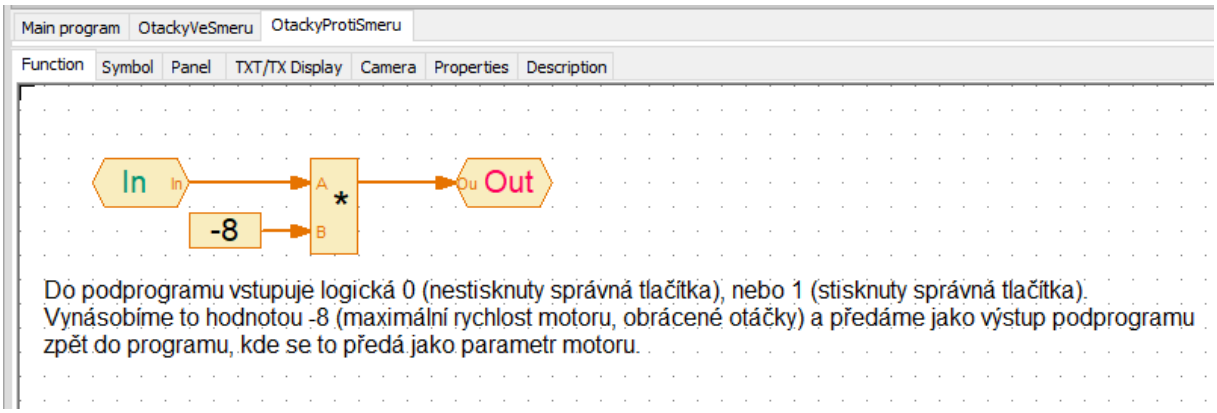


Obrázek 88: Logický motor - podprogram pro otáčky motoru ve směru hodinových ručiček

Přehled použitých elementů je v následující tabulce.

Tabulka 12: Logický motor - podprogram otáčky ve směr. hod. ručiček - přehled použitých elementů

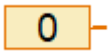
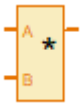
Použitý element	Základní zobrazení elementu	Počet	Umístění elementu v nabídce	Připojení na vstup (Input), výstup (Output/Motor)	Poznámka
Constant		1x	Program elements – Variables, timers…	-	Násobíme konstantou 8 – maximální rychlost motoru ve směru hodinových ručiček
Subprogram command input		1x	Program elements – Subprogram I/O	-	-
Subprogram command output		1x		-	-
Operators - Times		1x	Program elements - Operators	-	-

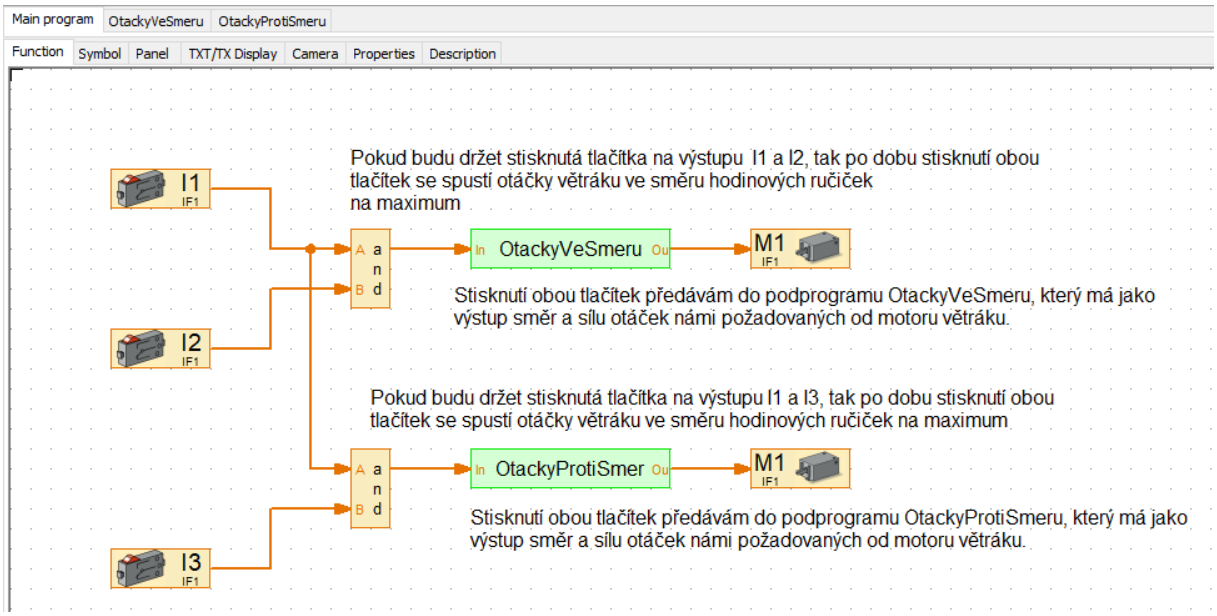


Obrázek 89: Logický motor - podprogram pro otáčky motoru v protisměru hodinových ručiček

Přehled použitých elementů je v následující tabulce.

Tabulka 13: Logický motor - podprogram otáčky proti směru hod. ručiček - přehled použitých elementů

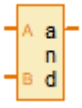
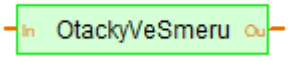
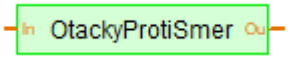
Použitý element	Základní zobrazení elementu	Počet	Umístění elementu v nabídce	Připojení na vstup (Input), výstup (Output/Motor)	Poznámka
Constant		1x	Program elements – Variables, timers…	-	Násobíme konstantou = -8, maximální rychlost motoru proti směru hodinových ručiček
Subprogram command input		1x	Program elements – Subprogram I/O	-	-
Subprogram command output		1x		-	-
Operators - Times		1x	Program elements - Operators	-	-



Obrázek 90: Logický motor – Program

Přehled použitých elementů je v následující tabulce.

Tabulka 14: Logický motor - Program - přehled použitých elementů

Použitý element	Základní zobrazení elementu	Počet	Umístění elementu v nabídce	Připojení na vstup (Input), výstup (Output/Motor)	Poznámka
Universal input		3x	Program elements – Inputs, outputs	I1	-
Motor output		2x		M1	-
Operator AND		2x	Program elements - operators	-	-
Subprogram OtackyVeSmeru		1x	Loaded Programs – Logický motor	-	Podprogramy se nabízí v Loaded programs až po jejich vytvoření!
Subprogram OtackyProtiSmeru		1x		-	

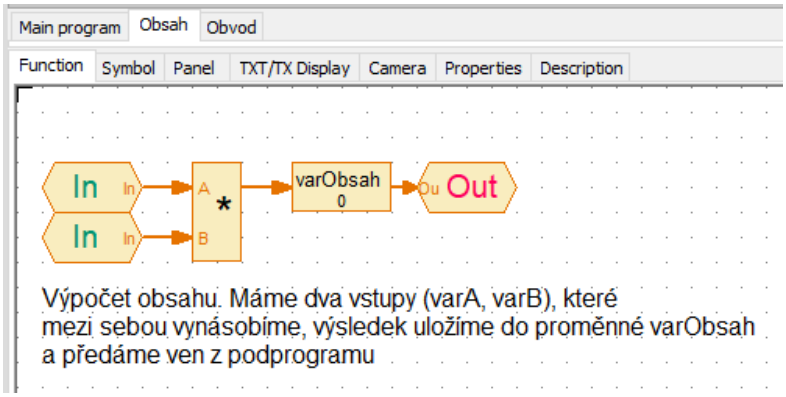
3.11 Matematika

V tomto cvičení si ukážeme, že lze provádět i matematické operace a vytvořit program, který např. pouze provádí potřebný výpočet i bez potřeby modelu. Zopakujeme si proměnné, podprogramy, předávání parametrů do podprogramu, vracení parametru z podprogramu.

3.11.1 Matematika – řídící program

Jak chceme, aby program fungoval…

1. Programu se zadají dvě hodnoty, které se uloží do proměnných.
2. Z hodnot se vypočte obsah a obvod pravoúhlého čtyřúhelníka (čtverec, obdélník), které se uloží do adekvátních proměnných.
3. Obsah a obvod se vhodně zobrazí.

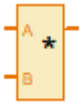


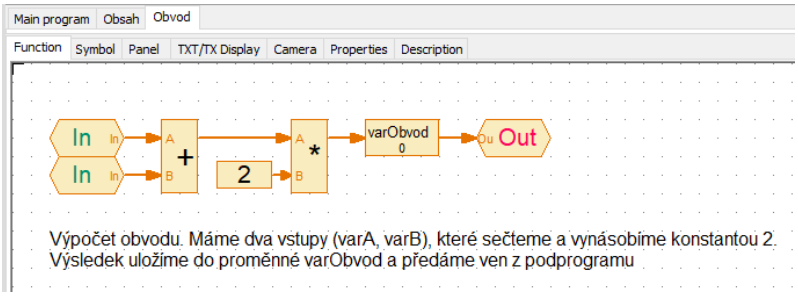
Obrázek 91: Matematika - podprogram pro obsah

Přehled použitých elementů je v následující tabulce.

Tabulka 15: Matematika - podprogram Obsah - přehled použitých elementů

Použitý element	Základní zobrazení elementu	Počet	Umístění elementu v nabídce	Připojení na vstup (Input), výstup (Output/Motor)	Poznámka
Variable	var 0	1x	Program elements – Variables, timers…	-	Výsledek ukládáme do proměnné varObsah. Protože je používáme v programu i podprogamech, je potřebné je nastavit jako Globální.
Subprogram command input	In	2x	Program elements – Subprogram I/O	-	-
Subprogram command output	Out	1x		-	-

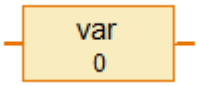
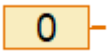
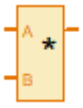
Operators - Times		1x	Program elements - Operators	-	-

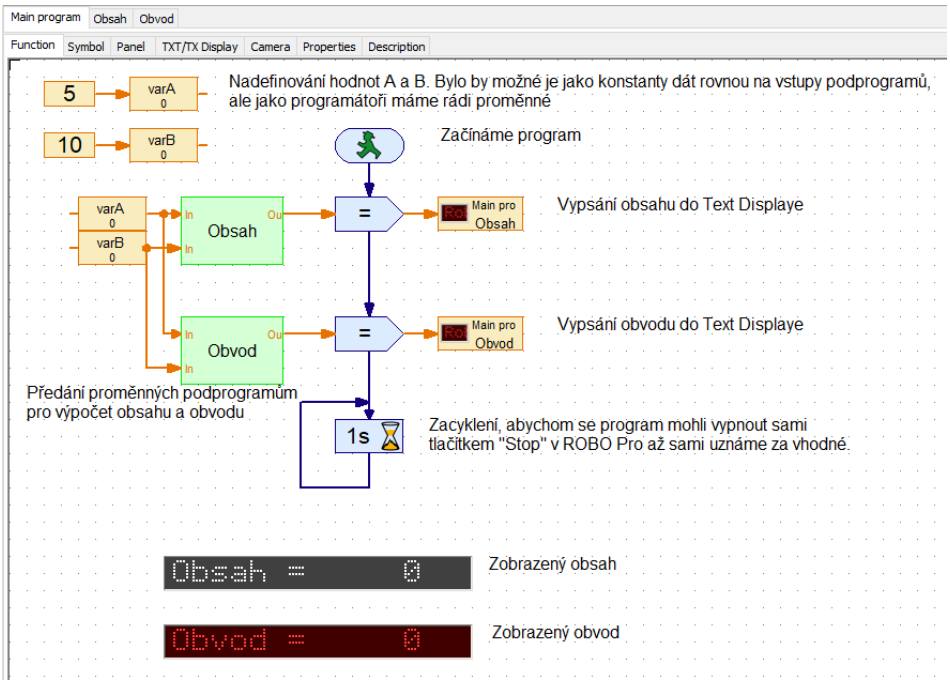


Obrázek 92: Matematika - podprogram pro obvod

Přehled použitých elementů je v následující tabulce.

Tabulka 16: Matematika - podprogram Obvod - přehled použitých elementů

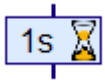
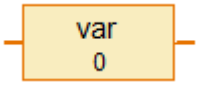
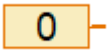
Použitý element	Základní zobrazení elementu	Počet	Umístění elementu v nabídce	Připojení na vstup (Input), výstup (Output/Motor)	Poznámka
Variable		1x	Program elements – Variables, timers…	-	Výsledek ukládáme do proměnné varObvod. Protože je používáme v programu i podprogamech, je potřebné je nastavit jako Globální.
Constant		1x		-	Násobíme konstantou 2
Subprogram command input		2x	Program elements – Subprogram I/O	-	-
Subprogram command output		1x		-	-
Operators – Times, Plus		2x	Program elements - Operators	-	1x násobíme, 1x sčítáme

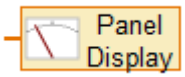
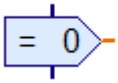
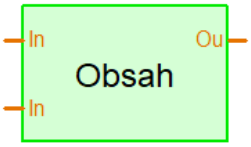
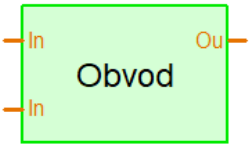


Obrázek 93: Matematika – program

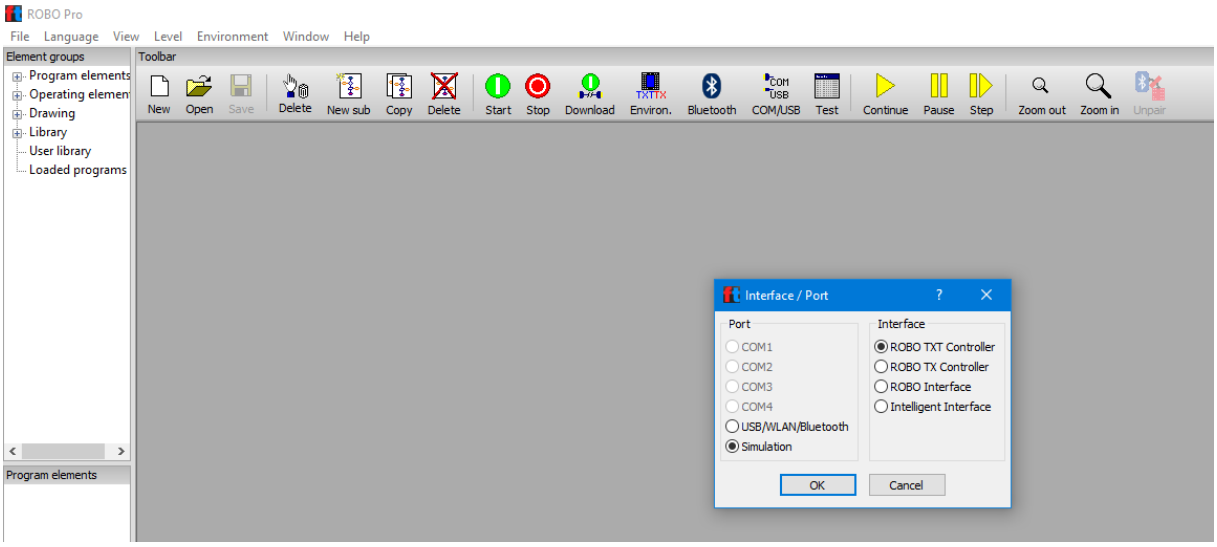
Přehled použitých elementů je v následující tabulce.

Tabulka 17: Matematika - program - přehled použitých elementů

Použitý element	Základní zobrazení elementu	Počet	Umístění elementu v nabídce	Připojení na vstup (Input), výstup (Output/Motor)	Poznámka
Start		1x	Program elements – Basics elements	-	-
Time delay		1x		-	Využijeme k zacyklení
Variable		4x	Program elements – Variables, timers…	-	Pro uložení a použití délek stran A a B, tedy varA a varB. Protože je používáme v programu i podprogamech, je potřebné je nastavit jako Globální.
Constant		2x		-	Přiřazení vstupních hodnot proměnným varA a varB

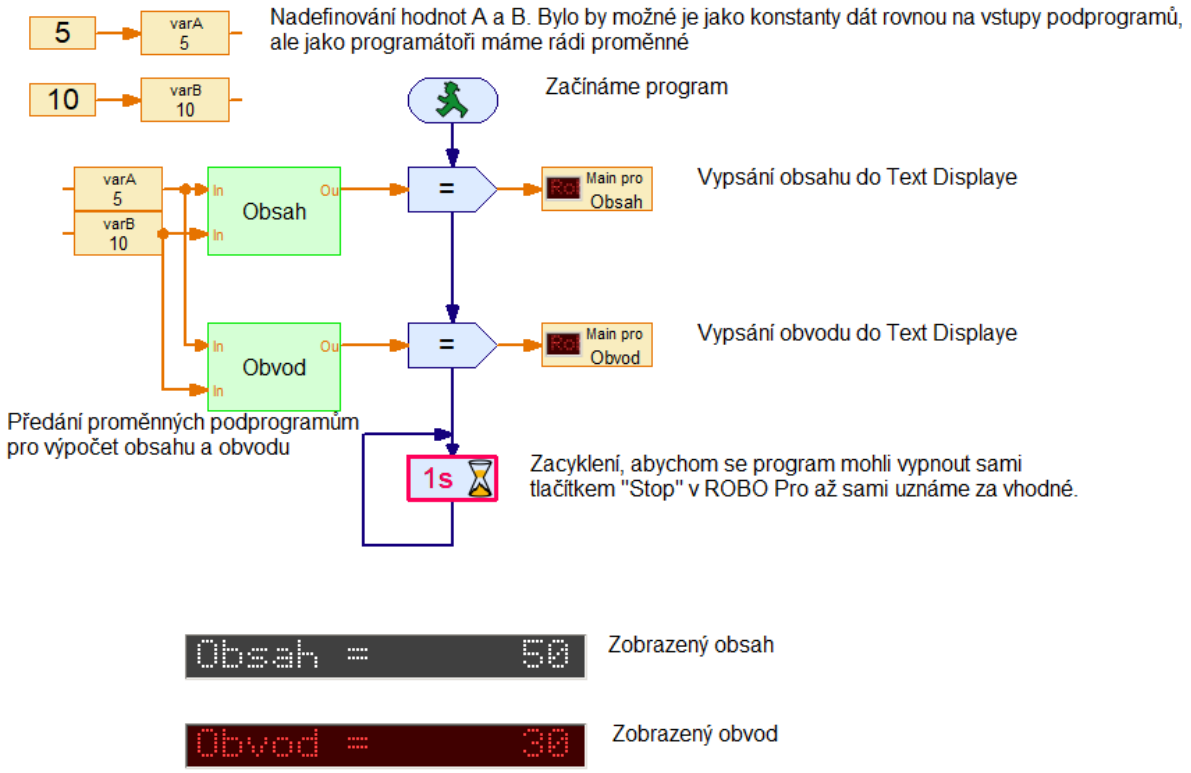
Panel display		2x	Program elements – Inputs, outputs	-	Pro předání Obsahu a Obvodu příslušným Text displejům
Text display		2x	Operating elements - Displays	-	Zobrazení spočítaného Obvodu a Obsahu
Assignment		2x	Program elements - Commands	-	Přiřazení hodnoty Panel displeji, budou pouze „=“
Subprogram Obsah		1x	Loaded Programs – Matematika	-	Podprogramy se nabízí v Loaded programs až po jejich vytvoření!
Subprogram Obvod		1x		-	

Program lze spustit i bez připojeného modelu, resp. TXT Controlleru. Pro takové spuštění je potřeba přes volbu COM/USB v horním menu zvolit položku Simulation viz následující obrázek. Poté je potřeba se přepnou opět zpět na volbu USB/WLAN/Bluetooth aby fungovalo propojení ROBO Pro s modely.



Obrázek 94: Spuštění programu bez připojeného TXT Controlleru

Na následujícím obrázku je vidět běžící program.



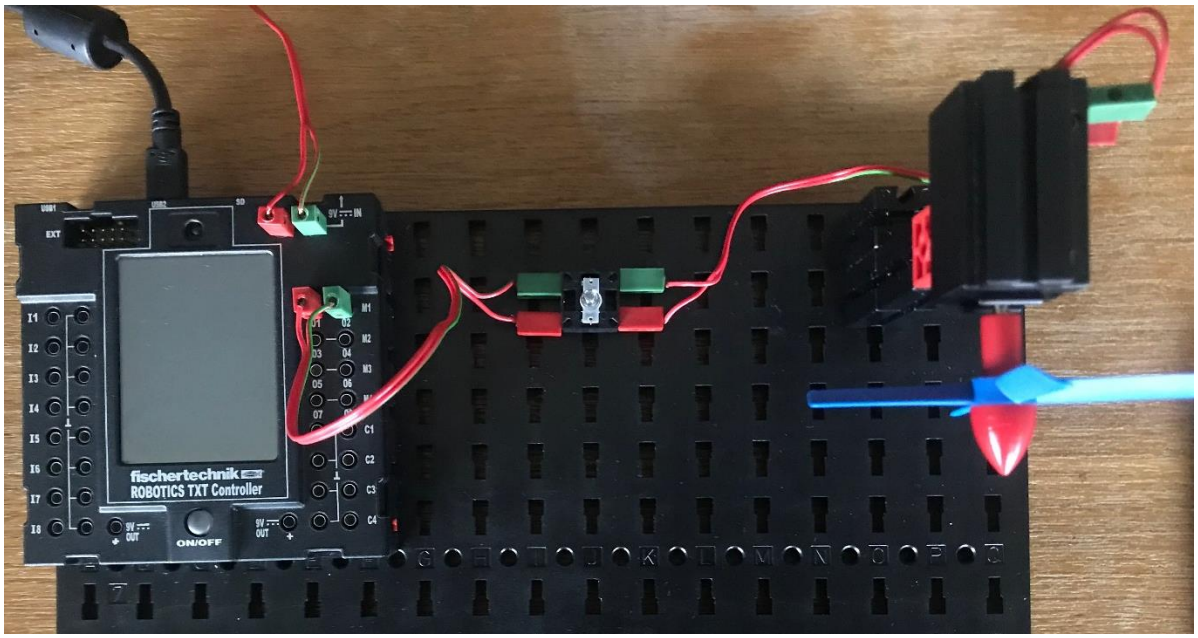
Obrázek 95: Spuštěný program bez připojeného TXT Controlleru

3.12 Větráček II– návrat k základům a jejich rozšíření

V tomto cvičení se naučíme pracovat s ukazatelem, Slidery (posuvníky), vytvářet GUI pro ovládání modelu na dotykovém displeji řídící jednotky a nahrání do řídící jednotky tak, aby program fungoval samostatně bez propojení řídící jednotky s počítačem. Navíc si ukážeme, že lze zapojovat komponenty paralelně.

3.12.1 Větráček II – zapojení komponent

Zapojení komponent je následující, viz obrázek. Na výstup IM1 je přivedena světelná komponenta a ta je dále propojena s motorem.



Obrázek 96: Větráček II - zapojení komponent

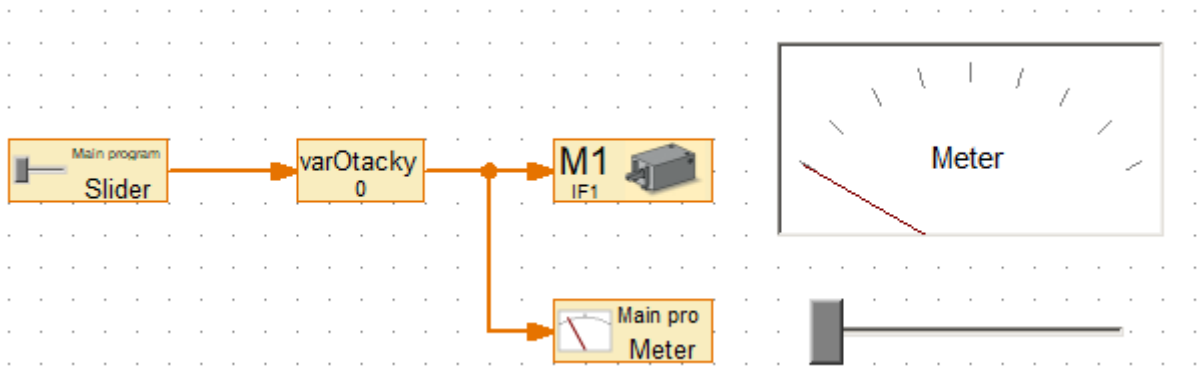
3.12.2 Větráček II – řídící program v režimu online propojení s počítačem

Jak chceme, aby se kontrolor barvy choval…

1. Při použití Slideru (posuvníku) na počítači se bude adekvátně rozsvěcet/zhasínat světelná komponenta a roztáčet větráček.
2. Intenzita emitace světla světelnou komponentou a rychlost otáček větráčku bude zobrazen rafičkovým ukazatelem na počítači.
3. Program bude v cyklu, aby bylo možné jej spouštět opakovaně.

Co v tomto případě potřebujeme vědět…

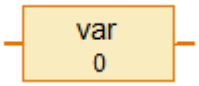
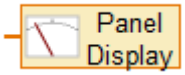
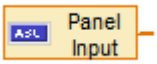
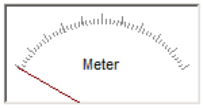
Intenzita svícení světelné komponenty a rychlost otáček motoru jsou dány rozsahem 0 – 8. Kdy při hodnotě 0 jsou komponenty vypnuty, při hodnotě 8 běží na plný výkon (největší svítivost světelné komponenty a nejvyšší otáčky motoru). Celý program je pak na následujícím obrázku.



Obrázek 97: Větráček 2 - řídící program pro online režim

Přehled použitých elementů je v následující tabulce.

Tabulka 18: Větráček II v režimu Online - přehled použitých elementů

Použitý element	Základní zobrazení elementu	Počet	Umístění elementu v nabídce	Připojení na vstup (Input), výstup (Output/Motor)	Poznámka
Variable		1x	Program elements – Variables, timers…	-	-
Panel display		1x	Program elements – Inputs, outputs	-	-
Motor output		1x		M1	-
Panel Input		1x		-	-
Meter		1x	Operating elements - Displays	-	-
Slider		1x	Operating elements – Control elements	-	-

Následuje ukázka nastavení vybraných elementů

3.12.2.1 Meter

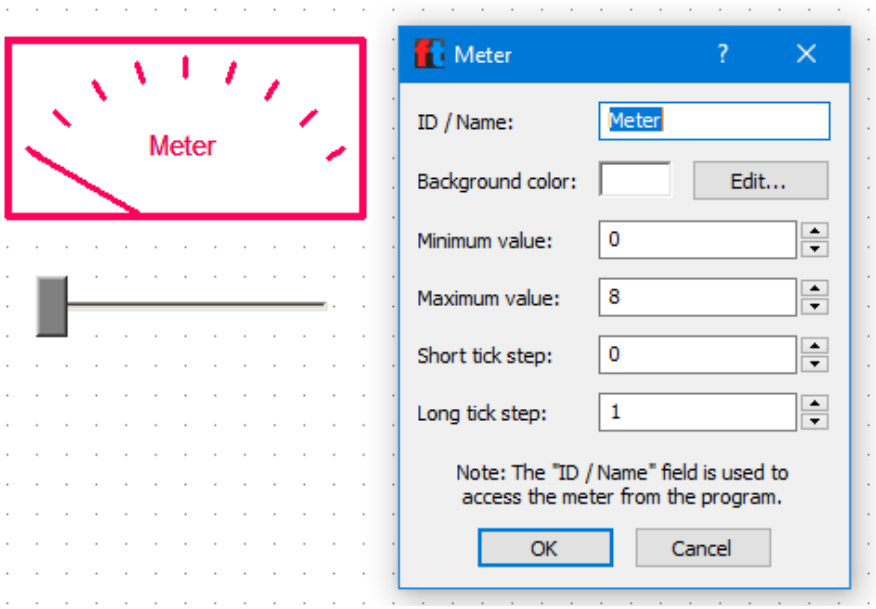
Element ze sekce Operating elements – displays. Elementu nastavíme:

Minimum value = 0 hodnota vypnutí komponent

Maximum value = 8 maximální výkon komponent

Short tick step = 0 při rozsahu 0 – 8 nevyužijeme vedlejší krok na stupnici

Long tick step = 1 hlavní hodnoty ukazatele budou mít krok na stupnici = 1



Obrázek 98: Větráček II - element Meter

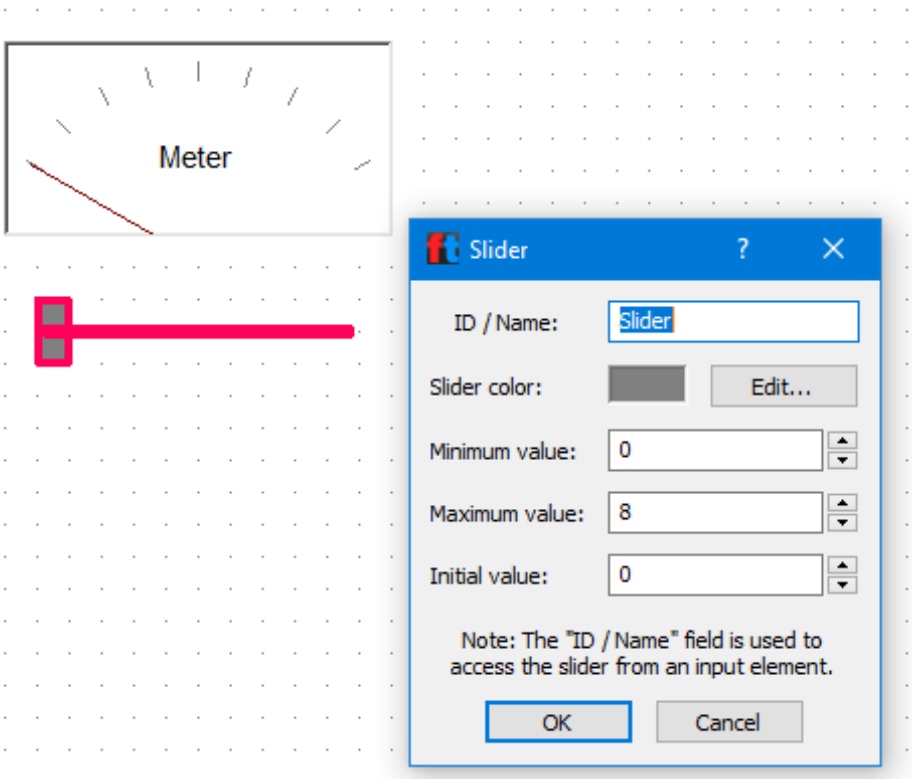
3.12.2.2 Slider

Element ze sekce Operating elements – Control elements. Elementu Slider nastavíme:

Minimum value = 0 Při pozici jezdce úplně vlevo budou komponenty vypnuty.

Maximum value = 8 Při pozici úplně vpravo budou komponenty puštěny na maximum.

Initial value = 0 Při spuštění programu bude jezdec na pozici úplně vlevo.



Obrázek 99: Větráček II - element Slider

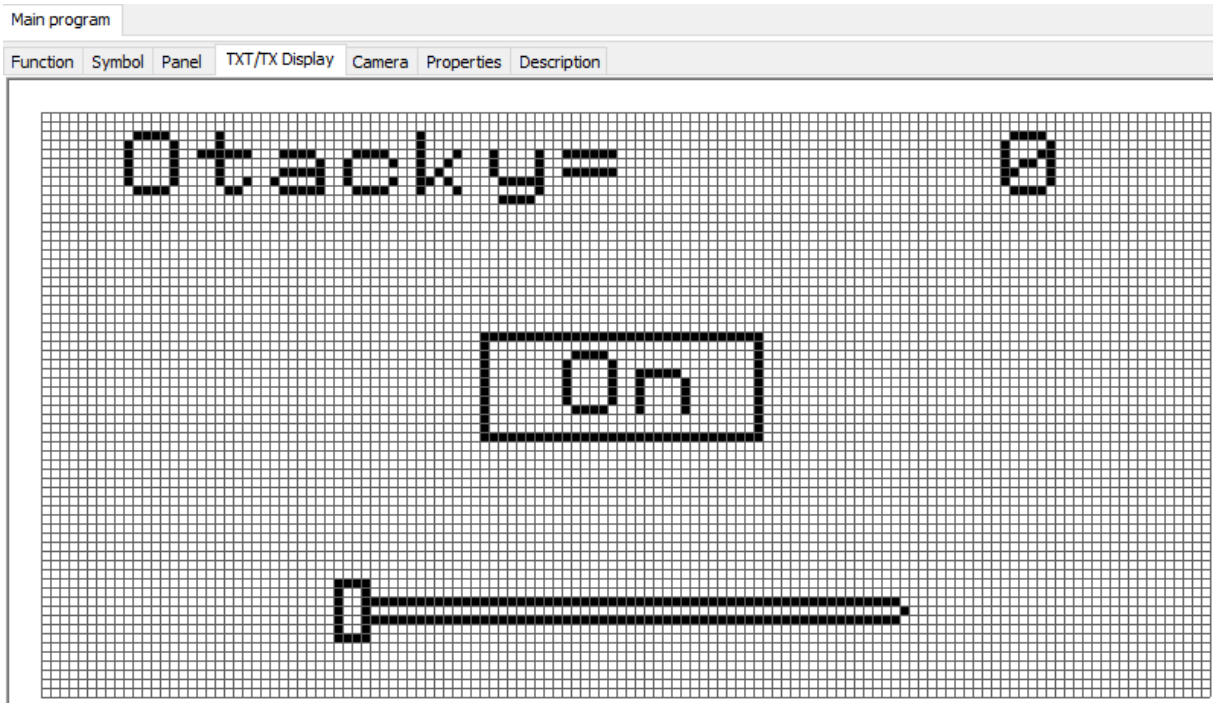
3.12.3 Větráček II – řídící program v samostatném režimu

Jak chceme, aby se kontrolor barvy choval…

1. Řídící program se nahraje přímo do řídící jednotky. Model tak bude fungovat i bez propojení s počítačem a bude se ovládat dotykovým displejem.
2. Na dotykovém displeji bude zobrazena hodnota odpovídající aktuálnímu výkonu komponent, tedy 0 pro vypnuté komponenty až 8 pro jejich maximální výkon.
3. Na dotykovém displeji bude tlačítko současně sloužící jak pro spuštění komponent na maximální výkon, tak i pro vypnutí komponent.
4. Na dotykovém displeji bude slider pro řízení výkonu komponent.
5. Nebude se vytvářet nový program, použije se program z předchozí varianty úkolu a jen se rozšíří o požadované.

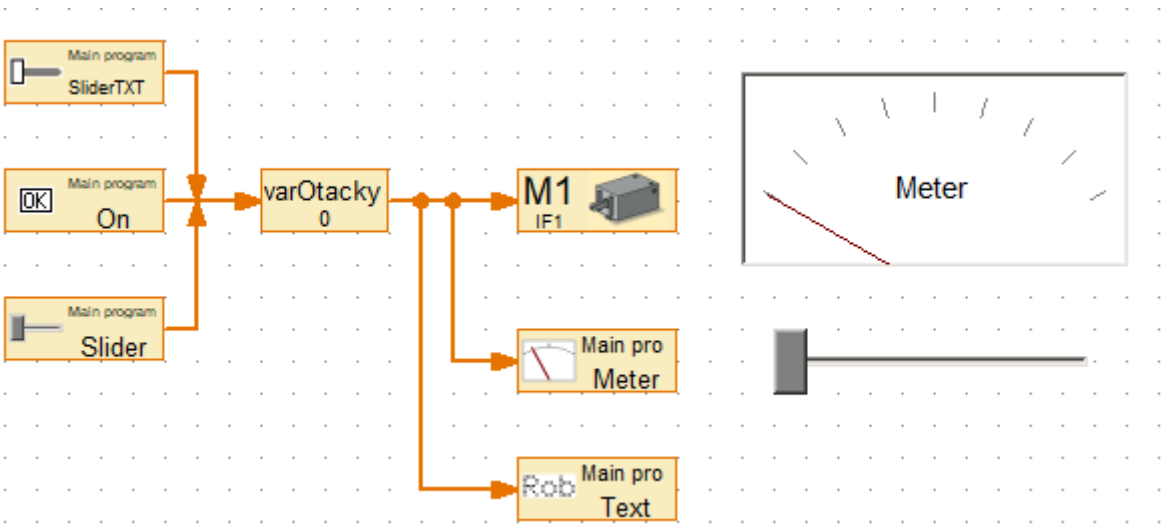
Co v tomto případě potřebujeme vědět…

GUI pro ovládání modelu na dotykovém displeji řídící jednotky se vytváří na záložce TXT/TX Display, kde vložíme požadované elementy, tedy element Display ze sekce Displays a elementy Slider a tlačítko OK. Panel pak může vypadat jako na následujícím obrázku. Elementy se nastavují obdobně jako pro zobrazení v programu, nebo na ovládacím panelu. Pro tlačítko OK je vhodné si přečíst ve slovníku elementů, jaké má vlastnosti.



Obrázek 100: Větráček II - GUI pro řídící jednotky

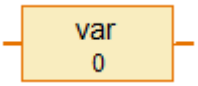
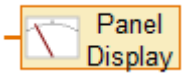
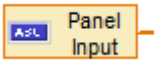
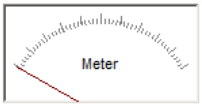
Program pak rozšíříme pouze o potřebné části. Tedy do proměnné varOtacky dva vstupy, jeden bude nastavený pro Slider na obrazovce řídící jednotky a druhý pro tlačítko na obrazovce řídící jednotky. Proměnou pak ještě nastavíme jako vstup do elementu Panel display, kterému ve vlastnostech přiřadíme Text display na obrazovce řídící jednotky (zobrazení otáček na displeji).

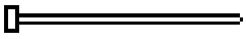


Obrázek 101: Větráček II - řídící program pro samostatný režim

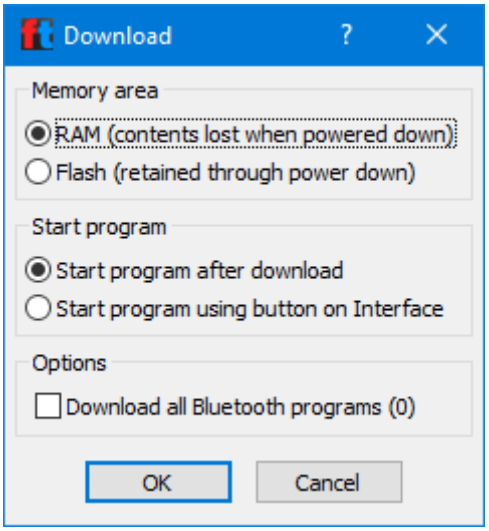
Přehled použitých elementů je v následující tabulce.

Tabulka 19: Větráček II v samostatném režimu - přehled použitých elementů

Použitý element	Základní zobrazení elementu	Počet	Umístění elementu v nabídce	Připojení na vstup (Input), výstup (Output/Motor)	Poznámka
Variable		1x	Program elements – Variables, timers…	-	-
Panel display		2x	Program elements – Inputs, outputs	-	1x meter 1x displej řídící jednotky
Motor output		1x		M1	-
Panel Input		3x		-	1x slider na monitoru 1x tlačítko na displeji řídící jednotky 1x slider na displeji řídící jednotky
Meter		1x	Operating elements - Displays	-	-

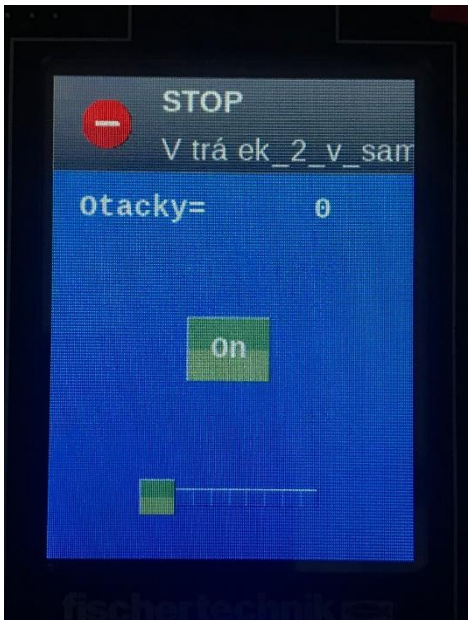
Slider		1x	Operating elements – Control elements	-	-
Display (TXT/TX Display)		1x	Interface display elements - Displays	-	Záložka TXT/TX Display, zobrazení rychlosti (0-8)
Display - Button		1x	Interface display elements – Control elements	-	Záložka TXT/TX Display, spuštění tlačítko
Slider		1x		-	Záložka TXT/TX Display, Slider

Nahrání programu do řídící jednotky se pak provede ikonou Download v horní liště programu. Zvolíme možnost nahrát program do RAM (to znamená, že po vypnutí řídící jednotky bude program zapomenut) a aby se program spustil ihned po nahrání, viz následující obrázek.



Obrázek 102: Větráček II - nahrání programu do řídící jednotky

Po nahrání programu se na displeji řídící jednotky zobrazí námi vytvoření GUI a můžete dotykem model ovládat.



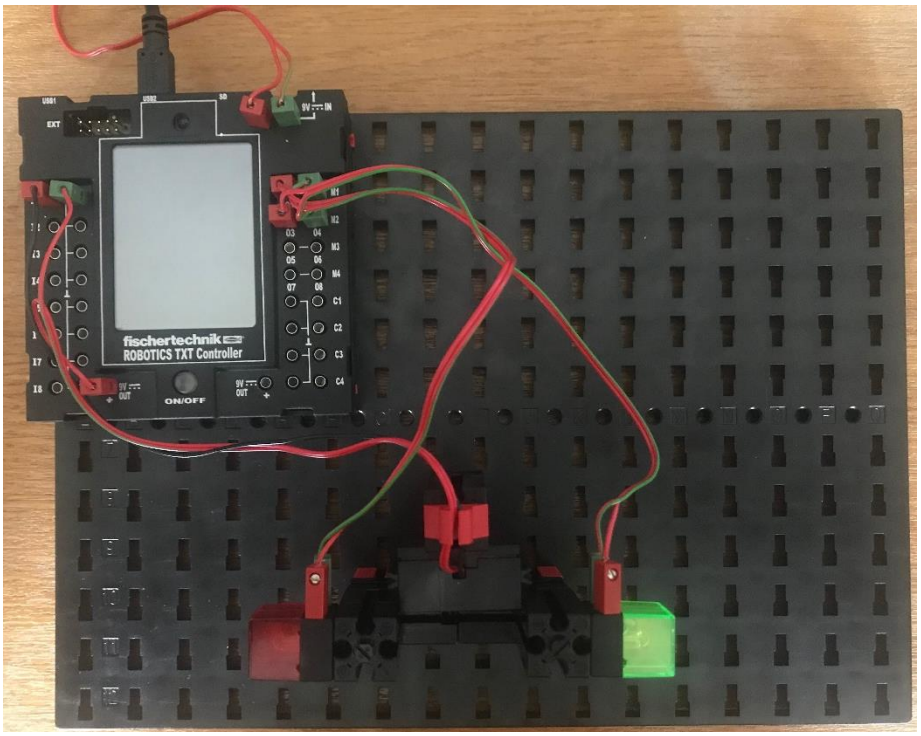
Obrázek 103: Větráček II - GUI pro ovládání dotykovým displejem na řídící jednotce

3.13 Kontrola barvy

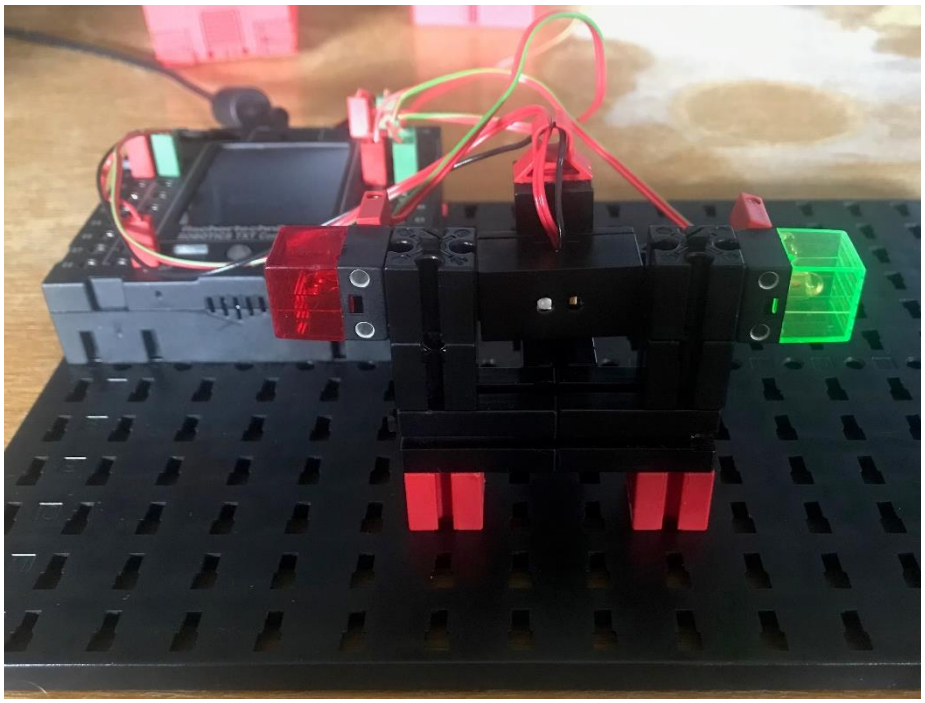
V tomto cvičení se naučíme vyhodnocovat barvy za pomoci světelného senzoru a zopakujeme si zavedení proměnné a její zobrazení pomocí Text display, abychom si mohli „nakalibrovat“ světelný senzor. Toto cvičení je možné vypracovat i při nahrazení optického barevného senzoru webkamerou. Jen by se musel model i program adekvátně upravit.

3.13.1 Kontrolor barvy – zapojení komponent

Zapojení komponent je následující, viz obrázky. Na vstup I1 a +9V je přivedený optický barevný senzor, na výstup O1/O2 je přivedena žárovka s červenou krytkou, na výstup O3/O4 je přivedena žárovka se zelenou krytkou.



Obrázek 104: Kontrola barvy - zapojení komponent, pohled shora



Obrázek 105: Kontrola barvy - zapojení komponent, pohled zpředu

3.13.2 Kontrolor barvy – řídící program

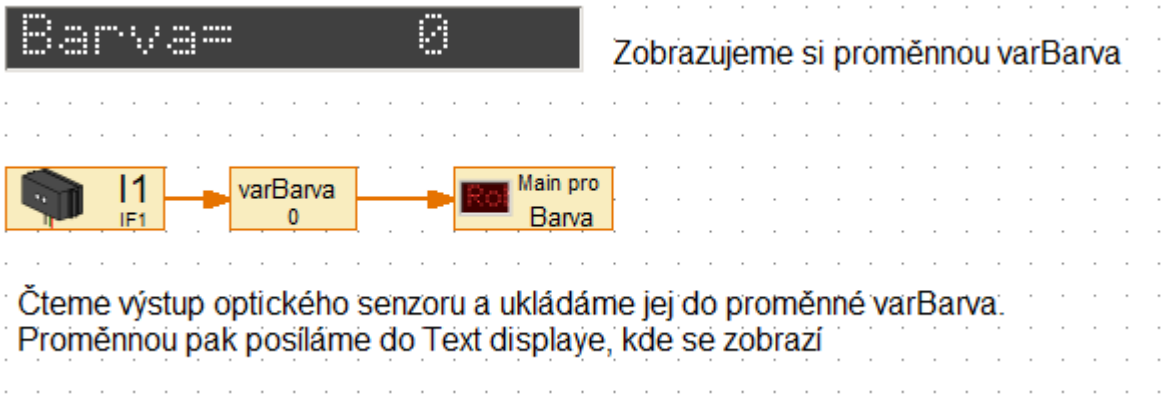
Jak chceme, aby se kontrolor barvy choval…

1. Při vložení zeleného papíru, se rozsvítí zelená kontrolka.
2. Při vložení červeného papíru se rozsvítí červená kontrolka.
3. Při nevloženém papíru, nebo vložení papíru jiné barvy než červené, nebo zelené, se neděje nic a model čeká na další vložení papíru.

4. Program bude v cyklu, aby vyhodnocování barev probíhalo neustále.

Co v tomto případě potřebujeme vědět…

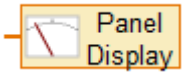
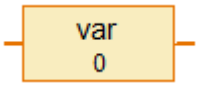
Optický barevný senzor vrací určité číslo, které odpovídá barvě, která je před ním vložena. My musíme zjistit, jaké číslo čte senzor pro červený papír a pro zelený papír. Pomůžeme si velmi jednoduchým programem, který bude pouze vypisovat číslo odečtené ze senzoru, viz obr.



Obrázek 106: Kontrolor barvy - zjištění hodnot testovaných barev pro kalibraci

Přehled použitých elementů je v následující tabulce.

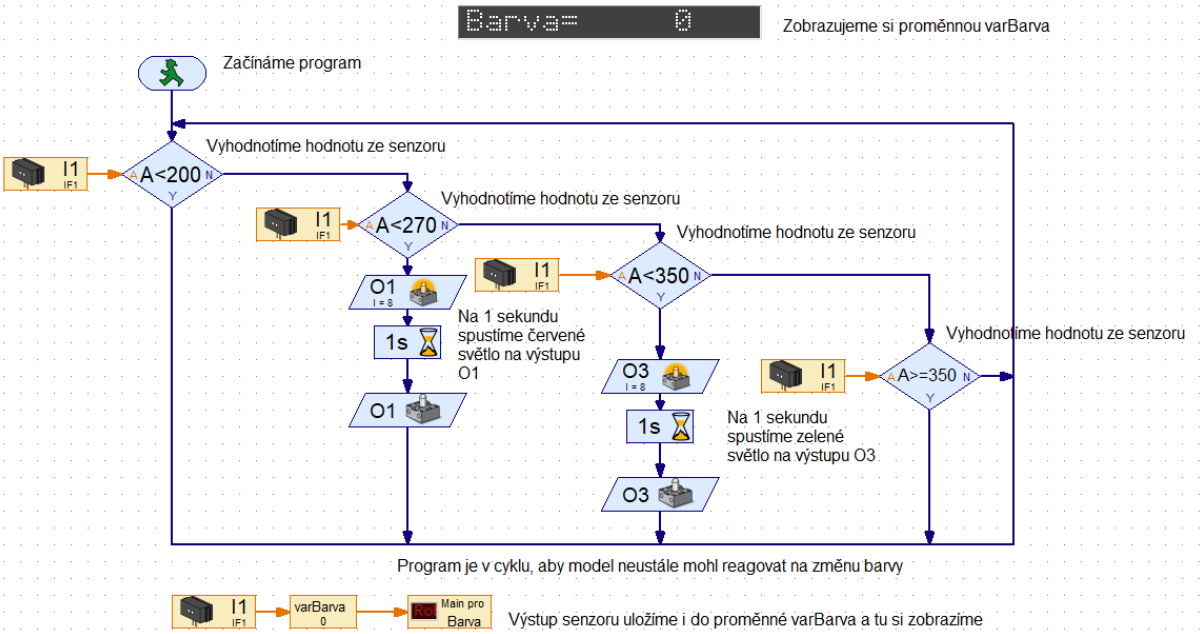
Tabulka 20: Kontrolor barvy - pomocný program - přehled použitých elementů

Použitý element	Základní zobrazení elementu	Počet	Umístění elementu v nabídce	Připojení na vstup (Input), výstup (Output/Motor)	Poznámka
Universal input		1x	Program elements – Inputs, outputs	I1	Ve vlastnostech Sensor type nastaveno na Color sensor
Panel display		1x		-	-
Variable		1x	Program elements – Variables, timers…	-	-
Text display		1x	Operating elements - Displays	-	-

Díky tomuto jednoduchému programu se zjistí, jaká čísla vrací optický senzor pro barvy, které potřebujeme vyhodnocovat. V tomto konkrétním případě při tvorbě dokumentu, který čtete, vrací optický senzor pro zelenou barvu hodnoty kolem 237 a pro zelenou barvu hodnoty kolem 325. S tím, že hodnoty kolísají přibližně ±50.

1. Známe přibližné hodnoty pro obě barvy. Abychom dosáhli výsledku, který potřebujeme, musíme hodnoty, které nám optický senzor může vracet nějakým způsobem třídit, využijeme tedy následujícího třídícího algoritmu:
    - Při odečtení hodnot optickým senzorem v intervalu (0,200) nebudeme dělat nic a budeme dále vyhodnocovat odečtenou barvu. Taková hodnota totiž neodpovídá námi kontrolovaným barvám při naší kalibraci.
    - Při odečtení hodnot optickým senzorem v intervalu (200,270) to vyhodnotíme jako červenou barvu a na 1 sekundu rozsvítíme červenou žárovku. Rozsvícení na 1s je dostačující, při delším přidržení barvy nebudeme okem vnímat, že se žárovka rozsvěcuje/zhasíná, ale budeme vnímat neustálý svit. Při odebrání barvy pak dojde nejpozději do 1s ke zhasnutí žárovky, což je pro nás dostatečně rychlá reakce systému.
    - Při odečtení hodnot optickým senzorem v intervalu (270, 350) to vyhodnotíme jako zelenou barvu a na 1 sekundu rozsvítíme zelenou žárovku, podobně, jako pro předchozí interval odečtených barev.
    - Při odečtení hodnot optickým senzorem 350+ nebudeme dělat nic a budeme dále vyhodnocovat odečtenou barvu. Taková hodnota totiž neodpovídá námi kontrolovaným barvám při naší kalibraci.
2. Uvedené hodnoty pro zelenou a červenou barvu jsou/byly platné za podmínek, které panovaly při tvorbě modelu potřebného pro tento dokument. Vámi odečtené hodnoty pro kalibraci budou určitě jiné a podle toho je potřeba upravit intervaly pro třídící (vyhodnocovací) algoritmus.

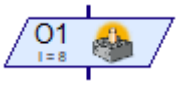
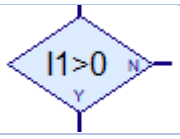
Celý program je pak na následujícím obrázku. Zobrazení odečtené barvy optickým senzorem jsem ponechal jako jeho součást. Při běhu programu tak můžeme na monitoru počítače kontrolovat, jaké hodnoty zrovna optický senzor odečítá.



Obrázek 107: Kontrolor barvy - řídící program

Přehled použitých elementů je v následující tabulce.

Tabulka 21: Kontrolor barvy -přehled použitých elementů

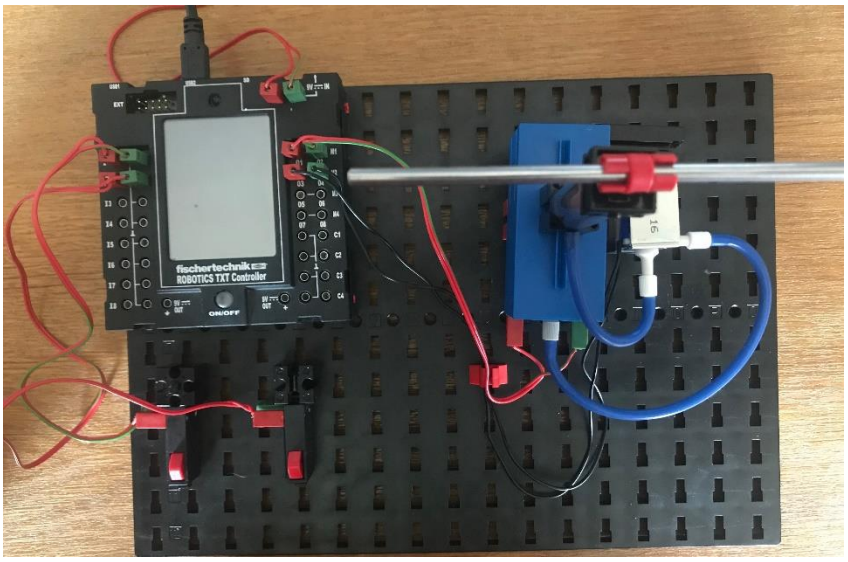
Použitý element	Základní zobrazení elementu	Počet	Umístění elementu v nabídce	Připojení na vstup (Input), výstup (Output/Motor)	Poznámka
Start		1x	Program elements – Basics elements	-	-
Lamp output		4x		O1, O3	-
Time delay		2x		-	-
Analog branch		4x		-	-
Universal input		4x	Program elements – Inputs, outputs	I1	Ve vlastnostech Sensor type nastaveno na Color sensor

3.14 Pneumatika

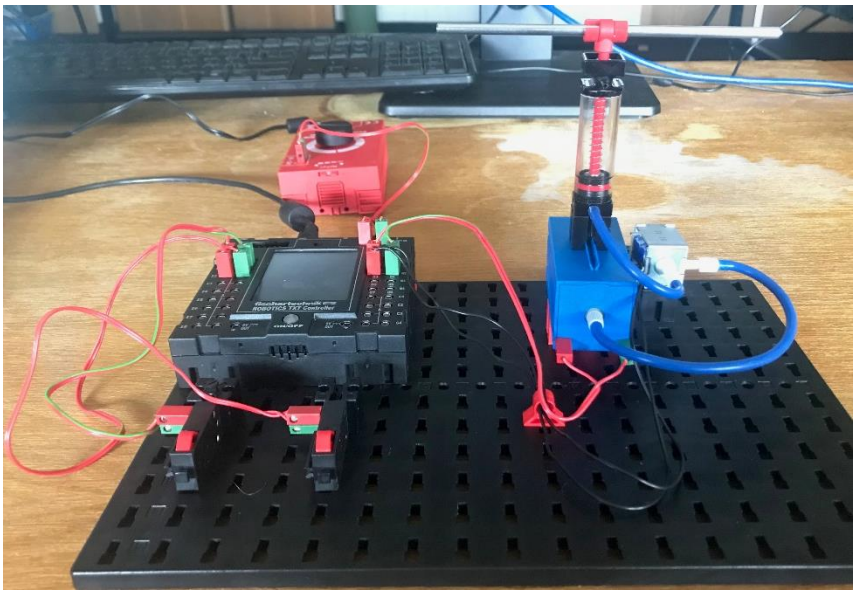
V tomto cvičení si ukážeme, že v jednom programu lze vytvořit libovolný počet funkcí, které se vykonávají současně, ovládání solenoidového ventilu a jednocestného pneumatického válce.

3.14.1 Pneumatika – zapojení komponent

Zapojení komponent je následující, viz obrázek. Na vstupy I1 a I2 jsou přivedeny spínače. Na výstup M1 je přivedený kompresor a na výstup M2 je přivedený solenoidový ventil.



Obrázek 108: Pneumatika - zapojení komponent, pohled shora



Obrázek 109: Pneumatika - zapojení komponent, pohled zpředu

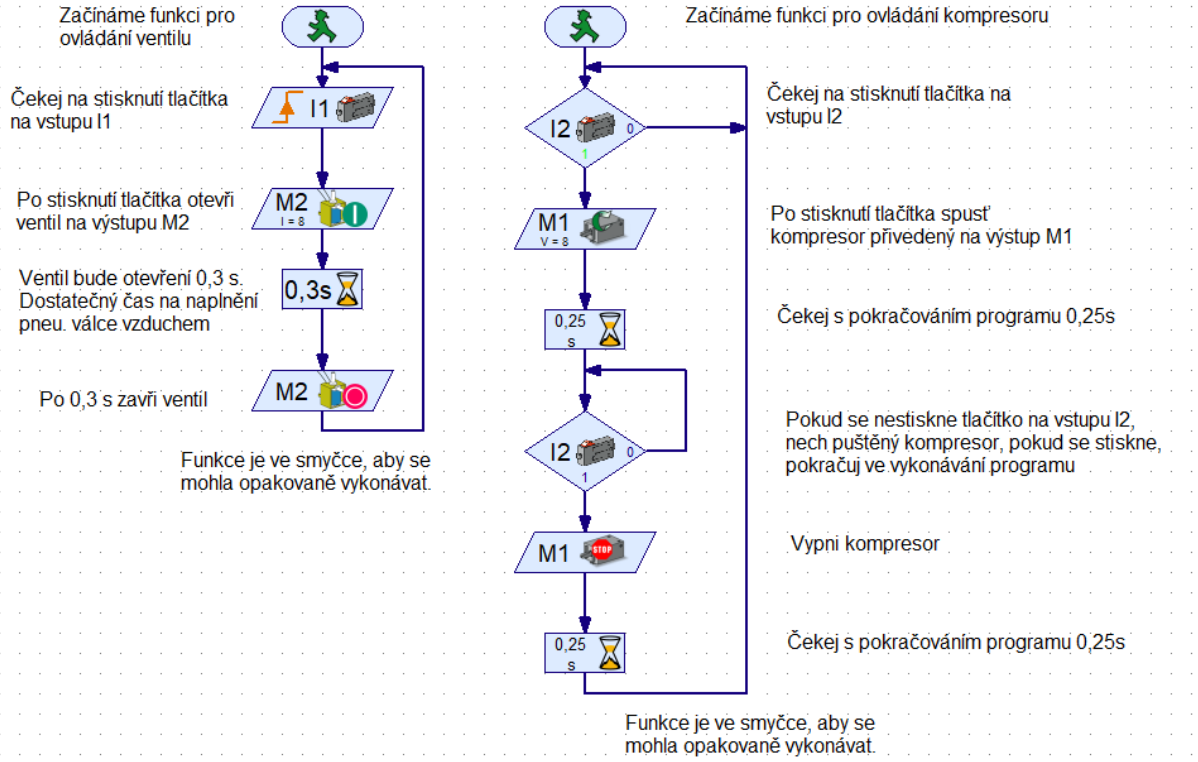
3.14.2 Pneumatika – řídící program

Jak chceme, aby se pneumatický model choval…

1. Chceme nezávisle ovládat běh kompresoru a ventil.
2. Pokud chce uživatel se systémem pracovat, musí nejprve spustit manuálně kompresor spínačem na vstupu I2. Po ukončení práce s modelem musí opětovným stisknutím spínače kompresor vypnout.
3. K ovládání ventilu bude sloužit spínač přivedený na vstup I1.
4. Obě funkce budou v cyklu, aby se dali opakovaně spouštět.

Co je potřeba vědět…

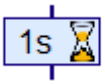
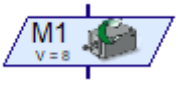
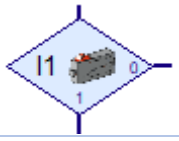
1. V tomto případě vytvoříme v Main programu dvě funkce nezávislé na sobě, viz obrázek s programem. Jednu pro ovládání kompresoru, jednu pro ovládání ventilu.
2. V případě využívání stejného tlačítka na více místech funkce, je potřeba přidávat časové prodlevy, jinak program nefunguje podle zadání, ale dochází k náhodným reakcím na stisknutí tlačítka. Proto vloženy časové prodlevy 0,25 sekundy.



Obrázek 110: Pneumatika - řídící program

Přehled použitých elementů je v následující tabulce.

Tabulka 22: Pneumatika - přehled použitých elementů

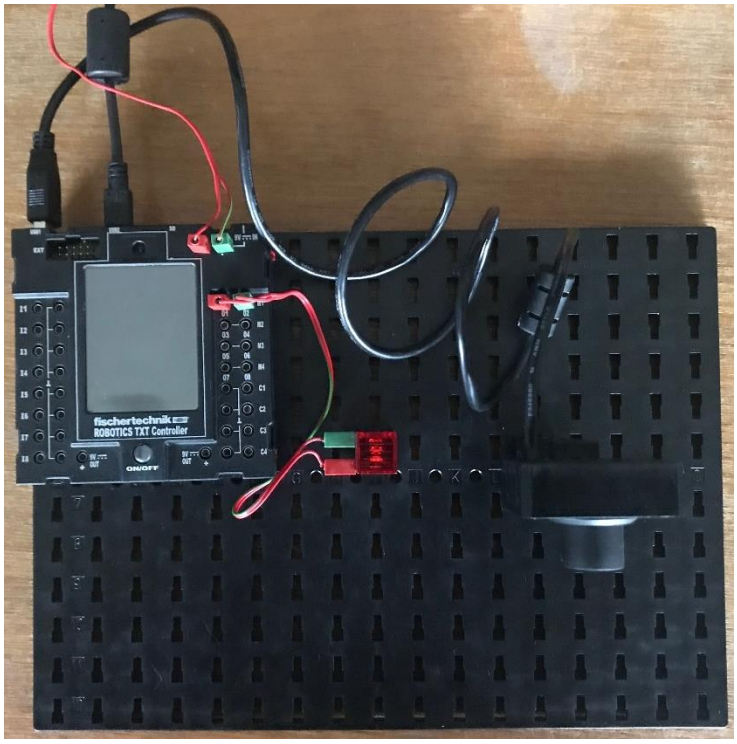
Použitý element	Základní zobrazení elementu	Počet	Umístění elementu v nabídce	Připojení na vstup (Input), výstup (Output/Motor)	Poznámka
Start		2x	Program elements – Basics elements	-	Paralelně běží 2 funkce
Time delay		3x		-	-
Wait for input		1x		I1	Nastaveno čekání na hodnotu „1“
Motor output		4x		M1, M2	M1 – 2x motor, spuštění a zastavení. M2 – 2x solenoid ventil, otevření a uzavření
Digital Branch		2x		I2	-

3.15 Alarm

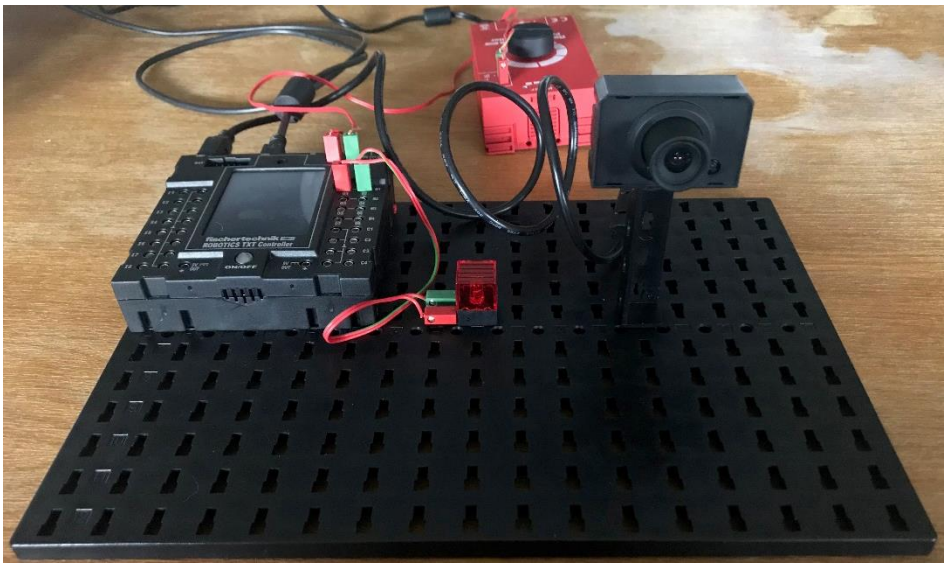
V tomto cvičení si ukážeme vyhodnocení pohybu před kamerou, využívání zvuků a zopakujeme si podprogramy.

3.15.1 Alarm – zapojení komponent

Zapojení komponent je následující, viz obrázek. Do TXT Controlleru je zapojena kamera, na výstup O1 je přivedena LED s červenou krytkou. V tomto případě určitě nepoužijeme žárovku z důvodu potřeby rychlého blikání alarmu. Žárovka by neměnila stavy tak rychle a zároveň by k ní požadované chování nebylo šetrné.



Obrázek 111: Alarm - zapojení komponent, pohled shora



Obrázek 112: Alarm - zapojení komponent, pohled zpředu

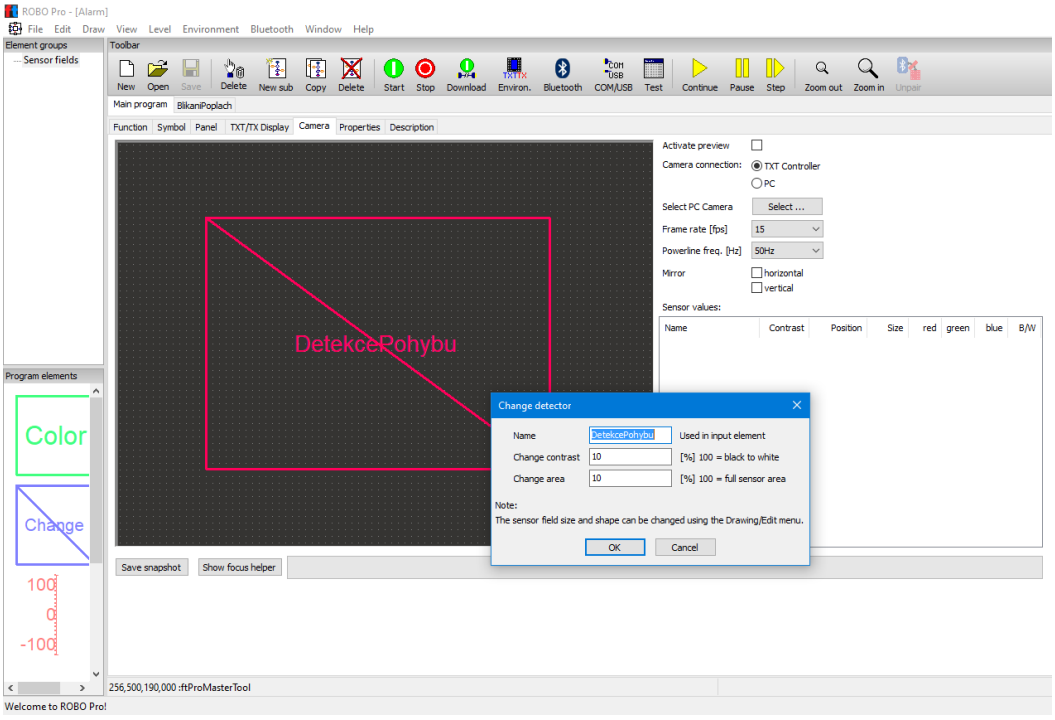
3.15.2 Alarm – řídící program

Jak chceme, aby se alarm choval…

1. Při vyhodnocení pohybu před kamerou se spustí zvuková výstraha a zároveň bude blikat červené světlo.
2. Program poběží ve smyčce, aby se neustále vyhodnocoval pohyb před kamerou a alarm mohl reagovat.

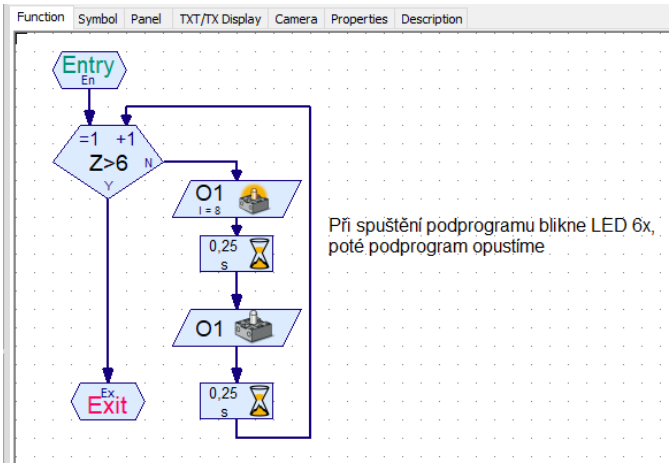
Nejprve je potřeba nastavit, co má kamera vyhodnocovat. Postup:

1. V hlavním programu, na záložce Camera, vložíme komponentu Change, kterou zakreslíme do plochy symbolizující snímaný prostor kamery. Lze si při tom zaškrtávátkem „Activate Preview“ zobrazit místo černé plochy obraz z kamery.
2. Podle toho, jakou oblast označíme (velikost, umístění), tak tuto oblast bude kamera hlídat. Přes pravé tlačítko myši nad označením vybrané plochy si pak zobrazíme možnosti nastavení a nastavíme si název oblasti a lze nastavit, jak citlivá má být kamera při vyhodnocování pohybu. Pohyb je vyhodnocován na základě změny kontrastu obrazu, předdefinovaná hodnota 10 je naprosto dostačující pro citlivé vyhodnocování.



Obrázek 113: Alarm - nastavení kamery pro využití v programu

Po nastavení kamery si vytvoříme podprogram pro blikání LED a samotný program.

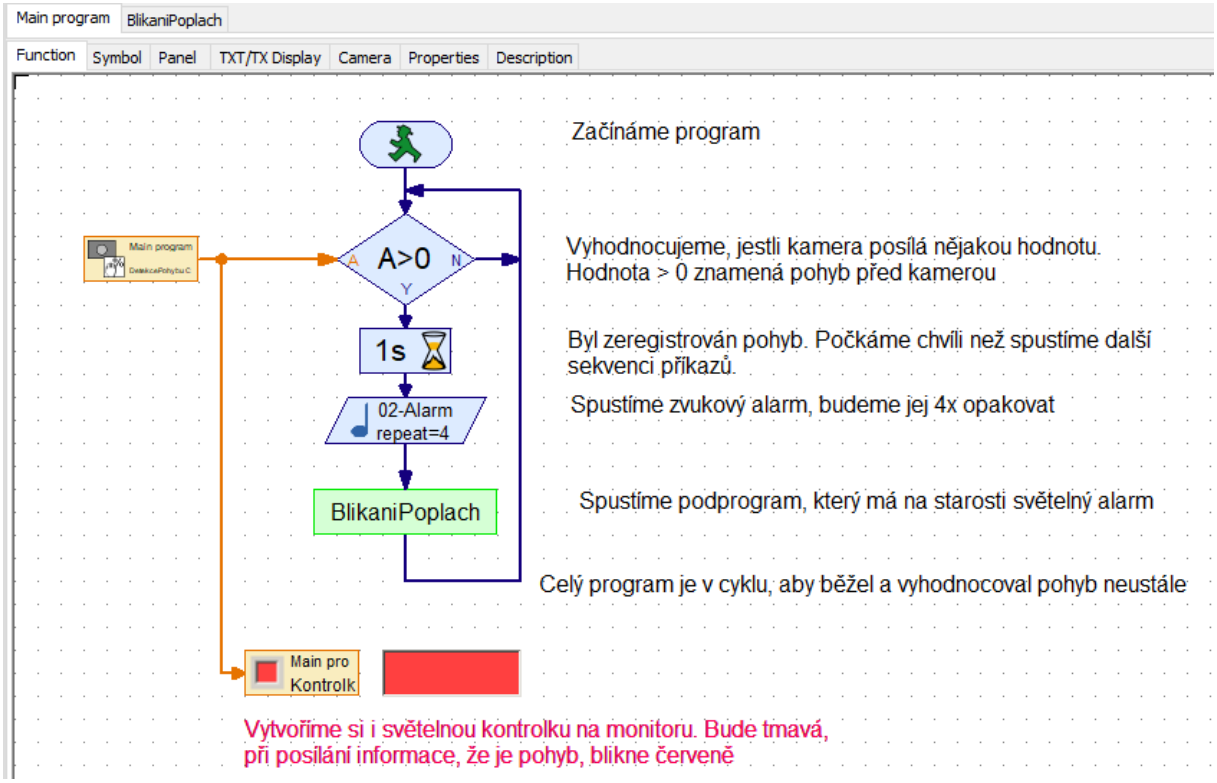


Obrázek 114: Alarm - podprogram pro blikání LED

Přehled použitých elementů je v následující tabulce.

Tabulka 23: Alarm - podprogram pro blikání LED - přehled použitých elementů

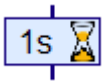
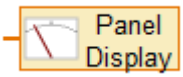
Použitý element	Základní zobrazení elementu	Počet	Umístění elementu v nabídce	Připojení na vstup (Input), výstup (Output/Motor)	Poznámka
Lamp output		2x	Program elements – Basic elements	O1	1x zapnutí, 1x vypnutí
Subprogram Entry		1x	Program elements – Subprogram I/O	-	-
Subprogram Exit		1x		-	-

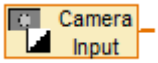
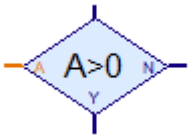
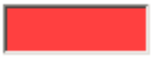
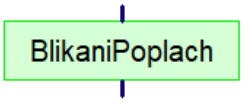


Obrázek 115: Alarm – program

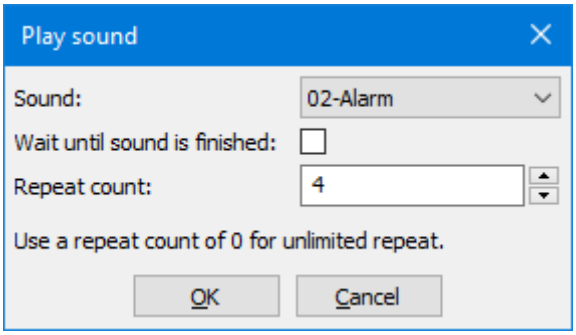
Přehled použitých elementů je v následující tabulce.

Tabulka 24: Alarm - program - přehled použitých elementů

Použitý element	Základní zobrazení elementu	Počet	Umístění elementu v nabídce	Připojení na vstup (Input), výstup (Output/Motor)	Poznámka
Start		2x	Program elements – Basics elements	-	Paralelně běží 2 funkce
Time delay		3x		-	-
Sound		1x		-	Přehrávaný zvuk 02 – Alarm, nečekat na dohrání zvuku, 4x zvuk opakovat
Panel display		1x	Program elements – Inputs, outputs	-	-

Camera Input		1x		-	-
Analog branch with data input		1x	Program elements – Branch, wait	-	-
Display lamp		1x	Operating elements - Displays	-	Navázáno na Panel display jako softwarová kontrolka
Podprogram BlikaniPoplach		1x	Loaded programs - Alarm	-	Podprogramy se nabízí v Loaded programs až po jejich vytvoření!
Movement detector		1x	Sensor fields	-	Označení oblasti na záložce Camera

Nově je tu vložena komponenta pro přehrání zvuku. V tomto případě je zvolený zvuk 02-Alarm, který se 4x opakuje a je odškrtnuta možnost „Wait until sound is finished“. To proto, aby se následující podprogram spustil ihned po spuštění zvuku a nečekalo se na jeho přehrání. Pro lidi, co mají rádi co nejjednodušší vyjadřování to začne blikat a houkat najednou.



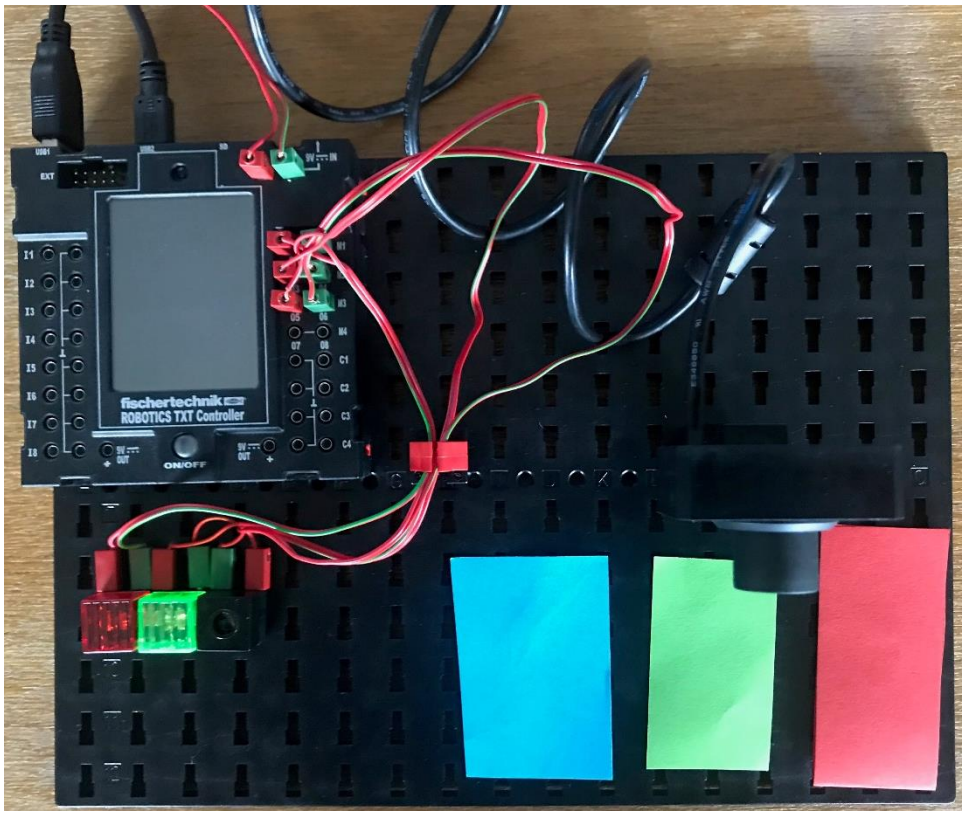
Obrázek 116: Alarm - nastavení komponenty pro přehrávání zvuku

3.16 Vyhodnocení barvy

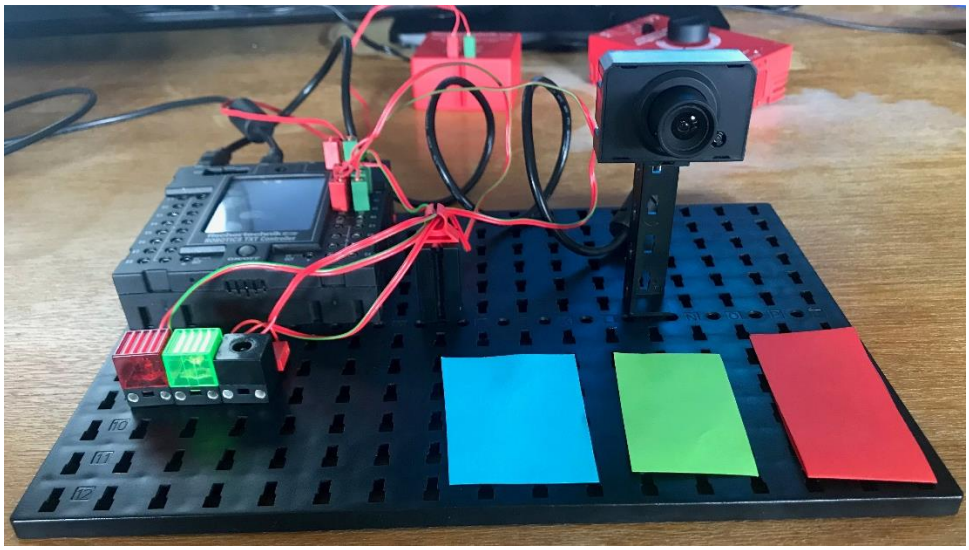
V tomto cvičení se naučíme zjišťovat převládající barvu v zorném poli kamery a zopakujeme si práci s podprogramy a matematické a logické operace.

3.16.1 Vyhodnocení barvy – zapojení komponent

Zapojení komponent je následující, viz obrázky. Do TXT Controlleru je zapojena kamera, na výstup M1 světelná komponenta s červenou krytkou, na výstup M2 světelná komponenta se zelenou krytkou a na výstup M3 světelný element s krytkou neměnící barvu.



Obrázek 117: Vyhodnocení barvy - zapojení komponent, pohled shora



Obrázek 118: Vyhodnocení barvy - zapojení komponent, pohled zpředu

3.16.2 Vyhodnocení barvy – řídící program

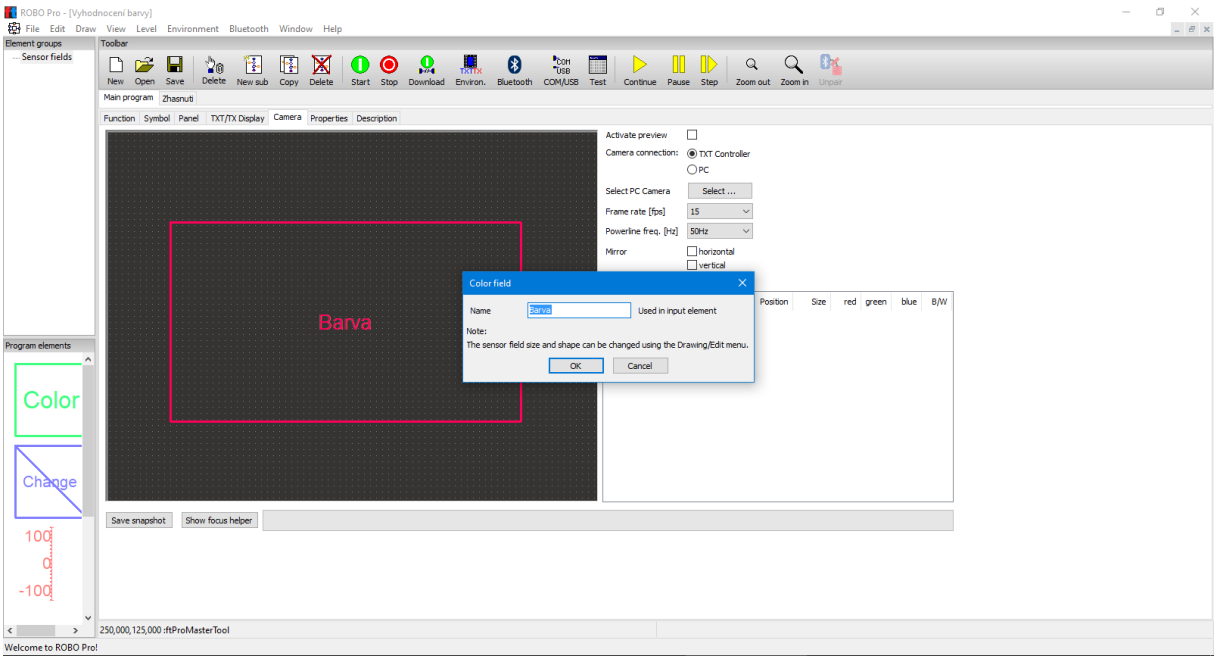
Jak chceme, aby vyhodnocování barvy probíhalo…

1. Kamera neustále pozoruje prostor před sebou a vyhodnocuje, která barva v daném prostoru převládá
2. Pokud převládá červená barva, rozsvítí se světelná komponenta s červenou krytkou přivedená na výstup M1.
3. Pokud převládá zelená barva, rozsvítí se světelná komponenta se zelenou krytkou přivedená na výstup M2.

4. Pokud převládá modrá barva, rozsvítí se světelná komponenta přivedená na výstup M3.
5. Hodnoty barev zobrazíme i na monitoru pomocí elementu Text Display
6. Program poběží ve smyčce a proto bude docházet k neustálému vyhodnocování barev.

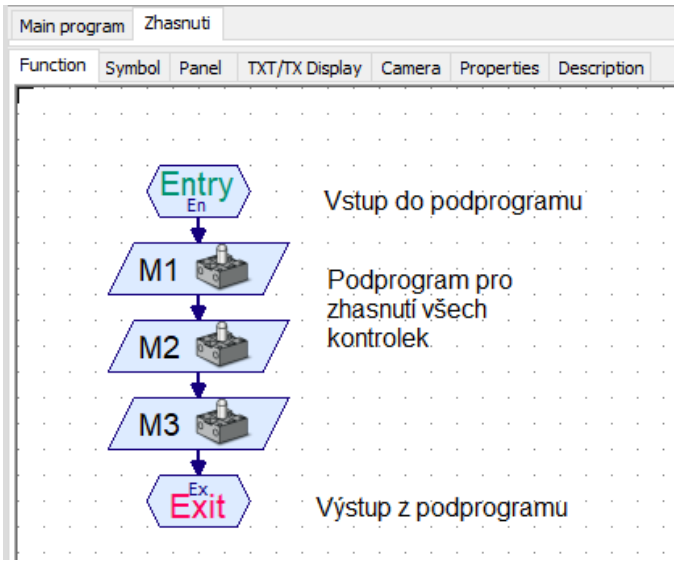
Nejprve je potřeba nastavit, co má kamera vyhodnocovat. Postup:

1. V hlavním programu, na záložce Camera, si tedy vložíme komponentu Color, kterou zakreslíme do plochy symbolizující snímaný prostor kamery. Lze si při tom zaškrtávátkem „Activate Preview“ zobrazit místo černé plochy obraz z kamery.
2. Podle toho, jakou oblast označíme (velikost, umístění), tak tuto oblast bude kamera hlídat. Přes pravé tlačítko myši nad označením vybrané plochy si pak zobrazíme možnosti nastavení, kde si můžeme oblast přejmenovat.



Obrázek 119: Vyhodnocení barvy - nastavení kamery pro využití v programu

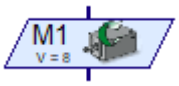
Po nastavení kamery vytvoříme podprogram, který bude vypínat všechny světelné komponenty a samotný program.

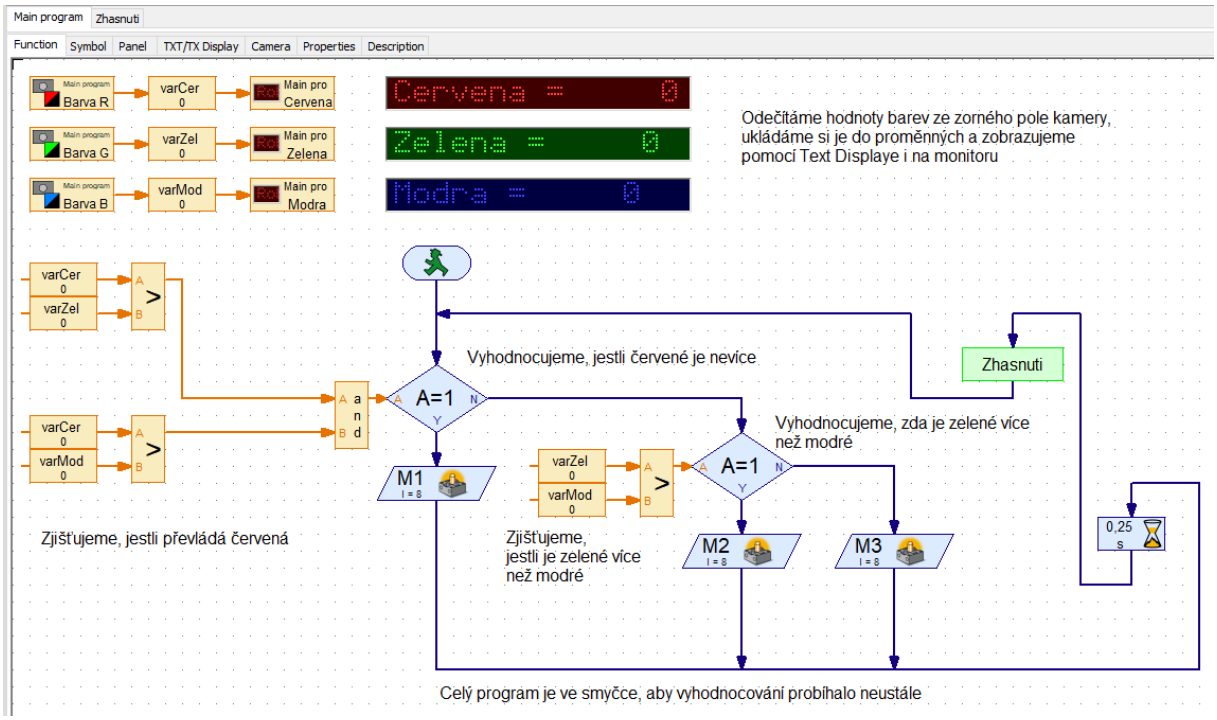


Obrázek 120: Vyhodnocení barvy - podprogram pro zhasnutí světelných elementů

Přehled použitých elementů je v následující tabulce.

Tabulka 25: Vyhodnocení barvy - podprogram pro zhasnutí - přehled použitých elementů

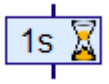
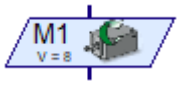
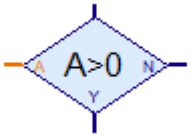
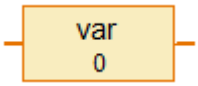
Použitý element	**Základní zobrazení elementu**	**Počet**	**Umístění elementu v nabídce**	**Připojení na vstup (Input), výstup (Output/Motor)**	**Poznámka**
Subprogram Entry		1x	Program elements – Subprogram I/O	-	-
Subprogram Exit		1x		-	-
Motor output		3x	Program elements – Basics elements	M1, M2, M3	Zhasnutí všech kontrolek (světelných komponent)

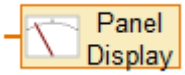
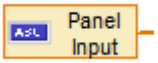
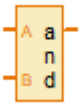


Obrázek 121: Vyhodnocení barvy - hlavní program

Přehled použitých elementů je v následující tabulce.

Tabulka 26: Vyhodnocení barvy - program - přehled použitých elementů

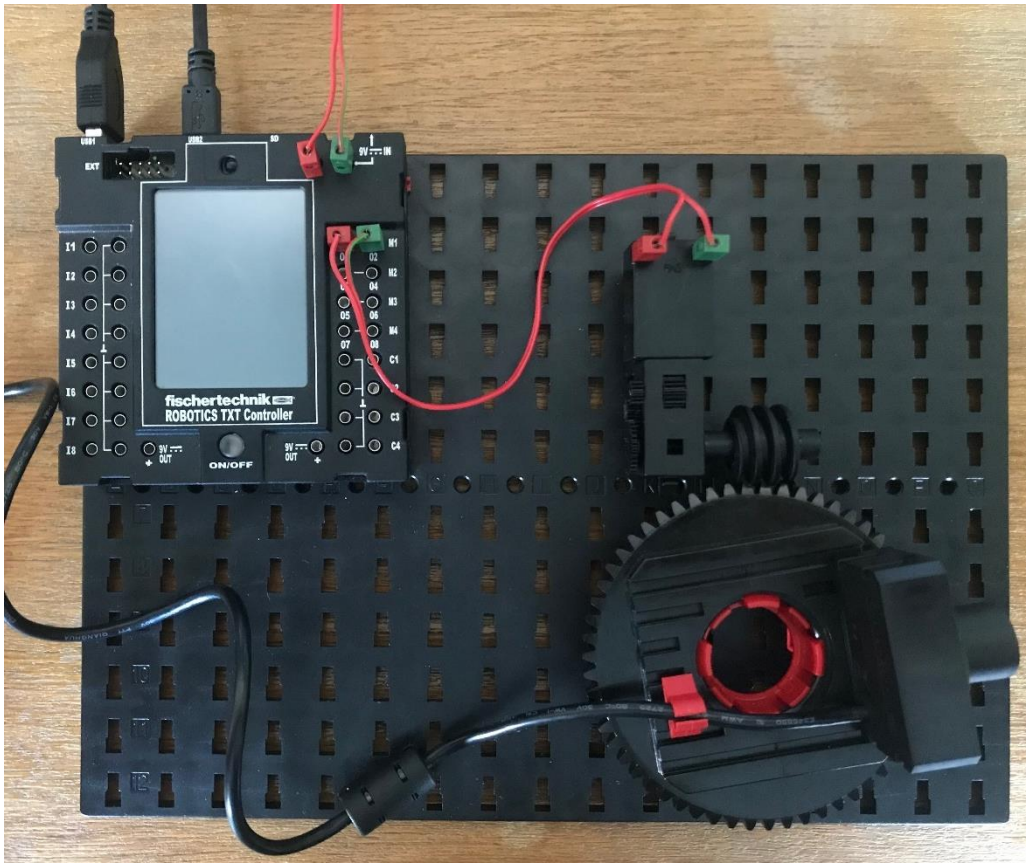
Použitý element	Základní zobrazení elementu	Počet	Umístění elementu v nabídce	Připojení na vstup (Input), výstup (Output/Motor)	Poznámka
Start		1x	Program elements – Basics elements	-	Paralelně běží 2 funkce
Time delay		1x		-	-
Motor output		3x		M1, M2, M3	Nastavené spouštění daných světelných komponent
Analog branch with data input		2x	Program elements – Branch, wait…	-	-
Variable		9x	Program elements – Variables, timers…	-	-

Panel display		3x	Program elements – Inputs, outputs	-	-
Panel Input		3x	Program elements – Inputs, Outputs	-	-
Text display		3x	Operating elements - Displays	-	-
Operator AND a Operator greater		4x	Program elements - operators	-	3x greater („>“) 1x and

3.17 Vyhledávání objektu

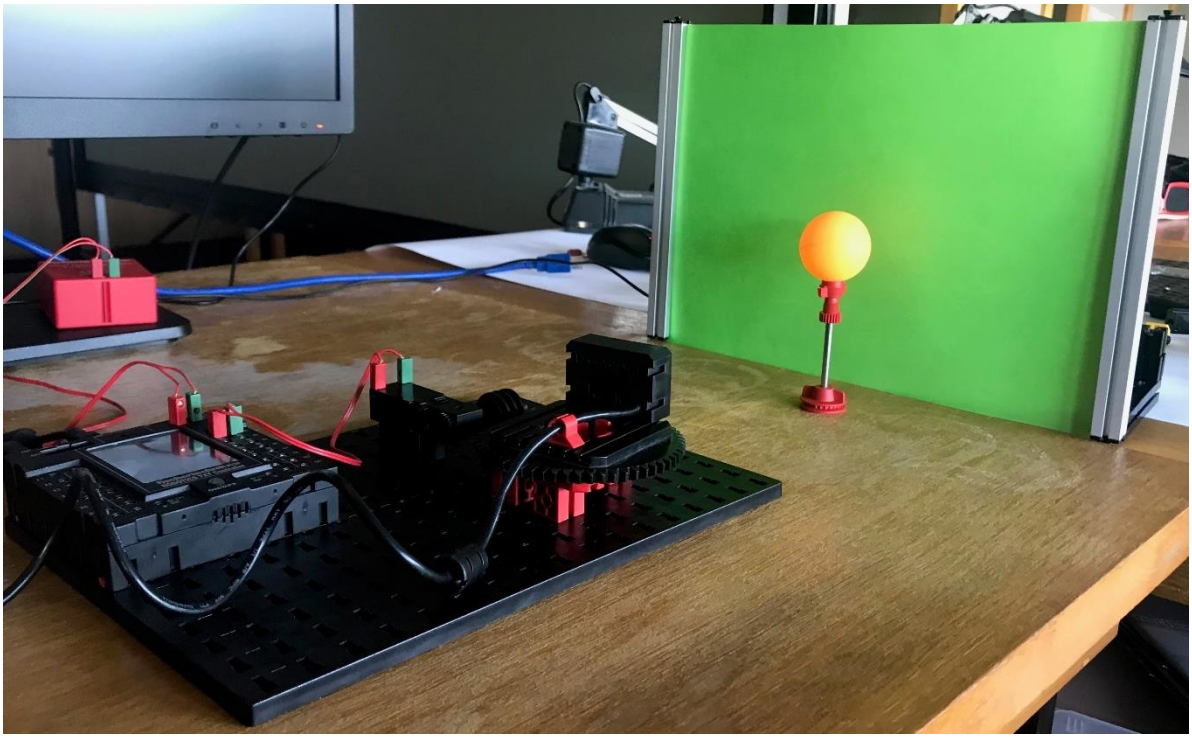
V tomto cvičení se naučíme vyhledávat objekt pomocí USB kamery, vyloučit prostor, který nechceme pomocí kamery vyhodnocovat a zopakujeme si práci s text display a ovládacími elementy. Kontrolor barvy – zapojení komponent

Zapojení komponent je následující, viz obrázek. Na výstup M1 je přivedený motor s převodovkou, která má na výstupu šnek. Šnek je pak v kontaktu s ozubením otočného „stolu“, na kterém je umístěna USB kamera.



Obrázek 122: Vyhledávání objektu - zapojení komponent

K testování modelu ještě potřebujeme míček s podstavcem a je vhodné použít jednobarevné pozadí, které nekoresponduje s barvou míčku, viz obr.



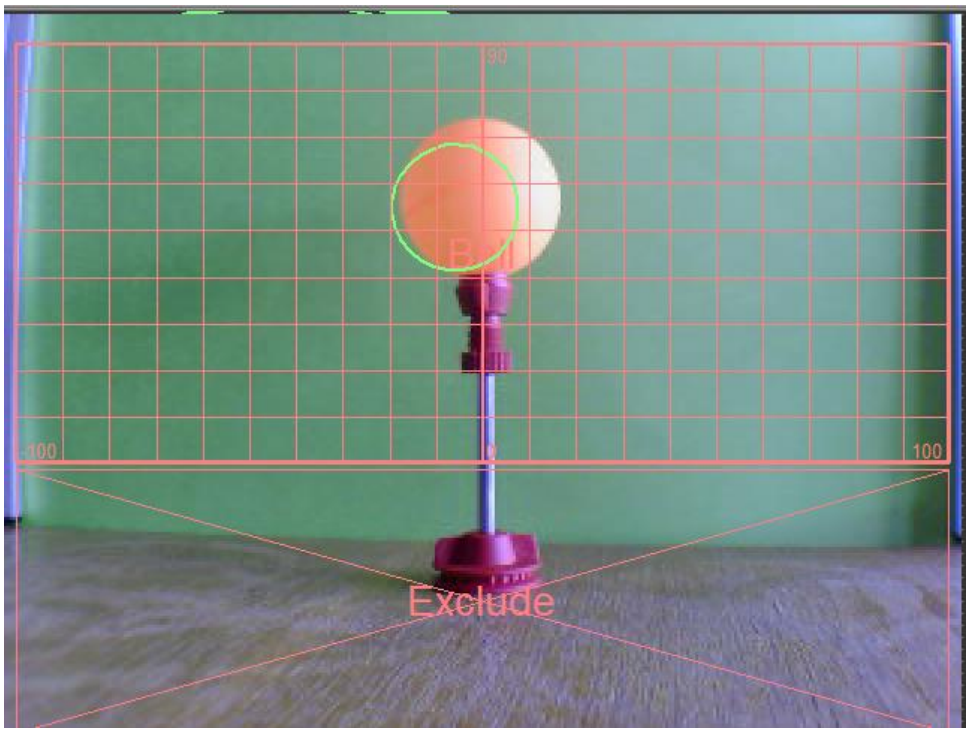
Obrázek 123: Vyhledávání objektu - kompletní model

3.17.1 Nastavení Camera

Nejprve je nutné na záložce Camera nastavit, jaké oblasti a jakým způsobem má kamera vyhodnocovat. Využijeme elementy:

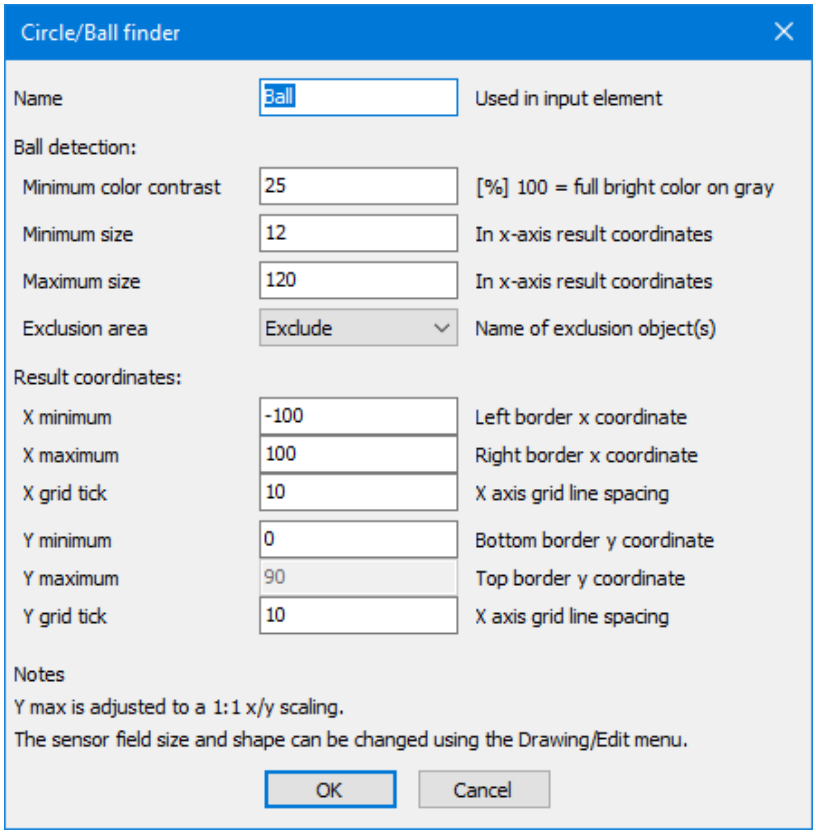
- **Exclude**, pro vyloučení spodní části obrazu z vyhodnocování
- **Ball finder**, pro vyhledávání objektu (míčku) v požadované části obrazu snímaného USB kamerou

Při jejich vytváření je vhodné zapnout si „Activate preview“, který usnadní výběr vhodných oblastí. V tomto případě jsem volil oblasti viz následující obrázek.



Obrázek 124: Vyhledávání objektu - nastavení panelu Camera

Elementu **Ball finder** nastavíme **Exclusion area**, jinak jej necháme ve výchozím nastavení, viz obr. V případě potřeby, nebo špatné detekce objektu lze upravit chováním změnou parametrů. Popis parametrů viz Ball finder.



Obrázek 125: Vyhledávání objektu - nastavení Ball finder

3.17.2 Kontrolor barvy – pomocná část programu

V tomto případě si vytvoříme jednoduchý pomocný program, který nám bude vypisovat všechny zajímavé hodnoty z vyhodnocení obrazu USB kamerou a zobrazí nám i obraz, který kamera snímá. Pokud je stále aktivovaný stav „Activate preview“, tak se nám bude zobrazovat v programu místo černé plochy přímo obraz snímaný kamerou (aniž bychom náš program spustili).

Program se skládá pouze z elementů **Camera input**, které předávají hodnoty elementům **Panel display** a ten je zobrazuje elementem **Text display**. Navíc je přidán element **Camera viewer**, abychom viděli obraz snímaný kamerou.

Obrázek 126: Vyhledávání objektu - pomocná část programu

Po spuštění tohoto programu a posunováním stojánku s míčkem před kamerou se nám zobrazují hodnoty odečítané naším pomocným programem.

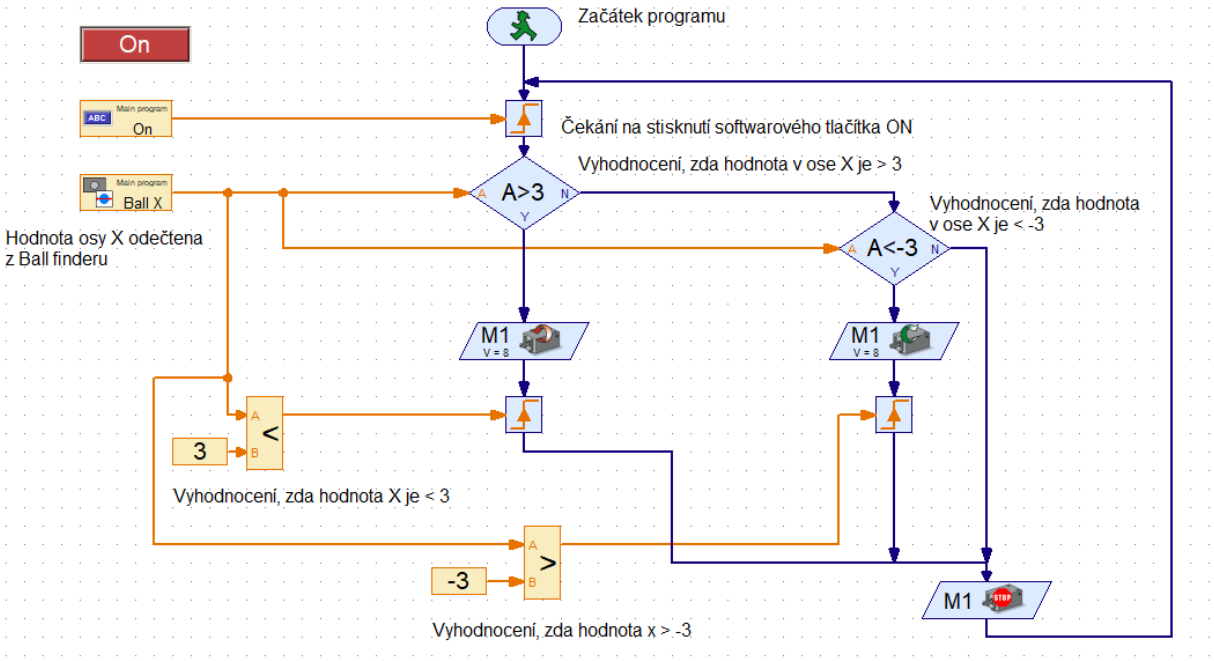
3.17.3 Vyhledávání objektu – řídící program

Jak chceme, aby se model choval…

1. Program se spustí stisknutím softwarového tlačítka.
2. Při stisknutí softwarového tlačítka program vyhledá míček a upraví pozici kamery tak, aby byl míček přibližně ve středu snímaného prostoru (ve středu obrazu v rámci osy X).
3. Program bude v cyklu, aby se mohl opakovaně spouštět softwarovým tlačítkem.
4. Stanovíme si, že chceme objekt v rámci osy X vystředit do +- 3 bodů (přesnost je potřeba brát s rezervou, je ovlivněna vzdáleností od kamery, rozlišením kamery, kontrastem míčku proti

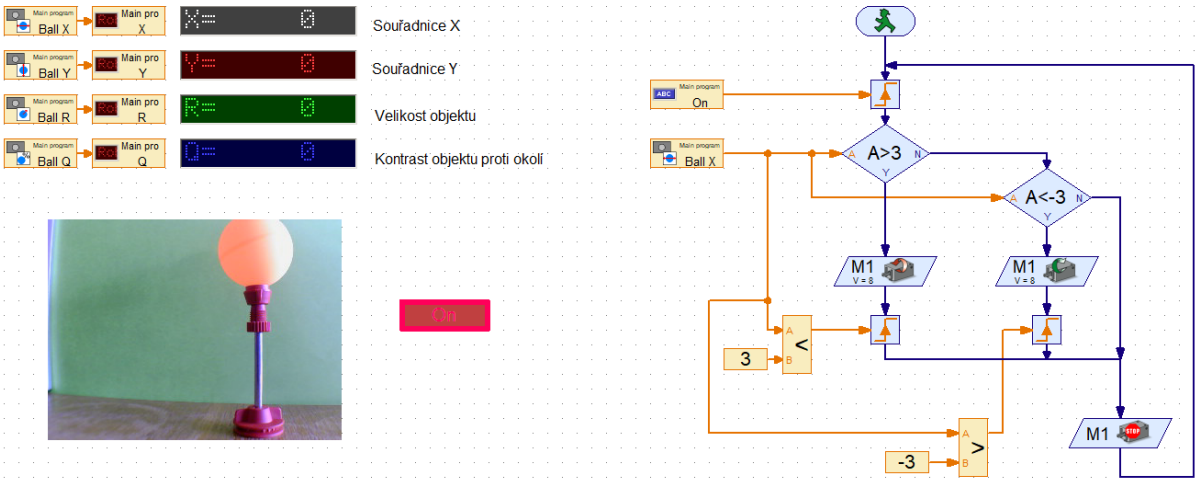
pozadí, použitým elektromotorem,…). Proto souřadnice X zobrazovaná pomocí Text displeje nebude odpovídat hodnotám v intervalu (-3,3).

Řídící program pak může vypadat následovně:



Obrázek 127: Vyhledávání objektu - řídící program

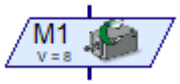
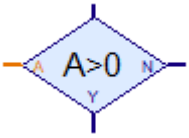
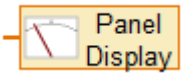
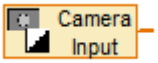
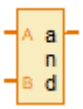
Řídící program včetně pomocné části programu můžeme mít dohromady, viz následující obr. To nám umožní kontrolovat si hodnoty získané z obrazu USB kamery, ale i samotný obraz snímaný kamerou.

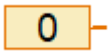


Obrázek 128: Vyhledávání objektu - kompletní program včetně pomocné části programu

Přehled použitých elementů je v následující tabulce.

Tabulka 27: Vyhledávání objektu - přehled použitých elementů

Použitý element	Základní zobrazení elementu	Počet	Umístění elementu v nabídce	Připojení na vstup (Input), výstup (Output/Motor)	Poznámka
Start		1x	Program elements – Basics elements	-	Paralelně běží 2 funkce
Motor output		3x		M1	-
Analog branch with data input		2x	Program elements – Branch, wait	-	-
Wait for…		3x		-	Čekání na splnění podmínky pro spuštění navazující části programu
Text display		4x	Operating elements - Displays	-	-
Button		1x	Operating elements – Control elements	-	Pro spuštění
Panel display		4x	Program elements – Inputs, outputs	-	-
Panel Input		1x		-	-
Camera Input		4x		-	-
Logical operator (Less)		2x	Program elements - Operators	-	Jakýkoliv logický, nebo mat. operátor nastavený na „<“
Camera viewer		1x	Operating elements – Camera viewer	-	Zobrazení náhledu kamery

Constant		2x	Program elements – Variables, timers…	-	Pro nastavení přesnosti

4 Slovníček nejčastěji používaných elementů (objektů) při tvorbě programů

Proces/program se tvoří sestavováním elementů a nastavováním jejich vlastností. Elementy se vkládají z menu po pravé straně ROBO Pro, kde je několik sekcí (Program elements, Operating elements,…). Vlastnosti zvoleného elementu se nastavují kliknutím pravého tlačítka myši na element vložený do procesu/programu. Na každém obrázku elementu je vidět i menu pro nastavení jeho vlastností. Kapitola je rozdělena a strukturována stejně jako elementy v nástroji ROBO Pro pro snazší orientaci a práci s programovacím prostředím ROBO Pro.

4.1 Program elements

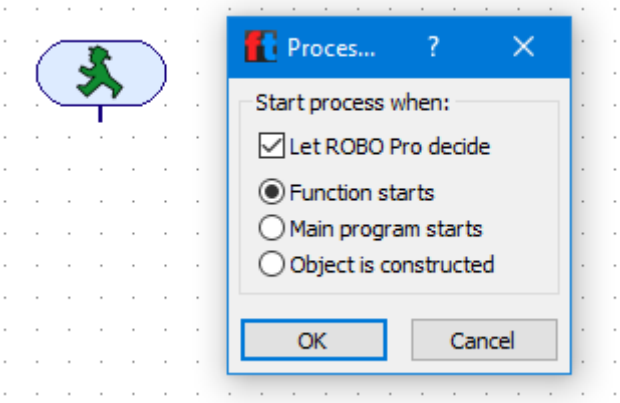
Sekce obsahující elementy pro tvorbu procesu/programu (podmínky, proměnné, práce se vstupy/výstupy,…).

4.1.1 Basic elements

Jedná se o nabídku nejzákladnějších elementů pro tvorbu programů.

4.1.1.1 Process start

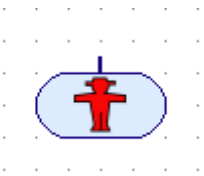
Každý program, nebo proces musí nějakým způsobem začínat, k tomu slouží element Start. V nastavení elementu se volí, kdy se má daný proces spustit, viz obr.



Obrázek 129: Start

4.1.1.2 End

Stejně jako každý proces/program začíná, tak i může končit (pokud neběží v cyklu). Pro to slouží element End.

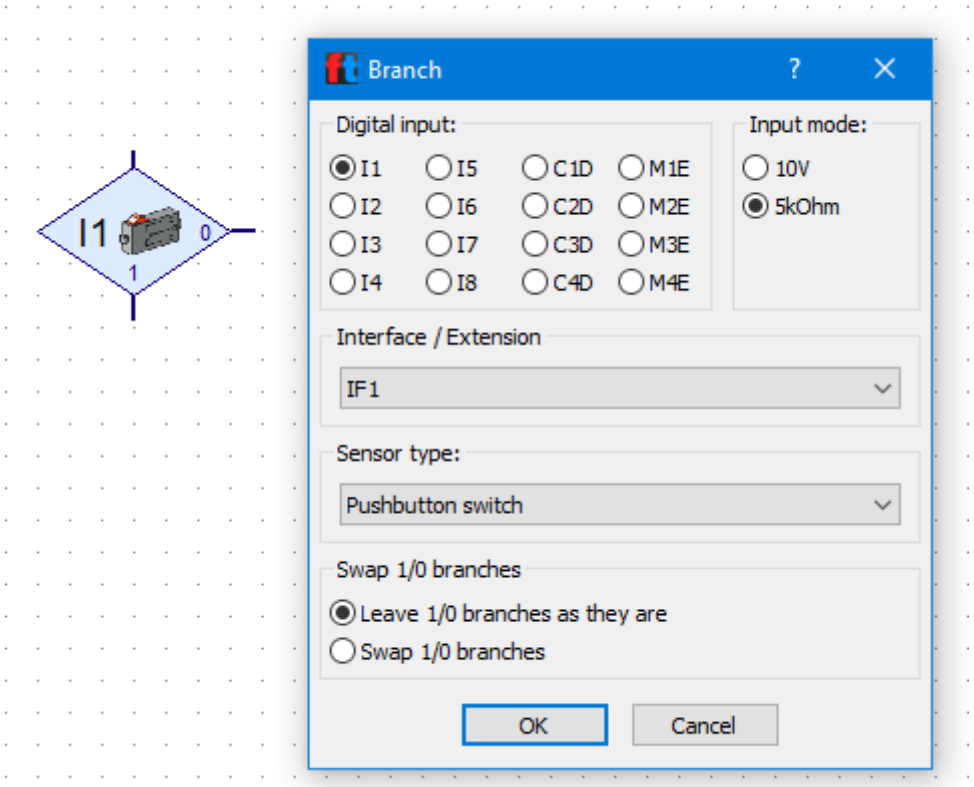


Obrázek 130: End

4.1.1.3 Digital branch (digitální větvení)

Element Digital branch, slouží k digitálnímu větvení programu. Přijímá tedy hodnoty 1/0 a podle přijaté hodnoty rozhoduje o tom, jaká větev v programu se bude dále vykonávat. Komponenta může být připojena na vstupy I1 – I8, C1-C4 a M1 – M4, podle typu komponenty. Pomocí rozbalovacího seznamu „Sensor type“ lze zvolit, jestli se pracuje se spínačem, nebo fototranzistorem. Ostatní možnosti jsou

s naší aktuální stavebnicí nevyužitelné. Možnost „Swap 1/0 branches“ slouží k prohození výstupů 1 a 0 elementu.

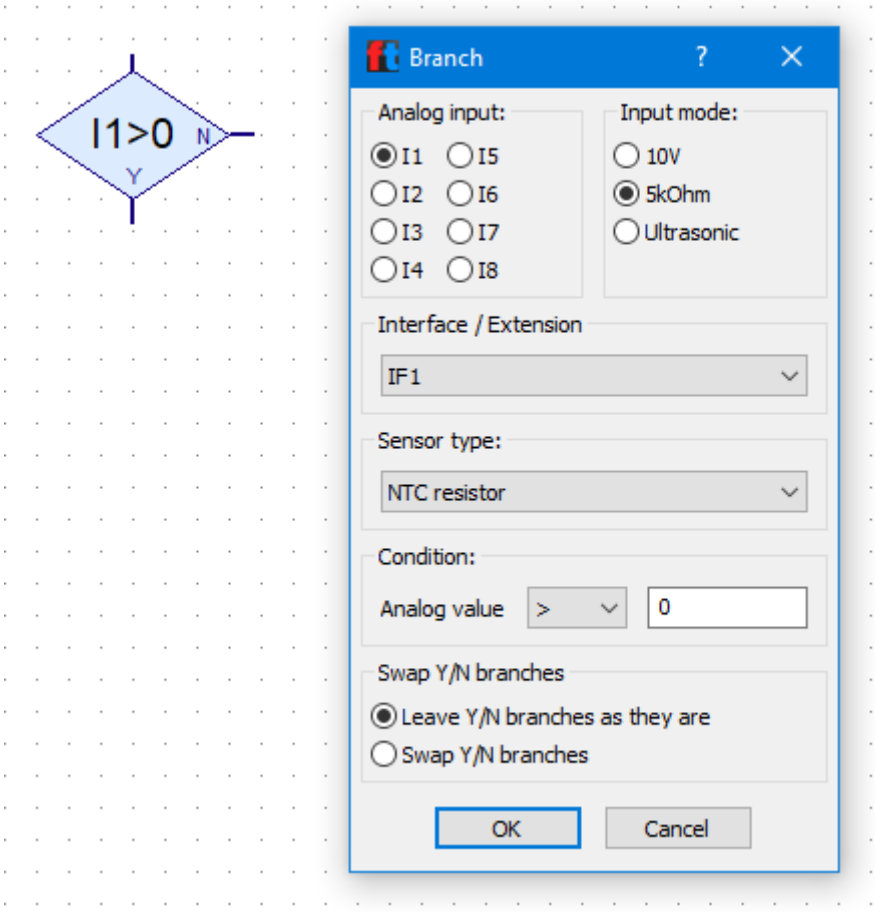


Obrázek 131: Digital branch

4.1.1.4 Analog branch

Element Analog branch, slouží k analogovému větvení programu. Nastaví se tedy podmínka a element vyhodnocuje, jestli je splněna - podle výsledku vyhodnocení rozhoduje o tom, jaká větev v programu se bude dále vykonávat. Komponenta může být připojena na vstupy I1 – I8.

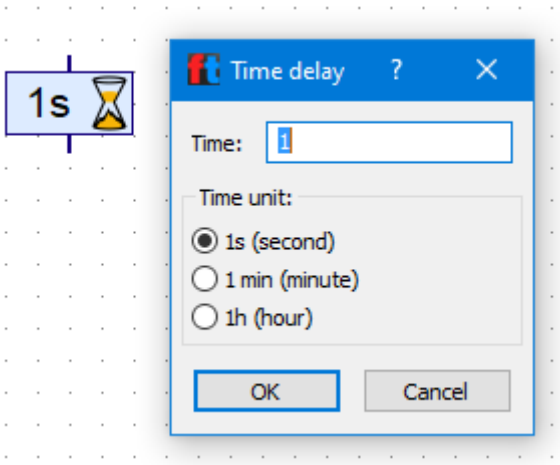
Pomocí rozbalovacího seznamu „Sensor type“ lze zvolit, jestli se pracuje s hodnotami z NTC rezistorem, fototranzistorem, nebo optickým detektorem barev. Ostatní možnosti jsou s naší aktuální stavebnicí nevyužitelné. Pod položkou „Condition“ lze zadat podmínku, kterou element vyhodnocuje. Možnost „Swap 1/0 branches“ slouží k prohození výstupů 1 a 0 elementu.



Obrázek 132: Analog branch

4.1.1.5 Time delay

Element Time delay slouží k vložení časové prodlevy do programu. Lze mu nastavit číselnou hodnotu a zvolit, jestli je v sekundách, minutách, nebo hodinách.



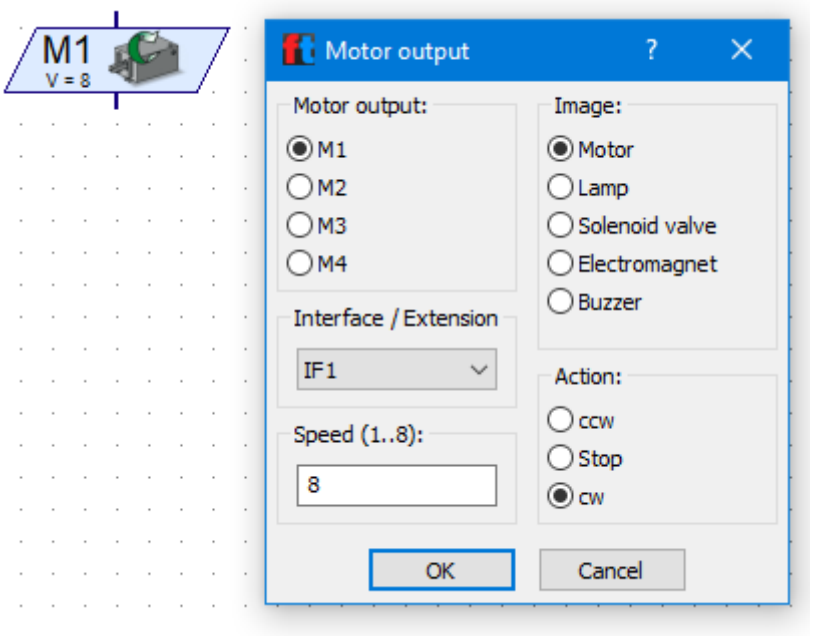
Obrázek 133: Time delay

4.1.1.6 Motor output

Element slouží k ovládání komponent připojených na výstupy M1 – M4 a může ovládat komponenty Mini motor, XS motor, Krokový motor (pokud je bez řízení otáček), světelné komponenty (LED i žárovka) a selenoidový ventil. Ostatní možnosti jsou s naší aktuální stavebnicí nevyužitelné. Pro daný element lze nastavit potřebné vlastnosti, tedy:

- Pro motory rychlost otáček, a akci (směr otáček, popř. zastavení motoru).
- Pro světelné komponenty intenzitu světla a zapnut, popř. vypnutí.
- Pro ventil zapnutí/vypnutí ventilu.

Po nastavení volby „Image“ na možnost odpovídající ovládané komponentě se obrázek elementu v programu změní podle této volby.

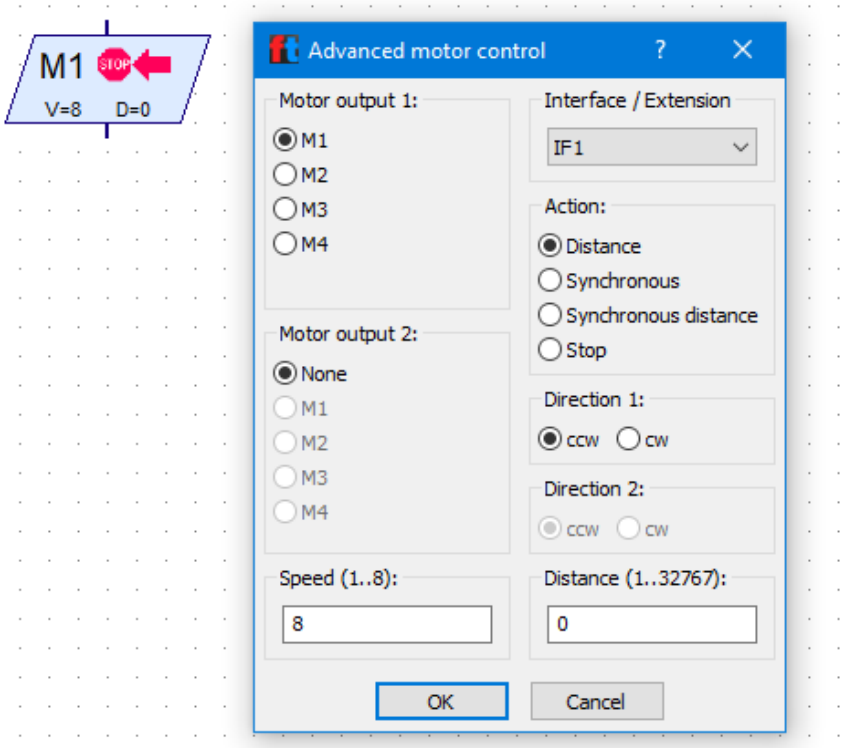


Obrázek 134: Motor output

4.1.1.7 Encoder motor output

Element slouží k řízení krokového motoru, nebo motorů. Ovládají se jím krokové motory na výstupech M1 – M4.

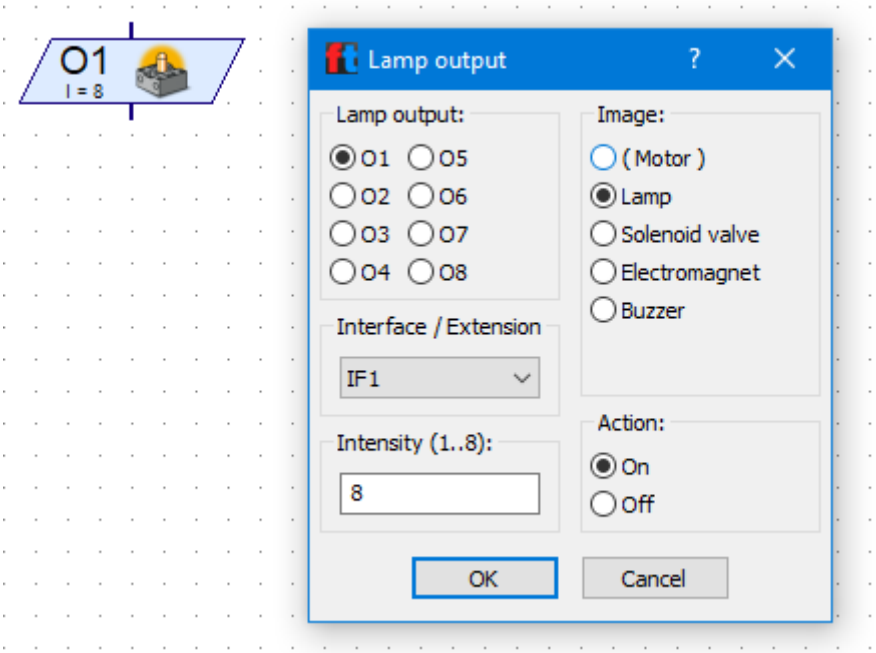
- Action – možnost nastavení chování motoru, konkrétně možnost běhu motoru po určitou vzdálenost (je potřeba si jej nakalibrovat), synchronnost s jiným motorem, synchronní vzdálenost běhu s jiným motorem a zastavení motoru.
- Motor output 1 – Volba, na který výstup M je motor přivedený
- Motor output 2 – V případě volby synchronnosti motoru se zde zpřístupní možnost zvolit, na který výstup M je přivedený druhý krokový motor.
- Speed – nastavení rychlosti otáček motoru v rozsahu 1 – 8.
- Direction 1 – nastavení směru otáček motoru.
- Direction 2 – nastavení směru otáček druhého motoru v případě volby synchronního běhu.
- Distance – nastavení vzdálenosti, kterou má motor ujet. Zadává se číselnou hodnotou a je potřeba si hodnotu vždy zjistit kalibrací.



Obrázek 135: Encoder motor output

4.1.1.8 Lamp output

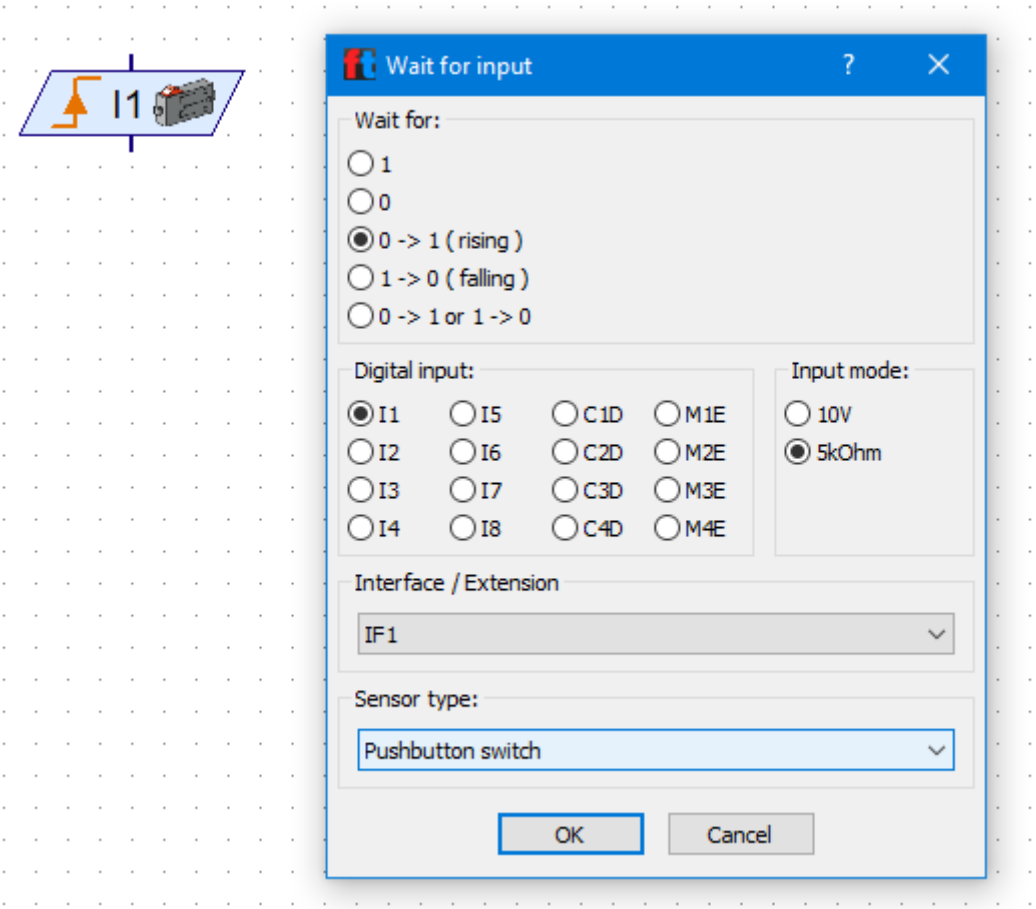
Element slouží k ovládání komponent připojených na výstup O1 – O8 a zem. V našich případech slouží pro ovládání světelných komponent (LED, žárovka) a solenoidový ventil. Podle zvolené komponenty se překreslí obrázek v elementu. Pro světelnou komponentu lze nastavit intenzitu světla a zapnutí/vypnutí. Pro ventil je možné nastavit zapnutí/vypnutí.



Obrázek 136: Lamp output

4.1.1.9 Wait for input

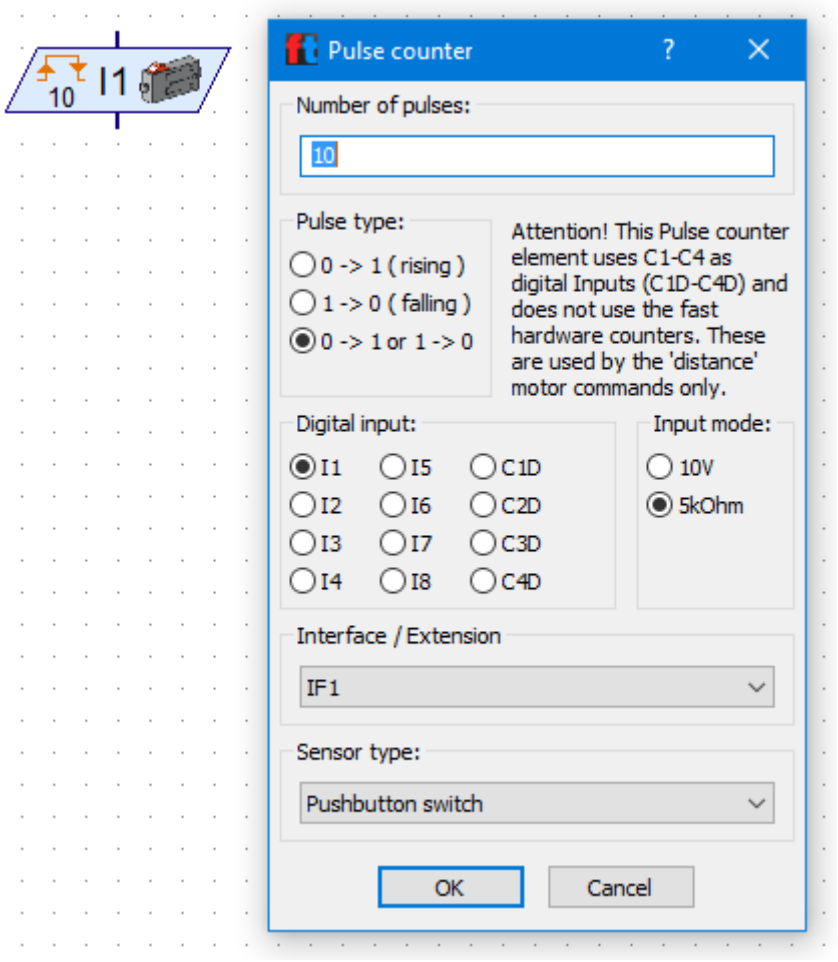
Element slouží k čekání na požadovaný vstup. Např. na stisknutí spínače apod. Nastavuje se, na jakou událost se musí počkat, na jakém vstupu, tedy I1 – I8, C1 – C4 a M1 – M4 se má v rámci procesu/programu vyčkat. Volbou „Sensor type“ říkáme, jaký typ je na daný vstup přivedený. Lze ovládat komponenty spínač a fototranzistor. Ostatní možnosti jsou s naší aktuální stavebnicí nevyužitelné. Element lze nahradit vhodně zacykleným elementem Digital branch.



Obrázek 137: Wait for input

4.1.1.10 Pulse counter

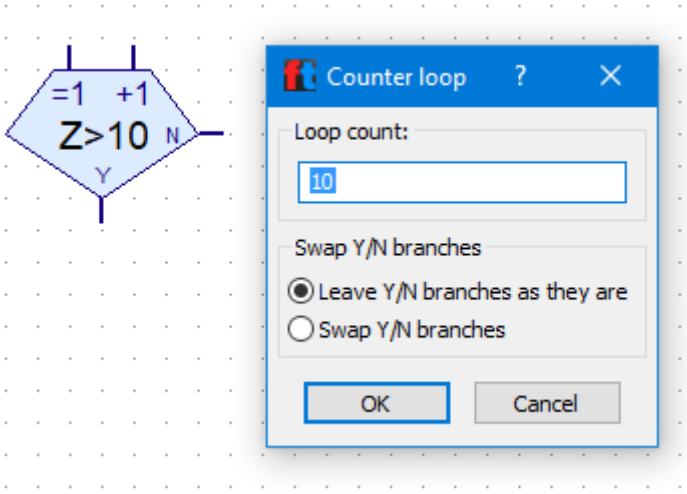
Element slouží k vyčkání na zadaný počet pulzů a až poté se vykonávají další části procesu/programu. Zadává se počet pulzů, jejich typ (změna na 1, změna na 0, jakákoliv změna), na jaký vstup je přivedený odpovídající element (I1 – I8 a C1 – C4) a typ používaného senzoru, tedy spínač a fototranzistor. Ostatní možnosti jsou s naší aktuální stavebnicí nevyužitelné.



Obrázek 138: Pulse counter

4.1.1.11 Counter loop

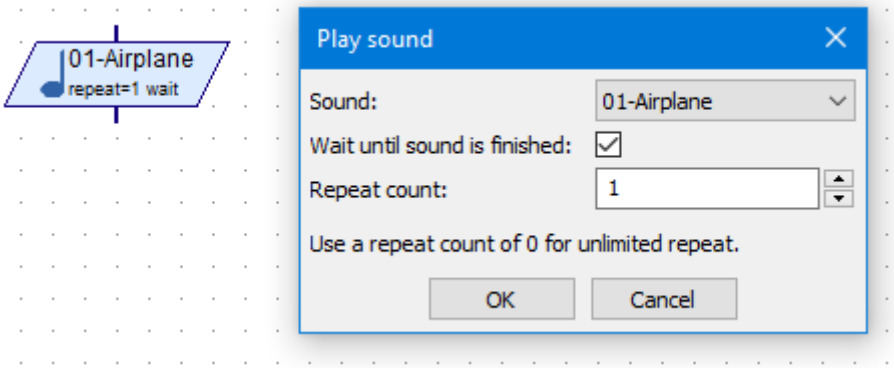
Element slouží k omezení počtu cyklů hodnotou. Nastavuje se pouze počet požadovaných cyklů a je možnost prohodit výstupy při splnění/nesplnění požadovaného počtu cyklů.



Obrázek 139: Counter loop

4.1.1.12 Sound

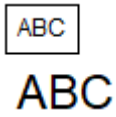
Element slouží k přehrání zvuku. Lze si vybrat ze seznamu zvuků, nastavit, jestli proces/program dál pokračuje hned po spuštění přehrávání zvuku, nebo se čeká až se přehrávání ukončí a počet opakování zvuku.



Obrázek 140: Sound

4.1.1.13 Vložení textu

Element umožňuje do programu vkládat textové popisky, poznámky, komentáře. Text může být víceřádkový (odřádkovává se standardně klávesou Enter). Vytvořený text nelze editovat. V případě potřeby úpravy textu se musí smazat a vytvořit nový.



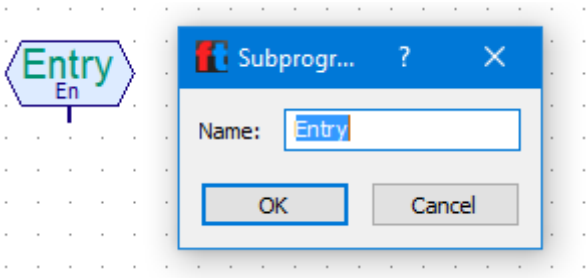
Obrázek 141: Vložení textu, možnost 2 velikostí

4.1.2 Subprogram I/O

Jedná se o nabídku elementů pro tvorbu podprogramů.

4.1.2.1 Subprogram Entry

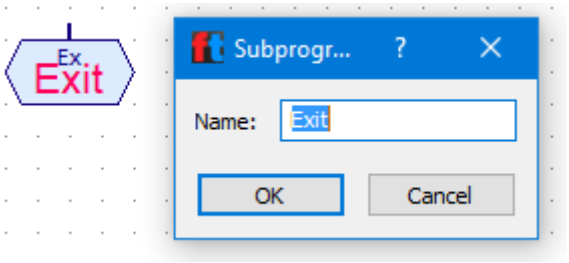
Elementem Subprogram Entry začíná podprogram. Lze mu nastavit pouze jméno.



Obrázek 142: Entry

4.1.2.2 Subprogram Exit

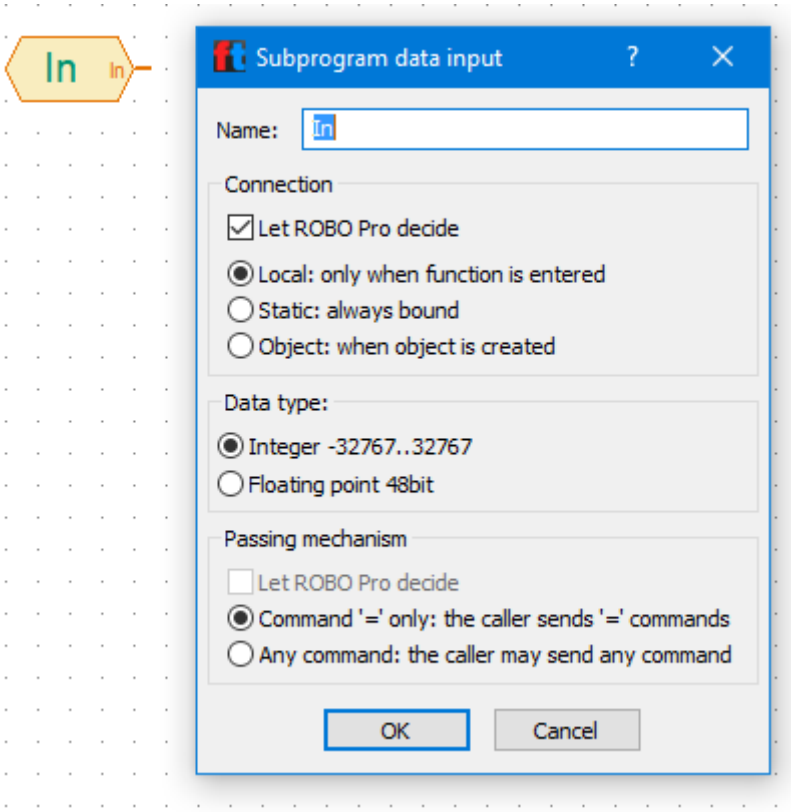
Elementem Subprogram Exit končí podprogram. Lze mu nastavit pouze jméno.



Obrázek 143: Exit

4.1.2.3 Subprogram command input

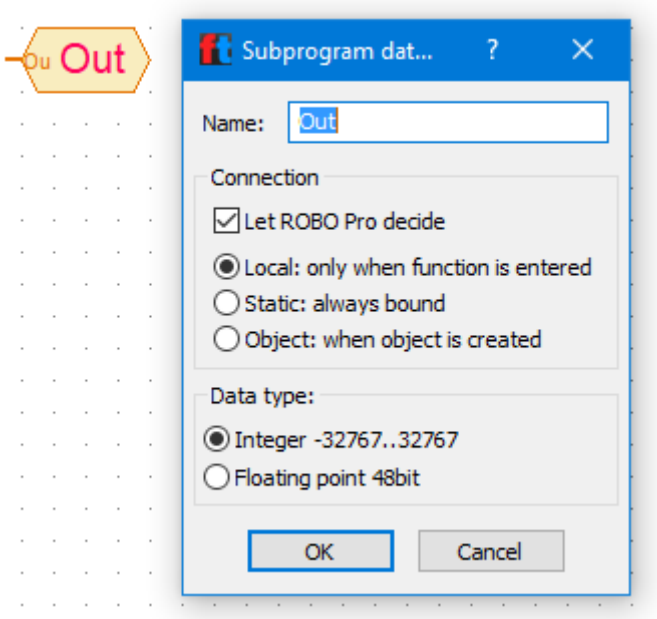
Element In slouží ke vstupu do podprogramu. Využívá se v případě, že se podprogramu předávají nějaké atributy, nebo proměnné. Dále lze nastavit, jestli pracuje pouze s hodnotami typu integer (celá čísla), nebo floating point (desetinná místa). Volbou „Passing mechanism“ se říká, zda vstup akceptuje pouze příkaz „=“, nebo jakýkoliv.



Obrázek 144: In

4.1.2.4 Subprogram command output

Element Out slouží k výstupu z podprogramu. Využívá se v případě, že se z podprogramu předávají nějaké atributy, nebo proměnné zpět programu. Dále lze nastavit, jestli pracuje pouze s hodnotami typu integer (celá čísla), nebo floating point (desetinná místa).



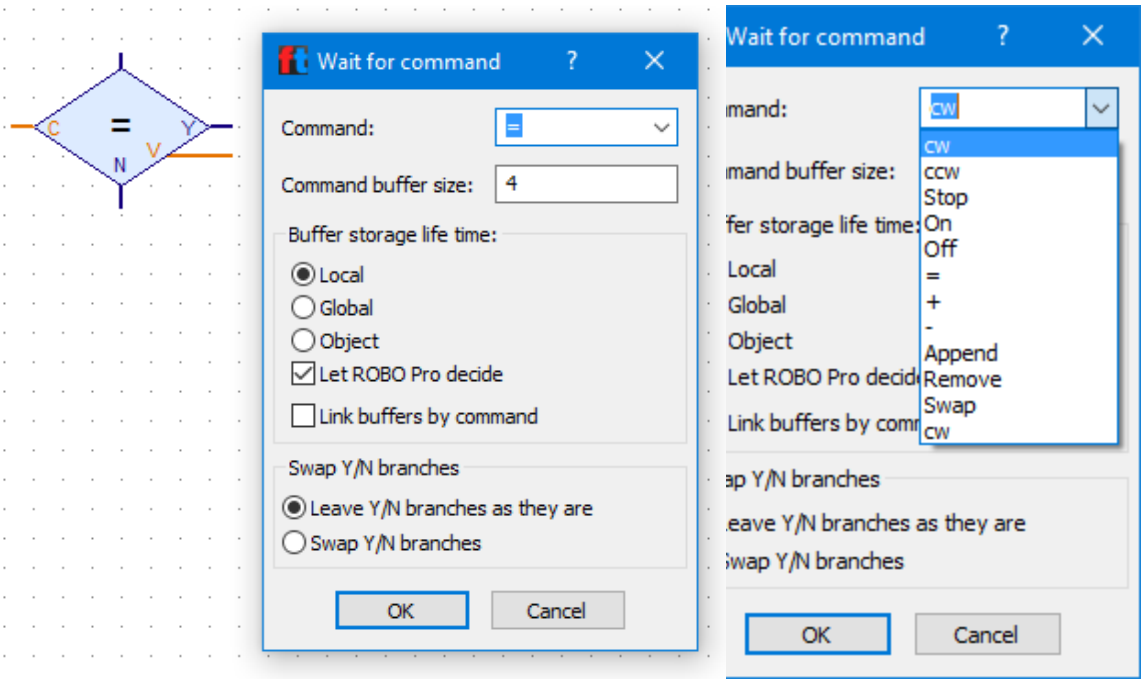
Obrázek 145: Out

4.1.3 Send, receive

Jedná se o nabídku elementů sloužících k přijetí a zpracování příkazů, nebo jejich modifikaci.

4.1.3.1 Wait for command

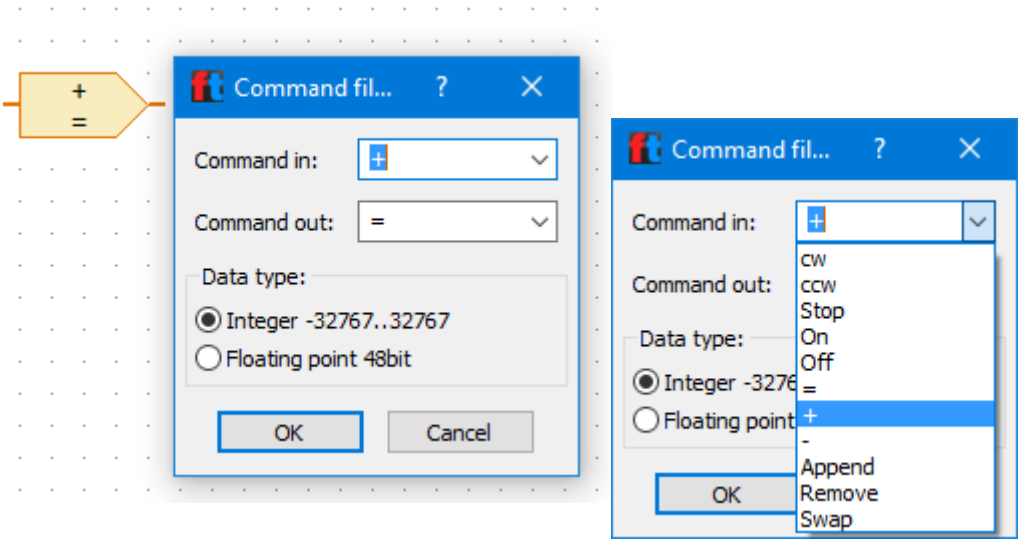
Element slouží k pozdržení vykonávání procesu/programu, do přijetí příkazu, jeho zpracování a poslání příkazu dále (C – vstup příkazu, V – výstup příkazu). Elementu lze nastavit příkaz (Command), kde je na výběr z několika typů příkazů, jako otáčky cw, otáčky ccw, Stop, On, Off, … viz obr. Dále pak Command Buffer size, tedy kolik příkazů může max. zpracovat (jejich počet, ne typ). Lze nastavit Buffer storage life time, zda je lokální, nebo globální a lze prohodit výstupy Y/N.



Obrázek 146: Wait for command

4.1.3.2 *Command filter*

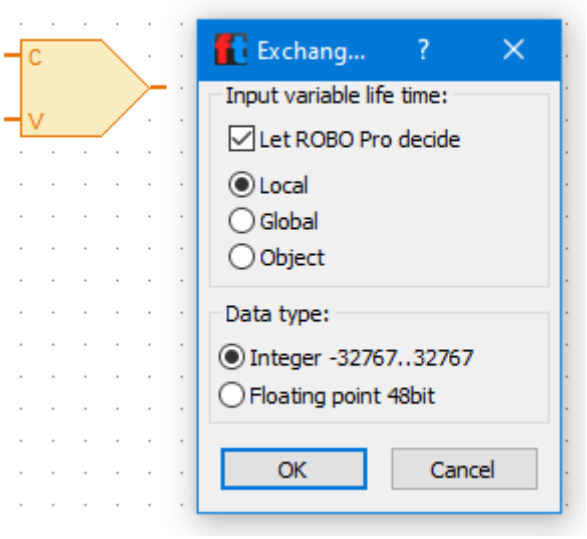
Element slouží ke změně příchozího příkazu na jiný. Zadává se vstupující příkaz (Command in) a vystupující příkaz (Command out). Navíc je možné nastavit, zda typ dat bude Integer, nebo Floating point.



Obrázek 147: Command filter

4.1.3.3 *Exchange message value*

Element je obdobou předchozího elementu „Command filter“. Mění vstupující příkaz (vstup C) na požadovaný příkaz (vstup V) a tento posílá dále na výstup. Elementu lze nastavit, zda vstupující proměnná je lokální, nebo globální (Life time) a jestli se jedná o datový typ integer, nebo floating point (Data type).



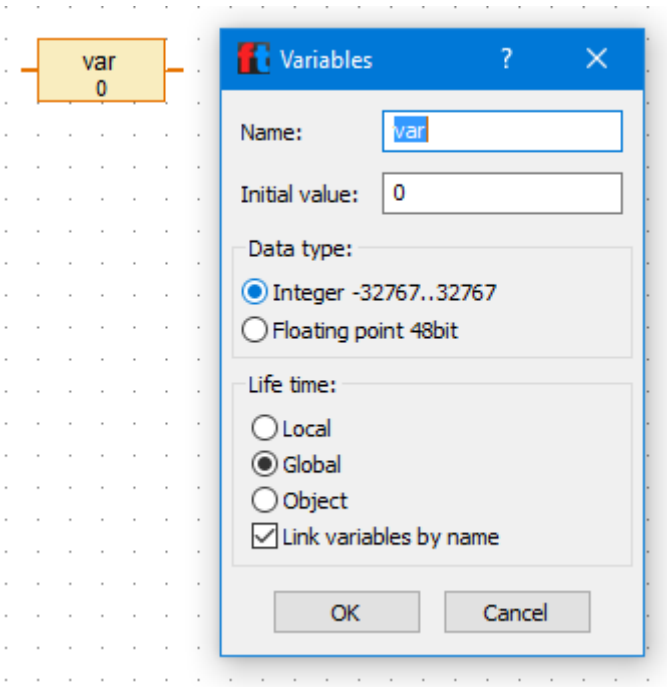
Obrázek 148: Exchange message value

4.1.4 Variables, timers,...

Jedná se o nabídku elementů pro definování proměnných, konstant a listů.

4.1.4.1 Variables

Element slouží k vytvoření proměnné. Lze nastavit název proměnné (Name), počáteční hodnotu (Initial value), Data type (Integer, nebo floating point) platnost proměnné (hlavně tedy zda je proměnná lokální nebo globální). Připojením konstanty zleva (nemusí být jen konstanta, ale může se jednat o část programu starající se o výpočet hodnoty a pod) přiřazujeme proměnné její obsah.



Obrázek 149: Variables

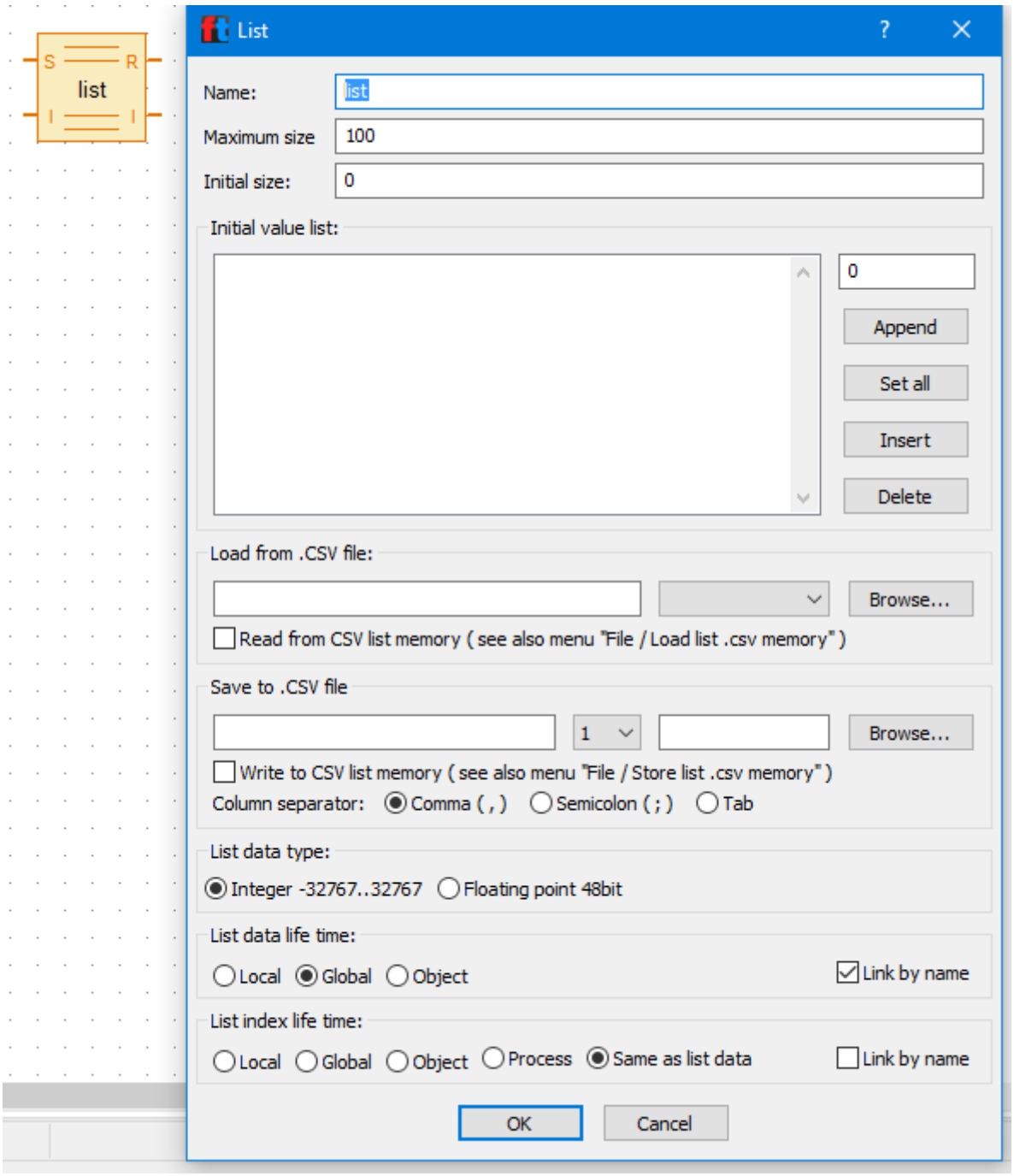
4.1.4.2 List

Element List slouží ke čtení dat ze souboru, nebo zapisování dat do souboru. Do listu lze přidávat hodnoty (příkaz Append), odebírat hodnoty (příkaz Delete), nebo měnit jim pořadí (příkaz Exchange). Můžeme jej parametrizovat:

- Name – název listu. Pozor, není myšlen list v programu Excel (který umí s .CSV soubory pracovat, ve skutečnosti se jedná o seznam)
- Maximum Size – lze nastavit maximální počet zadaných hodnot v listu. Tato velikost se nemění příkazy typu Append, Delete, Exchange.
- Initial size – hodnota, s kolika zadanými údaji je list inicializován. Většinou začínáme s prázdným a zapisujeme do něj.
- List of initial values – pokud nezačínáme s prázdným listem, můžeme zde nadefinovat inicializační hodnoty.
- Load from .CSV file – zde vybíráme .CSV soubor, který chceme načítat a pořadí sloupce, se kterým chceme pracovat.
- Save to .CSV file - nastavujeme .CSV soubor, do kterého se hodnoty budou zapisovat, do jakého sloupce (jeho pořadí) a jakým způsobem budou data v .CSV souboru oddělena (budeme používat středník „;“).
- Data type – volíme, zda hodnoty budou typu Integer, nebo floating point.
- List data life time – definujeme, zda je element složen z globálních, nebo lokálních proměnných. Pro velké listy (s velikostí přes 100 hodnot) se doporučuje používat globální, protože jim je vyhrazeno více paměti.

Vstupy a výstupy elementu List

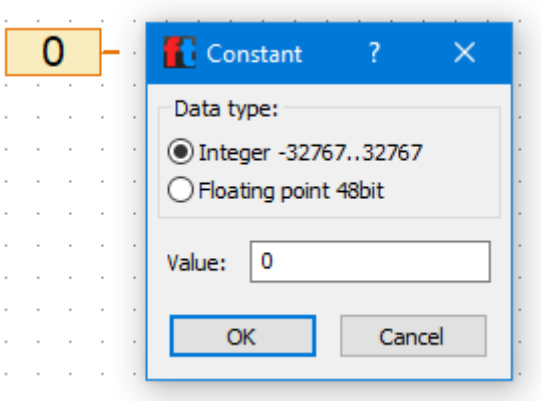
- Vstup S – hodnota k zapsání do listu
- Vstup I – možnost zadat index k hodnotě zapisované do listu přiřazením „=“ a požadované hodnoty indexu
- Výstup R – čtení hodnot z listu
- Výstup I – čtení počtu hodnot/sloupců, možnost např. zkontrolovat, jak se zapisují další hodnoty a mění se počet zapsaných hodnot listu.



Obrázek 150: List

4.1.4.3 Constant

Element slouží k nadefinování konstantní hodnoty. Element se používá k předání konstanty proměnné, nebo jiným elementům, které s ní mají pracovat. Nastavuje se Data type (hodnota typu Integer, nebo Floating point) a hodnota konstanty (Value).



Obrázek 151: Constant

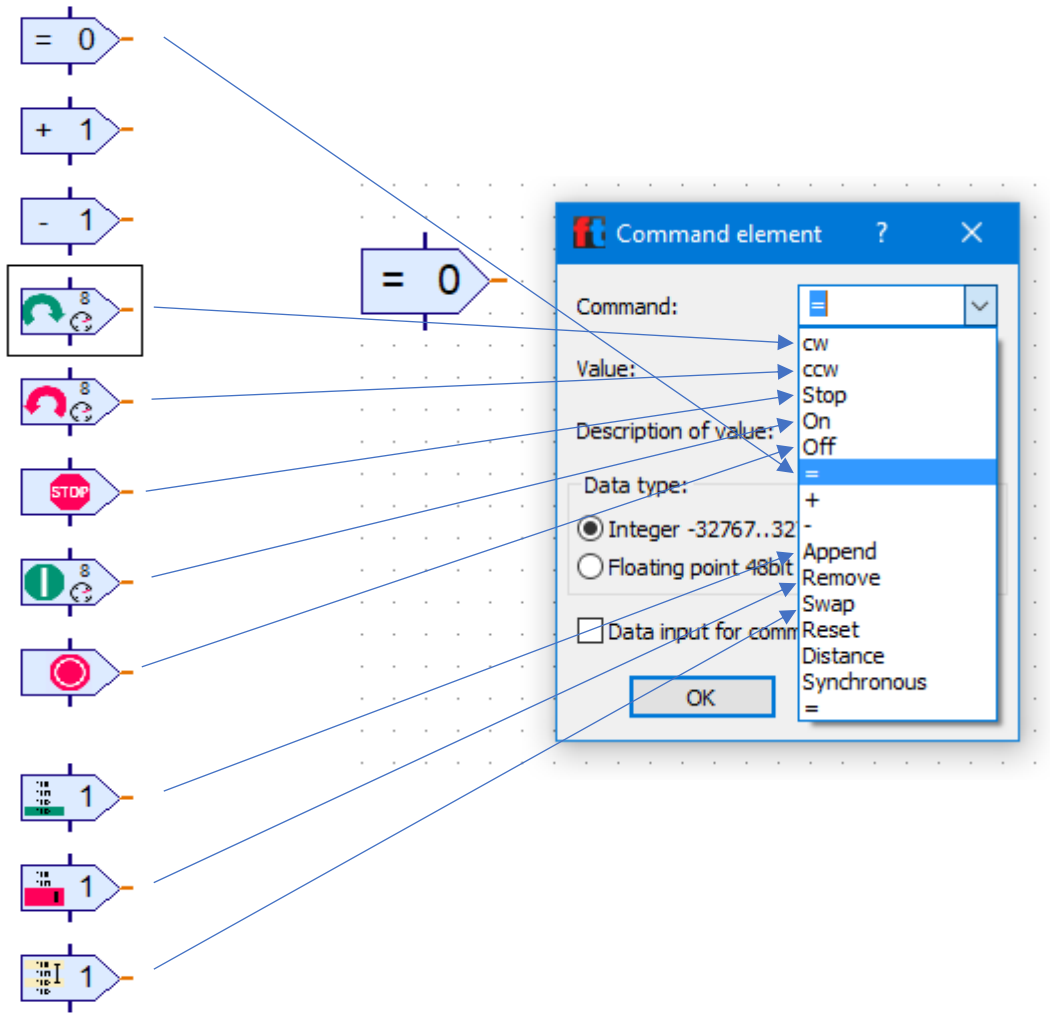
4.1.5 Commands

Jedná se o nabídku elementů příkazů.

4.1.5.1 Assign, command elements

Elementů Assign command element je podle ikon v nástroji ROBO Pro více, ale jedná se o jeden element, který mění ikonu dle nastavení přes pravé tlačítko myši. Mezi elementy „Command“ lze tedy přepínat pouze jejich nastavením. Princip elementu je takový, že se elementu nastaví, příkaz, který se má vykonat a na výstup vpravo se napojí část programu pro další práci s jeho výstupem, např. uložení do proměnné, nebo v případě že příkaz např. nastavuje otáčky CW, tak se na výstup napojí element typu Input pro motor. Tím se motoru předají parametry, podle kterých bude fungovat.

U elementu lze tedy zvolit jaký příkaz vykonává, parametr hodnota (Value), která je mu přiřazena (v případě motoru příkaz cw a hodnota např. 8 – maximální otáčky motoru). Příkazy Append, Delete a Exchange slouží pro práci s listy. U elementů je navíc možnost ve vlastnostech zapnout, jestli má vstup (Data input for command value). V takovém případě přibyde na levé straně elementu vstup a hodnota (Value) se nezadává ve vlastnostech elementu, ale přivádí se jako input zleva.

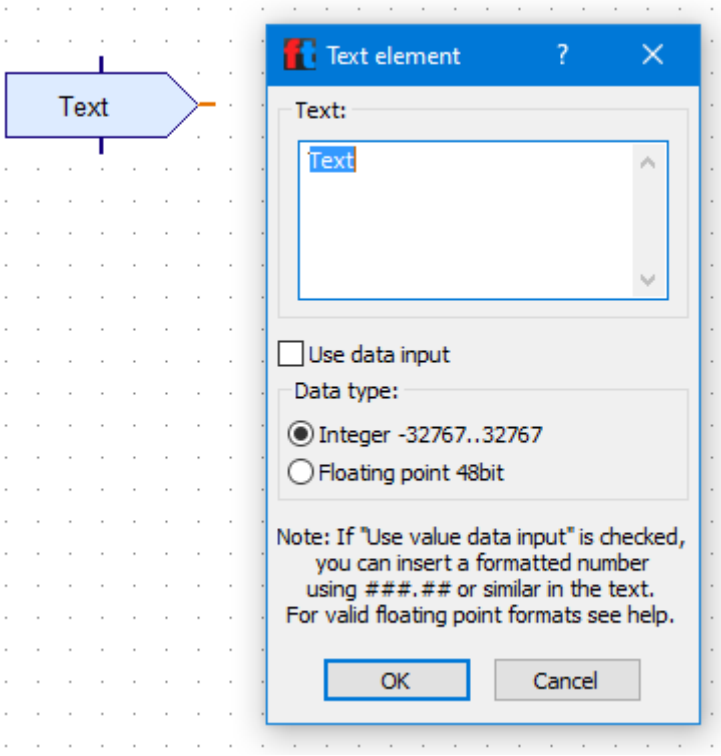


Obrázek 152: Command element - náznak vztahů mezi ikonami a příkazem

4.1.5.2 Text command

Element Text command se od předchozích lehce odlišuje. Tato odlišnost spočívá v tom, že předchozí „Assign command elements“ pracují s číselnými hodnotami, zatímco Text command pracuje s textem (stringem), který se předává dále přes výstup vpravo.

U elementu lze tedy zvolit Text, který se předává dále a v případě číselných hodnot je možné nastavit, zda se jedná o typ Integer, nebo Floatin point. U elementu je navíc možnost ve vlastnostech zapnout, jestli má vstup (Data input for command value). V takovém případě přibyde na levé straně elementu vstup a text se nezadává ve vlastnostech elementu, ale přivádí se jako input zleva.



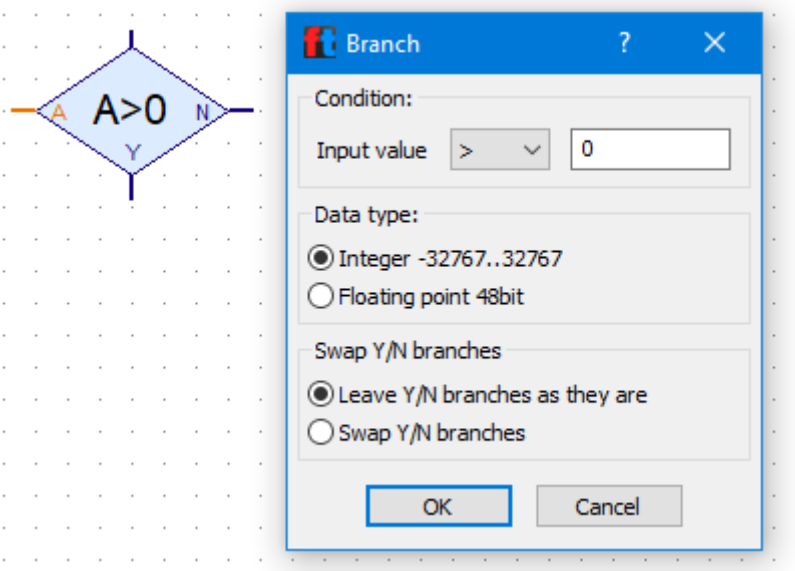
Obrázek 153: Text element

4.1.6 Branch, wait,…

Jedná se o nabídku elementů pro rozhodování, nebo porovnávání.

4.1.6.1 Branch with data input

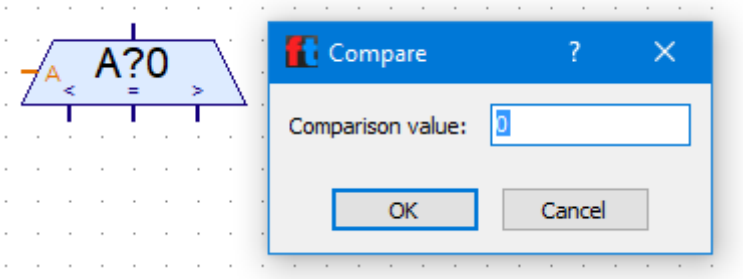
Element slouží k zadání podmínky, kdy se vyhodnocuje vstupující hodnota do elementu oproti komparační hodnotě. V nastavení můžeme říci, jaká je komparační hodnota a jaký se mezi hodnotami požaduje vztah (menší, větší, rovno,…). Podle vyhodnocení podmínky se pak dál vykonává část procesu/programu na výstupu Y (Yes, podmínka splněna), nebo N (No, podmínka nesplněna).



Obrázek 154: Branch with data input

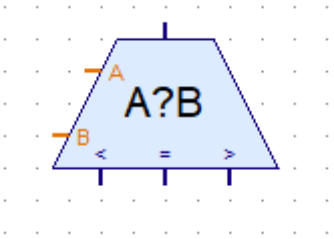
4.1.6.2 Compare

Element slouží porovnání vstupu oproti komparační hodnotě. V nastavení elementu můžeme říci, jaká je komparační hodnota. Následně porovnáním vstupní hodnoty s komparační hodnotou volí element vhodný výstup (vstup < komparační hodnotě, vstup = komparační hodnotě, vstup > komparační hodnota).



Obrázek 155: Compare

4.1.6.3 Compare – 2 inputs

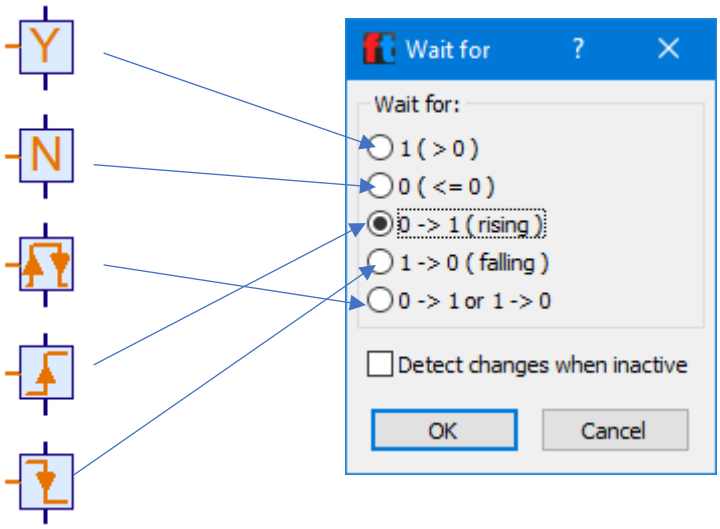


Obrázek 156: Compare - 2 inputs

Element je modifikací předchozího elementu „Compare“. Slouží k porovnání 2 vstupů oproti sobě. Porovnáním vstupních hodnot volí element vhodný výstup (vstup < komparační hodnotě, vstup = komparační hodnotě, vstup > komparační hodnota). Elementu se nic nenastavuje přes pravé tlačítko myši.

4.1.6.4 Wait for…

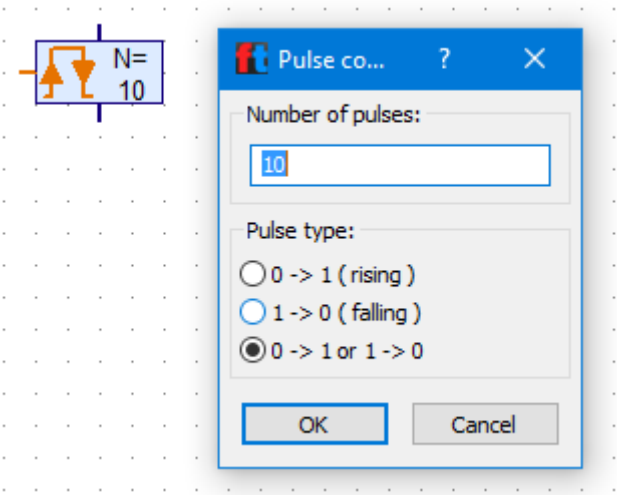
Elementů Wait for je podle ikon v nástroji ROBO Pro více, ale jedná se o jeden element, který mění ikonu dle nastavení přes pravé tlačítko myši. Mezi elementy „Wait for“ lze tedy přepínat pouze jejich nastavením. Princip elementu je takový, že vykonávání programu se na elementu pozastaví do té doby, dokud na bočním vstupu elementu nenastane požadovaná změna (podle typu elementu), viz obr.



Obrázek 157: Wait for

4.1.6.5 Pulse counter

Element Pulse counter je speciálním typem předchozího elementu Wait for. Princip elementu je takový, že vykonávání programu se na elementu pozastaví do té doby, dokud na bočním vstupu elementu neproběhne požadovaný počet pulsů. V nastavení elementu se nastavuje požadovaný počet pulsů (Number of pulses) a typ pulsů, které se mají započítávat (Pulse type), tedy zda se počítá pouze pulz (změna) 0 – 1, 1 – 0. nebo jakákoliv změna.



Obrázek 158: Pulse counter

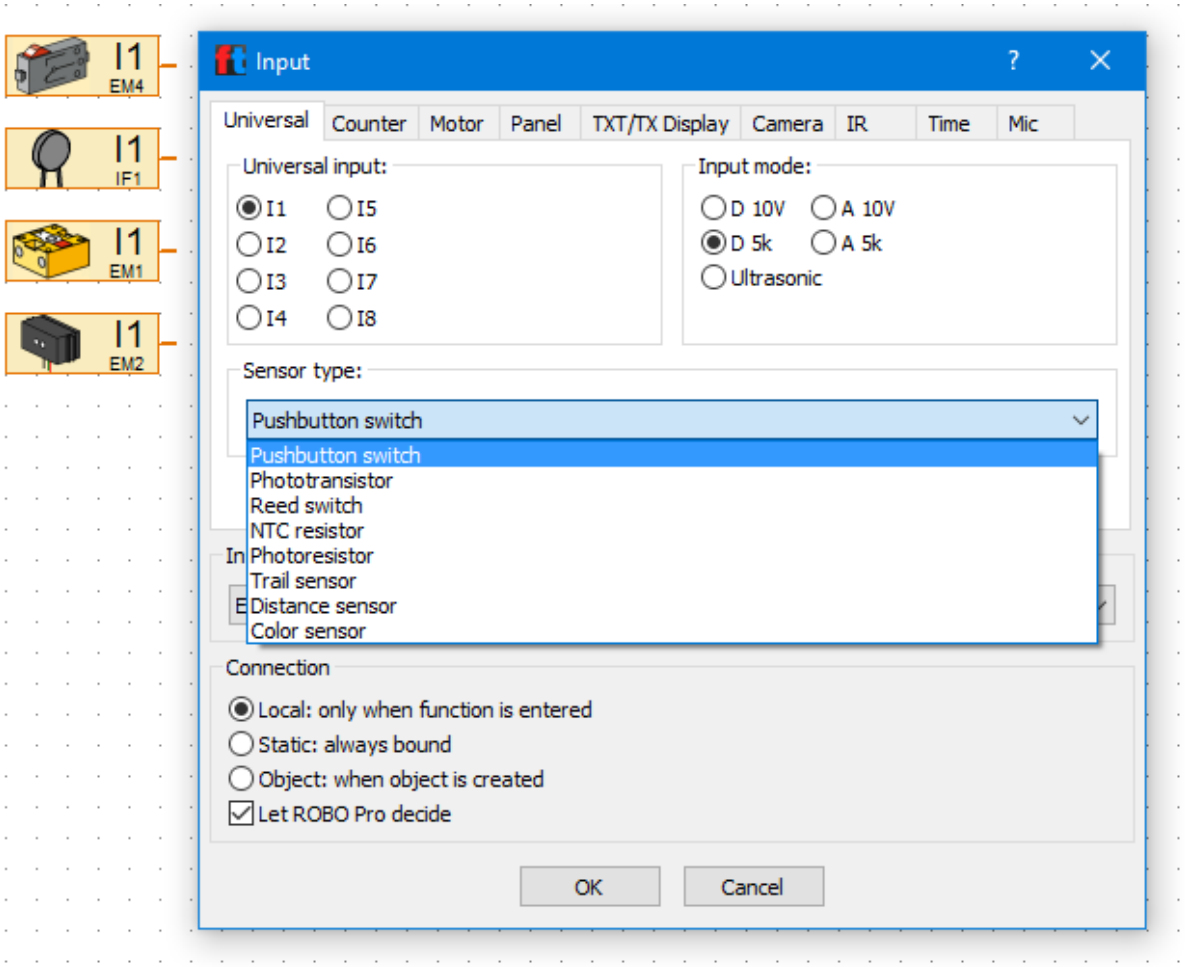
4.1.7 Inputs, outputs

Jedná se o nabídku elementů sloužících jako vstupy (Inputs), nebo výstupy (Outputs). Na oknech k nastavení vstupů/výstupů je vidět, že se vždy otevře stejné okno, pokaždé jen s jinou (elementu odpovídající) záložkou. Vložíme-li tedy nějaký element typu Input/output, jsme schopni jej přepnout na jiný. Tím se nám změní i zobrazení elementu.

4.1.7.1 Universal input

Element, který slouží jako univerzální vstup. Lze mu nastavit pomocí Sensor type, zda se jedná o spínač, NTC rezistor, fototranzistor, nebo optický barevný senzor. Ostatní možnosti nejsou u naší stavebnice dostupné (hardwarově). U univerzálních vstupů lze nastavit na jakém vstupu řídící jednotky je

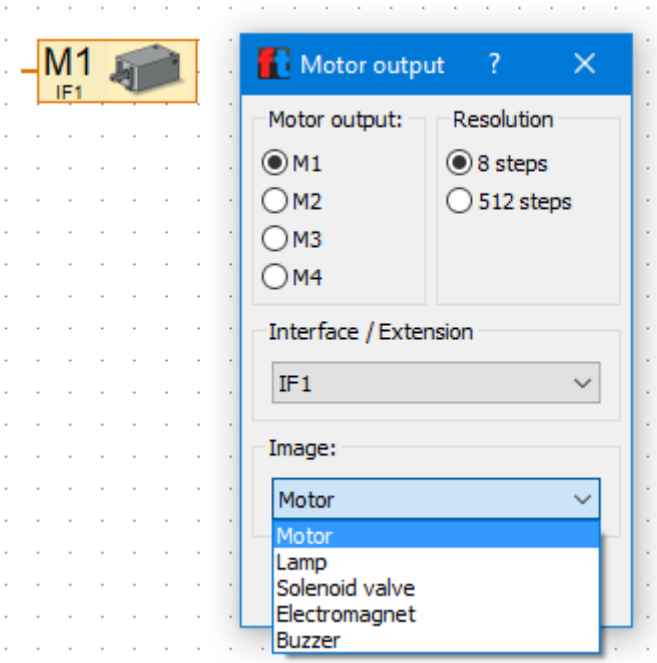
odpovídající komponenta (I1 – I8). Hodnoty odečtené z dané komponenty jsou pak odesílány na výstup elementu.



Obrázek 159: Universal input

4.1.7.2 Motor output

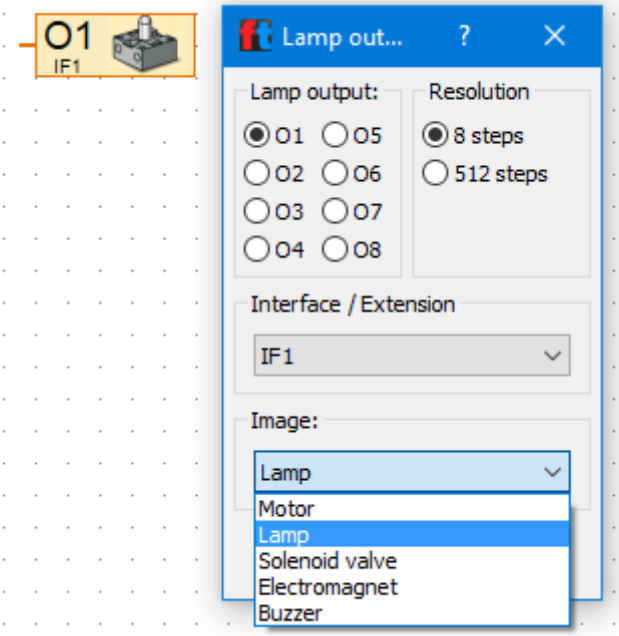
Element, který slouží jako výstup pro XS motory a mini motory, popř. lze změnit na výstup pro světelnou komponentu (LED, žárovka), nebo selenoidový ventil. Podle zvolené možnosti se změní i obrázek elementu v ROBO Pro. Ostatní možnosti nejsou s naší stavebnicí dostupné (hardwarově). Elementu se pak ještě nastavuje, na který výstup typu M je komponenta přivedena, tedy M1 – M4. Výstupu se z leva předávají parametry, tedy např. v případě motoru směr otáček (cw, cww) a rychlost otáček (1 – 8).



Obrázek 160: Motor output

4.1.7.3 Lamp output

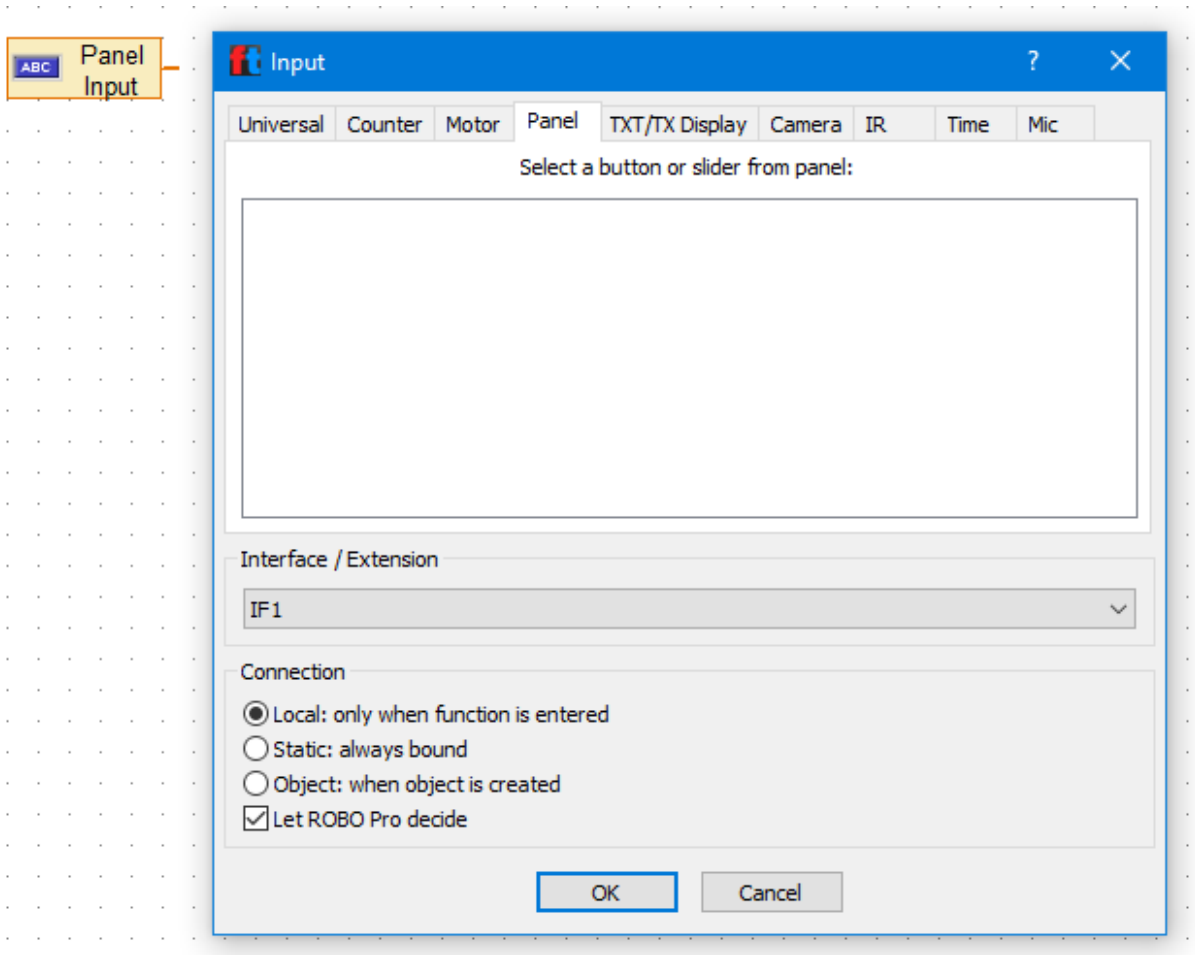
Element, který slouží jako výstup pro XS motory a mini motory, popř. lze změnit na výstup pro světelnou komponentu (LED, žárovka), nebo selenoidový ventil. Podle zvolené možnosti se změní i obrázek elementu v ROBO Pro. Ostatní možnosti nejsou s naší stavebnicí dostupné (hardwarově). Elementu se pak ještě nastavuje, na který výstup typu O je komponenta přivedena, tedy O1 – O8 + zem. Výstupu se z leva předávají parametry, tedy např. v případě světelného elementu zapnutí/vypnutí (on/off) a intenzita svícení (1- 8).



Obrázek 161: Lamp output

4.1.7.4 Panel Input

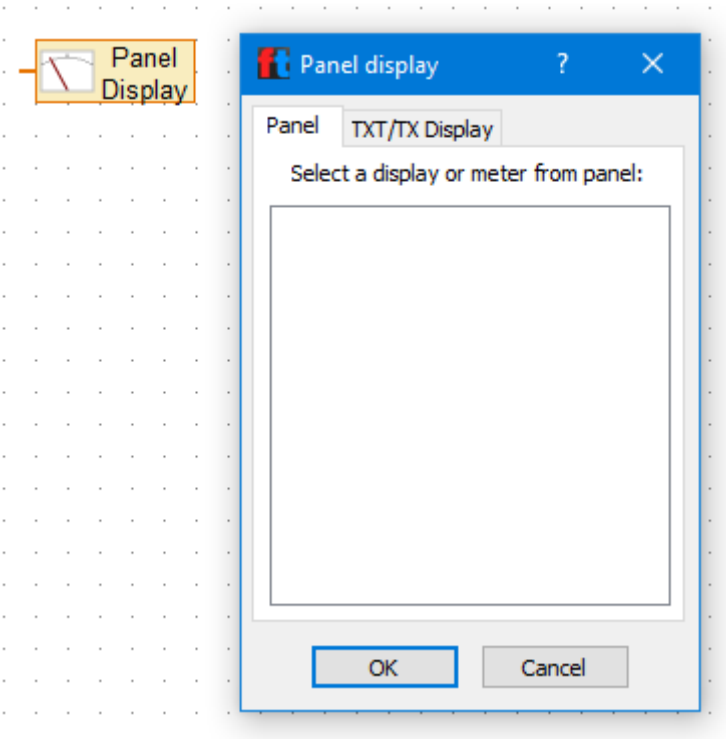
Element slouží jako vstup. Pokud je někde v programu použité např. tlačítko v panelu (viz příkaz na pojezd mezi dvěma pevnými body), tak v elementu Panel Input na záložce pane se volí, který element na něj je navázaný a jeho hodnotu/hodnoty posílá dále na svůj výstup.



Obrázek 162. Panel input

4.1.7.5 Panel display

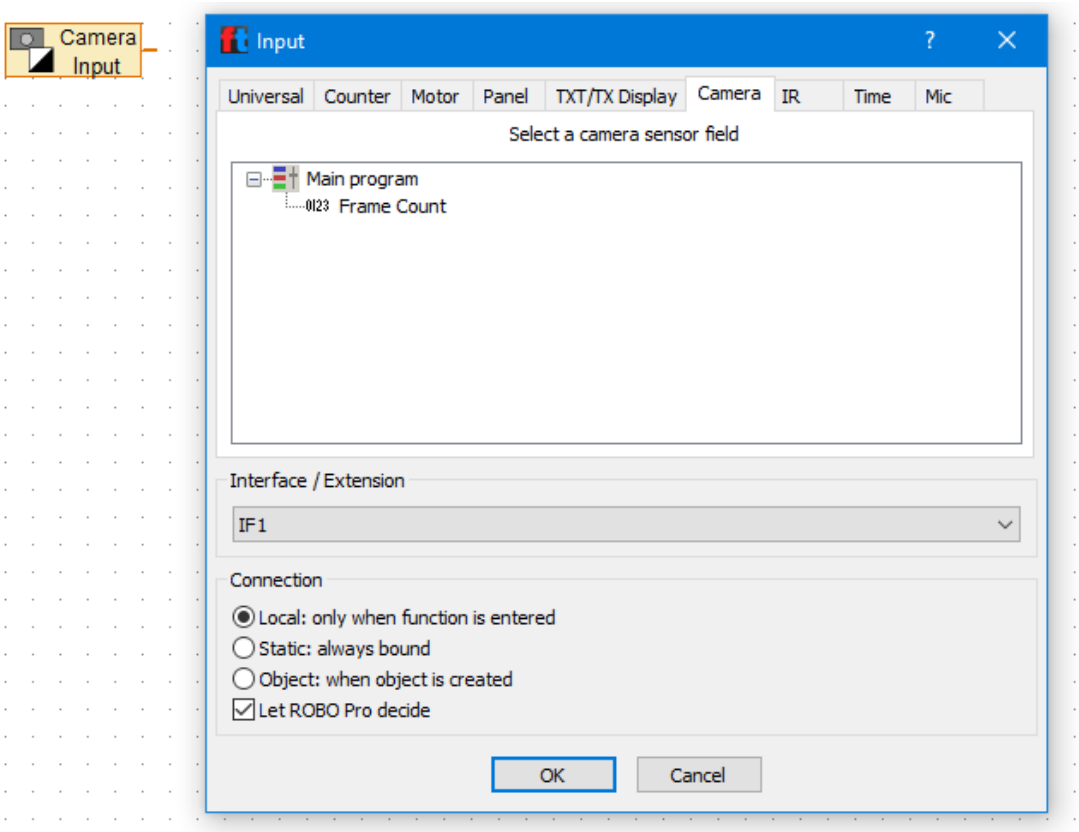
Elementu se předává hodnota/parametr a dále je jím předávána tato hodnota elementu použitého např. na Text displeji. Pokud tedy je někde v programu použitý např. element Text display, který má zobrazovat nap. nějakou hodnotu, tak zde se volí, na jaký Text display budou data předávána.



Obrázek 163: Panel display

4.1.7.6 Camera input

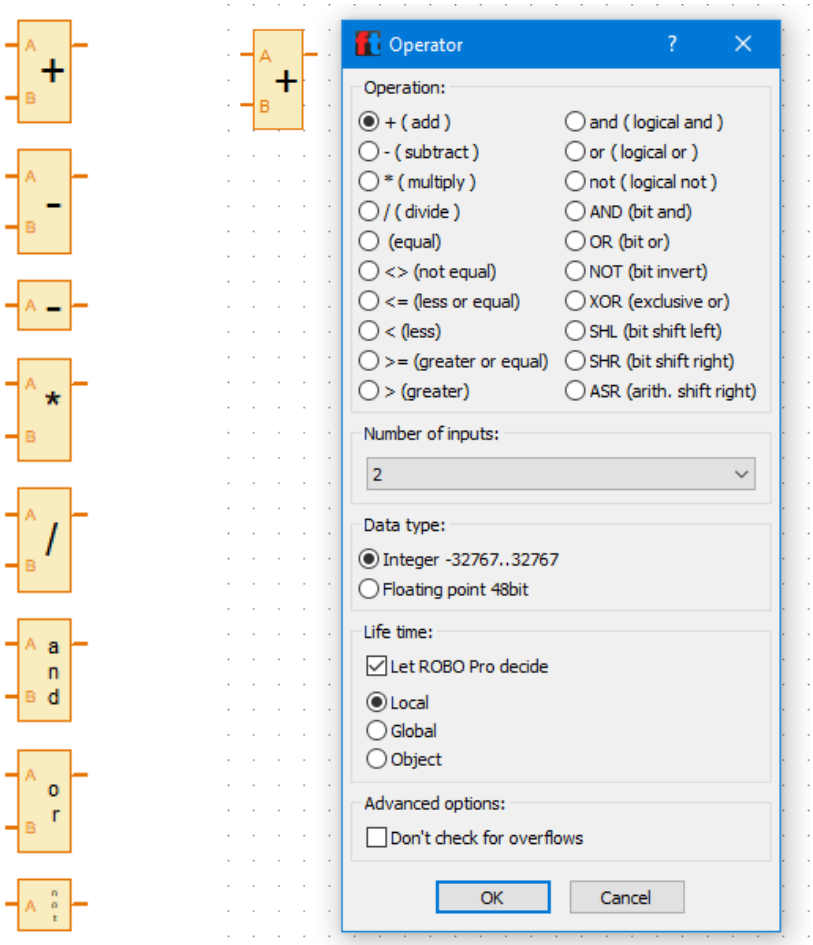
Element čte data z kamery a předává je dále programu pro další zpracování.



Obrázek 164: Camera input

4.1.8 Operators

Elementů Operators je podle ikon v nástroji ROBO Pro více, ale jedná se o jeden element, který mění ikonu dle nastavení přes pravé tlačítko myši. Mezi elementy „Operators“ lze tedy přepínat pouze jejich nastavením. Princip elementu je takový, že se vstupy provede takovou operaci, jaká je v nastavení elementu zvolena, viz obr. V nastavení lze tedy vyprat typ operace, kterou má element se vstupy provést (sečíst, odečíst, porovnat, logické and, logické or, …), lze nastavit počet vstupů operátoru (Number of inputs, standardně je nastaveno na 2 vstupy), Data type (zda je hodnota typu integer, nebo floating point) a life time, tedy zda se jedná o lokální, nebo globální element.



Obrázek 165: Operators

4.2 Operating elements

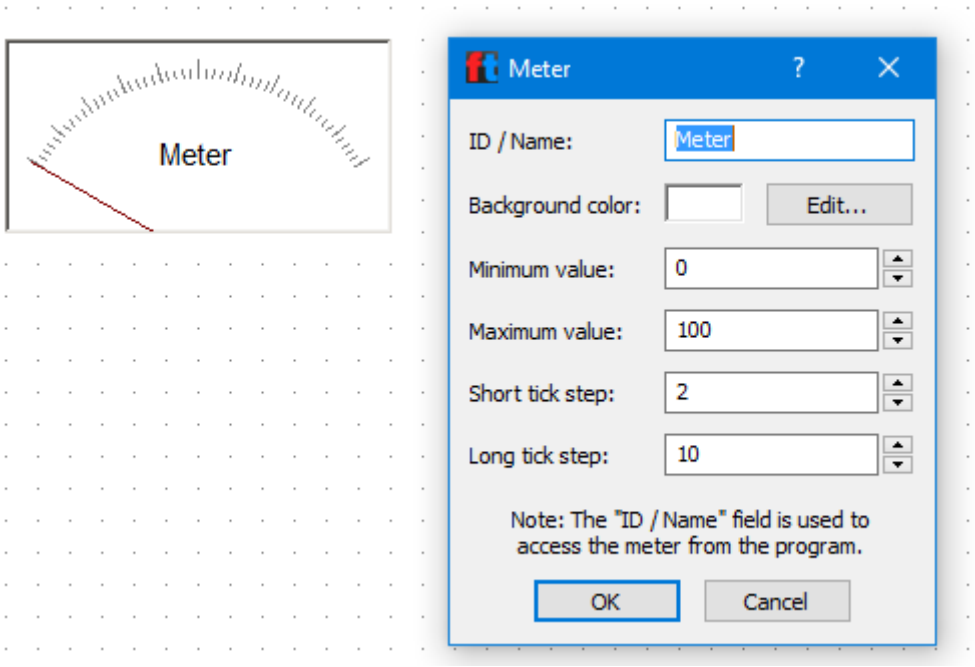
Sekce obsahující elementy pro tvorbu softwarových ovládacích prvků a zobrazovacích prvků. Lze je používat jak přímo v procesu/programu na záložce „Function“, tak na záložce „Panel“ v prostředí ROBO Pro.

4.2.1 Displays

Jedná se o nabídku softwarových zobrazovacích elementů.

4.2.1.1 Meter

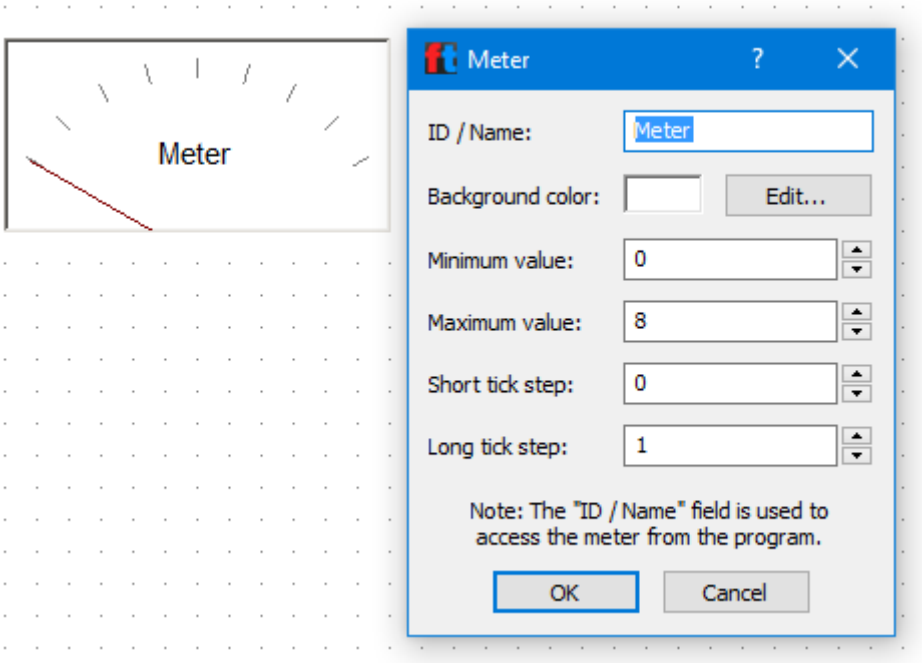
Element slouží k zobrazení hodnoty formou ručičkového ukazatele. Lze mu nastavit Název, barvu pozadí, minimální hodnotu (na jaké hodnotě ručička začíná), maximální hodnotu (na jaké maximální hodnotě ručička může končit), malý krok a velký krok, viz následující obrázek. Element se opět musí propojit s elementem Panel display, ze kterého čerpá údaje k zobrazení.



Obrázek 166: Meter

Např. pro motor, který má maximální otáčky rychlostí 8 a nastavuje se po 1, by to mohlo být:

- Minimum value = 0
- Maximum value = 8
- Short tick step = 0
- Long tick step =1



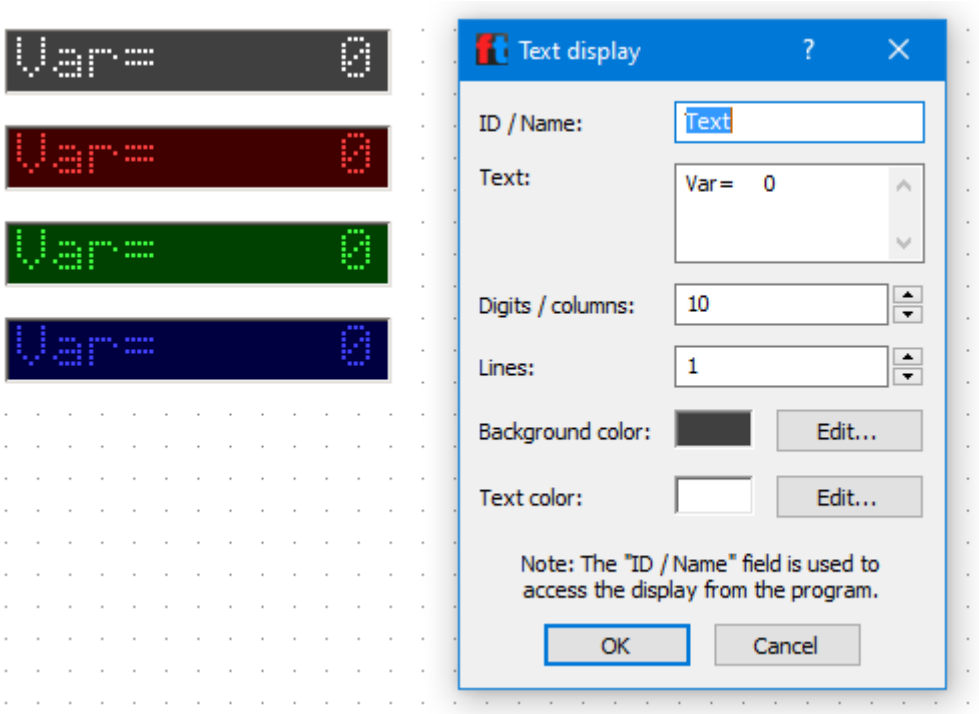
Obrázek 167: Meter s nastavením pro XS motor

4.2.1.2 Text display

Element slouží k zobrazení hodnoty formou textového informačního pole. Lze mu nastavit:

- Název – název elementu
- Text - text který se zobrazuje v Text display)
- Digits/columns – počet pozic grafického elementu. Potřeba vždy vyzkoušet, aby se hodnoty zobrazovaly celé.
- Lines – počet řádků pole.
- Background color – barva pozadí pole
- Text color – barva textu v poli.

Barvu pozadí i textu lze nastavit libovolnou, v nabídce jsou 4 předdefinované možnosti (šedé, červené, zelené a modré grafické pole), viz obr. Element se opět musí propojit s elementem Panel display, ze kterého čerpá údaje k zobrazení.



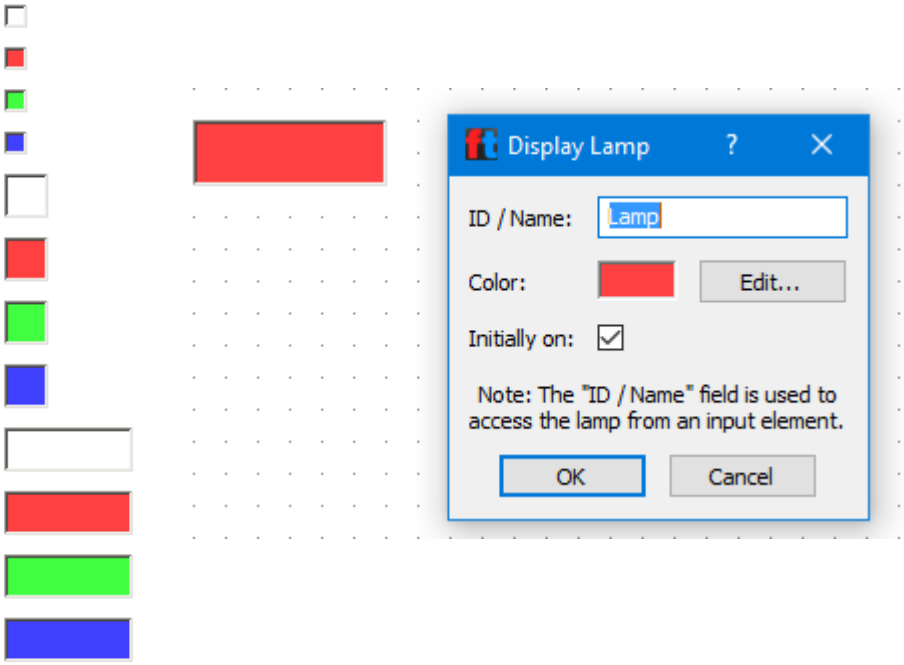
Obrázek 168: Text display

4.2.1.3 Display lamp

Element funguje podobně jako např. kontrolní žárovka/LED. Při předání informace elementu Display lamp se změní jeho barva, nebo blikne požadovanou barvu. Lze vybírat ze 3 velikostí a č předdefinovaných barev. Velikost je pevně daná nabídkou, barvu lze ale následně ve vlastnostech elementu změnit na jakoukoliv. Lze mu nastavit:

- Název – název elementu
- Color – barvu, na kterou se má přepínat
- Initialy on – nastavení, zda má být barva inicializována při startu programu, nebo až při předání informace za běhu programu.

Element se opět musí propojit s elementem Panel display, ze kterého čerpá údaje k zobrazení.



Obrázek 169: Display lamp

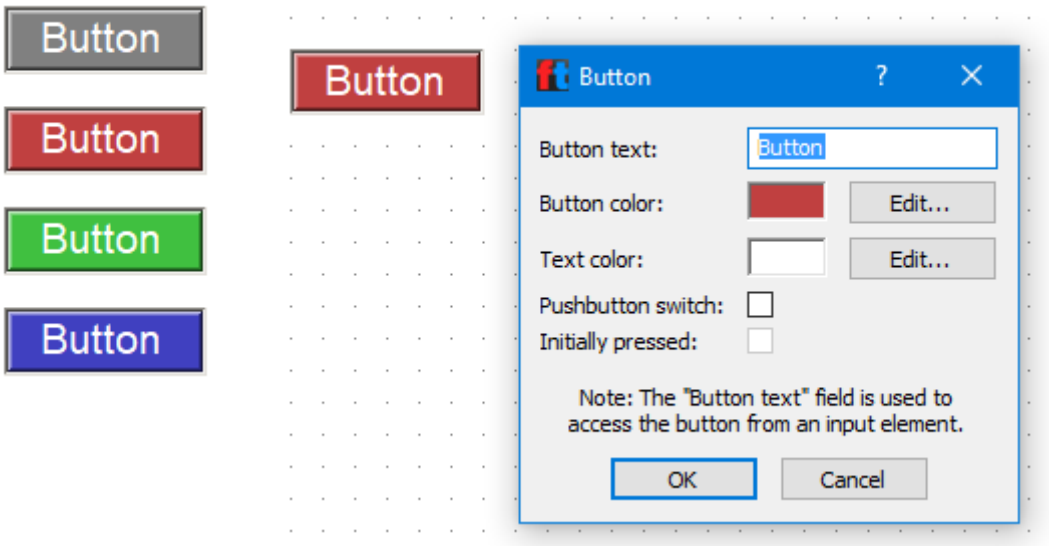
4.2.2 Control elements

Jedná se o nabídku softwarových ovládacích elementů.

4.2.2.1 Button

Element slouží k ovládání programu formou tlačítka. Velikost ovládacího tlačítka je pevně dána, z barev je na výběr (šedá, červená, zelená, modrá), ale v nastavení prvku lze zvolit libovolnou barvu.

Elementu lze nastavit Název, barvu tlačítka, barvu textu a zvolit, zda má fungovat jako přepínač (standardně se chová jako spínač), a v případě přepínače, zda má mít výchozí stav „stisknuto“. Element se pak propojí buď s požadovaným outputem, nebo proměnnou, která se používá v procesu/programu.

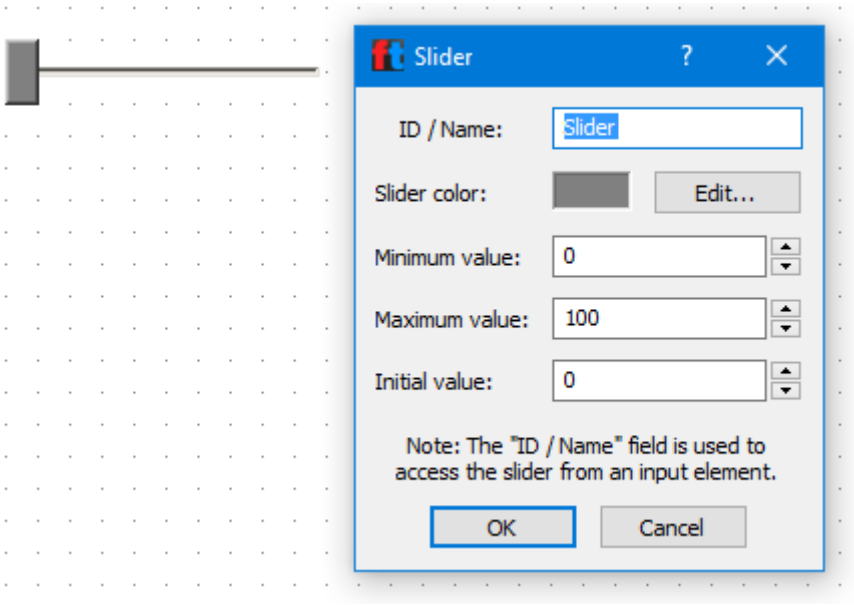


Obrázek 170: Button

4.2.2.2 Slider

Element slouží k ovládání komponent formou posuvníku a je možné volit mezi vertikálním a horizontálním provedením.

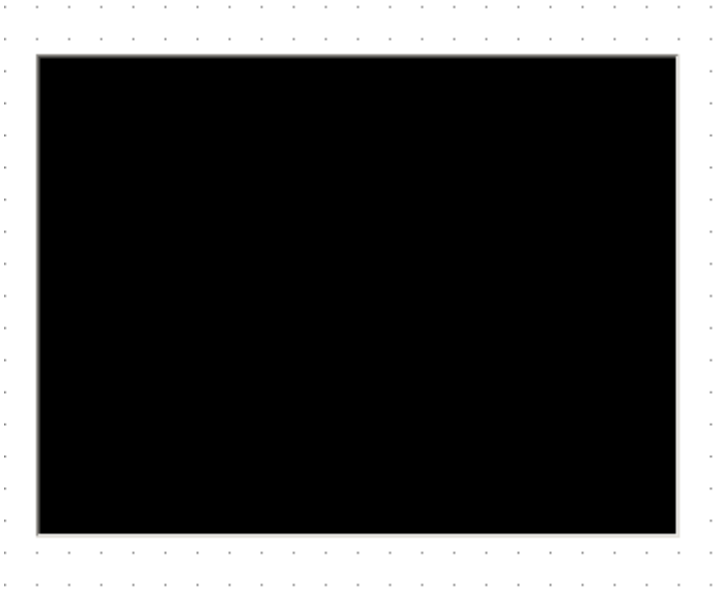
Lze mu nastavit Název, barvu, minimální hodnotu, maximální hodnotu a počáteční hodnotu. Např. pro XS motor by mohla být minimální hodnota 0, maximální 8 a počáteční 0. Slider by se tak používal pro ovládání otáček motoru s možností úplného vypnutí motoru. Element se pak propojí buď s požadovaným outputem, nebo proměnnou, která se používá v procesu/programu.



Obrázek 171: Slider

4.2.3 Camera viewer

Element slouží k zobrazení dat z USB kamery. Při používání kamery tak zobrazuje obraz, který kamera snímá. Po aktivaci kamery zobrazuje obraz, který kamera snímá. V nastavení Camera lze nastavit i používání jiné kamery, např. webkamery v notebooku.



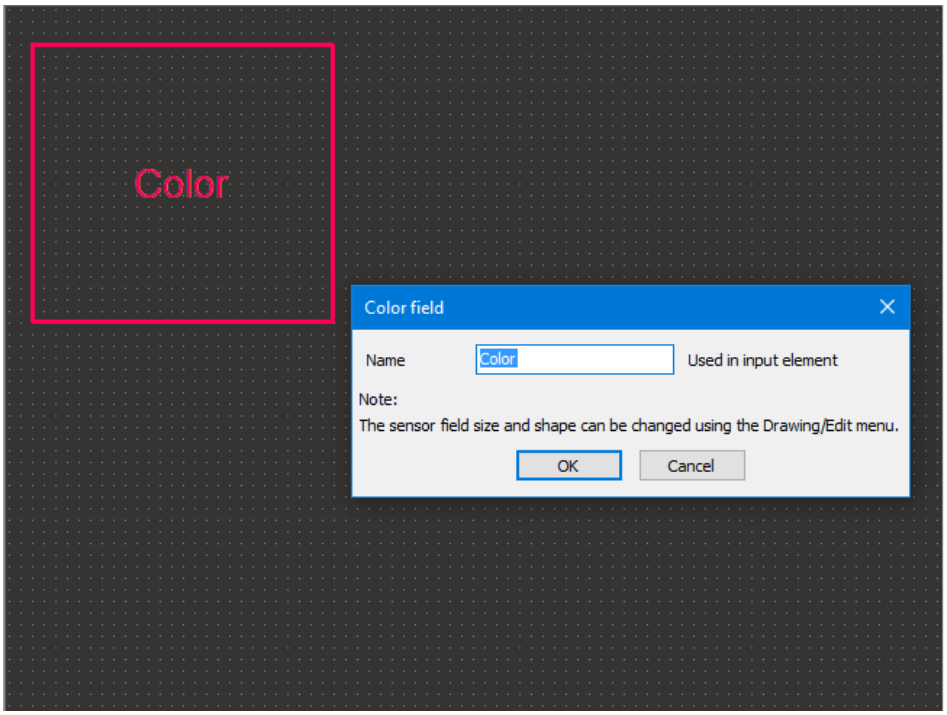
Obrázek 172: Camera viewer

4.3 Camera panel – Sensor fields

Sekce obsahující elementy pro práci s obrazem snímaným kamerou. Není potřeba využívat pouze kameru ze stavebnice připojenou do řídící jednotky, je možné použít jakoukoliv kameru – např. kameru notebooku. S elementy se pracuje na záložce Camera, kde se elementy nabízí v nabídce. Elementy lze v jednom programu libovolně kombinovat a mohou se překrývat.

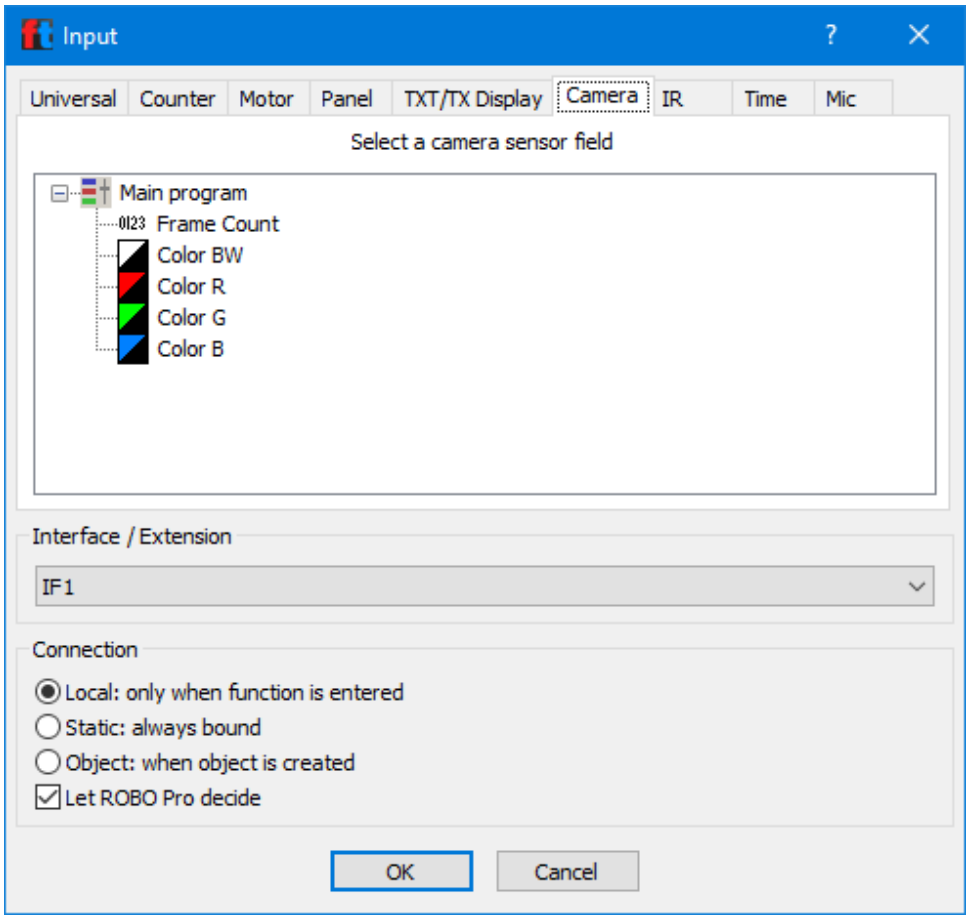
4.3.1.1 Color detector

Element Color detector slouží k vyhodnocování barev v označené části kamerou snímaného prostředí. S nasnímanými hodnotami se pak pracuje v programu, kde je lze zobrazit, uložit do proměnné apod., viz předchozí elementy jako outputy, inputy, … Velikost a místo vyhodnocované části obrazu se volí nakreslením obdélníku elementu Color myší. Lze nastavit pouze název.



Obrázek 173: Color detector

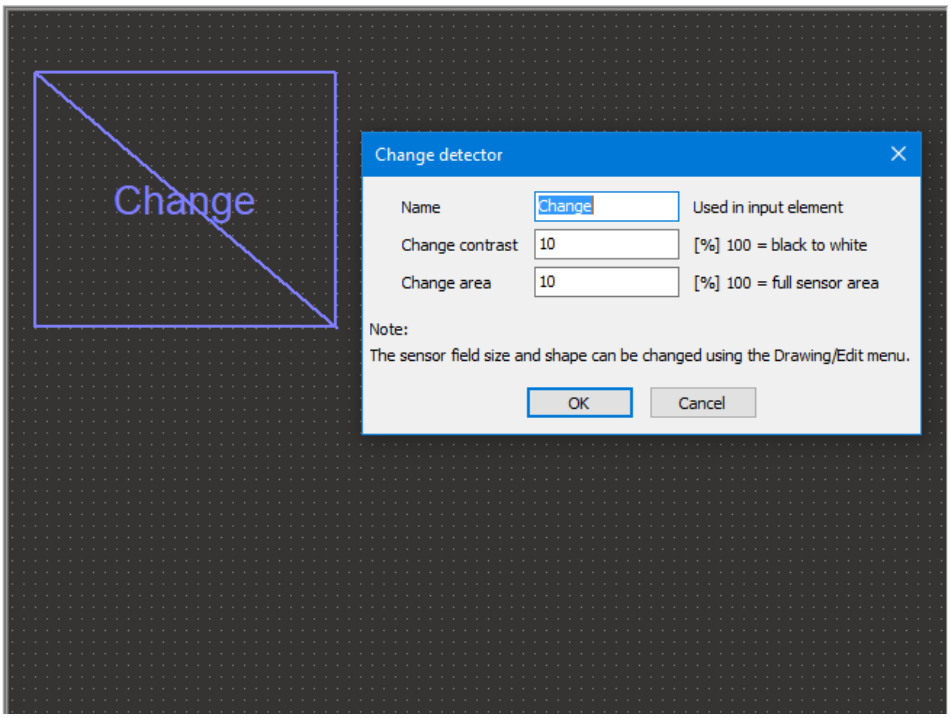
Camera input pak vrací hodnoty viz obr., se kterými lze dále pracovat v rámci procesu/programu/podprogramu. Jedná se o hodnoty jednotlivých základních barev v rámci snímaného obrazu.



Obrázek 174: Color detector – parametry pro použití v Camera input

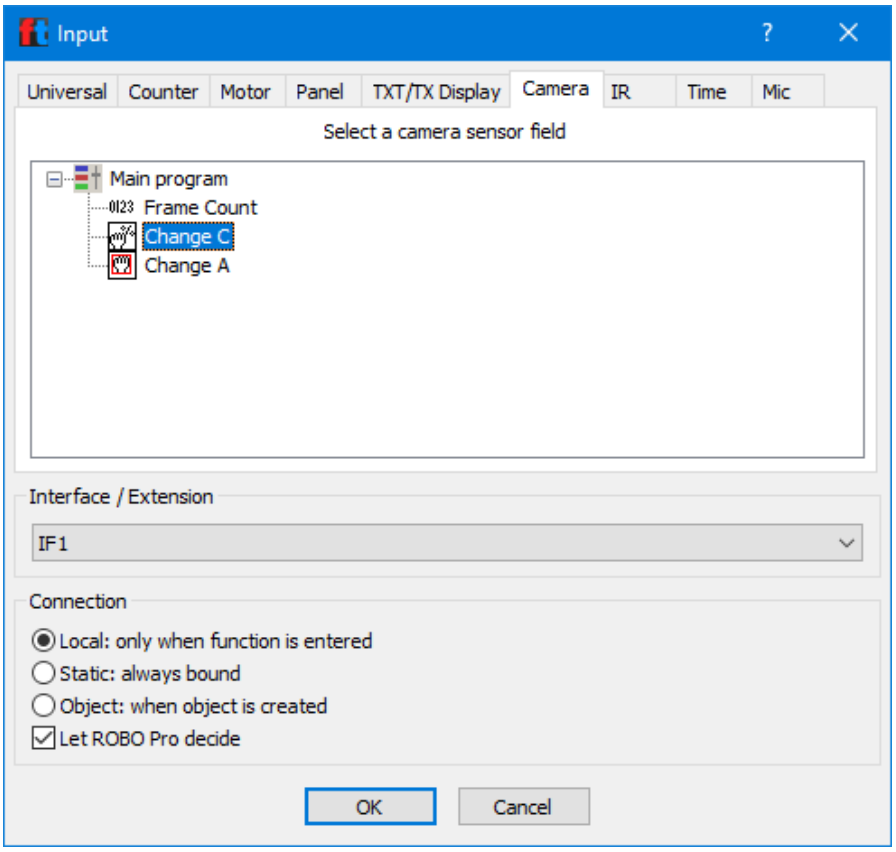
4.3.1.2 Movement detector (Change)

Element Movement detector slouží k vyhodnocování změn v kamerou snímaném prostředí, viz např. využití v praktickém příkladu „Alarm“. K vyhodnocování změn v označené části kamerou snímaného prostředí se provádí hlídáním změn v kontrastu. S nasnímanými hodnotami se pak pracuje v programu, kde je lze zobrazit, uložit do proměnné apod., viz předchozí elementy jako outputy, inputy, … Velikost a místo vyhodnocované části obrazu se volí nakreslením čtverce elementu Change myší. Lze nastavit název, citlivost na kontrast a oblast.



Obrázek 175: Movement detector (Change)

Camera input pak vrací hodnoty viz obr., se kterými lze dále pracovat v rámci procesu/programu/podprogramu. Change C hlídá změnu snímaného obrazu v rámci zadaných %, Change A hlídá jakoukoliv změnu.

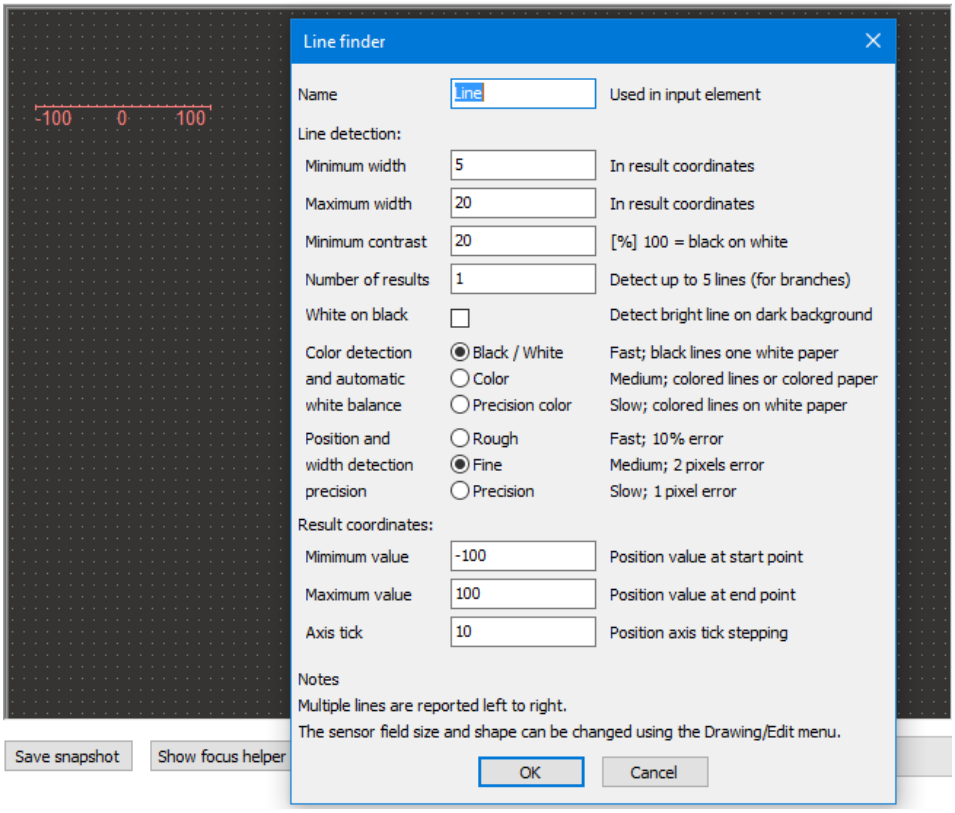


Obrázek 176Line finder – parametry pro použití v Camera input

4.3.1.3 Line finder

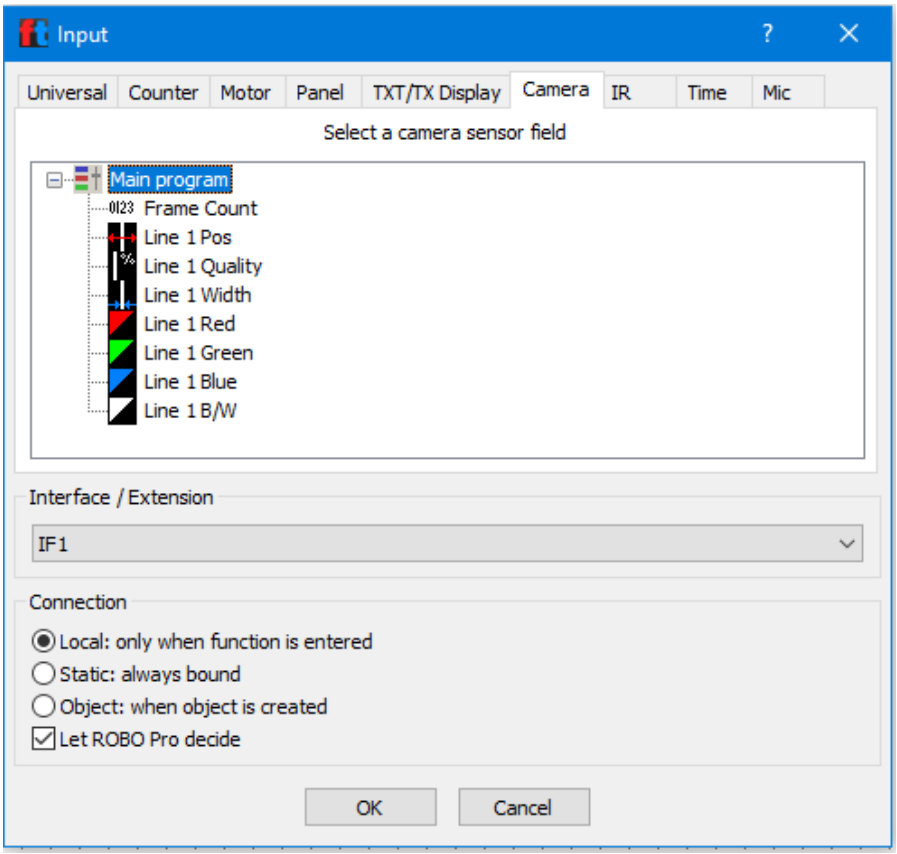
Element Line finder slouží k vyhledávání křivky, resp. Trasy (např. u robota jezdícího podle křivky na zemi). Lze nastavit:

- Name – název
- **Sekce Line detection**
  - Minimum width – minimální šířka křivky
  - Maximum width– maximální šířka křivky
  - Minimum contrast – minimální kontrast křivky proti podkladu, na kterém je namalována
  - Number of results – maximální počet křivek
  - White on black – detekce světlých křivek na tmavém podkladu
  - Color detection:
    - Black/White – rychlé vyhodnocení křivky, např. černá křivka na bílém papíru
    - Color – středně rychlé vyhodnocení křivky, např. barevná křivka, nebo barevný podklad
    - Precision color – pomalé vyhodnocení, barevná křivka na bílém papíru
  - Position and width detection precision (rychlost a přesnost detekce):
    - Fast – rychlé, 10% chybovost
    - Medium – střední, chybovost v rámci 2 pixelů
    - Slow – pomalá, chybovost v rámci 1 pixelu
- **Result coordinates (nastavení hodnot pro vyhodnocování, střed je uprostřed křivky)**
    - Minimum value – minimální hodnota, hodnota vlevo
    - Maximum value – maximální hodnota, hodnota vpravo
    - Axis tick – rozdělení na úseky



Obrázek 177: Line finder

Camera input pak vrací hodnoty viz obr., se kterými lze dále pracovat v rámci procesu/programu/podprogramu.

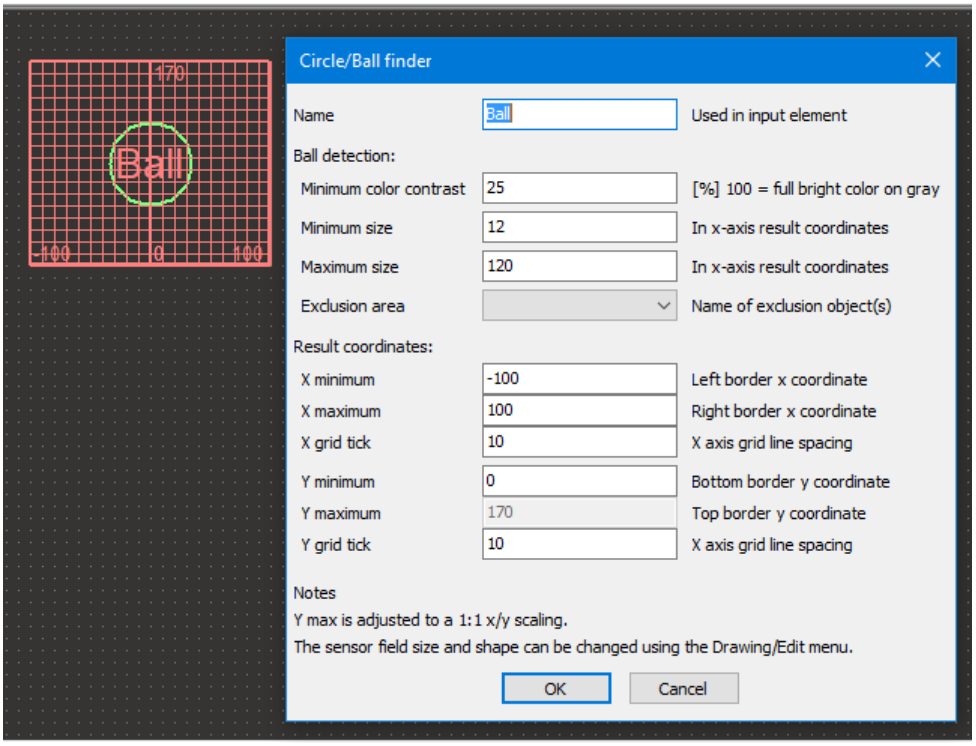


Obrázek 178: Line finder – parametry pro použití v Camera input

4.3.1.4 Ball finder

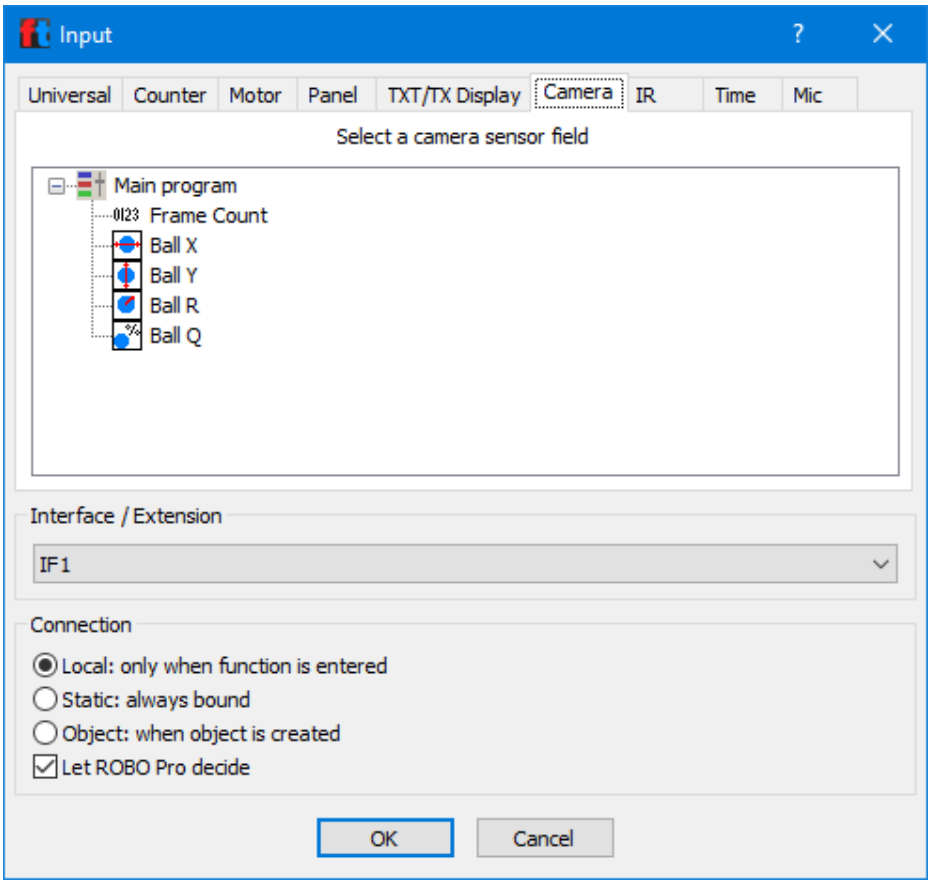
Element Ball finder slouží k vyhodnocení, zda je v zorném poli kamery míček a pokud ano, tak jaké má souřadnice X, Y, kontrast a rozměr. Velikost a místo vyhodnocované části obrazu se volí nakreslením obdélníku elementu Ball finder myší. Lze nastavit:

- Name – název
- **Sekce Ball detection**
  - Minimum color contrast – minimální kontrast objektu, aby byl detekován
  - Minimum size – minimální velikost objektu, aby byl detekován
  - Maximum size – maximální velikost objektu, aby byl detekován
  - Exclusion area – oblast, která se má vyloučit ve vyhodnocování. Vytváří se elementem Exclude. Takových oblastí může být použito více. Pokud chceme, aby bylo z vyhledávání objektu vyloučeno více oblastí, musí být tyto oblasti pojmenovány stejně.
- **Sekce Result coordinates**
  - X minimum – maximální rozsah v ose X doleva (levá hrana vloženého pole)
  - X maximum – maximální rozsah v ose X doprava (pravá hrana vloženého pole)
  - X grid tick – hustota mřížky v ose X
  - Y minimum – hodnota v dolní hraně vloženého pole v ose X
  - Y maximum – maximální hodnota v ose Y (horní hrana vloženého pole)
  - Y grid tick – hustota mřížky v ose Y



Obrázek 179: Ball finder

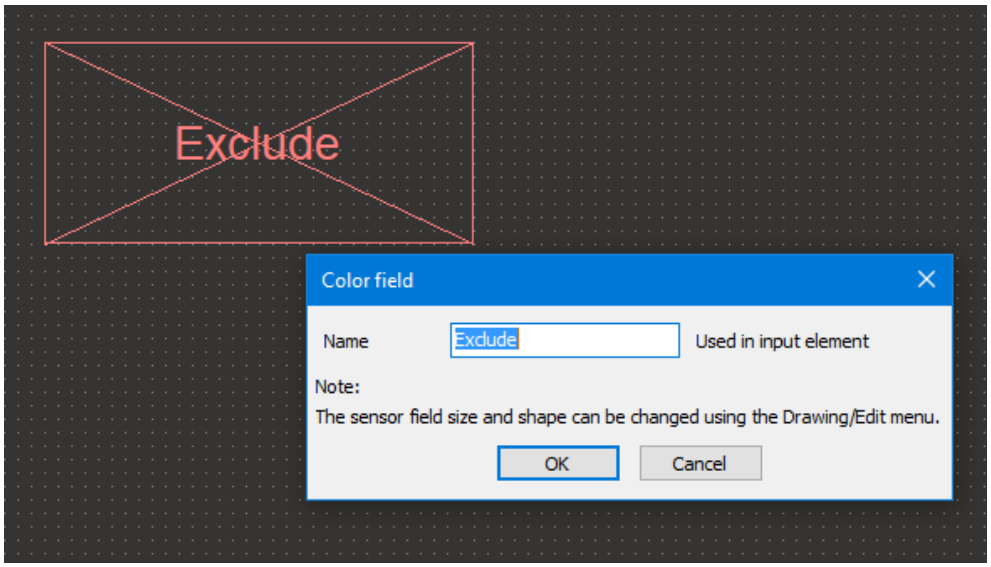
Camera input pak vrací hodnoty viz obr., se kterými lze dále pracovat v rámci procesu/programu/podprogramu. Tedy pozice v osách X a Y, průměr (R) a kontrast objektu oproti pozadí (Q).



Obrázek 180: Ball finder – parametry pro použití v Camera input

4.3.1.5 Exclude

Element Exclude slouží k vyznačení oblastí obrazu, které nemají být vyhodnocovány při použití Ball finderu. Velikost a místo vyhodnocované části obrazu se volí nakreslením čtverce elementu Exclude myší. Lze nastavit pouze název. Pokud se použije více takových elementů a mají se vztahovat všechny k jednomu Ball finderu, musí míst stejný názvy.



Obrázek 181: Exclude

4.4 TXT/TX display

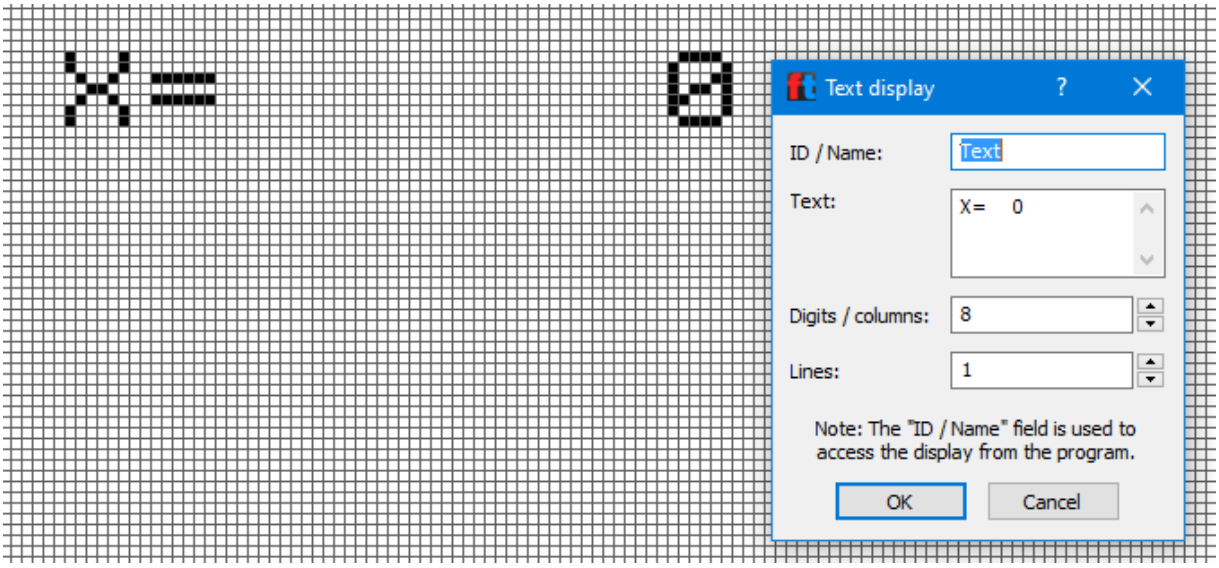
Sekce obsahující elementy pro tvorbu softwarových ovládacích prvků a zobrazovacích prvků, které se zobrazují na dotykovém displeji řídící jednotky. Lze je používat na záložce TXT/TX Display, která prezentuje displej řídící jednotky.

4.4.1 Displays – Text display

Element je obdobou Text displeje ze sekce Operating elements. Jediný rozdíl je, že tento element neslouží k zobrazení na monitoru počítače, ale na displeji řídící jednotky. Lze mu nastavit:

- Název – název elementu.
- Text - text který se zobrazuje v Text display).
- Digits/columns – počet pozic grafického elementu. Potřeba vždy vyzkoušet, aby se hodnoty zobrazovaly celé.
- Lines – počet řádků pole.

Element se opět musí propojit s elementem Panel display, ze kterého čerpá údaje k zobrazení.

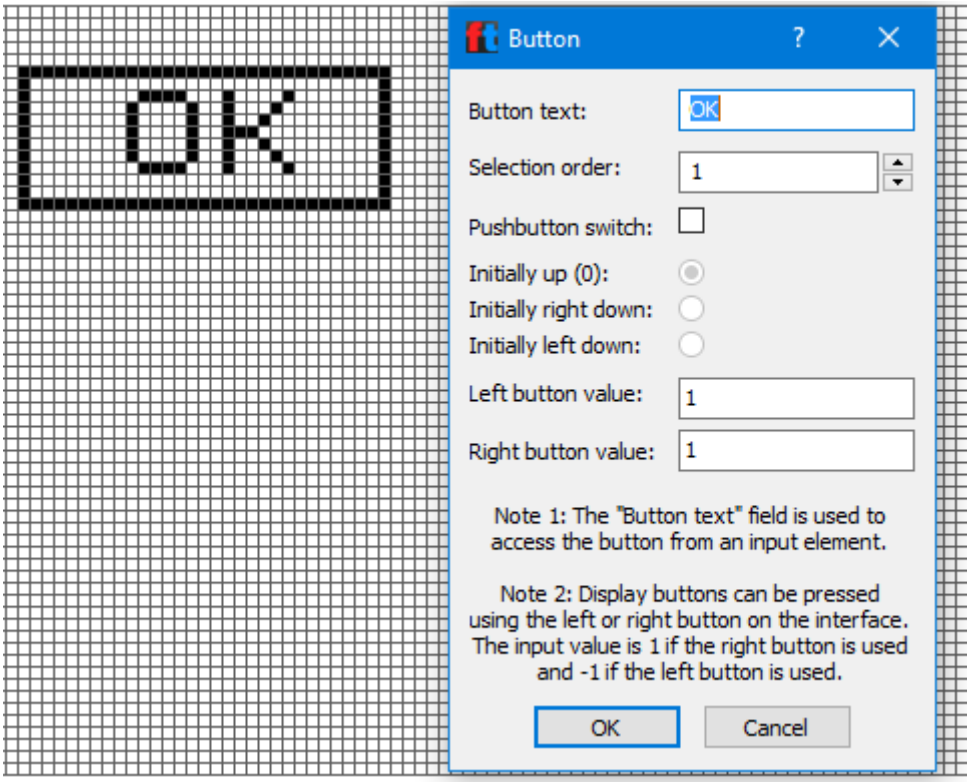


Obrázek 182: Text display pro zobrazení na displejii řídící jednotky

4.4.2 Control elements – Button

Element je obdobou elementu Button ze sekce Operating elements. Slouží k ovládání programu formou tlačítka. Velikost ovládacího tlačítka je pevně dána.

Elementu lze nastavit Název, barvu tlačítka, barvu textu a zvolit, zda má fungovat jako přepínač (standardně se chová jako spínač), a v případě přepínače zda má mít výchozí stav „stisknuto“. Tlačítko je navíc pomyslně rozděleno na levou a pravou část. Lze tedy nastavit, co se stane při stisknutí tlačítka v levé části a co v pravé části – proto lze nastavovat Left button value a Right button value. Element se pak propojí buď s požadovaným inputem, nebo proměnnou, která se používá v procesu/programu. Pokud se má tlačítko používat bez tohoto pomyslného rozdělení, tak nejlepším řešením je zadat pro levou i pravou stranu stejnou hodnotu do políčka Raight button value.

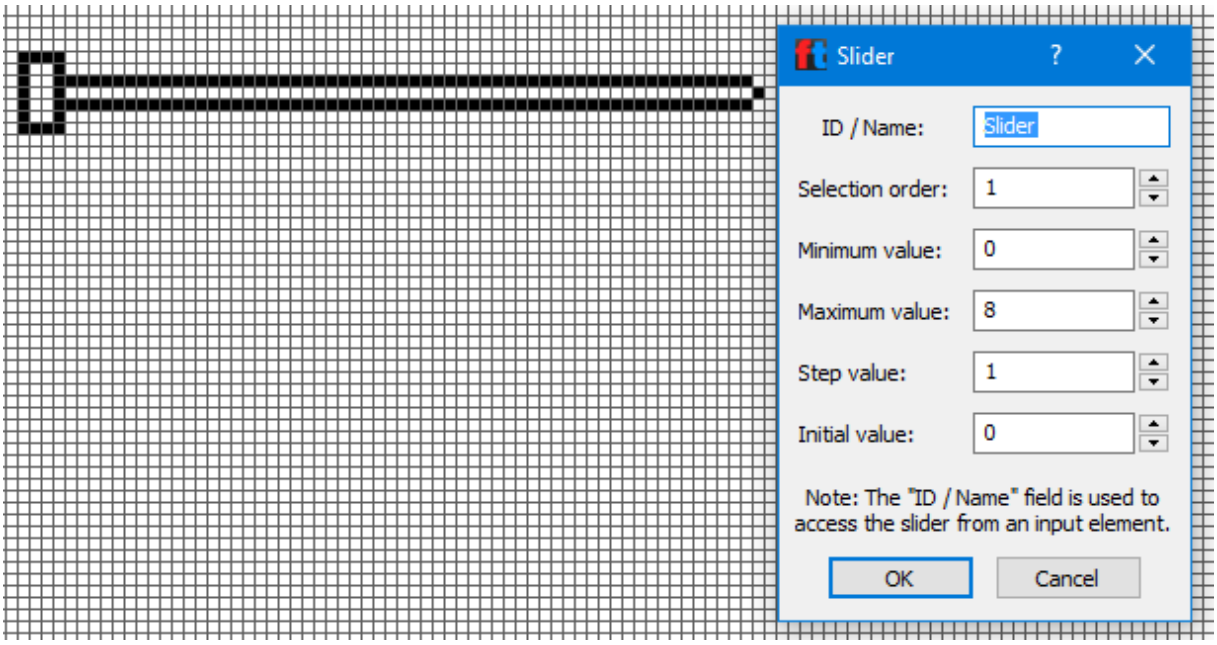


Obrázek 183: Control elements - Button

4.4.3 Control elements – sliders

Element je obdobou elementu Slider ze sekce Operating element. Slouží k ovládání komponent formou posuvníku a je možné volit mezi vertikálním a horizontálním provedením.

Lze mu nastavit Název, barvu, minimální hodnotu, maximální hodnotu a počáteční hodnotu. Např. pro XS motor by mohla být minimální hodnota 0, maximální 8 a počáteční 0. Slider by se tak používal pro ovládání otáček motoru s možností úplného vypnutí motoru. Element se pak propojí buď s požadovaným inputem, nebo proměnnou, která se používá v procesu/programu.



Obrázek 184: Control elements – Slider

Poděkování

Tento dokument byl vytvořen za podpory interního grantu Západočeské univerzity, číslo projektu SGS-2018-031, s názvem „Optimalizace parametrů udržitelného výrobního systému“.

5 Seznam obrázků

6 Seznam tabulek

7 Zdroje

[1] ROBOTICS TXT Electropneumatic - fischertechnik. [online]. Copyright © [cit. 16.09.2019]. Dostupné z: https://www.fischertechnik.de/en/service/elearning/playing/txt-electropneumatic