

Konfigurace vstupů a výstupů na cílových platformách

Uživatelská příručka

REX Controls s.r.o.

Verze 3.0.4
27.3.2025
Plzeň

Obsah

1	Úvod	2
2	Fyzická připojení	3
2.1	WAGO PFC100/200	3
2.2	Raspberry Pi	5
2.3	Monarco HAT	5
2.4	REX M527, Unipi Patron/Axon/Neuron	5
2.5	Weidmüller u-OS PLC (WL2000, M3000, M4000)	6
3	Přidání vstupů a výstupů do projektu	8
4	Práce se vstupními a výstupními signály	9
4.1	Obecné kroky pro všechny platformy	9
4.2	Specifické parametry pro jednotlivé platformy	10
4.2.1	WAGO PFC100/200	10
4.2.2	Raspberry Pi	11
4.2.3	Monarco HAT	12
4.2.4	REX M527, Unipi Patron/Axon/Neuron	13
4.2.5	Weidmüller u-OS PLC (WL2000, M3000, M4000)	14
5	Interakce s algoritmem	16
6	Aktualizace HMI	17
7	Příklady	19
8	Další kroky	20
	Literatura	20

Kapitola 1

Úvod

Pro demonstraci konfigurace vstupů a výstupů na různých platformách využijeme projekt vytvořený v uživatelské příručce [1] a budeme v něm pokračovat. Dosud algoritmus neinteragoval s vnějším světem, nebyl připojen k žádnému fyzickému signálu ani externím datům. Nyní budeme používat vstupní a výstupní signály platformy pro interakci se skutečným světem. Pro Raspberry Pi použijeme GPIO piny, které mohou být nakonfigurovány jako vstupy nebo výstupy.

Jako vstupy budou použity dva fyzické spínače a softwarový časovač bude řídit jeden výstupní signál. Jinými slovy, tato příručka je věnována přechodu od obecného příkladu 0101-01 k výchozímu projektu pro každou platformu ukázaném v [2].

Kapitola 2

Fyzická připojení

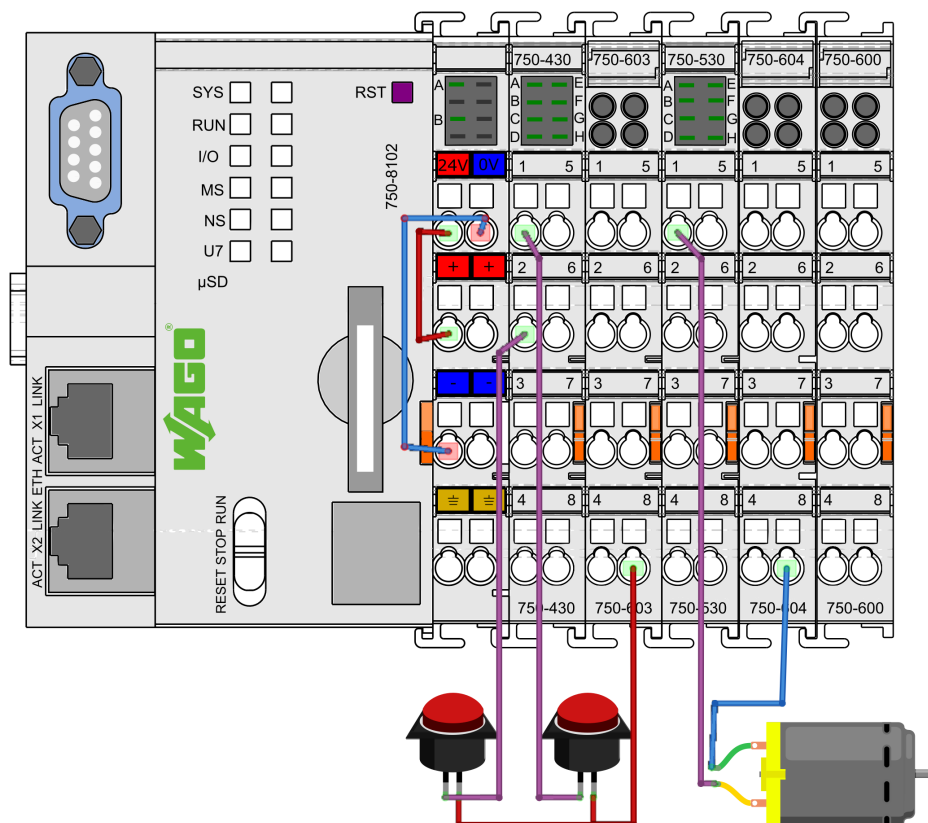
Tato část postupu je pro každé cílové zařízení individuální. Přejděte na sekci určenou pro vaši platformu a po zapojení pokračujte Kapitolou [3](#).

2.1 WAGO PFC100/200

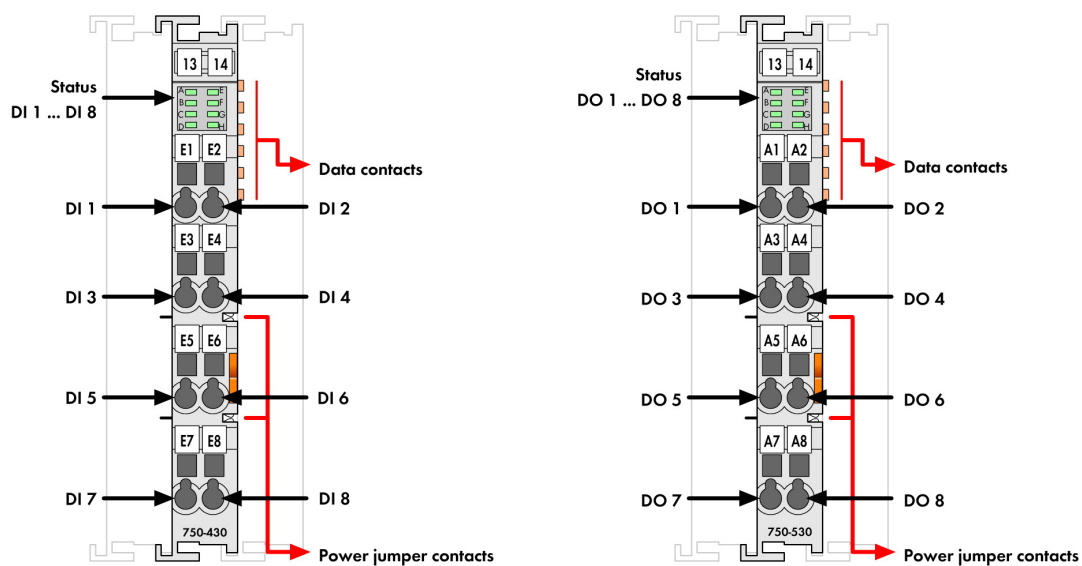
Pro tento ukázkový projekt se předpokládá následující jednoduchá sestava:

- 1x 750-8102 PFC100 nebo 750-8202 PFC200 řadič s napájením
- 1x 750-430 modul (8 digitálních vstupů)
- 1x 750-603 modul (24V potenciálová distribuce)
- 1x 750-530 modul (8 digitálních výstupů)
- 1x 750-604 modul (0V potenciálová distribuce)
- 1x 750-600 povinný koncový modul

Připojte spínače a externí relé, jak je znázorněno níže. Relé je volitelné, jelikož stav výstupu indikuje i vestavěná LED modulu.



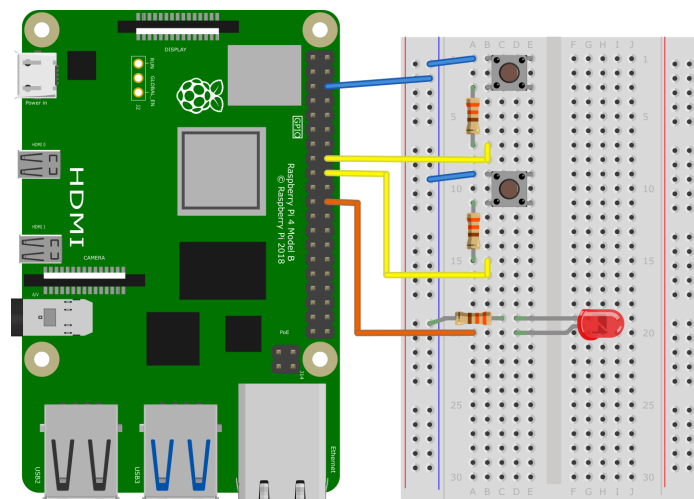
DŮLEŽITÉ: Upozorňujeme, že číslování terminálů na modulech WAGO (výše) se neshoduje s číslováním kanálů (níže)! Vždy používejte označení DI1..8, DO1..8, AI1..4 atd. podle dokumentace WAGO.



Pro podrobnosti a příklady zapojení pro platformu se podívejte do dokumentace modulu I/O systému WAGO 750/753.

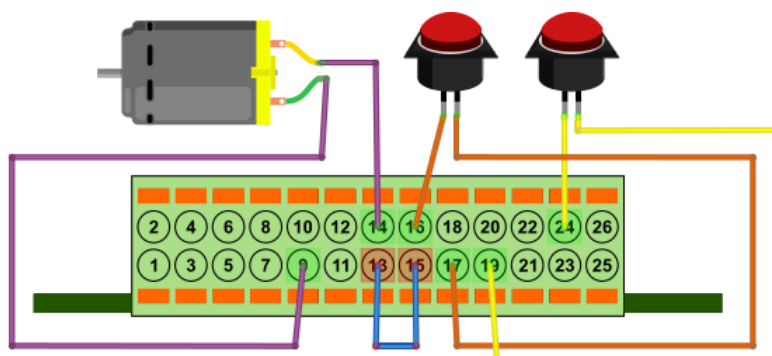
2.2 Raspberry Pi

Pripojte spínače, ochranné odpory (330 Ohm) a indikátor LED, jak je znázorněno níže.



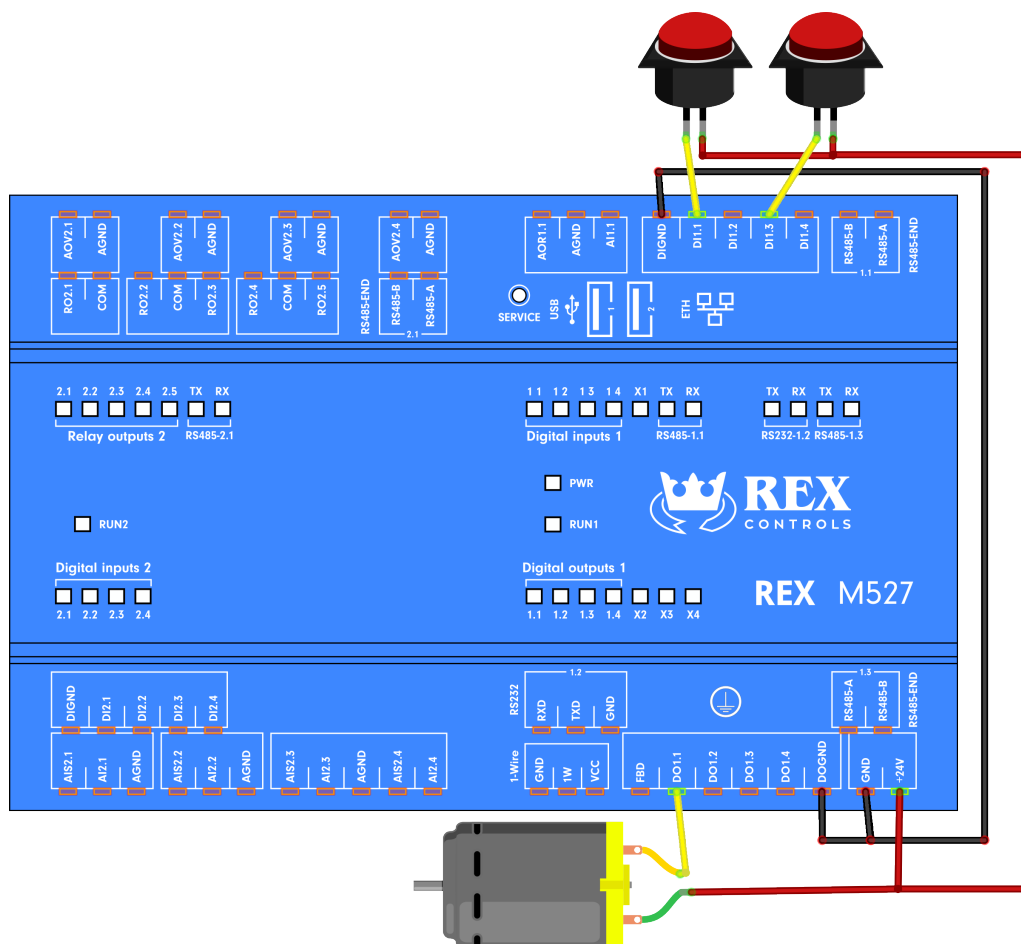
2.3 Monarco HAT

Připojte spínače, jak je znázorněno níže. Motor je volitelný, jelikož stav výstupu indikuje i vestavěná LED.



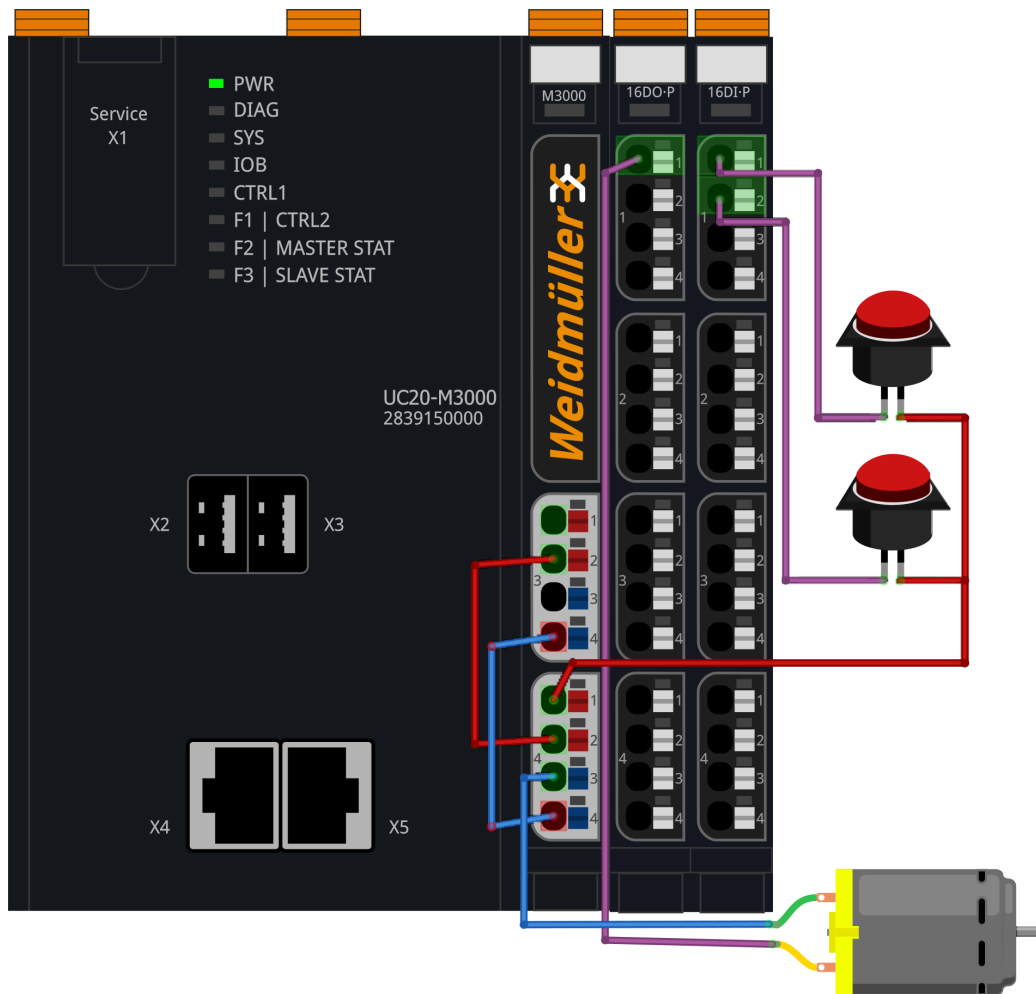
2.4 REX M527, Unipi Patron/Axon/Neuron

Připojte spínače, jak je znázorněno níže. Motor je volitelný, jelikož stav výstupu indikuje i vestavěná LED.



2.5 Weidmüller u-OS PLC (WL2000, M3000, M4000)

Připojte spínače, jak je znázorněno níže. Motor je volitelný, jelikož stav výstupu indikuje i vestavěná LED.

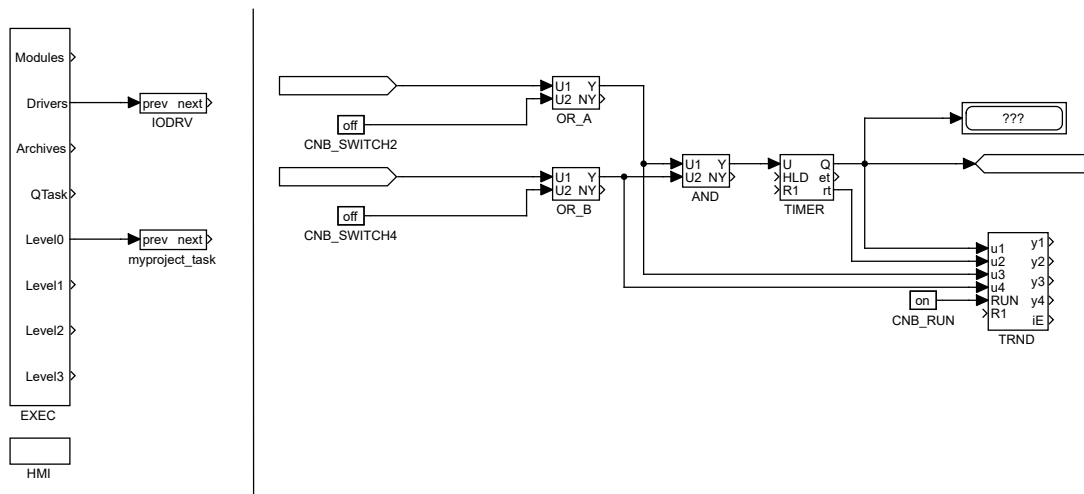


Kapitola 3

Přidání vstupů a výstupů do projektu

Nyní, když máme všechno zapojené, je čas zahrnout fyzické signály do algoritmu. Je třeba rozšířit hlavní soubor projektu o další funkční blok pro přístup k vstupům a výstupům z řídicího algoritmu ve vašem projektu. Vložte blok EXEC/IODRV z Knihovny bloků (Block Library) a připojte jej k bloku EXEC, jak je znázorněno níže.

V TASKu odstraňte bloky CNB_SWITCH1 a CNB_SWITCH3 a nahraďte je bloky INOUT/From, což budou vstupní signály. Také přidejte jeden blok INOUT/Goto, který bude sloužit jako výstup a který bude řízen časovačem. Připomínáme, že nová větev čáry se vytváří tažením pravým tlačítkem myši.



Kapitola 4

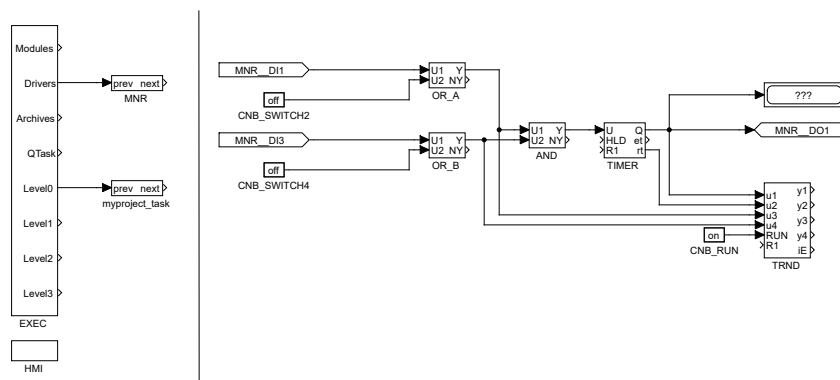
Práce se vstupními a výstupními signály

Nyní řekneme kompilátoru, aby používal konkrétní ovladač I/O. K tomu je třeba upravit parametry bloku `IODRV` podle následujících kroků:

4.1 Obecné kroky pro všechny platformy

1. Nastavte blok `IODRV`, aby komunikoval s příslušným modulem pro cílovou platformu, upravením parametrů bloku:
 - `module=<název modulu>`
 - `classname=<název třídy>`
 - `cfgname` – nastavte podle potřeby (viz níže).
 - `factor=1`
 - Ostatní parametry nechte beze změny.
2. Přejmenujte blok `IODRV` podle doporučení pro jednotlivé platformy (např. `WG` pro `WAGO`, `GPIO` pro `Raspberry Pi`). Toto jméno bude zároveň prefixem pro všechny signály I/O tohoto ovladače.
3. Pokud bude potřeba, vytvořte konfigurační soubor pro ovladač I/O. Klikněte na tlačítko *Configure* a vyplňte formulář podle potřeby. Poté klikněte na *OK* a soubor bude uložen do složky projektu.
4. V blocích `From` a `Goto` nastavte parametr `GotoTag` pro vstupní/výstupní signály. Parametr `GotoTag` musí být ve formátu `<prefix>__<název signálu>`, kde prefix je název bloku `IODRV`. Všimněte si dvojitého znaku podtržení. Například `WG__S1M430C1` pro kanál 1 modulu 750-430 v paměťovém slotu 1.

Na konci této Kapitoly by Váš projekt měl vypadat, jak je vidět na obrázku níže s tím, že jméno bloku `IODRV` a prefixy pro signály budou odpovídat platformě, kterou používáte.



4.2 Specifické parametry pro jednotlivé platformy

4.2.1 WAGO PFC100/200

- `module=WagoDrv`
- `classname=WagoDrv`
- `cfgname` – prázdné.
- Doporučené jméno `IODRV` a prefix pro signály: **WG**

Pro potřeby projektu nastavte pro bloky **From** parametry **GotoTag** jako:

- **WG__S1M430C1** – První fyzický přepínač je připojen ke kanálu 1 (DI1) modulu 750-430, který je ve slotu 1.
- **WG__S1M430C3** – Druhý fyzický přepínač je připojen ke kanálu 3 stejného modulu.

Kanál 1 modulu 750-530 ve slotu 2 bude sloužit jako řídicí signál pro externí relé. Nastavte tedy pro blok **Goto** (**GotoTag** = **WG__S2M530C1**).

DŮLEŽITĚ: Moduly pro rozdělení potenciálu neposkytují žádná data, proto nezbírají žádnou paměť a nepočítají se při číslování slotů modulů!

Podobně pro ostatní I/O můžeme použít následující vlajky:

- **Goto**, **WG__S2M530C8** – kanál 8 modulu 750-530 ve slotu 2
- **From**, **WG__S3M555C2** – kanál 2 modulu 750-555 ve slotu 3 (v tomto příkladu není přítomný)
- atd.

Podrobný popis ovladače I/O pro WAGO PFC100/200 je k dispozici v samostatném manuálu [3].

4.2.2 Raspberry Pi

- `module=GpioDrv`
- `classname=GpioDrv`
- `cfgname=gpio.rio`
- Doporučené jméno IODRV a prefix pro signály: GPIO

Nezapomeňte kliknout na tlačítko *Configure*. Tím vytvoříte výchozí soubor konfigurace s názvem `(.rio)`. Pro Raspberry Pi 5 zadejte parametr `GPIO Chip=gpiochip4`. Pro starší verze `GPIO Chip=gpiochip0`. Poté zavřete dialog.

Pro potřeby projektu nastavte pro bloky **From** parametry **GotoTag** jako:

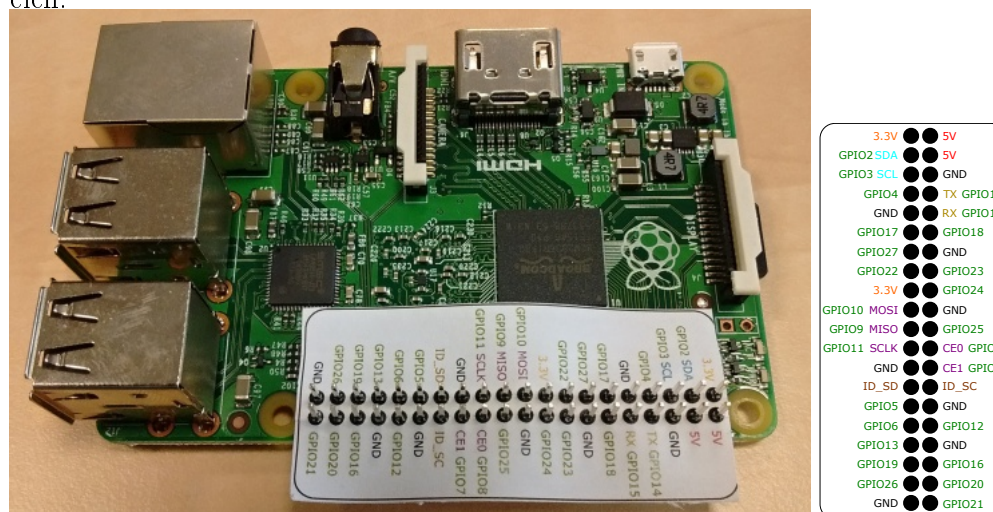
- `GPIO__GPIO23UI` – První fyzický přepínač bude připojen k `GPIO#23`, bude aktivován interní pull-up rezistor a logika vstupů bude invertována.
- `GPIO__GPIO24UI` – Druhý fyzický přepínač bude připojen k `GPIO#24`, bude aktivován interní pull-up rezistor a logika vstupů bude invertována.

Výstup časovače bude připojen k `GPIO#25` a bude sloužit jako výstupní signál. Nastavte tedy pro blok **Goto** (`GotoTag = GPIO__GPIO25`).

Podobně pro ostatní piny můžeme použít následující vlajky:

- `Goto, GPIO__GPIO22` – digitální výstup 22
- `From, GPIO__GPIO7U` – digitální vstup 7 s interním pull-up rezistorem
- `From, GPIO__GPIO8D` – digitální vstup 8 s interním pull-down rezistorem
- `From, GPIO__GPIO21` – digitální vstup 21 bez pull-up/down rezistoru

Číslování pinů na Raspberry Pi B+ a novějších je zobrazeno na následujících obrázcích:

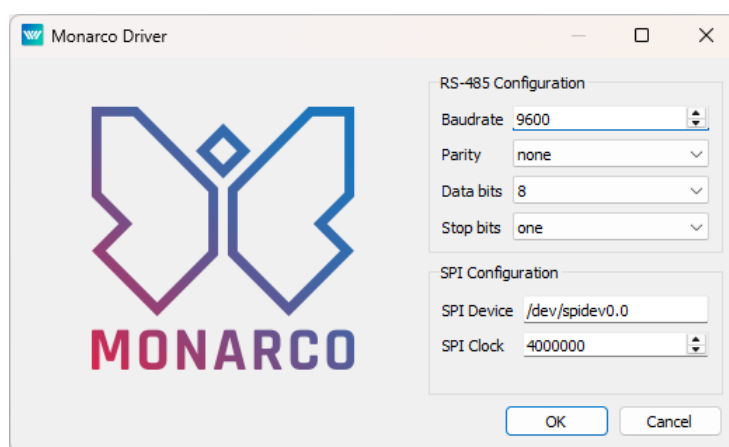


Navštivte stránku http://elinux.org/RPi_Low-level_peripherals pro podrobné informace o jednotlivých GPIO pinech.

4.2.3 Monarco HAT

- `module=MonarcoDrv`
- `classname=MonarcoHatDrv`
- `cfgname=monarcohat.rio`
- Doporučené jméno IODRV a prefix pro signály: **MNR**

Nezapomeňte kliknout na tlačítko *Configure*. Tím vytvoříte výchozí soubor konfigurace s názvem (.rio). Ponechte výchozí hodnoty a zavřete dialog.



Pro potřeby projektu nastavte pro bloky **From** parametry **GotoTag** jako:

- **MNR__DI1** – První fyzický přepínač je připojen k DI1.
- **MNR__DI3** – Druhý fyzický přepínač je připojen k DI3.

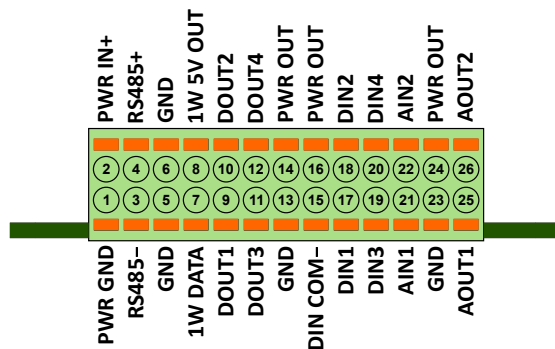
Výstup časovače bude připojen k DO1 a bude sloužit jako výstupní signál. Nastavte tedy pro blok **Goto** (**GotoTag** = **MNR__D01**).

Podobně pro ostatní piny můžeme použít následující vlajky:

- **Goto**, **MNR__D04** – digitální výstup 4
- **From**, **MNR__DI3** – digitální vstup 3
- **From**, **MNR__AI1** – analogový vstup 1

Podrobný popis ovladače I/O pro Monarco HAT je k dispozici v samostatném manuálu [4].

Pinout zařízení Monarco HAT je vyobrazen na následujícím obrázku:



4.2.4 REX M527, Unipi Patron/Axon/Neuron

- `module=MBDrv`
- `classname=MtmDrv`
- `cfgname=patron_cfg.rio`
- Doporučené jméno IODRV a prefix pro signály: PTN

Musíte vytvořit soubor `patron_cfg.rio`. Klikněte na tlačítko *Configure* a vyplňte formulář, jak je uvedeno níže. Poté klikněte na *OK* a soubor bude uložen do složky projektu.

Name	Address	Port	Max. requests	Subaddress
1 slaveMB	127.0.0.1	502	16	0

Name	Slave	Item Address	Type	Count	Init Value	Flags	Function code
1 DO1_1	slaveMB	0	COIL	1		W	5 - write single coil
2 DI1_1	slaveMB	4	COIL	1		R	1 - read coils
3 DI1_3	slaveMB	6	COIL	1		R	1 - read coils

Pro potřeby projektu nastavte pro bloky *From* parametry *GotoTag* jako:

- PTN__DI1_1 – První fyzický přepínač je připojen k první skupině digitálních vstupů - vstup 1.
- PTN__DI1_3 – Druhý fyzický přepínač je připojen k digitálnímu vstupu 1.3.

Výstup časovače bude připojen k digitálnímu výstupu 1.1 a bude sloužit jako výstupní signál. Nastavte tedy pro blok **Goto** (**GotoTag** = PTN__DO1_1).

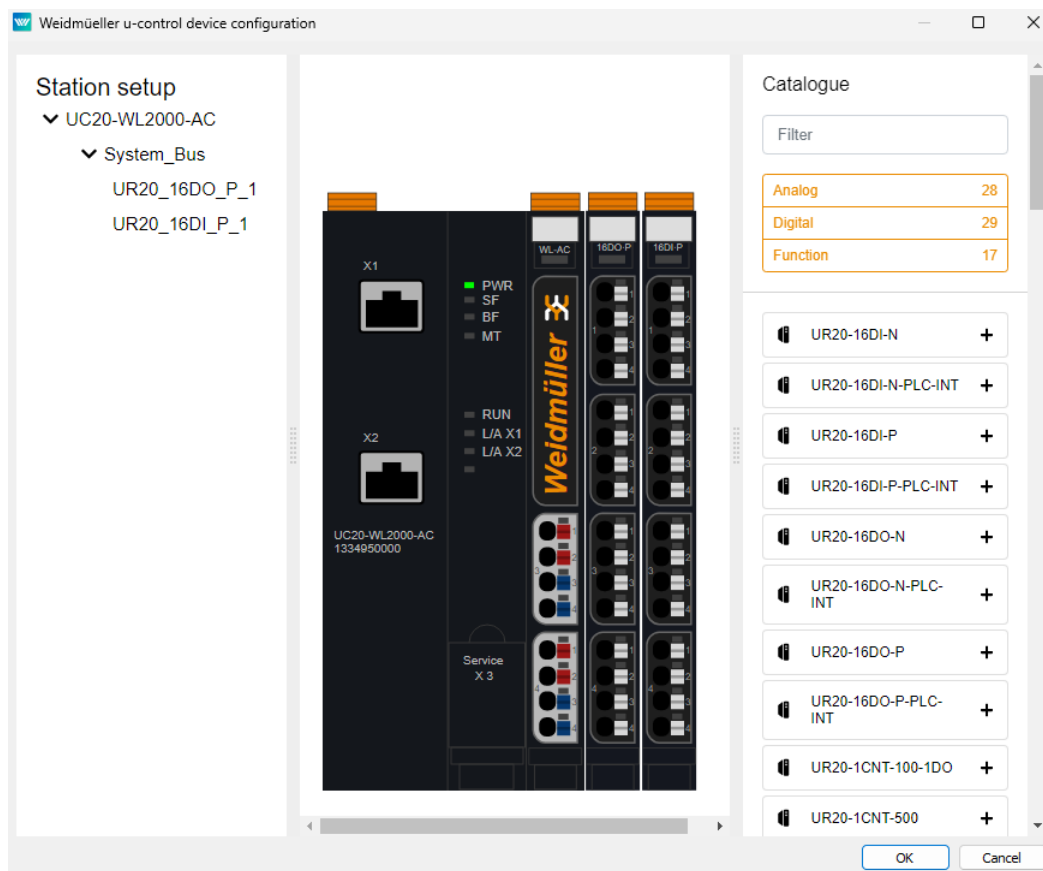
Všimněte si, že komunikace se zřízením probíhá pomocí protokolu Modbus TCP. Podrobný popis ovladače pro Modbus TCP je k dispozici v samostatném manuálu [5].

Poznámka: Pokud používáte starší Unipi Axon nebo Neuron, je nastavení stejné. Můžete pro přehlednost použít prefix **AXN**, resp. **NRN** místo **PTN**.

4.2.5 Weidmüller u-OS PLC (WL2000, M3000, M4000)

- `module=UControlDrv`
- `classname=UControlDrv`
- `cfgname=weidmueller_cfg.rio`
- Doporučené jméno IODRV a prefix pro signály: **WM**

Musíte vytvořit soubor `weidmueller_cfg.rio`. Klikněte na tlačítko *Configure*, v katalogu po pravé straně postupně vyberte I/O moduly, které máte připojené a naskládejte je za PLC v prostřední části ve správném pořadí. Poté klikněte na *OK* a soubor bude uložen do složky projektu.



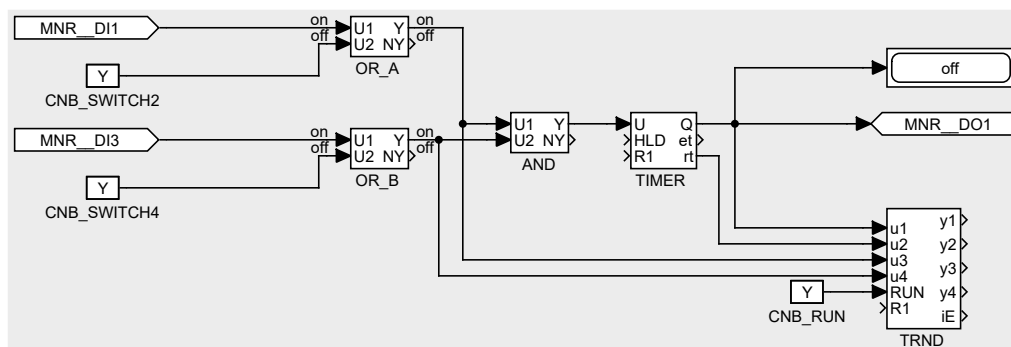
Nyní byste měli mít v Knihovně bloků připravenou novou knihovnu se jménem driveru, tedy WM. Uvnitř najdete vygenerované bloky **From** a **Goto** podle modulů, které jste přidali do konfigurace driveru. Vyberte **From** bloky:

- WM__UR20_16DI_P_1_Value_0 – První fyzický přepínač je připojen k modulu 16DI_P k prvnímu vstupu.
- WM__UR20_16DI_P_1_Value_1 – Druhý fyzický přepínač je připojen k modulu 16DI_P k druhému vstupu.

Výstup časovače bude připojen k modulu 16DO_P k prvnímu výstupu. Vyberte tedy pro blok WM__UR20_16DO_P_1_Value_0.

Interakce s algoritmem

Po kompilaci projektu a jeho stažení na platformu interaguje řídicí algoritmus s fyzickým světem. Opět je možné přepnout do **Watch mode** a pozorovat signály v reálném čase nebo analyzovat trendy signálů. Přepněte fyzické přepínače a sledujte signály.



Kapitola 6

Aktualizace HMI

Je také nutné aktualizovat HMI. Bloky CNB_SWITCH1 a CNB_SWITCH3 již nejsou v algoritmu přítomné. Navíc je potřeba nahradit virtuální vstupní prvky (DW) indikátory (DR). Otevřete tedy soubor `index.hmi.js` a nahraďte

```
{type: 'DW', alias: 'switch1', desc: 'Spinac 1', cstring: 'myproject_task.  
CNB_SWITCH1:YCN'},
```

tímto

```
{type: 'DR', alias: 'switch1', desc: 'Spinac 1', cstring: 'myproject_task.OR_A:  
U1'},
```

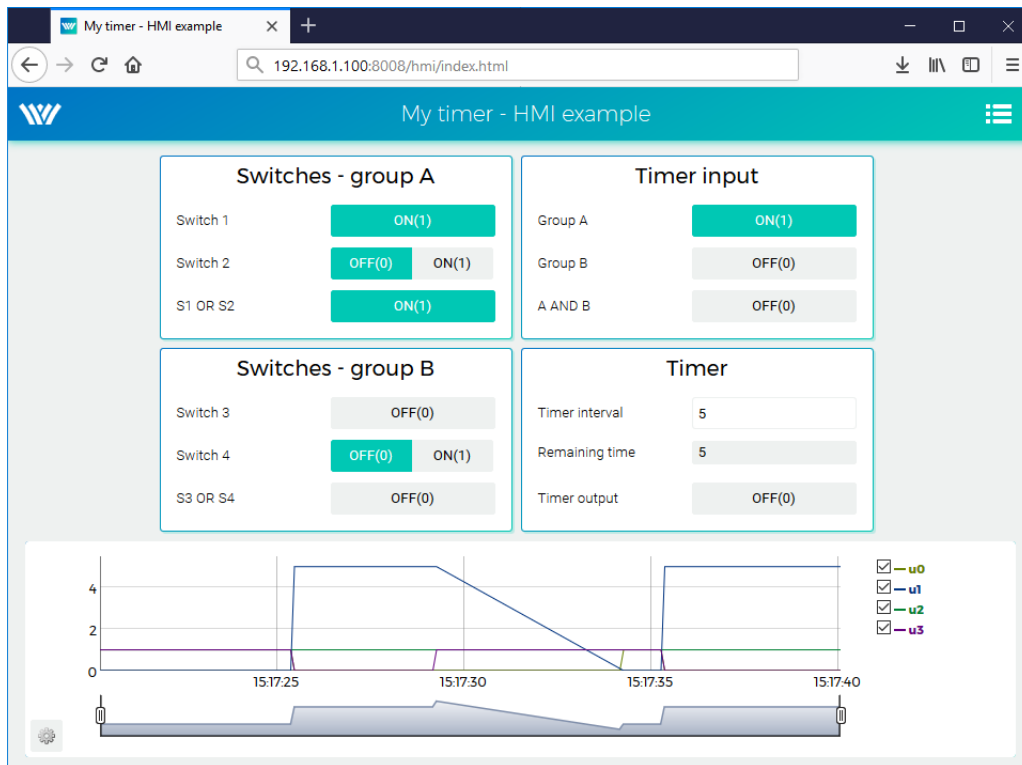
Podobně pro přepínač č. 3, nahraďte

```
{type: 'DW', alias: 'switch3', desc: 'Spinac 3', cstring: 'myproject_task.  
CNB_SWITCH3:YCN'},
```

tímto

```
{type: 'DR', alias: 'switch3', desc: 'Spinac 3', cstring: 'myproject_task.OR_B:  
U1'},
```

Uložte soubor, zkompilujte a stáhněte projekt znovu a otevřete webové rozhraní. Stiskněte dva fyzické přepínače a čekejte, až časovač spustí výstup. Alternativně můžete stále používat virtuální přepínače. To ukazuje, že můžete kombinovat fyzické a virtuální vstupní prvky.



Kapitola 7

Příklady

Ukázkové projekty a sada všech podporovaných I/O vlajek jsou zahrnuty v instalačním balíčku vývojových nástrojů REXYGEN. V REXYGEN Studio přejděte do nabídky **File** → **Start** → **Start from an Example Project** a vyberte jeden z příkladů platformy. Jak již bylo zmíněno, nejaktuálnější informace o příkladech jsou k dispozici na <https://www.rexygen.com/example-projects/>

Kapitola 8

Další kroky

Gratulujeme, vytvořili jste svůj první ukázkový projekt od začátku! Naučili jste se, jak vyvíjet, kompilovat a spouštět algoritmy na své platformě. Zvládli jste konfiguraci a práci s vstupně-výstupním ovladačem systému REXYGEN pro interakci se senzory a aktuátory. Také jste získali zkušenosti s tvorbou uživatelských rozhraní pomocí WebBuDi.

Impozantní pokrok za tak krátkou dobu, nemyslíte? Cílem těchto tutoriálů bylo rychle vás seznámit s klíčovými kroky a nástroji potřebnými k rozvoji projektů. Pro systém REXYGEN existuje také nástroj pro tvorbu vizualizací - REXYGEN HMI Designer [6], který je vhodný pro složitější vizualizace u větších projektů. Podrobnosti o tvorbě vizualizací v REXYGEN HMI Designer naleznete v samostatném manuálu [7].

Nyní je čas zaměřit se na svůj vlastní projekt a prohloubit své znalosti. Existují funkční bloky mnohem výkonnější než ty zmíněné v této příručce, mnoho inspirativních příkladových projektů a další vstupně-výstupní ovladače, které můžete využít k rozšíření vašeho projektu. Objevíte také různé způsoby výměny dat s externími systémy a zařízeními, a mnoho dalšího.

Pamatujte, že kdykoliv dosáhnete nějakého úspěchu, o který se chcete podělit, rádi si o něm poslechneme. A kdykoliv narazíte na nějaké potíže, rádi vám pomůžeme. Kontaktujte nás kdykoliv na support@rexygen.com.

Literatura

- [1] REX Controls s.r.o.. *První projekt*, 2024. [→](#).
- [2] REX Controls s.r.o.. *Začínáme s řídicím systémem REXYGEN*, 2024. [→](#).
- [3] REX Controls s.r.o.. *Ovladač WagoDrv systému REXYGEN pro Wago PFC100/PFC200 – Uživatelská příručka*, 2020. [→](#).
- [4] REX Controls s.r.o.. *Ovladač MonarcoDrv systému REXYGEN – Uživatelská příručka*, 2020. [→](#).
- [5] REX Controls s.r.o.. *Ovladač systému REXYGEN pro komunikaci Modbus – Uživatelská příručka*, 2020. [→](#).
- [6] REX Controls s.r.o.. *REXYGEN HMI – Uživatelská příručka*, 2024. [→](#).
- [7] REX Controls s.r.o.. *Tvorba HMI v REXYGEN HMI Designer*, 2024. [→](#).