

# Průvodce programováním AMiNi-E jazykem STL

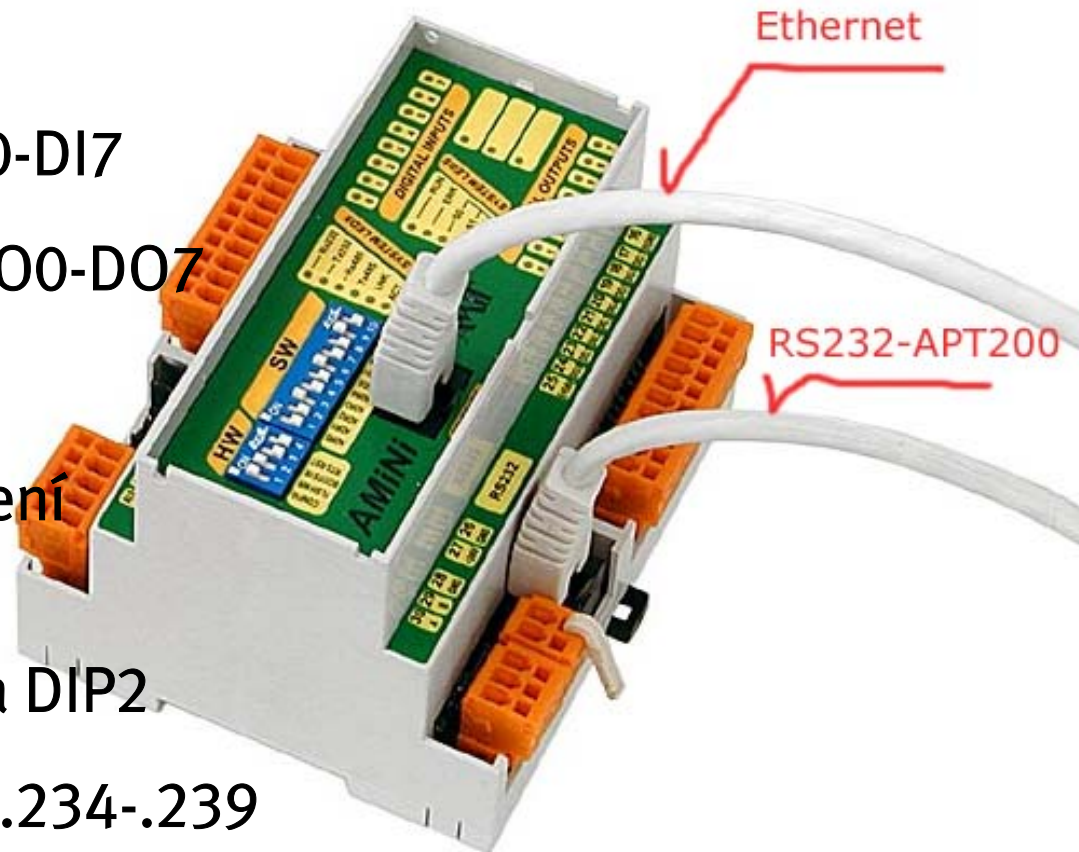
Dipl. Ing. Pavel Votrubec

## Základní body:

- HW popis PLC AMiNi-E
- Definice potřebných vstupů
- Definice potřebných výstupů
- Definice potřebných proměnných
- Definice potřebných aliasů
- Vlastní tvorba procesů a algoritmů
- Odlad'ování úlohy

## AMiNi-E – relevantní hardware parametry

- 8 digitálních vstupů DI0-DI7
- 8 digitálních výstupů DO0-DO7
- Ethernet přípojku
- RS232 přípojku (připojení Terminálu APT200)
- HW Adresy ŘS 4 až 9 na DIP2
- IP adresy ŘS 192.168.1.234-.239



## SW – DIP pro HW adresu AMiNi-E – 5 bitová adresa

HW Adresa ŘS na 10 bitovém DIP SW:

ADR0, ADR1, ADR2, ADR3, ADR4

Váhy: 1 2 4 8 16

4 – 0 0 1 0 0

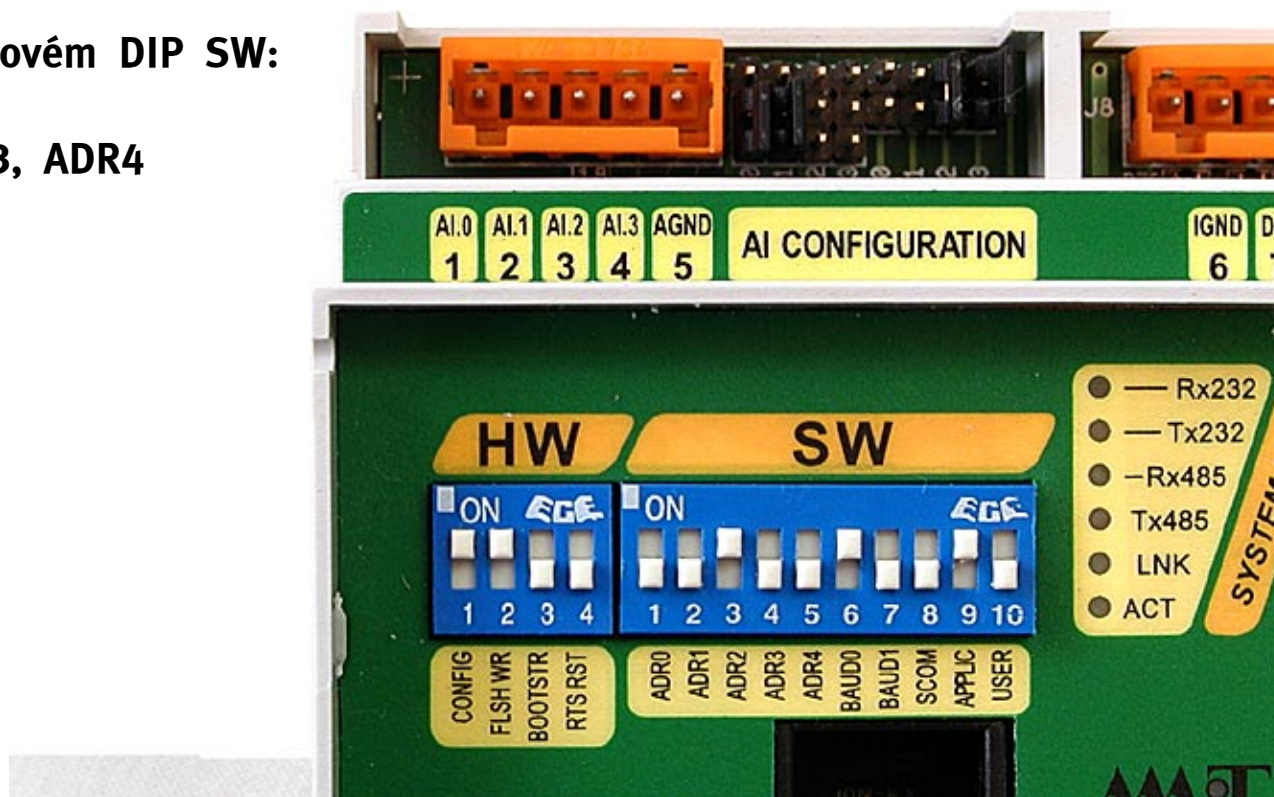
5 – 1 0 1 0 0

6 – 0 1 1 0 0

7 – 1 1 1 0 0

8 – 0 0 0 1 0

9 – 1 0 0 1 0



## Vývojové prostředí DetStudio

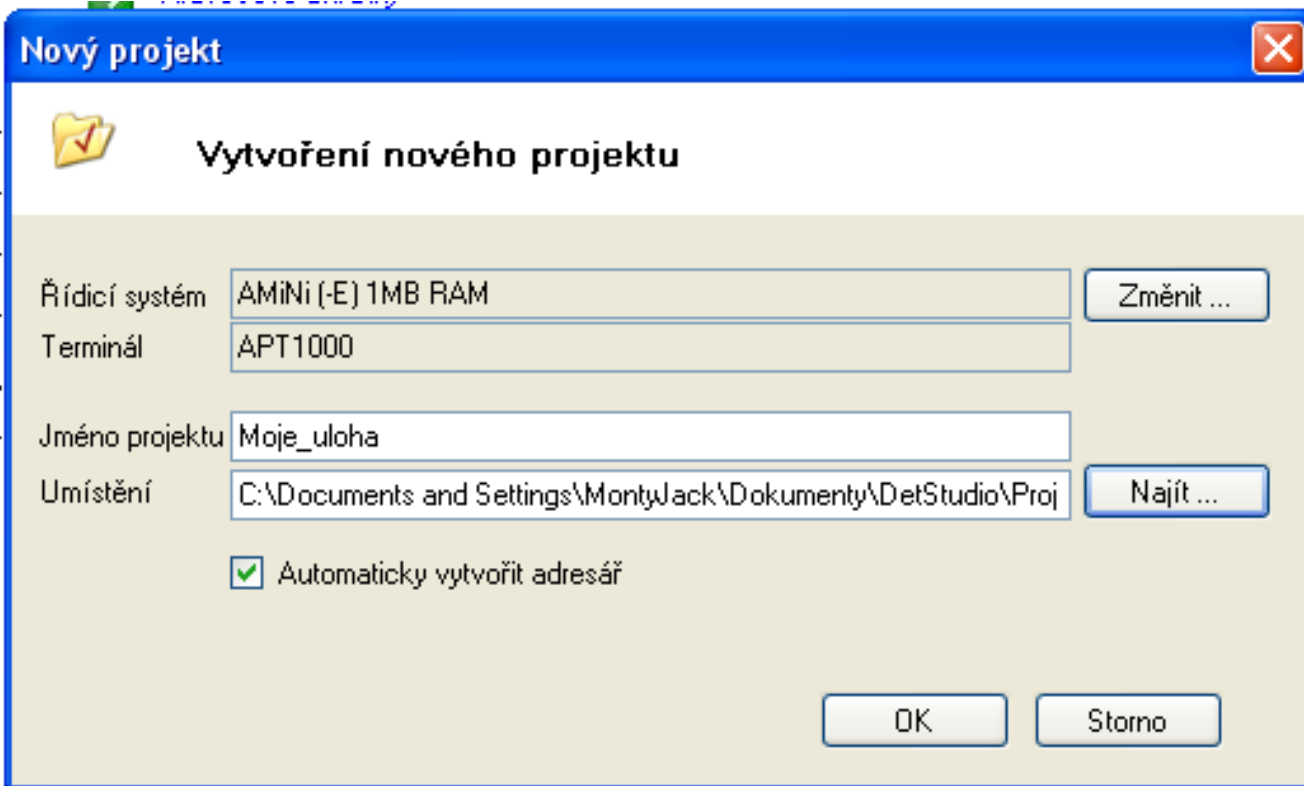
### HW popis PLC AMiNi-E

1) Řídící systém


2) Terminál

3) Jméno projektu

4) Adresa umístění



**Nový projekt**

 **Vytvoření nového projektu**

Řídící systém: AMiNi (-E) 1MB RAM Změnit ...

Terminál: APT1000

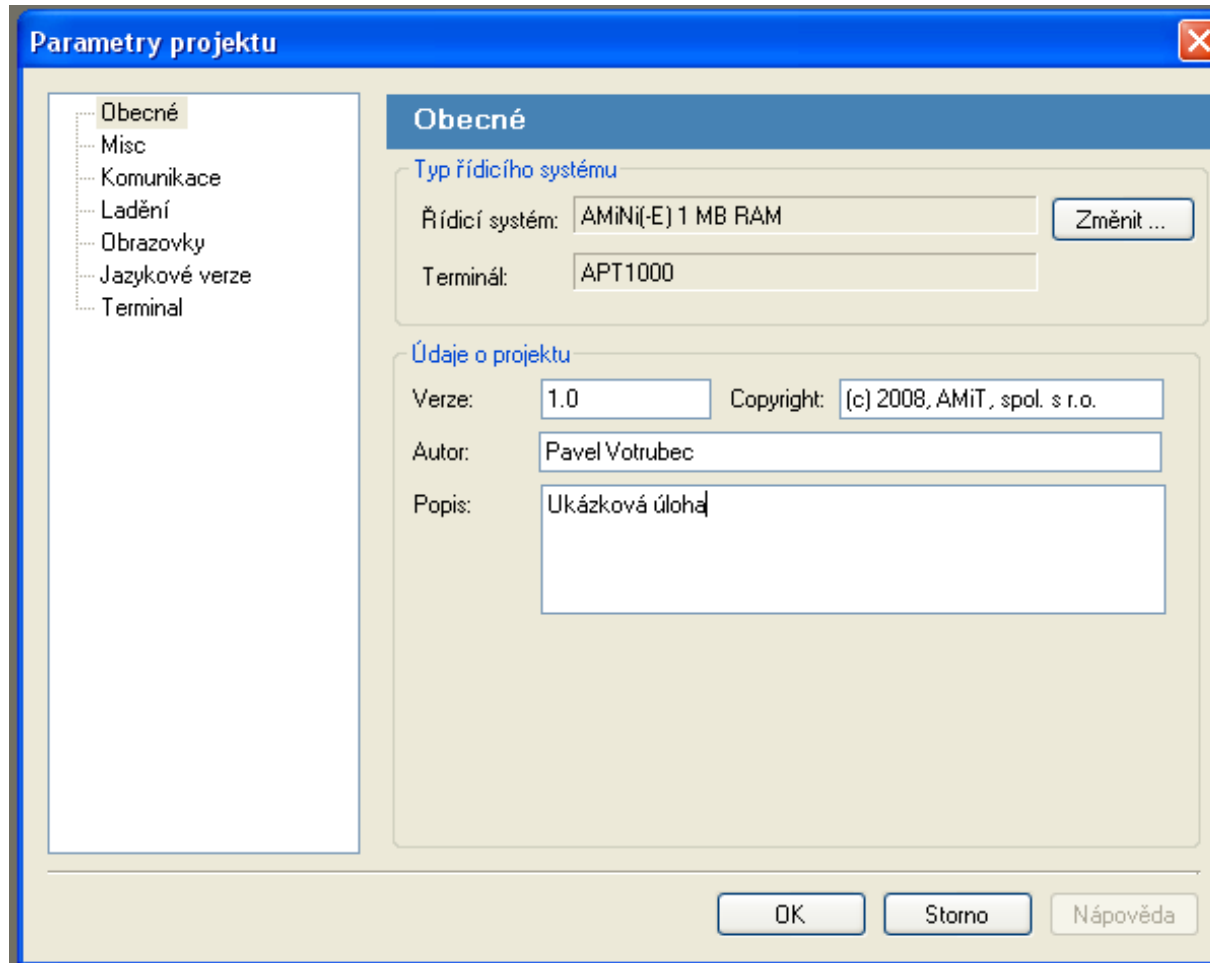
Jméno projektu: Moje\_uloha

Umístění: C:\Documents and Settings\MontyJack\Dokumenty\DetStudio\Proj Najít ...

Automaticky vytvořit adresář

OK Storno

## Parametry projektu - obecné



The screenshot shows a dialog box titled "Parametry projektu" with a close button (X) in the top right corner. On the left is a tree view with the following items: Obecné (selected), Misc, Komunikace, Ladění, Obrazovky, Jazykové verze, and Terminal. The main area is titled "Obecné" and contains two sections:

- Typ řídicího systému**:
  - Řídicí systém: AMiNi(-E) 1 MB RAM (with a "Změnit ..." button)
  - Terminál: APT1000
- Údaje o projektu**:
  - Verze: 1.0 (with a "Copyright: (c) 2008, AMiT, spol. s r.o." label)
  - Autor: Pavel Votrubec
  - Popis: Ukázková úloha

At the bottom of the dialog are three buttons: "OK", "Storno", and "Nápověda".

## Parametry projektu - Misc

Parametry projektu

Obecné  
Misc  
Komunikace  
Ladění  
Obrazovky  
Jazykové verze  
Terminal

Misc

Identifikace v síti

Id1: Ukazkova uloha

Id2: Dipl. Ing. Pavel Votrubec

Protokol

Pasivní stanice  
 Aktivní stanice  
 Mapování vidů

OK Storno Nápověda

## Parametry projektu – Komunikace „použitá stanice 4“

Adresa stanice 4  
SW DIP 00100  
IP 192.168.1.234

**Parametry projektu**

**Komunikace**

Adresa stanice: 4

Způsob komunikace: Ethernet

**IP konfigurace**

Heslo	0
IP Stanice	<b>192.168.1.234</b>
<b>Maska</b>	<b>255.255.255.0</b>
Offset	0
PC heslo	0
PC port	59
Port	59
TimeOut	3000

**Přesměrování**

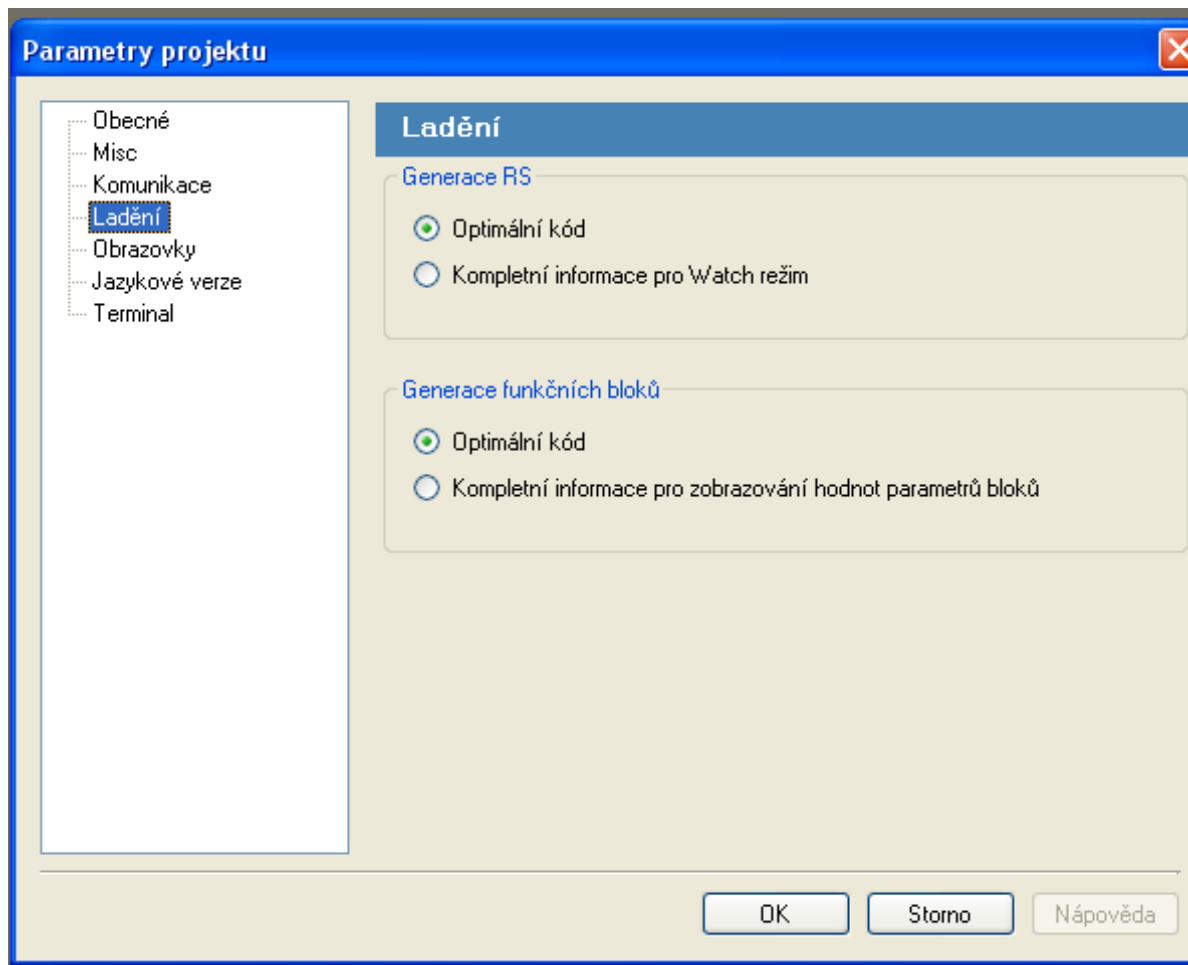
Adresa	<b>0.0.0.0</b>
--------	----------------

**Maska**  
Maska sítě.

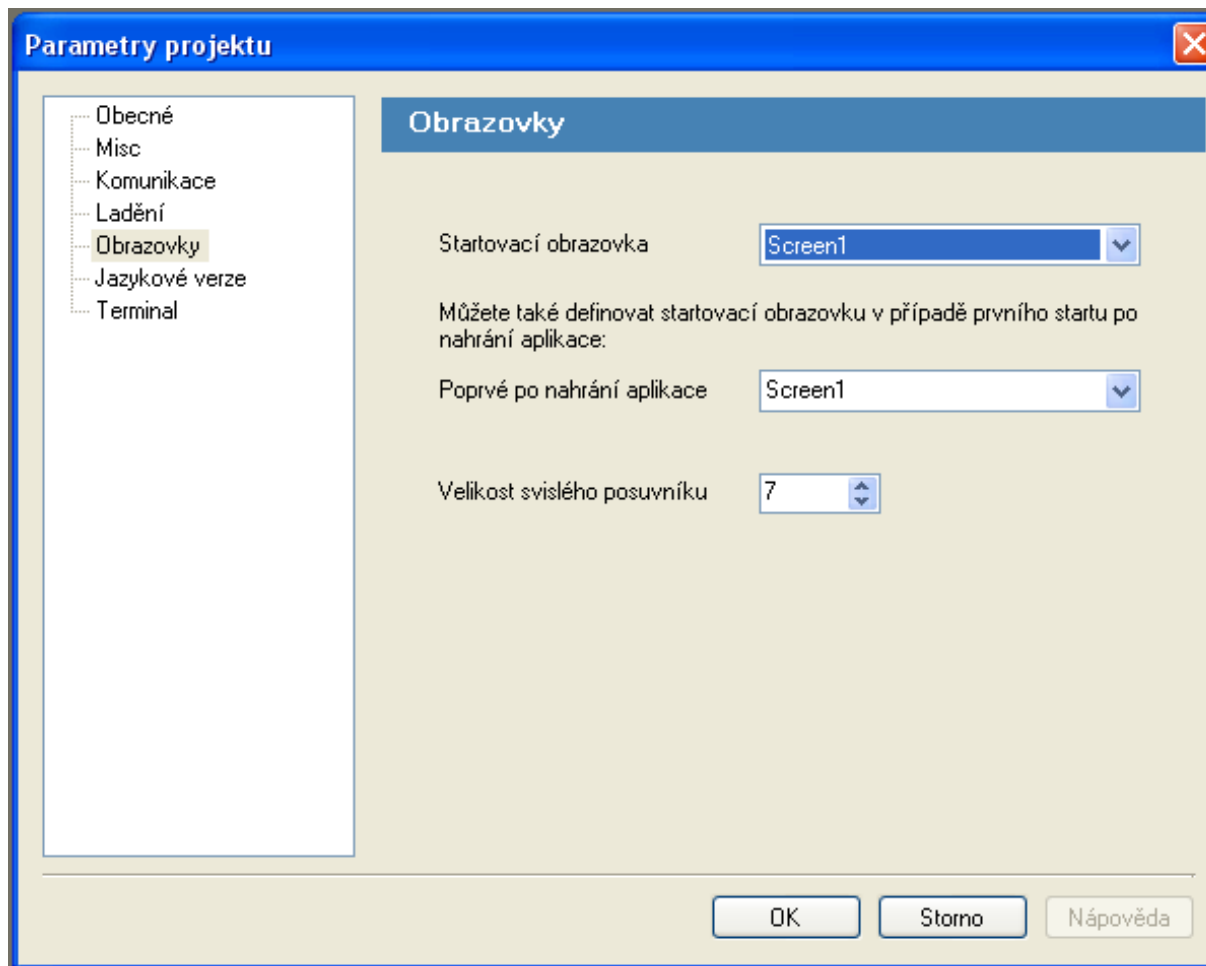
OK Storno Nápověda



## Parametry projektu – Ladění „vše default“



## Parametry projektu - Obrazovky

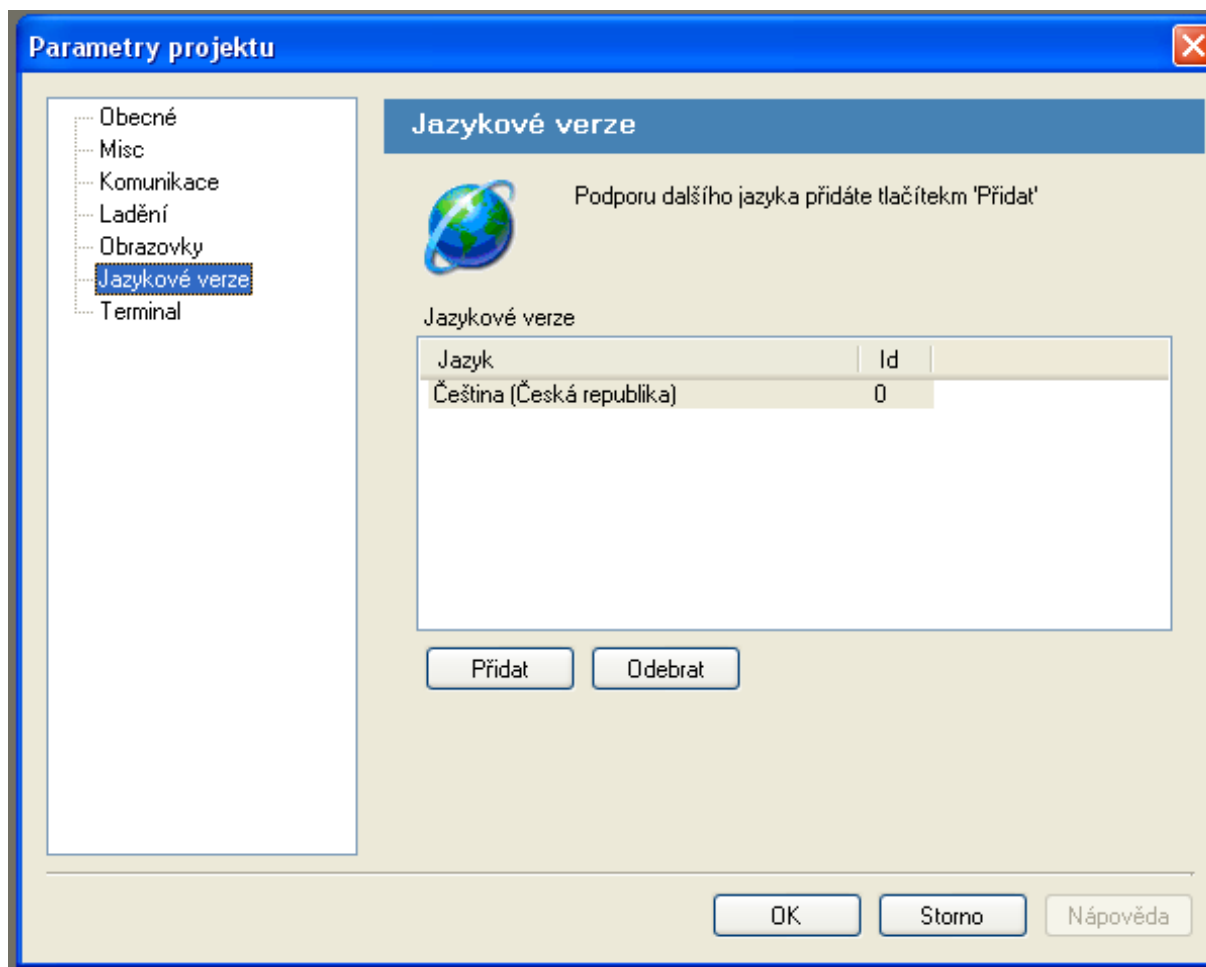


The screenshot shows a dialog box titled "Parametry projektu" with a close button (X) in the top right corner. On the left is a tree view with the following items: Obecné, Misc, Komunikace, Ladění, Obrazovky (highlighted), Jazykové verze, and Terminal. The main area is titled "Obrazovky" and contains three settings:

- Startovací obrazovka: Screen1 (dropdown menu)
- Můžete také definovat startovací obrazovku v případě prvního startu po nahrání aplikace: (text label)
- Poprvé po nahrání aplikace: Screen1 (dropdown menu)
- Velikost svislého posuvníku: 7 (spin box)

At the bottom of the dialog are three buttons: OK, Storno, and nápověda.

## Parametry projektu – Jazyková verze „vše default“




## Parametry projektu - Terminál

**Parametry projektu** ✕

- Obecné
- Misc
- Komunikace
- Ladění
- Obrazovky
- Jazykové verze
- Terminal**

### Terminal

 Nastavení parametrů připojení externího terminálu

Sériový port:

Baudová rychlost:

Typ připojení:

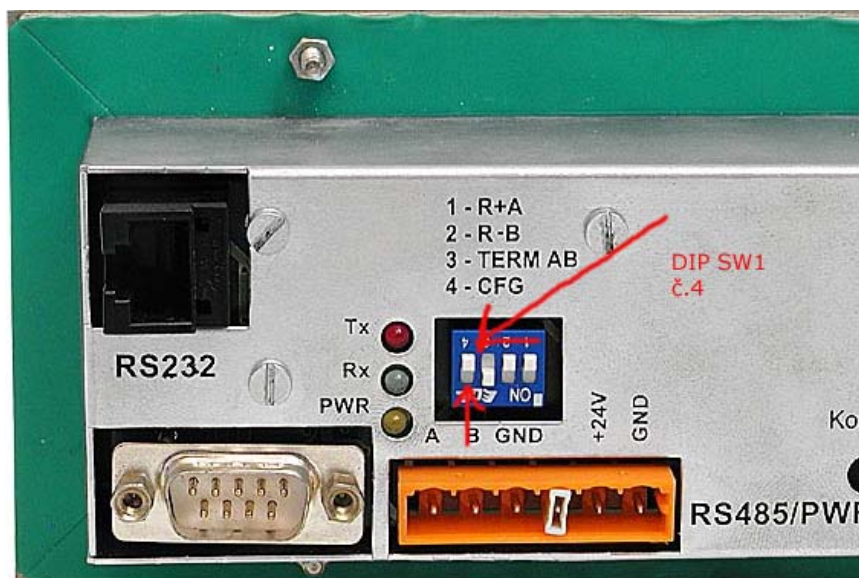
Ostatní parametry komunikace nastavte na 8 bitů, parita=EVEN, stopbit=1

## Terminál APT200

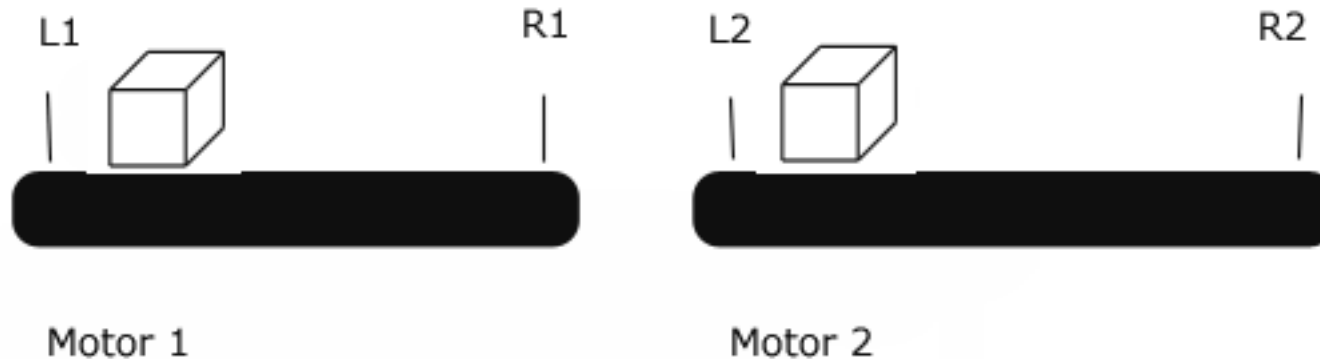


### Nastavení parametrů:

Terminál lze nastavit pomocí ovládání terminálu a současně nastavení DIPu terminálu SW1 č.4 a následného reset terminálu



# Úloha



1. Úloha se spustí stisknutím spínače Tl1
2. Obě kostičky se rozjedou doprava pokud jsou obě na snímači L
3. Dojedou na snímače R a čekají na tu druhou
4. Jakmile dojedou obě na R, tak se rozjedou doleva
5. Dojedou na L a čekají na tu druhou
6. Pokud jsou obě na L a je stisknuto Tl1, úloha se znova rozjede

# Definice potřebných vstupů

IO konfigurace			
Typ	Jméno	Log	Komentář
DIO		0	
DI.00	D100_0		T11 startovací tlačítko
DI.01	D100_1		L1 senzor kostka 1 vlevo
DI.02	D100_2		R1 senzor kostka 1 vpravo
DI.03	D100_3		L2 senzor kostka 2 vlevo
DI.04	D100_4		R2 senzor kostka 2 vpravo
DI.05	D100_5		NC
DI.06	D100_6		NC
DI.07	D100_7		NC
DIO_AC		1	
DI.00	DIO_AC1_0		NC
DI.01	DIO_AC1_1		NC

# Definice potřebných výstupů

Musím zvolit:

Např.:

M1 ON/OFF

SMERM1 – 0 vlevo  
– 1 vpravo

M2 ON/OFF

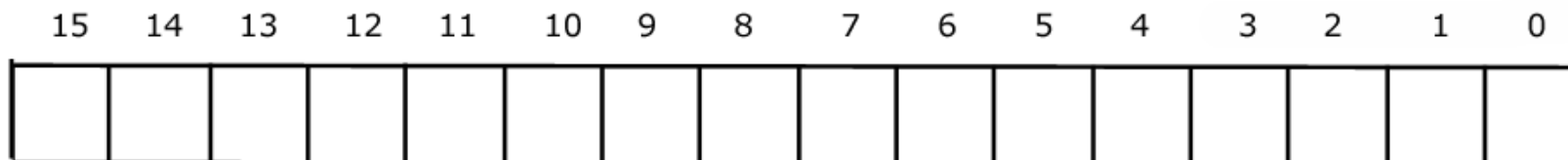
SMERM2 – 0 vlevo  
– 1 vpravo

IO konfigurace			
Typ	Jméno	Log	Komentář
DI.00	DAIO_AC3_0		NC
DI.01	DAIO_AC3_1		NC
DI.02	DAIO_AC3_2		NC
DI.03	DAIO_AC3_3		NC
DO		0	
DO.00	DO0_0		M1
DO.01	DO0_1		SMERM1
DO.02	DO0_2		M2
DO.03	DO0_3		SMERM2
DO.04	DO0_4		NC
DO.05	DO0_5		NC
DO.06	DO0_6		NC
DO.07	DO0_7		NC
LED		1	
DO.00	LED1_0		NC

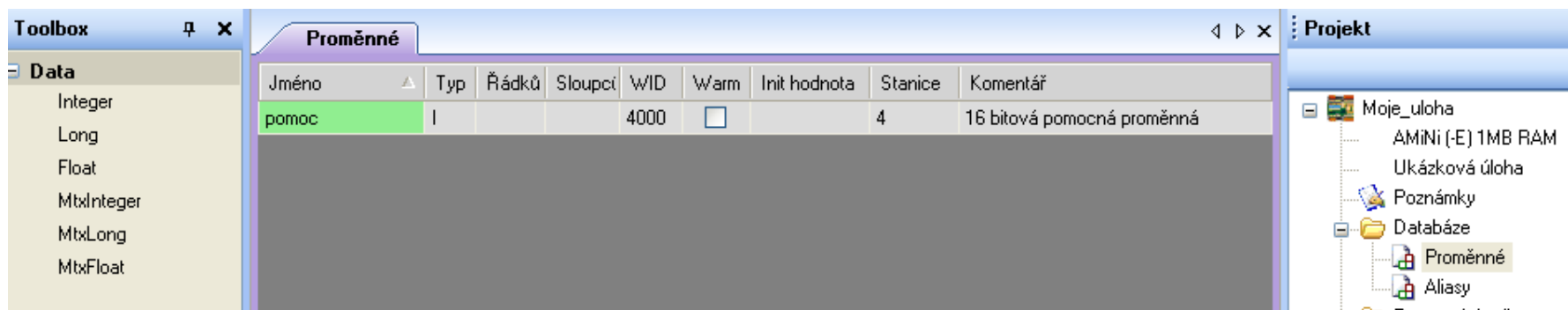


## Definice potřebných proměnných

Na bitové operace nám stačí jedna proměnná typu I. Poté máme k dispozici 16 bitů (0-15) adresovatelné pomocí nástroje alias.

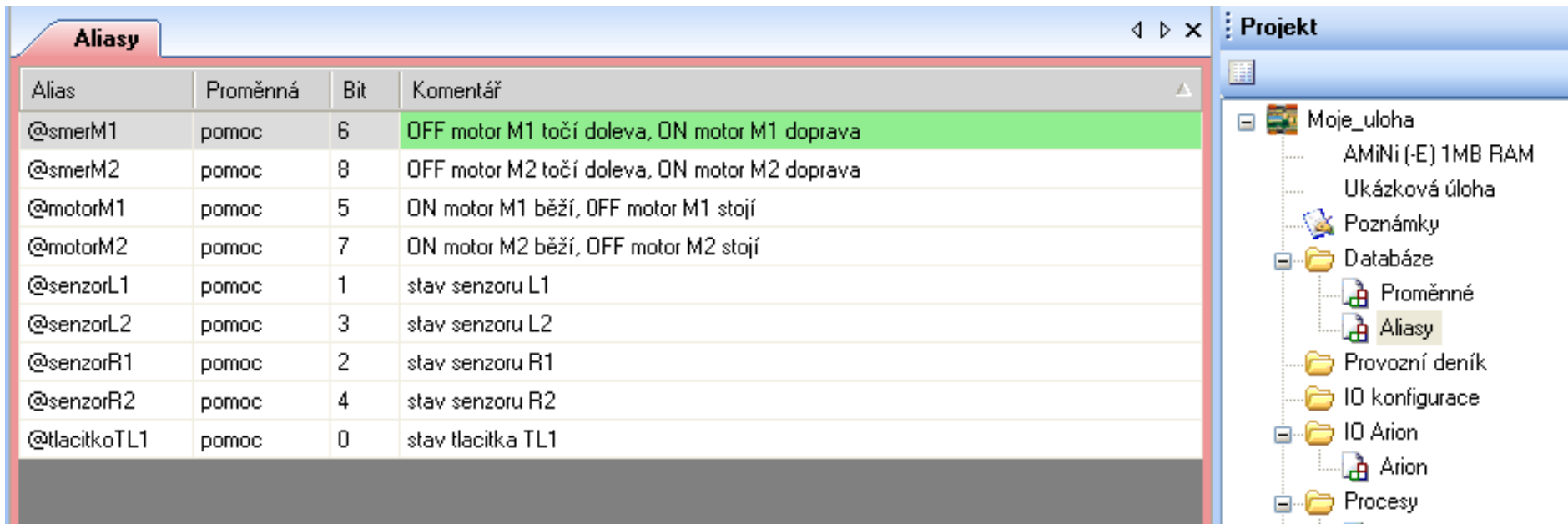


Pomoc



## Definice potřebných aliasů

Abychom mohli zpracovat vnější informace tak je nejprve musíme uložit do paměti. Pro bitové vstupy DI a bitové výstupy DO si tudíž nadefinujeme potřebné aliasy – paměťová místa pro bity.



The screenshot shows a software interface with a table of aliases and a project tree on the right.

Alias	Proměnná	Bit	Komentář
@smerM1	pomoc	6	OFF motor M1 točí doleva, ON motor M1 doprava
@smerM2	pomoc	8	OFF motor M2 točí doleva, ON motor M2 doprava
@motorM1	pomoc	5	ON motor M1 běží, OFF motor M1 stojí
@motorM2	pomoc	7	ON motor M2 běží, OFF motor M2 stojí
@senzorL1	pomoc	1	stav senzoru L1
@senzorL2	pomoc	3	stav senzoru L2
@senzorR1	pomoc	2	stav senzoru R1
@senzorR2	pomoc	4	stav senzoru R2
@tlacitkoTL1	pomoc	0	stav tlacitka TL1

The project tree on the right shows the following structure:

- Moje\_uloha
  - AMiNi (-E) 1MB RAM
  - Ukázková úloha
  - Poznámky
  - Databáze
    - Proměnné
    - Alias
  - Provozní deník
  - IO konfigurace
  - IO Arion
    - Arion
  - Procesy

# A můžeme psát vlastní algoritmy procesů...

Máme k dispozici:

## 1. Procesy spouštěné časově – periodicky

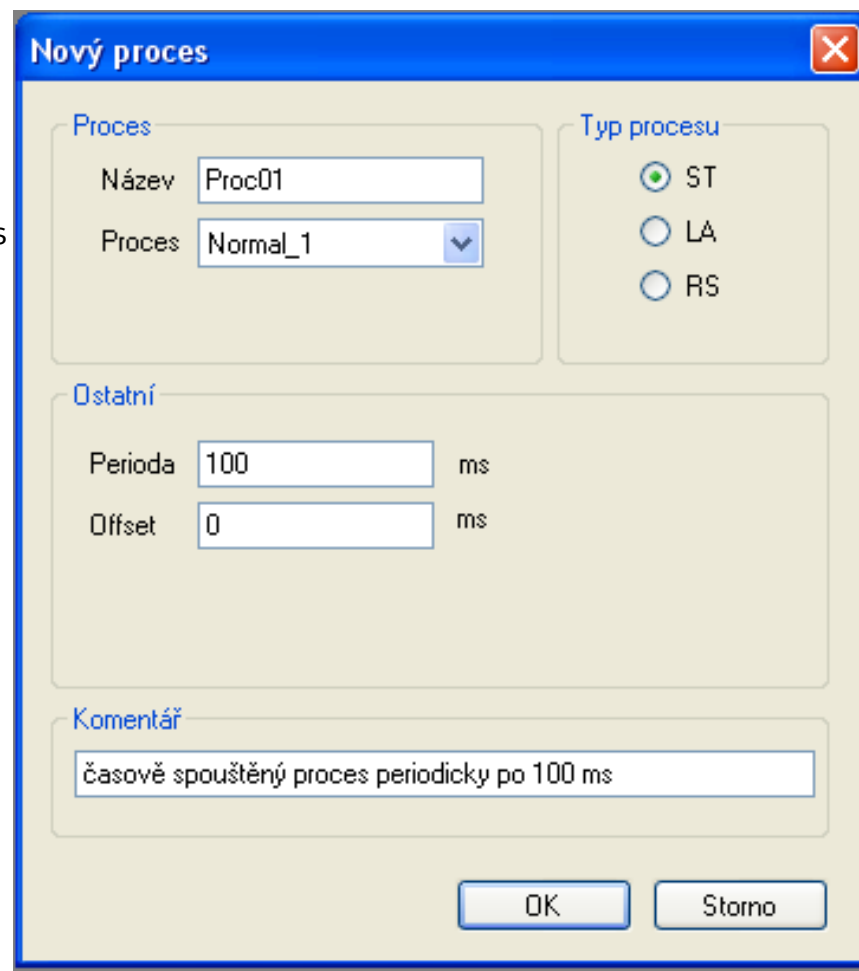
1. Normal\_0 až Normal\_16 perioda od 100 ms do 1000000 s
2. Quick perioda 5, 10, 20, 50, 100 ms
3. Hi\_0 a Hi\_1 s periodou od 1 ms do 1677 ms

## 2. Proces spuštěn jednou při startu ŘS

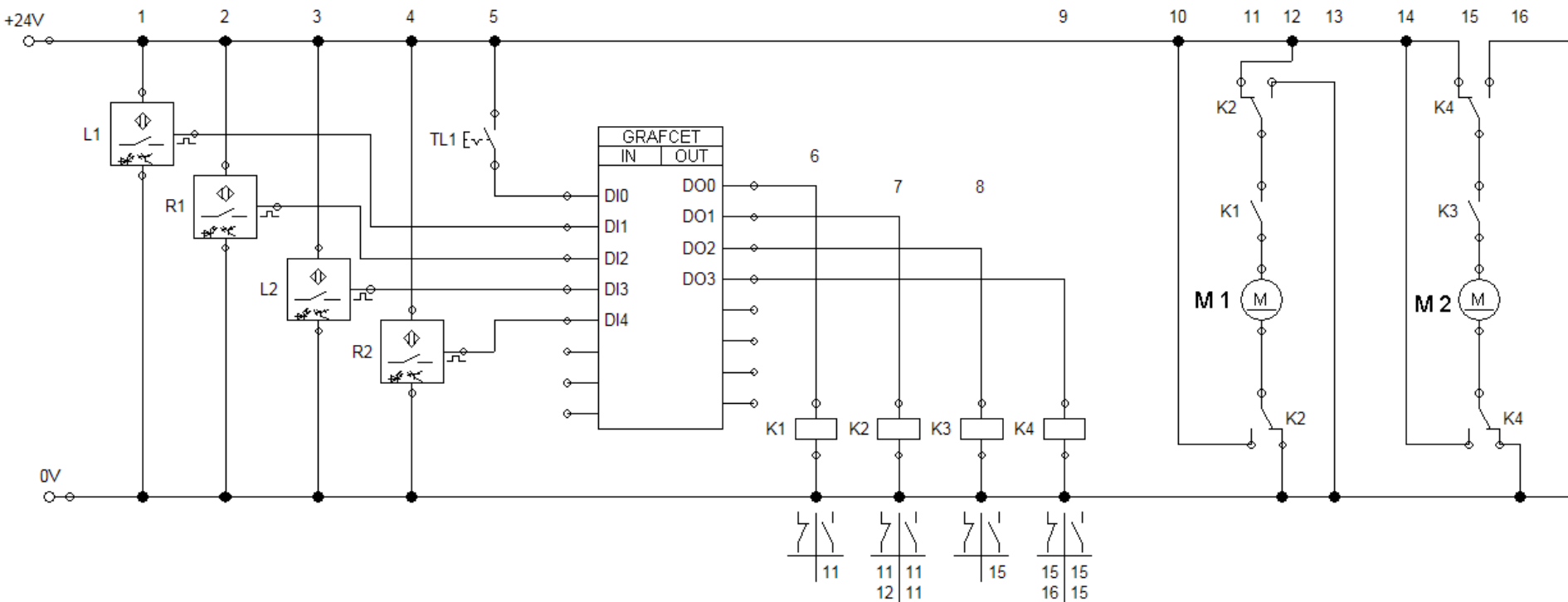
1. Init

## 3. Procesy spouštěné při aktivitě na vstupu

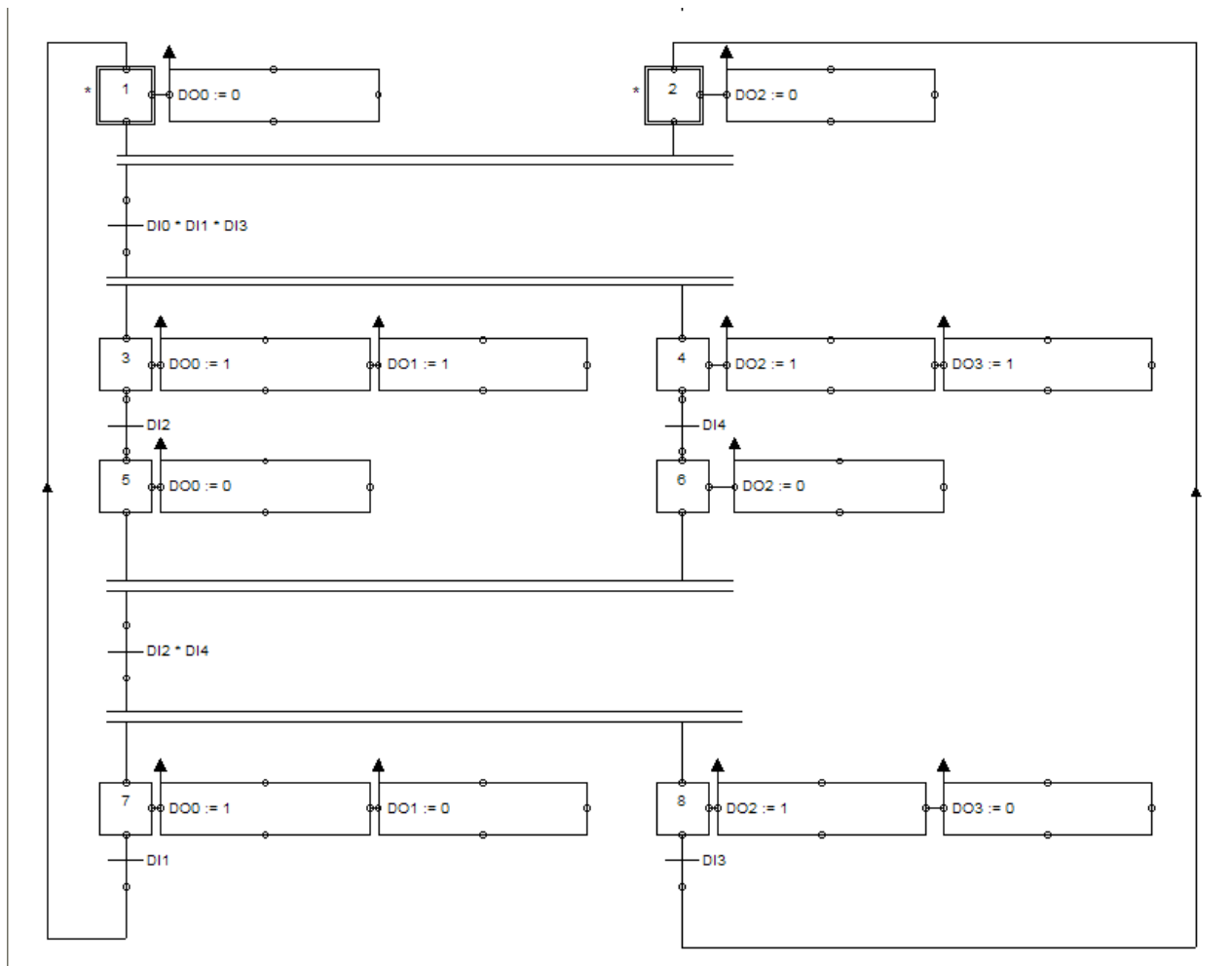
1. ITR0 vstupem DI0.0
2. ITR1 vstupem DI0.1
3. ITR2 vstupem DI0.2
4. ITR3 vstupem DI0.3
5. ITR4 vstupem DI0.4
6. ITR5 vstupem DI0.5
7. ITR6 vstupem DI0.6
8. ITR7 vstupem DI0.7 a to buď náběžnou, nebo doběžnou hranou a nebo oběma hranama vstupního signálu.



# Schéma zapojení ss motorů, jejich reversace, tlačítka a senzorů



# Algoritmus ŘS zapsaný pomocí GRAFCET



## Ošetření vstupních signálů pomocí procesů ITR

Aktuální ITR proces se spustí na jednu nebo druhou hranu

Stavy DI se nám ukládají při změně do paměti (do příslušných aliasů)

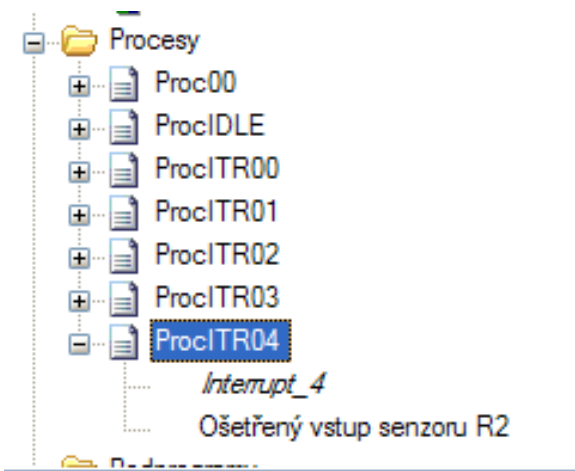
```

Procesy ProcITR00
BinIn #DIO0_0, 0x0000, @tlacitkoTL1
    
```

**Vlastnosti**

Objekt: Proc: ProcITR04

Comment	Ošetřený vstup senzoru R2
InterruptEdge	Both
Name	ProcITR04
Type	Interrupt_4



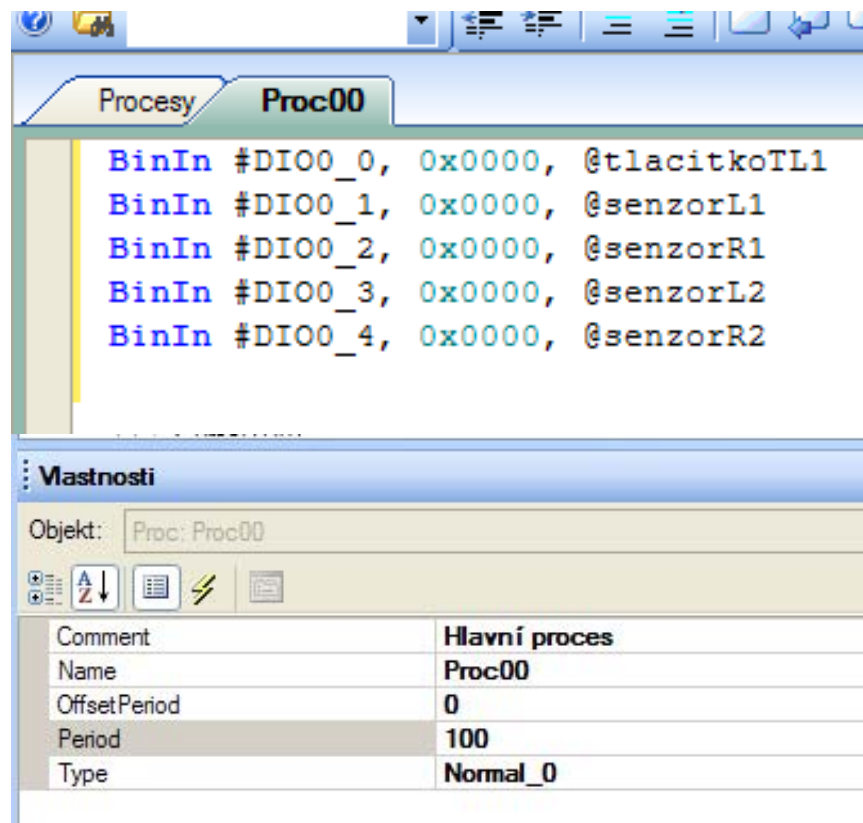
Název	Jazyk	Typ	Perioda	Offset	Komentář
Proc00	ST	Normal_0	1000	0	Hlavní proces
ProcIDLE	ST	Idle	-	-	Obsluha obrazovek
ProcITR00	ST	Interrupt_0	-	-	Ošetřený vstup tlačítka TL1
ProcITR01	ST	Interrupt_1	-	-	Ošetřený vstup senzoru L1
ProcITR02	ST	Interrupt_2	-	-	Ošetřený vstup senzoru R1
ProcITR03	ST	Interrupt_3	-	-	Ošetřený vstup senzoru L2
ProcITR04	ST	Interrupt_4	-	-	Ošetřený vstup senzoru R2

## Stavy na vstupech - alternativní řešení

Stavy na vstupech můžete snímat periodicky v časově spouštěném procesu.

V tomto případě jsme navolili periodu procesu 100 ms.

Tudíž stavy na senzorech se čtou každých 100 ms a poté ukládají do paměti (příslušných aliasů).



The screenshot shows a software interface with a 'Procesy' (Processes) tab. The selected process is 'Proc00'. The main area displays a list of binary inputs (BinIn) with their addresses and aliases:

- BinIn #DIO0\_0, 0x0000, @tlacitkoTL1
- BinIn #DIO0\_1, 0x0000, @senzorL1
- BinIn #DIO0\_2, 0x0000, @senzorR1
- BinIn #DIO0\_3, 0x0000, @senzorL2
- BinIn #DIO0\_4, 0x0000, @senzorR2

Below the list, the 'Vlastnosti' (Properties) window is open, showing the following details for 'Proc: Proc00':

Comment	Hlavní proces
Name	Proc00
OffsetPeriod	0
Period	100
Type	Normal_0

## Proces pro vozík č.1 přepis podle Grafcet algoritmu

**Switch** pomoc2

**Case** 1

**Let** @motorM1 = False

**Let** @logika = @tlacitkoTL1 **AND** @senzorL1 **AND** @senzorL2

**If** @logika

**Let** pomoc2 = 3

**Let** @motorM1 = True

**Let** @smerM1 = True

**Else**

**EndIf**

**EndCase**

**Case** 3

**If** @senzorR1

**Let** pomoc2 = 5

**Let** @motorM1 = False

**Else**

**EndIf**

**EndCase**

**Case** 5

**Let** @logika = @senzorR1 **AND** @senzorR2

**If** @logika

**Let** pomoc2 = 7

**Let** @motorM1 = True

**Let** @smerM1 = False

**Else**

**EndIf**

**EndCase**

**Case** 7

**If** @senzorL1

**Let** pomoc2 = 1

**Let** @motorM1 = False

**Else**

**EndIf**

**EndCase**

**EndSwitch**

**BinOut** @motorM1, 0x0000, #DO0\_0

**BinOut** @smerM1, 0x0000, #DO0\_1



## Proces pro vozík č.2 přepis podle Grafcet algoritmu

**Switch** pomoc3

**Case** 2

**Let** @motorM2 = False

**Let** @logika = @tlacitkoTL1 **AND** @senzorL1 **AND** @senzorL2

**If** @logika

**Let** pomoc3 = 4

**Let** @motorM2 = True

**Let** @smerM2 = True

**Else**

**EndIf**

**EndCase**

**Case** 4

**If** @senzorR2

**Let** pomoc3 = 6

**Let** @motorM2 = False

**Else**

**EndIf**

**EndCase**

**Case** 6

**Let** @logika = @senzorR1 **AND** @senzorR2

**If** @logika

**Let** pomoc2 = 8

**Let** @motorM2 = True

**Let** @smerM2 = False

**Else**

**EndIf**

**EndCase**

**Case** 8

**If** @senzorL2

**Let** pomoc3 = 2

**Let** @motorM2 = False

**Else**

**EndIf**

**EndCase**

**EndSwitch**

**BinOut** @motorM2, 0x0000, #D00\_2

**BinOut** @smerM2, 0x0000, #D00\_3

**A teď už jen správně zapojit HW, naloudovat SW a spustit úlohu**