

Semestrální práce z předmětu **KIV/TI**

Logické řízení - ohřívací nádrž

student: Jan Hinterholzinger
studijní číslo: A19B0052P
email: hintik@students.zcu.cz
datum: 19. 05. 2021

Obsah

1	Zadání	2
2	Analýza úlohy	3
2.1	Popis chování zařízení	3
2.2	Analýza	3
3	Automatový model	4
3.1	Popis vstupních signálů	4
3.2	Popis výstupních signálů	4
3.3	Seznam stavů	5
3.4	Přechodový graf automatu	5
4	Implementace	6
4.1	Algoritmický diagram programu	7
5	Uživatelská příručka	8
5.1	Spuštění programu	8
5.2	Ovládání programu	8
6	Závěr	8

1 Zadání

HINTERHOLZINGER

7. Logické řízení – ohřívací nádrž:

Na <http://home.zcu.cz/~vais/> v rozšiřujícím materiálu o konečných automatech prostudujte kapitoly Logické řízení a Principy softwarové implementace.

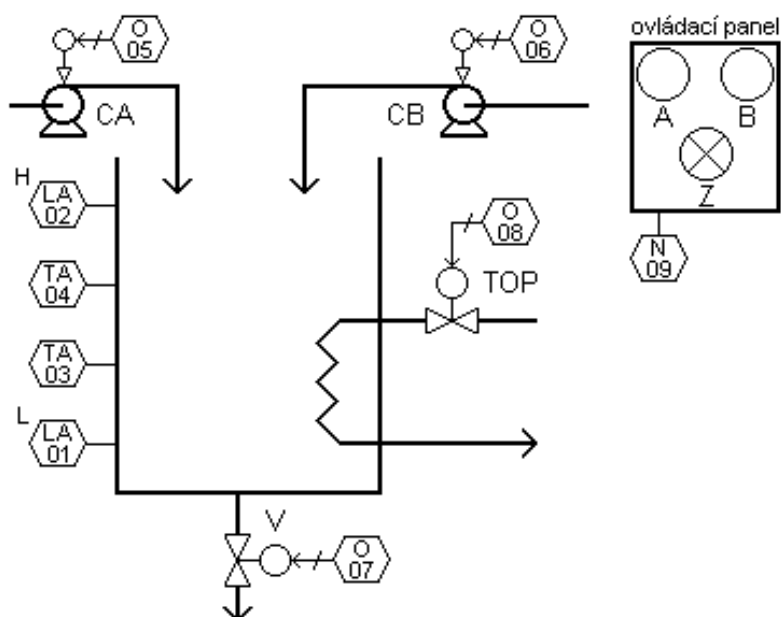
Navrhněte konečněautomatový model řídicího systému níže popsaného zařízení.

V nádrži, do níž se přivádí čerpadlem CA látka A, nebo čerpadlem CB látka B se tyto látky ohřívají na různou teplotu. Látka A se má vždy ohřát na teplotu, jejíž dosažení je signalizováno signálem TA z čidla TA/03 a látka B na jinou teplotu signalizovanou obdobně signálem TB z čidla TA/04. Po ohřátí na příslušnou teplotu se má látka vypustit. Operátor odstartuje technologický cyklus (napuštění, ohřev na příslušnou teplotu a vypuštění), stisknutím tlačítka A nebo B podle toho, pro kterou z látek bude cyklus proveden. Ohřev se uskuteční vždy až po naplnění celé nádrže. Ponoření čidla hladiny LA/01 u dna nádrže a čidla LA/02 u vrcholu nádrže je signalizováno signály L11 a L21, vymoření L10 a L20. Topení se zapíná (respektive vypíná) výstupním signálem TOP1 (respektive TOP0). Výstupní ventil je ovládán signály V1 (otevřít) respektive (V0 zavřít). Čerpadla se spouští (vypínají) příslušnými signály CA1, CA0 (pro čerpadlo A) nebo CB1, CB0 (pro čerpadlo B). Stisknutí tlačítek A, B na ovládacím panelu reprezentují signály A, B.

Předpokládejte, že na počátku je nádrž prázdná, čerpadla a topení jsou vypnuty a výstupní ventil uzavřen.

Model řídicího automatu realizujte softwarově na základě principů popsaných v materiálu. Všechny signály od čidel modelujte vstupy od klávesnice, řídicí signály a informaci o stavu vypisujte textově na obrazovku.

Technologické schéma nádrže:



2 Analýza úlohy

Zadání úlohy představuje úlohu pro logické řízení na základě vstupních (signálních) a výstupních (řídících) signálů. Chod zařízení lze rozdělit do několika stavů, mezi kterými bude přepínat na základě vstupních signálů. Tyto vstupní signály pocházejí dle zadání z různých čidel (např. měření správné teploty, hladiny látky nebo stisknutí tlačítka). Naopak řídící signály se spouští při změně stavu a ovládají tím různé mechanismy zařízení. Na základě těchto poznatků, bude navrhnout konečně automatový model systému.

2.1 Popis chování zařízení

Jedná se o zařízení ohřívací nádrže na dvě látky. Každou látku je potřeba ohřát na jinou teplotu. Zařízení dle stisknutí tlačítka napustí do nádrže příslušnou látku. Po naplnění nádrže se čerpání vypne a zapíná se topení. Poté co čidlo detekuje správnou teplotu pro napuštěnou látku, vyšle signál a zařízení vypne topení a začne vypouštět látku výstupním ventilem. Po vyprázdnění nádrže je výstupní ventil uzavřen a čeká se na další vstup z tlačítek.

2.2 Analýza

Celý proces lze tedy rozdělit na 3 části: cyklus pro látku A, cyklus pro látku B a vypuštění. Cykly pro látku A a B musí být oddělené, protože na základě této informace se odlišují signály pro otevírání ventilů pro napouštění a signály signalující správnou teplotu pro danou látku. Součástí cyklů látek je jejich napuštění a ohřátí na jejich příslušné teploty.

Automat přechází mezi stavy podle vstupních signálů, které inicializují změnu stavu a vyslání výstupních řídících signálů. Vstupní signály dle chování zařízení přicházejí v určitém logickém pořadí. Na základě toho jsme schopni sestavit přechodový graf (diagram) automatu. Jelikož přechod mezi stavy zahájí pouze některé signály, bylo by nepřehledné zakreslovat veškeré možné hrany, které jsou způsobeny vstupem jiného signálu. Proto výchozí značení diagramu je takové, že pokud z uzlu nevystupuje hrana s aktuálním vstupním signálem, automat setrvává ve svém stavu a nekazuje žádné výstupní signály.

3 Automatový model

3.1 Popis vstupních signálů

Tabulka 1: Tabulka vstupních signálů

Značka signálu	Popis signálu
TA1	teplota látky A stoupla nad určenou teplotu
TA0 (nepoužito)	teplota látky A klesla pod určenou teplotu
TB1	teplota látky B stoupla nad určenou teplotu
TB0 (nepoužito)	teplota látky B klesla pod určenou teplotu
L11 (nepoužito)	hladina látky stoupla nad spodní snímač
L21	hladina látky stoupla nad horní snímač
L10	hladina látky klesla pod spodní snímač
L20 (nepoužito)	hladina látky klesla pod horní snímač
A	tlačítko A stisknuto
B	tlačítko B stisknuto

3.2 Popis výstupních signálů

Tabulka 2: Tabulka výstupních řídicích signálů

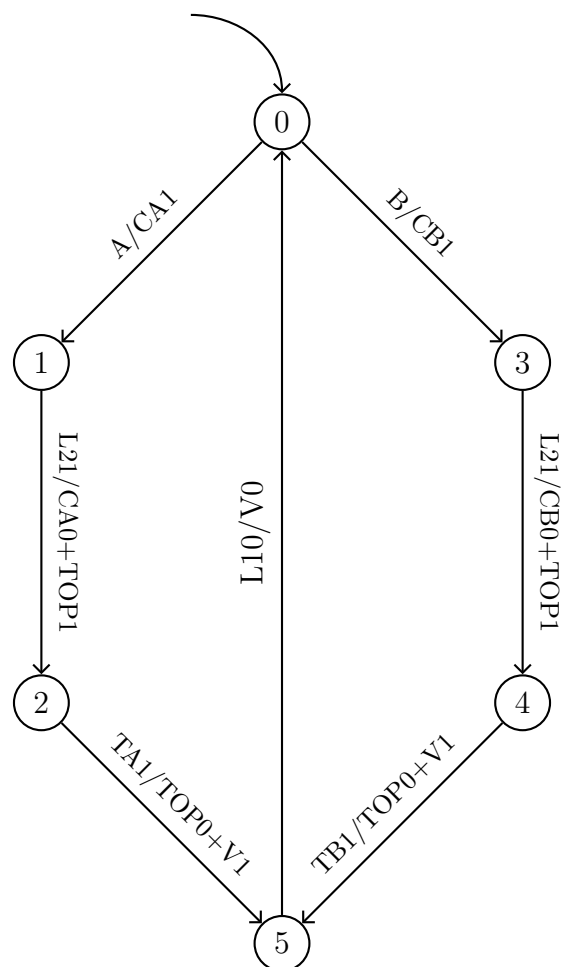
Značka signálu	Popis signálu
CA1	zapnout čerpadlo látky A
CA0	vypnout čerpadlo látky A
CB1	zapnout čerpadlo látky B
CB0	vypnout čerpadlo látky B
TOP1	zapnout topení
TOP0	vypnout topení
V1	otevřít výstupní ventil
V0	uzavřít výstupní ventil

3.3 Seznam stavů

Tabulka 3: Tabulka stavů automatu

Číslo stavu	Zkratka	Popis signálu
0	NOP	zařízení není v provozu (čeká se na stisknutí tlačítka)
1	TAF	cyklus A - napuštění (čeká se dokud nebude nádrž napuštěna)
2	TAH	cyklus A - ohřev (čeká se dokud látka nedosáhne potřebné teploty)
3	TBF	cyklus B - napuštění (čeká se dokud nebude nádrž napuštěna)
4	TBH	cyklus B - ohřev (čeká se dokud látka nedosáhne potřebné teploty)
5	DRA	vypuštění (čeká se dokud nebude nádrž prázdná)

3.4 Přechodový graf automatu



Obrázek 1: Diagram řídicího systému

V diagramu systému na obrázku 2 nosí jednotlivé hrany popisek udávající vstupní signál (signál před lomítkem), který vyvolal daný přechod a seznam výstupních řídicích signálů (signál za lomítkem, v případě více signálů se signály řetěží znakem +), které se při přechodu provednou.

4 Implementace

Implementace byla provedena v objektovém jazyce Java podle příkladu v ucelém materiálu o softwarové implementaci. Vstupní signály jsou nahrazeny vstupem z klávesnice od uživatele. Výstupní signály a informace o změně stavu jsou reprezentovány výpisy na obrazovku uživatele.

Pro stavy, vstupy a výstupy jsou pro jejich omezené množství vytvořeny výčtové typy **Stav**, **Vstup** a **Vystup**, které obsahují všechny použité signály a stavy. Výstupní signály, které jsou v diagramu (obrázek 2) zřetězeny (spojeny znakem +), jsou ve výčtovém typu uvedeny jako jedna společná hodnota. Například mezi stavy 2 a 5 se vyšlou řídicí signály **TOP0** pro vypnutí topení a **V1** pro otevření výstupního ventilu, ve výčtovém typu se jedná o jednu společnou hodnotu **TOP0V1**.

Vstup řídicích signálů je reprezentováno vstupem od uživatele. O tento vstup se stará metoda `void vstup()`. Uživatel zadává znaky, které reprezentují příslušný vstupní signál. Metoda vstupní znak převede na hodnotu výčtového typu, kterou uloží do globální proměnné `vstup`. Obdobně je řešena metoda `void vystup(Vystup)`, která vypíše hlášku přiřazenou k danému signálu.

Pro přechod mezi stavy je vytvořeno dvourozměrné pole `prech_tab` typu **Stav**, které reprezentuje přechodovou tabulku. První index v tomto poli je index aktuálního stavu a druhý index je index vstupního signálu. Na zadané pozici v tomto poli se nachází stav (typu **Stav**), do kterého systém přejde. Na základě diagramu přechodového grafu (Obrázek 2) byla vytvořena následující podoba této přechodové tabulky. Do tohoto stavu pole inicializuje metoda `inic_tab()`.

$$\begin{pmatrix} NOP & NOP & NOP & NOP & TAF & TBF \\ TAF & TAF & TAH & TAF & TAF & TAF \\ DRA & TAH & TAH & TAH & TAH & TAH \\ TBF & TBF & TBH & TBF & TBF & TBF \\ TBH & DRA & TBH & TBH & TBH & TBH \\ DRA & DRA & DRA & NOP & DRA & DRA \end{pmatrix}$$

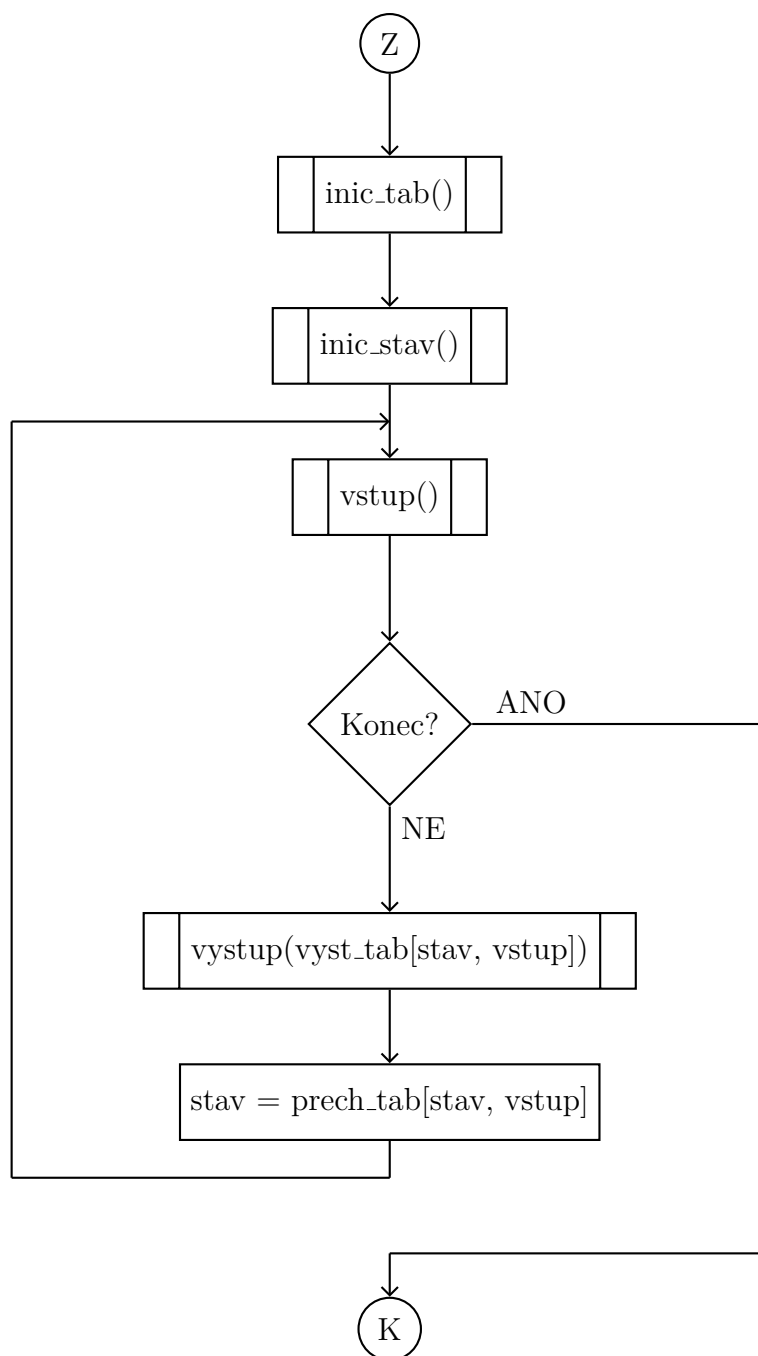
Obdobně je vytvořeno další dvourozměrné pole `vyst_tab` typu **Vystup**. První index pole je číselné označení aktuálního stavu. A druhý označuje index vstupního signálu. Na těchto indexech se nachází hodnota výčtového typu **Vystup**, která označuje konkrétní výstupní akci. Na základě diagramu přechodového grafu (Obrázek 2) byla vytvořena následující podoba této výstupní tabulky. Do tohoto stavu pole inicializuje metoda `inic_tab()`. Symbol `null` označuje, že žádný výstupní signál není proveden.

$$\begin{pmatrix} null & null & null & null & CA1 & CB1 \\ null & null & CA0TOP1 & null & null & null \\ TOP0V1 & null & null & null & null & null \\ null & null & CB0TOP1 & null & null & null \\ null & TOP0V1 & null & null & null & null \\ null & null & null & V0 & null & null \end{pmatrix}$$

poslední nezmněná metoda programu je `inic_stav`. Tato metoda přiřazuje hodnotě `stav` výchozí hodnotu. Podle zadání má program začít ve stavu 0 (systém neoperuje), proto je v této metodě proměnná `stav` nastavena na `NOP` (not operating).

4.1 Algoritmický diagram programu

Následující obrázek zobrazuje strukturu hlavní metody `run()`, která prochází programem po celou dobu běhu programu.



Obrázek 2: Diagram hlavního běhu programu

Program zahajuje inicializací přechodové tabulky, tabulky výstupů a stavové proměnné. Následně čte příkaz (simulovaný vstup) z klávesnice, který převede na hodnotu výčtového typu `vstup`. Program vstupuje do cyklu, ve kterém setrvává dokud příkaz nebude ukončovací. Program vybírá z tabulky výstupů (podle aktuálního stavu a vstupního signálu) příslušný výstup a pokud existuje, tak jej vyšle. Následně je aktualizována stavová proměnná hodnotou v přechodové tabulce na pozici aktuálního stavu a vstupního signálu. Poté program opět načítá příkaz z klávesnice a převádí jej a tím je ukončen jeden průběh cyklu. Po ukončení běhu cyklu, tak se ukončí program.

5 Uživatelská příručka

5.1 Spuštění programu

Pro spuštění programu je potřeba, aby na daném počítači bylo nainstalováno spouštěcí prostředí pro programy napsané v jazyce Java. Pokud toto prostředí není nainstalováno, je možné jej stáhnout a doinstalovat například na této adrese: <https://www.java.com/en/download/> (Java 8 JRE)

K této dokumentaci je přiložen soubor `TI-HINTER.jar`, který obsahuje vypracovaný program. Program je exportovaný jako spustitelné JAR, lze jej tedy spustit z příkazové řádky za pomoci následujícího příkazu, kde `cesta k JAR souboru` nahradíme cestou k umístění souboru `TI-HINTER.jar`.

```
java -jar <cesta k JAR souboru>
```

Pokud se v příkazové řádce nacházíme ve stejném adresáři jako JAR soubor s programem, můžeme poté zavolat následující příkaz, který spustí program.

```
java -jar TI-HINTER.jar
```

Po spuštění programu by měla být vypsána hláška „Simulujte vstup: “ a měl by být očekáván vstup z klávesnice.

Tímto je spuštění programu dokončeno a je možné zadávat jednotlivé simulované vstupy.

5.2 Ovládání programu

Program je ovládán příkazy, které simulují vstupní signály do systému, nebo signalizují ukončení aplikace. Příkazy simulující vstupní signály jsou shodné se značkami použitých vstupních signálů (viz tabulka 1). Porovnání příkazů je použito „case-insensitive“ (nerozlišují se malá a velká písmena). To znamená, že například příkaz pro simulaci stisknutí tlačítka A může být „A“ ale také „a“.

Běh programu lze kdykoli ukončit vložení příkazu `quit`.

Program po každém zadání příkazu vypíše popis vstupního příkazu, aby byla zajištěna přehlednost toho, co uživatel zaslal. V případě simulace vstupního signálu se navíc vypíše popisky výstupních signálů a případně i informaci o změně stavu.

6 Závěr

Program byl vypracován podle vyžádaných parametrů po vzoru uceleného materiálu. Je tedy použita struktura z tohoto materiálu. Osobně se mi příliš nezamlouvá implementační spojení více výstupních akcí do jedné, proto bych říste pravděpodobně použil formu

nějakého seznamu výstupních signálů, aby bylo jednodušší při větším projektu operovat s výstupními signály. Každopádně program se zdá být funkční a správně simuluje funkci zařízení.

Případné rozšíření pro tuto úlohu bych zvolil například tlačítko pro okamžité odstavení, pro odstavení zařízení a tlačítko pro opětovné oživení.