

Katedra informatiky a výpočetní techniky

**Semestrální práce z předmětu KIV/TI**

**Logické řízení - řízení otevírání garážových vrat**

Jonáš Dufek

A21B0111P

[jonasd@students.zcu.cz](mailto:jonasd@students.zcu.cz)

Stanislav Kafara

A21B0160P

[skafara@students.zcu.cz](mailto:skafara@students.zcu.cz)

18. 11. 2022

**Obsah**

[1 Zadání 3](#_Toc119262542)

[2 Analýza úlohy 4](#_Toc119262543)

[3 Automatový model 5](#_Toc119262544)

[3.1 Vstupní signály 5](#_Toc119262545)

[3.2 Výstupní signály 5](#_Toc119262546)

[3.3 Přechodový graf 6](#_Toc119262547)

[3.4 Popis stavů 7](#_Toc119262548)

[4 Popis implementace 8](#_Toc119262549)

[5 Uživatelská příručka 9](#_Toc119262550)

[5.1 Příprava a spuštění aplikace 9](#_Toc119262551)

[5.2 Komunikace s aplikací 9](#_Toc119262552)

[6 Závěr 11](#_Toc119262553)

[6.1 Zhodnocení 11](#_Toc119262554)

[6.2 Další možná vylepšení a rozšíření tématu 11](#_Toc119262555)

# Zadání

**12. Logické řízení - řízení otevírání garážových vrat**

Na [http://home.zcu.cz/~vais/](https://l.facebook.com/l.php?u=http%3A%2F%2Fhome.zcu.cz%2F~vais%2F%3Ffbclid%3DIwAR0g-UC-mkNvBwQtjwnKP2-6ddVUp8j-ONk4NOiw7QUsZ2Q1Qp8HvamLw6U&h=AT0acMJx0JeY0Nz_wVWZvFHxi76v7AxDBWr7NtArMJA9mCqi6ScMJFMviE_dAjP_gBUJzIzwCS8qFZ9kgBaecf954gbLSgwB5atDavXRKm_li9Ny-PAbaGRwl83k1vaiOI6hBA) v rozšiřujícím materiálu o konečných automatech prostudujte kapitoly Logické řízení a Principy softwarové implementace.

Navrhněte konečněautomatový model pro řízení otevírání garážových vrat podle zadání:

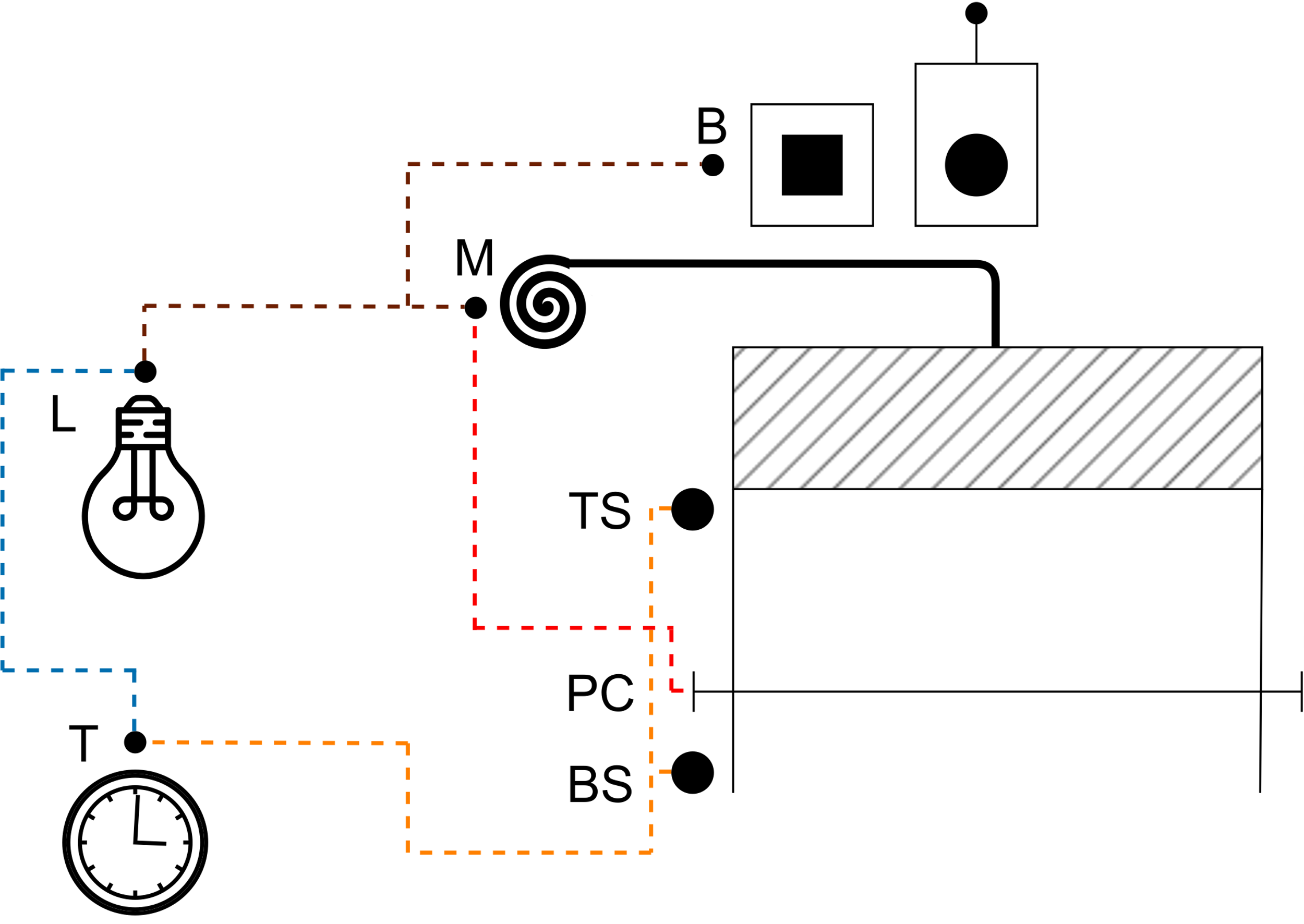
1. Vrata jsou řízena ovládacím tlačítkem v garáži a vzdáleným ovládáním ovladačem (chová se jako tlačítko).
2. Po příchodu signálu z tlačítka nebo ovladače (tj. po stisku tlačítka či ovladače) se dveře pohybují nahoru nebo dolů.
3. Když je tlačítko stisknuto jednou při běhu dveří, dveře se zastaví.
4. Když je tlačítko stisknuto podruhé, začnou se pohybovat opačným směrem.
5. Existují horní a dolní mezní spínače pro zastavení motoru ovládání vrat.
6. Je nainstalováno zařízení generující světelný paprsek dole přes šířku dveří fotobuňka, jestliže je paprsek přerušen, když se dveře zavírají, dveře se zastaví a začnou se pohybovat opačným směrem.
7. V garáži je světlo, které má svítit při pohybu dveří a pak ještě 5 minut potom co se dveře otevřou nebo zavřou.

Definujte všechny vstupní a výstupní signály, automat popište přechodovým grafem.

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.

# Analýza úlohy

Analýza úlohy probíhala tak, že jsme si nejprve načrtnuly diagram *Obrázek 1*.



*Obrázek 1 - Diagram řízeného systému*

Diagramem jsme si definovaly následující komponenty:

* B – Tlačítko, nebo vzdálený ovladač; Umožňuje uživateli ovládat systém.
* M – Motor; Může buď být vypnutý, nebo pohybovat vraty dolů, nebo nahoru.
* L – Světlo; Zapne se, jakmile se vrata začnou pohybovat. Vypne se, když buď vrata budou zastavena při otevírání, resp. zavírání, nebo když doběhne časovač.
* T – Časovač; Resetuje se a začne odpočítávat 5 minut, jakmile dojde k sepnutí  
  spínače TS nebo BS.
* TS – Horní mezní spínač vrat; Sepne při úplném otevření vrat.
* BS – Dolní mezní spínač vrat; Sepne při úplném zavření vrat.
* PC – Fotobuňka; Sepne při přerušení paprsku fotobuňky.

S pomocí diagramu *Obrázek 1* již je jednoduché nadefinovat vstupní a výstupní symboly (viz kapitola 3) a poté vytvořit přechodový graf automatu.

# Automatový model

## Vstupní signály

Všechny tyto vstupní signály jsou zadávány uživatelem. V reálné implementaci by uživatel „zadával“ pouze signál BP a zbytek by byl ovládán pomocí hardwarových komponent.

*Tabulka 1: Vstupní symboly automatu*

|  |  |
| --- | --- |
| **Vstupní symbol** | **Popisek** |
| BP | Tlačítko stisknuto / ovladač stisknut. |
| BS | Dolní spínač sepnut (dveře se zavřely). |
| TS | Horní spínač sepnut (dveře se otevřely). |
| PC | Paprsek fotobuňky přerušen (objekt brání zavření dveří). |
| TF5 | Časovač pěti minut doběhl. |

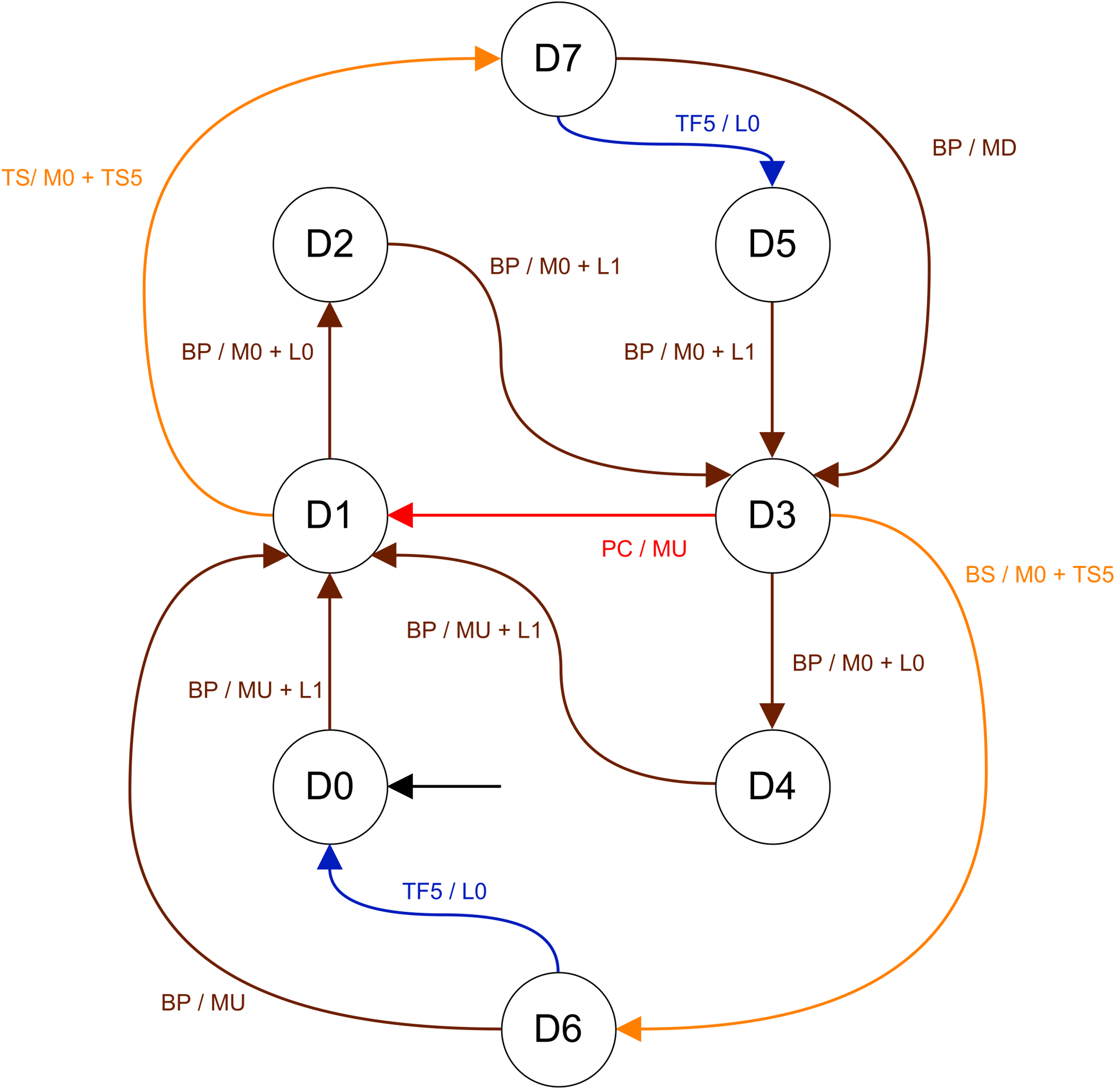
## Výstupní signály

Výstupní signály představují „akce“, které se vykonávají při změně stavu automatu (automat Mealyho typu).

*Tabulka 2: Výstupní symboly automatu*

|  |  |
| --- | --- |
| **Výstupní symbol** | **Popisek** |
| M0 | Vypni motor ovládání vrat. |
| MD | Pohybuj vraty dolů. |
| MU | Pohybuj vraty nahoru. |
| L0 | Zhasni světlo. |
| L1 | Rozsviť světlo. |
| TS5 | Resetuj časovač a spusť časování pěti minut. |

## Přechodový graf

**Oranžově** jsou označeny signály horního a dolního spínače dveří, **modře** jsou označeny signály časovače, **hnědě** jsou označeny signály stisknutí tlačítka, a **červeně** je označen signál přerušení paprsku fotobuňky.

*Obrázek 2 - Přechodový graf automatu*

## Popis stavů

*Tabulka 3* popisuje jednotlivé stavy automatu (viz *Obrázek 2*).

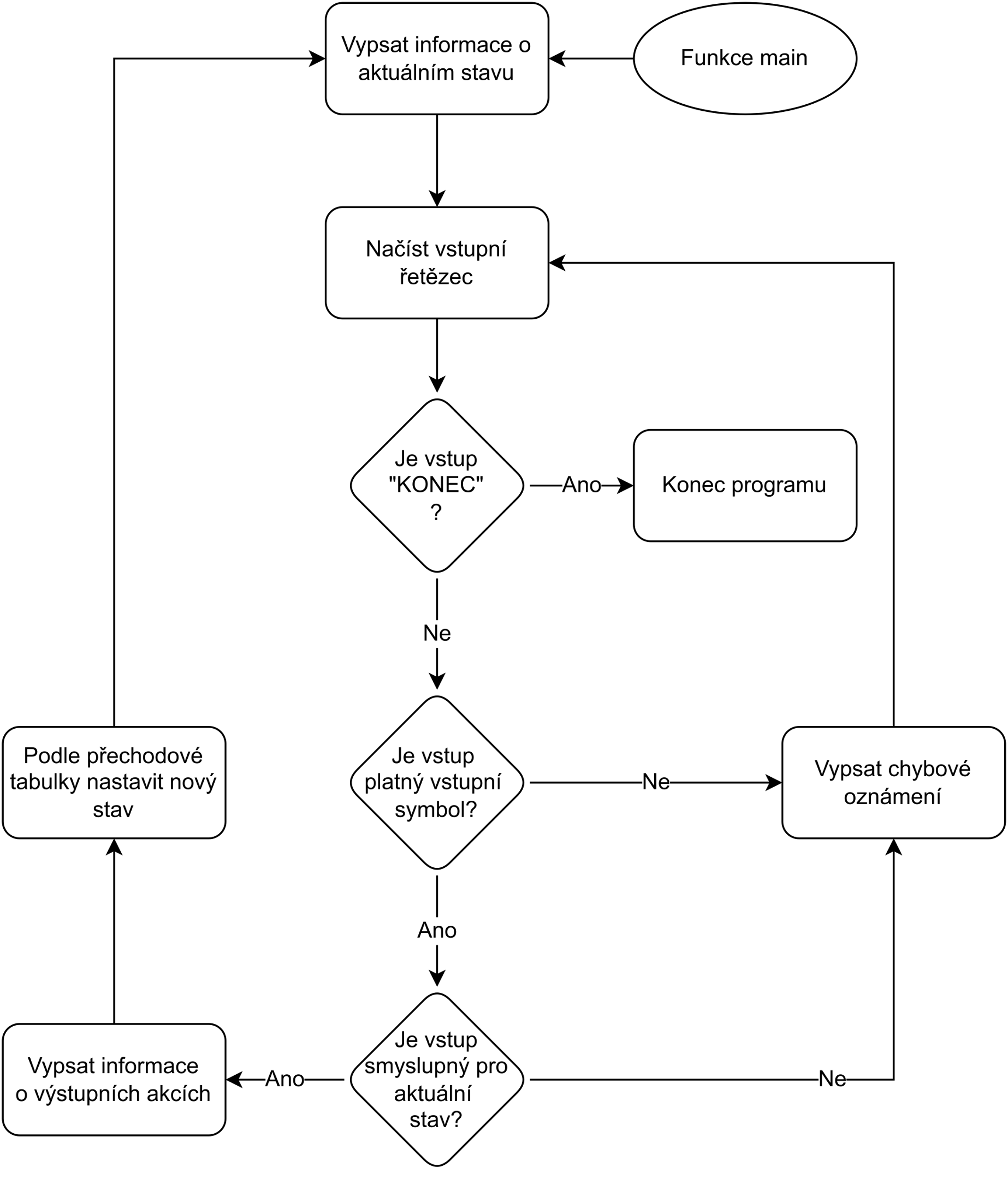
*Tabulka 3: Stavy automatu*

|  |  |
| --- | --- |
| **Stav** | **Popisek[[1]](#footnote-1)** |
| D0 | V zavřena, M vypnut, S nesvítí |
| D1 | V se pohybují nahoru, M zapnut, S svítí |
| D2 | V zastavena při pohybu nahoru, M vypnut, S nesvítí |
| D3 | V se pohybují dolů, M zapnut, S svítí |
| D4 | V zastavena při pohybu dolů, M vypnut, S nesvítí |
| D5 | V otevřena, M vypnut, S nesvítí |
| D6 | V zavřena, M vypnut, S nesvítí |
| D7 | V otevřena, M vypnut, S svítí |

# Popis implementace

Automat byl implementován standardním způsobem pro realizaci automatů Mealyho typu, tedy obsahuje přechodovou tabulku udávající následující stav, tabulku výstupních akcí udávající, jaká akce nastane po určitém vstupu, a pak obsahuje funkce, které s těmito tabulkami pracují.

Pro usnadnění práce s konzolovým uživatelským rozhraním byly přidány různá pole s popisky stavů nebo vstupních / výstupních symbolů (určeno pro výpis), dále bylo vytvořeno několik výčtových typů (pro stavy, vstupy, výstupy, …), jejichž účelem je zvýšení čitelnosti kódu.

Celý program je řízen smyčkou ve funkci loop(), jejíž diagram popisuje *Obrázek 3.*

*Obrázek 3 - Diagram funkce loop()*

Pro obsluhu „příkazových bloků“ diagramu byly implementovány příslušné funkce.

# Uživatelská příručka

## Příprava a spuštění aplikace

K softwarové implementaci byl použit jazyk C.

Kořenový adresář obsahuje soubor Makefile a Makefile.win, které mohou být použity pro překlad programu pomocí programu make. Program lze tedy přeložit příkazem  
make -f Makefile na platformě UNIX, resp. make -f Makefile.win na platformě Windows.

Jelikož je ale program obsažen pouze v jednom souboru, je možné program přeložit  
jednoduše pomocí překladače gcc (nebo alternativního). příkazem  
gcc -o fmmodel.exe src/fmmodel.c.

Program lze spustit příkazem ./fmmodel.exe na platformě UNIX, resp. fmmodel  
na platformě Windows.

Všechny příkazy jsou zadávány v kořenovém adresáři aplikace. Překlad a spuštění aplikace lze vidět na *Obrázku 4*.

## Komunikace s aplikací

Program komunikuje výhradně v konzoli. Uživatel zadává vstupní symboly (vypsané  
na začátku programu). Program vždy uživateli vypisuje informace o aktuálním stavu  
a vykonaných řídících signálech. Program lze kdykoliv ukončit zadáním řetězce „KONEC“.

Komunikace s aplikací lze vidět na *Obrázku 4*.

Obsah obrázku text

Popis byl vytvořen automaticky

*Obrázek 4: Překlad, spuštění aplikace a uživatelské rozhraní aplikace*

# Závěr

## Zhodnocení

Analyzovali jsme zadaný problém a vytvořili jeho schéma *Obrázek 1*. Pro problém jsme navrhly konečně-automatový model výše popsaný tabulkami vstupních symbolů *Tabulka 1*, výstupních symbolů *Tabulka 2* a stavů automatu *Tabulka 3* a obrázkem přechodového grafu automatu *Obrázek 2*.

Softwarový model jsme implementovaly v programovacím jazyce C. Vytvořili jsme soubory Makefile a Makefile.win umožňující pohodlný překlad programu na platformách UNIX  
a Windows pomocí programu make.

Práce splňuje všechny body zadání.

## Další možná vylepšení a rozšíření tématu

Dokážeme si představit vylepšené uživatelské rozhraní, které by bylo vizuálně přívětivější  
a intuitivnější, avšak myslíme si, že takové je dostačující.

Model by mohl být rozšířen o další komponenty, které má v garáži smysl ovládat.

Model by mohl být reálně využit k ovládání takového systému. Stačilo by jen vyměnit funkce výpisu výstupních signálů na obrazovku za výkonný kód vyslání takového signálu.

1. Pro zvýšení čitelnosti popisku jsou použity zkratky: V = vrata, M = motor, S = světlo. [↑](#footnote-ref-1)