

AMiTsys Junior



AMiTsys Expert

2011

## **Model inteligentního domu**

Autoři: *Dominik Hungr*

*Pavel Roček*

Škola: *Střední průmyslová škola elektrotechnická*

*Brno, Kounicova 16*

Konzultanti: *Ing. Miroslava Odstrčilíková*

*Antonín Veselý*

## Prohlášení

Prohlašujeme, že jsme tuto práci vypracovali samostatně a použili jsme literárních pramenů a informací, které citujeme a uvádíme v seznamu použité literatury a zdrojů informací

V Brně dne

24.4.2011

.....

## Poděkování

Rádi bychom poděkovali Antonínu Veselému, za obětavou pomoc při tvorbě této práce a za seznámení nás se světem programovatelných automatů. Bez jeho vedení by pro nás bylo jakkoliv nemyslitelné tento projekt dokončit. Další neméně významné poděkování patří Ing. Miroslavě Odstrčilíkové, pro ochotu zpřístupnit nám prostory školní laboratoře a zapůjčování školních pomůcek.

## Anotace

V rámci našeho školního oborového zaměření automatizace jsme počátkem školního roku 2010/2011 začali navštěvovat kroužek AMiT, který funguje jako dobrovolná studentská aktivita. Pod vedením Antonína Veselého jsme začali objevovat základy práce s programovatelným automatem (PLC) AMiNi2D od firmy AMiT. Jen co jsme se rozkoukali, už jsme si měli vybrat volné téma do soutěže AMiTsys Expert. Projekt Model inteligentního domu vznikl spojením dvou nápadů: Zabezpečením budovy a Automatickým ovládáním termoregulace a periférií (automatická brána, osvětlení zahrady) domu. Na modelu jsme pracovali společně a systémovou část jsme si rozdělili podle oblasti zájmu.

# Obsah

1	Úvod .....	8
2	Co je to inteligentní dům ?.....	9
3	Termoregulace budovy .....	10
3.1	System vytápění .....	10
3.1.1	Topný systém:.....	10
3.1.2	Topné větve: .....	11
3.2	Elektrokotel.....	12
3.2.1	Programové řešení .....	12
3.2.2	Modelové řešení.....	13
3.2.3	Reálné řešení.....	13
3.3	Akumulační nádrž .....	13
3.3.1	Modelové řešení.....	14
3.4	Čerpadlo kotlového topného okruhu .....	15
3.4.1	Programové řešení .....	15
3.4.2	Modelové řešení.....	15
3.4.3	Reálné řešení.....	15
3.5	Čerpadlo druhého topného okruhu.....	15
3.5.1	Programové řešení .....	15
3.5.2	Modelové řešení.....	17
3.6	Vytápění místností .....	17
3.6.1	Režimy vytápění .....	17
3.6.2	Programové řešení .....	18
3.6.3	Modelové řešení.....	18
3.6.4	Reálné řešení.....	19
3.7	Obrazovky.....	20

4	Automatická brána .....	24
4.1	Programové řešení.....	25
4.1.1	Rychlý proces pro ovládání brány .....	25
4.1.2	Pomalý proces (1000ms) .....	25
4.2	Obrazovky.....	26
4.3	Modelové řešení.....	28
4.4	Reálné řešení.....	30
5	Automatické osvětlení zahrady .....	31
5.1	Programové řešení.....	31
5.2	Modelové řešení.....	31
5.3	Obrazovky.....	32
6	Elektronický zabezpečovací systém EZS: .....	33
6.1	Názvosloví .....	<b>Chyba! Záložka není definována.</b>
6.2	Rozdělení detektorů.....	34
6.2.1	Venkovní obvodová (perimetrická) ochrana .....	34
6.2.2	Plášťová ochrana.....	34
6.2.3	Prostorová ochrana.....	35
6.3	Ústředna EZS .....	35
6.4	Regulace vytápění za pomoci EZS .....	<b>Chyba! Záložka není definována.</b>
6.5	Realizace EZS .....	36
6.5.1	Obrazovky .....	38
6.5.2	Programové řešení .....	<b>Chyba! Záložka není definována.</b>
6.5.3	Modelové řešení.....	44
7	Seznam použité literatury .....	46
7.1	Závěr.....	45
8	Příloha.....	47
8.1	Termoregulace, brána, osvětlení .....	47

8.1.1	Vstupy, výstupy .....	47
8.1.2	Databázové proměnné .....	47
8.1.3	Alias .....	49
8.1.4	Program .....	50
8.2	Elektronický zabezpečovací systém.....	63
8.2.1	Databázové proměnné .....	63
8.2.2	Alias .....	64
8.2.3	Program .....	65

# 1 Úvod

Naším úkolem bylo navrhnout a realizovat funkční model inteligentního domu. Aplikované systémy jsou řízeny pomocí dvou PLC AMiNi2D od firmy AMiT a rozšiřujícího modulu DM-UI8DO8 od stejného výrobce. Bezpečnostní systém je řízen prvním PLC samostatně a není spojen s ostatními systémy, z důvodu bezpečnosti. Druhé PLC ovládá termoregulaci domu, automatickou bránu a osvětlení zahrady. Je rozšířeno o vzdálené vstupy a výstupy, pro jejich nedostatek na samotném modulu. V modelu jsou nahrazeny reálné ovládací a indikační prvky za jednoduché levnější varianty, pracující na stejném, nebo podobném principu.

## 2 Co je to inteligentní dům ?

Inteligentní dům je soubor vzájemně spolupracujících zařízení zajišťuje optimální vnitřní prostředí pro komfort osob prostřednictvím stavební konstrukce, techniky prostředí, řídicích systémů, služeb a managementu. Inteligentní dům reaguje na potřeby obyvatel s cílem zvýšit jejich pohodlí, zpříjemnit jim zábavu, zaručit co nejvyšší bezpečí a snížit náklady na provoz.

Energický management se stará o řízení spotřeby domu a také nám dává možnost využít všech zdrojů energie např. solární panely na střeše, kotel na tuhé paliva který předehřívá vodu a uchovává jí a pak rozvádí do celého domu. Tímto pomáhá efektivně šetřit a rozdělit energii do domu.

Součástí inteligentních domů často bývá bezpečnostní systém, který včas informuje a hlídá neoprávněné vniknutí (narušení bezpečnosti). Stará se o správu přístupů do domu, identifikuje jednotlivé uživatele dle zadaného přístupového kódu. Pomocí SMS informuje uživatele o narušení bezpečnosti. Dokáže simulovat život v domě rozsvěcením vnitřního a venkovního osvětlení, případně spouštěním hudby či televize v nepřítomnosti obyvatel domu.

Inteligentní dům bývá vybaven kamerovým systémem, který střeží okolí a vnitřek domu. Při narušení bezpečnosti začne nahrávat záznam obrazu a zvuku, který může usvědčit pachatele.

Umí detekovat požár, který ohlašuje poplachem, zasláním zprávy na linku 112, aby ochránil obyvatele domu. Při vytopení domu dokáže vypínat elektrické zásuvky

Vše lze spravovat pomocí SMS nebo internetového prohlížeče.

V budovách s centrálním multimediálním systémem, můžeme poslouchat svojí oblíbenou hudbu kdekoli v domě např. připojením iPodu do systému a to díky zabudovaným reproduktorům v celém domě. Dále také sledování oblíbených pořadů v ložnici, kuchyni, kdekoli v době kde jsou instalovány LCD obrazovky.

Pomocí centrálního systému napojeného na kamerový systém můžeme zobrazit výstup z jednotlivých kamer hlídajících vnitřní i vnější prostory domu. Jedná se o základní prvky používané v inteligentních domech.

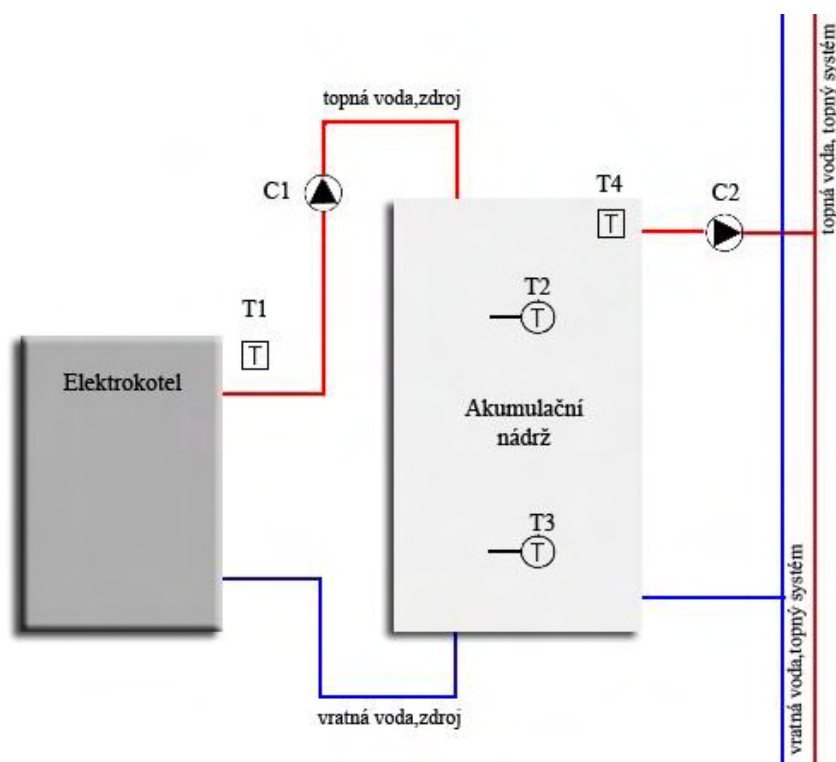
### 3 Termoregulace budovy

System vytápění reaguje na tepelné změny prostředí. Jeho úkolem je vytápět dům s co největší efektivitou a šetřit výdaje za odběr energie.

Byl aplikován topný systém skládající se z elektrokotle, akumulární nádrže, čerpadel a elektrotermických hlavice dvoucestných ventilů na radiátorech. Tento topný systém byl aplikován do modelu domu.

#### 3.1 Systém vytápění

##### 3.1.1 Topný systém:



Obrázek 1: Topný systém

T1-spínaný termostat, indikuje přetopení elektrokotle

T2-snímač teploty osazený u vrchu akumulární nádrže

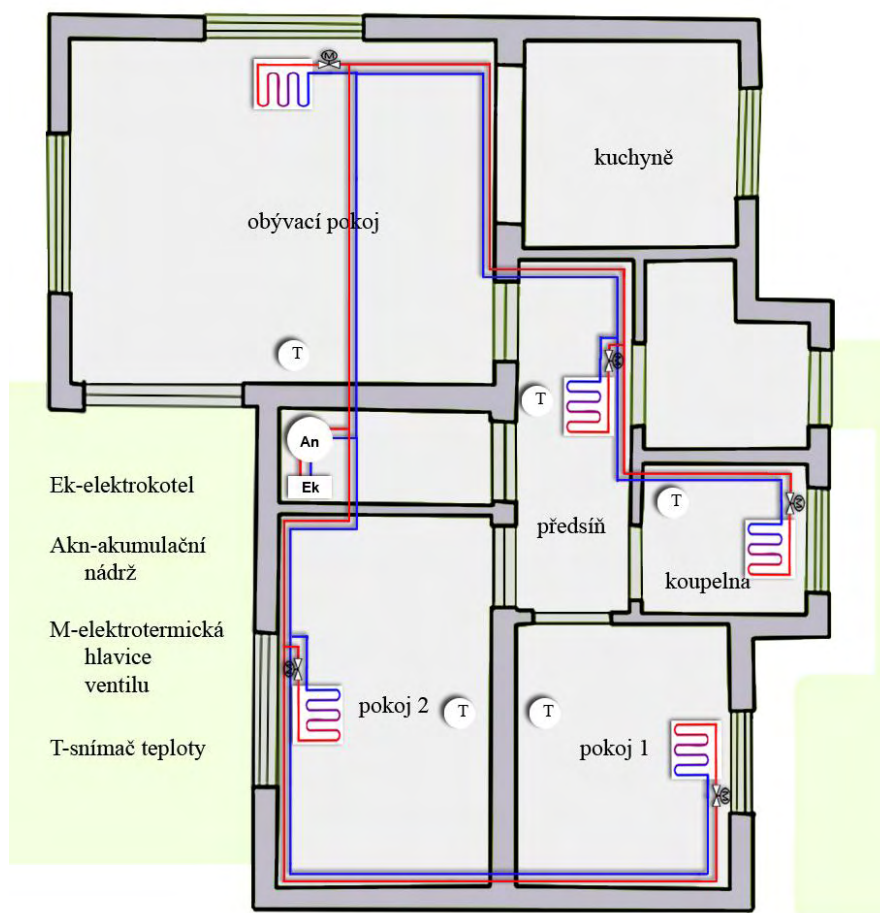
T3-snímač teploty osazený u dna akumulární nádrže

T4-spínaný termostat, indikuje přetopení akumulární nádrže

C1-čerpadlo 1

C2-čerpadlo 2

### 3.1.2 Topné větve:



Obrázek 2: Topný rozvod

Teplovodní radiátory jsou osazeny pod okny, pokud se v místnosti nachází. Jsou instalovány ve dvou topných větvích, od největší místnosti po nejmenší. Otevřená kuchyně je bez radiátoru, protože sama vydává hodně parazitního tepla.

Prostorové termostaty T snímají teplotu na opačném konci místnosti, než se nachází radiátory. Není vhodné, aby byly prostorové termostaty umístěny na venkovních stěnách budovy.

## 3.2 Elektrokotel

Obecně se říká, že topení elektrickými kotli je drahé. Není tomu tak, pokud seřídíme jeho činnost efektivně. Regulace elektrokotle spočívá v připínání jednotlivých topných spirál v závislosti na požadované výstupní teplotě topné vody na čase. Náš elektrokotel připíná sekundární topnou spirálu v případě, že je rozdíl mezi teplotou požadovanou a teplotou aktuální veliký.

### 3.2.1 Programové řešení

```
//elektrokotel se zapne při sepnutí vypínače
//pokud sepne buďto termostat přetopení kotle T1 nebo termostat
přetopení Akumulační nádrže...

//Spuštění kotle a topného systému
Let @Ek_run=if(@Ek_pswitch,true,false)

//...kotel bude nouzově vypnut
let @Ek_run=if((@T1) or (@T4)),false,@Ek_run)

if @Ek_run

    //primární spirála pojede pokud budeme chtít
dosáhnout požadované
    //teploty v akumulaci nádrži

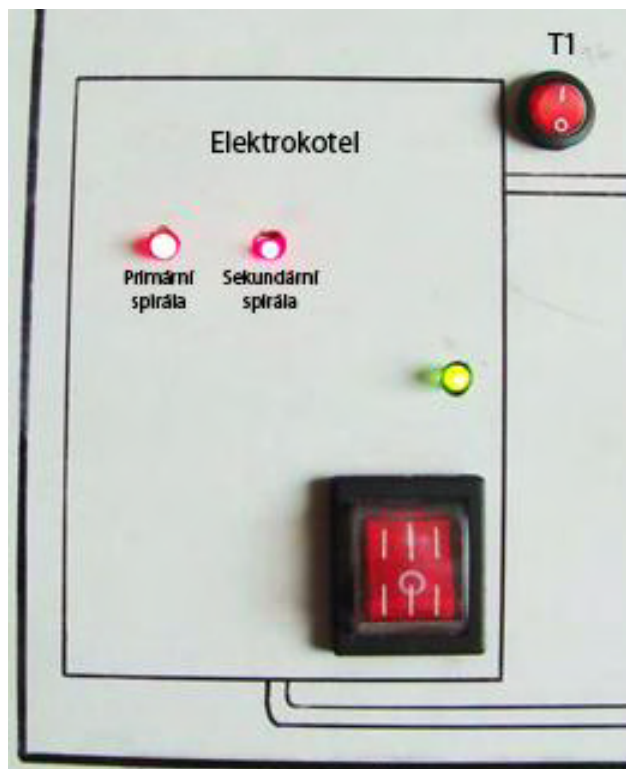
    Let @prim_tspir=if((T_ak_poz>T3),true,false)

    //sekundární spirála se bude spínat, pokud bude
rozdíl požadované teploty
    //a teploty skutečné větší než 30stupňů-pro rychlejší
vytopení Akn

    Let @sec_tspir=if((T_ak_poz-T3)>30),true,false)
else
    Let @sec_tspir=false
    let @prim_tspir=false
endif
```

### 3.2.2 Modelové řešení

V modelu je sepnutí elektrokotle indikováno zelenou LED diodou. Topné spirály jsou simulovány červenými LED diodami. Spínaný termostat je nahrazen spínačem.



Obrázek 3: Model elektrokotle

### 3.2.3 Reálné řešení

Ve skutečnosti by bylo vhodné pro regulaci elektrokotle (připínání spirál) použít sestavu PID regulátorů.

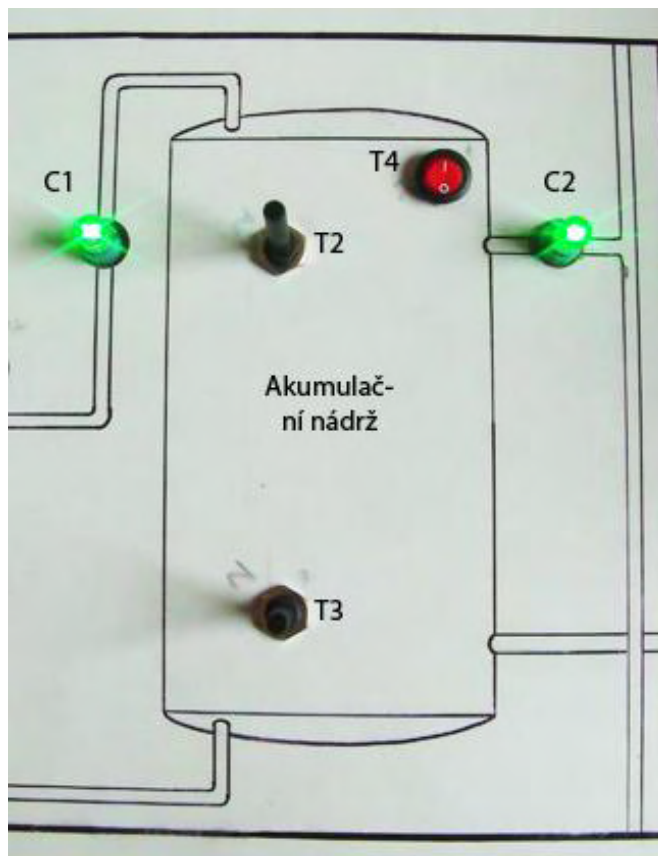
## 3.3 Akumulační nádrž

Akumulační nádrž šetří výdaje za energii, díky shromažďování teplé vody, která neztrácí tak rychle svoji tepelnou energii. V dnešní době již není pořízení vhodné akumulace nádrže drahé díky sériové výrobě. Je nutno také zvolit správný objem. Při malé nádrži bude docházet k častému přetopení, na druhou stranu příliš velká akumulace nádrž by se dlouho vytápěla, pokud by se to vůbec podařilo.

Je osazena spínaným termostatem pro přetopení v horní části. Dále jsou tu dva snímače teploty. T2 u vrchu, pro akumulční nádrž chovající se jako zdroj tepla a T3 u spodu pro akumulční nádrž chovající se jako spotřebič tepla.

### 3.3.1 Modelové řešení

V modelu je termostat nahrazen kolébkovým spínačem. Snímače teploty jsou nahrazeny dvěma potenciometry.



Obrázek 4: Model akumulční nádrže

### 3.4 Čerpadlo kotlového topného okruhu

Pro kotlový topný okruh lze použít standartní čerpadlo, protože se nebude skokově měnit tlak. Čerpadlo poběží, bude-li třeba vytopit akumulární nádrž na požadovanou hodnotu, nebo bude potřeba vychladit přehřátý kotel.

#### 3.4.1 Programové řešení

```
//čerpadlo poběží, pokud budeme chtít vytopit akumulární nádrž na požadovanou hodnotu
//čerpadlo poběží i v příladě, že se nouzově vypne Elektrokotel kvůli přetopení, snímaném termostatem T1 kvůli vychlazení kotle

If @Ek_run
    Let @Run_C1=if(((T_ak_poz>T3) or (@T1)), true, false)
    Let @Run_C1=if((T_ak_poz<T3), false, @Run_C1)
Else
    Let @Run_C1=false
Endif
```

#### 3.4.2 Modelové řešení

V modelu je chod čerpadla znázorněn zelenou LED diodou na 24V.

#### 3.4.3 Reálné řešení

Čerpadlo by se zapojilo s použitím časového relé pro plynulý pozvolný doběh.

### 3.5 Čerpadlo druhého topného okruhu

Do tohoto topného okruhu je nutné osadit elektronicky řízené čerpadlo, kvůli použití elektrotermických hlavic. Ty se budou v závislosti na prostorové teplotě zavírat a tlak v potrubí se bude skokově měnit.

#### 3.5.1 Programové řešení

Programové řešení regulace tlaku byla regulace výkonu čerpadla C2 v závislosti na počtu uzavřených ventilů, ovládaných elektrotermickými hlavicemi. Musely být sestaveny všechny kombinace veškerých otevřených a zavřených ventilů. Pro zjednodušení byla zaznamenána jedinečná hodnota proměnné pro každou kombinaci. Ta byla následně použita v podmínkách programu. Program viz příloha

stavy ventilů					uzavřených ventilů	dec hodnota Integeru					
eth5	eth4	eth3	eth2	eth1		0	1	2	3	4	5
0	0	0	0	0	0	0					
0	0	0	0	1	1	1	1				
0	0	0	1	0	1	1	2				
0	0	0	1	1	2	2	3				
0	0	1	0	0	1	1	4				
0	0	1	0	1	2	2	5				
0	0	1	1	0	2	2	6				
0	0	1	1	1	3	3	7				
0	1	0	0	0	1	1	8				
0	1	0	0	1	2	2	9				
0	1	0	1	0	2	2	10				
0	1	0	1	1	3	3	11				
0	1	1	0	0	2	2	12				
0	1	1	0	1	3	3	13				
0	1	1	1	0	3	3	14				
0	1	1	1	1	4	4	15				
1	0	0	0	0	1	1	16				
1	0	0	0	1	2	2	17				
1	0	0	1	0	2	2	18				
1	0	0	1	1	3	3	19				
1	0	1	0	0	2	2	20				
1	0	1	0	1	3	3	21				
1	0	1	1	0	3	3	22				
1	0	1	1	1	4	4	23				
1	1	0	0	0	2	2	24				
1	1	0	0	1	3	3	25				
1	1	0	1	0	3	3	26				
1	1	0	1	1	4	4	27				
1	1	1	0	0	3	3	28				
1	1	1	0	1	4	4	29				
1	1	1	1	0	4	4	30				
1	1	1	1	1	5	5	31				

Tabulka 1: Hodnoty promenné R\_Regprom v závislosti na kombinaci otevřených a zavřených ventilů

### **3.5.2 Modelové řešení**

Chod čerpadla C2 je v modelu indikován zelenou LED diodou na 24V.

Výkon čerpadla je zobrazen na obrazovce Čerpadlo.

## **3.6 Vytápění místností**

Vytápění místností je řízeno nastaveným režimem. Systém reaguje na prostorovou teplotu okolí a uzavírá, nebo otevírá dvoucestné ventily radiátorů v místnosti.

### **3.6.1 Režimy vytápění**

#### **Temperace:**

Při temperaci udržujeme teplotu domě tak, aby neklesla pod mez, kdy se výrazně ochladí stěny (14-16 stupňů ). Pokud se tak stane, bude energeticky náročnější(dražší) dům znovu vytopit. Dále se pak i po určitou dobu nebude dostavovat pocit tepla při klasické komfortní pokojové teplotě.

Režim temperace nastaví nízkou požadovanou teplotu akumulární nádrže a požadovanou teplotu ve všech místnostech nastaví na 16 stupňů.

#### **Útlum:**

Tento režim nastaví požadovanou teplotu v akumulární nádrži na 70 stupňů. Požadovaná teplota ve všech místnostech je 19stupňů. Uplatnění má zejména při opouštění domu na víkendy a podobně.

#### **Komfort:**

V akumulární nádrži je požadavek na teplotu maximální, kvůli efektivitě. Pro každou místnost je možné volit teplotu zvlášť od 20 do 25ti stupňů.

#### **Denní plán:**

Topný systém nastaven stejně jako u komfortu. Uživatel si může pro každou místnost nastavit teploty v časových zlomech dne. Teplotu je možno volit od 15ti do 25ti stupňů

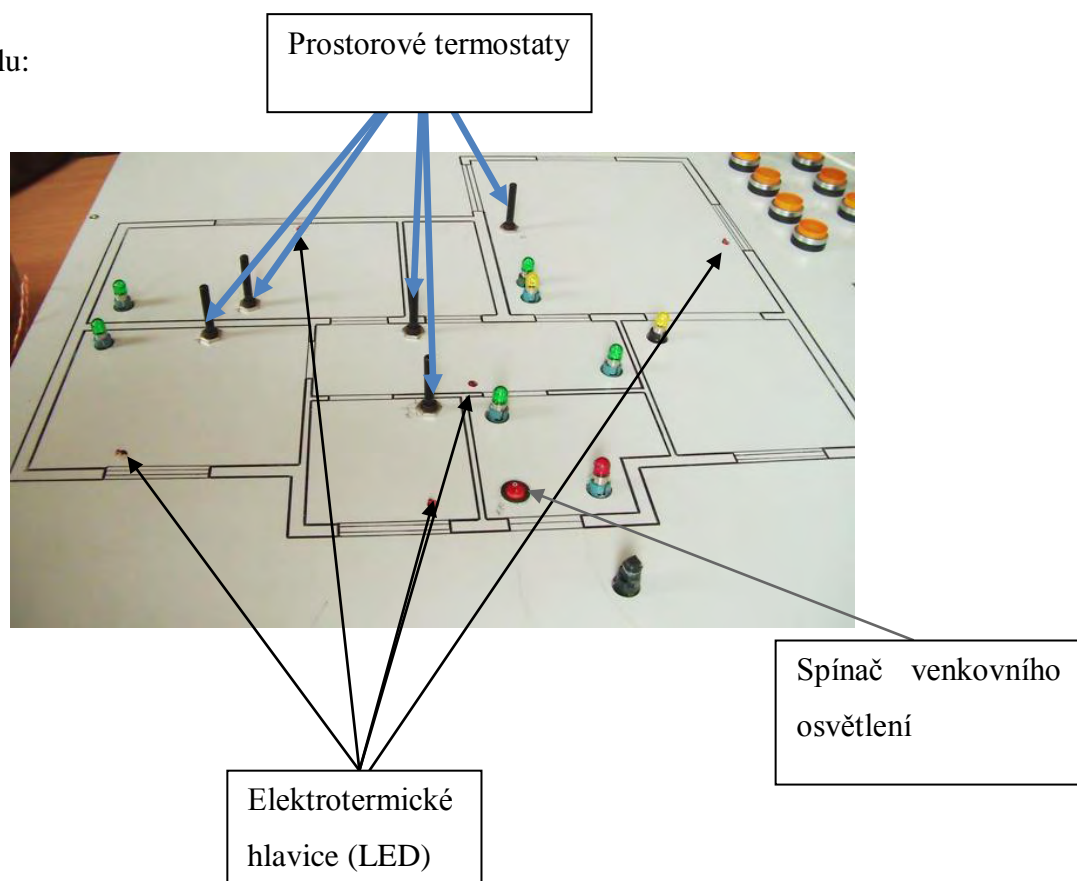
### 3.6.2 Programové řešení

V jednotlivých stavech se pomocí podmínky IF porovnávají teploty požadované s teplotou aktuální pro každou místnost. Pokud dosáhne teplota místnosti požadované hodnoty, uzavře se ventil. Dále jsou zde nastaveny parametry modulů Dayplan a přepínání stavů. Program viz příloha

### 3.6.3 Modelové řešení

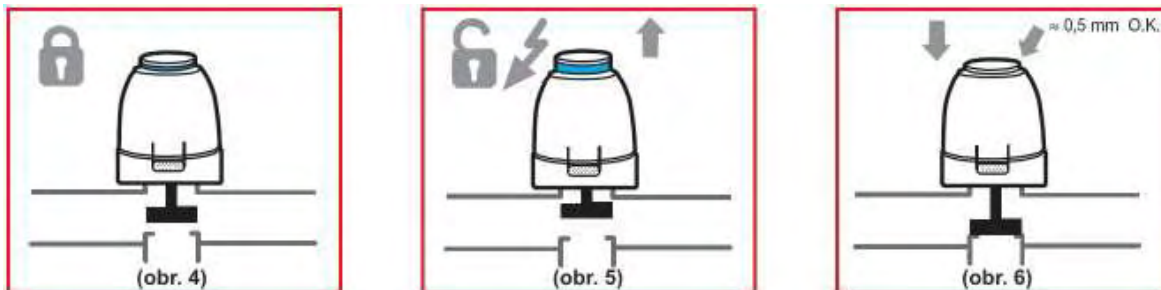
Elektrotermické hlavice eth\_x jsou simulovány červenými LED diodami, přičemž svítící dioda indikuje uzavřený ventil. Prostorové termostaty jsou simulačně nahrazeny 2,5k $\Omega$  potenciometry. Režimy vytápění se přepínají na displayi AMiNi2D.

Popis modelu:



### 3.6.4 Reálné řešení

Elektrotermické hlavice by byly přímo ovládány výstupem z PLC 24V. Přítomná jednička nebo nula na výstupu by pak otevírala/zavírala ventil.



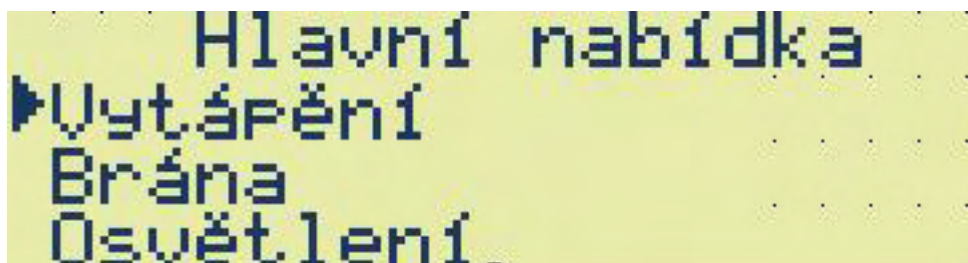
Obrázek 5: Správná instalace elektrotermické hlavice

Jako prostorové snímače by šlo použít typ Ni1000.

### 3.7 Obrazovky

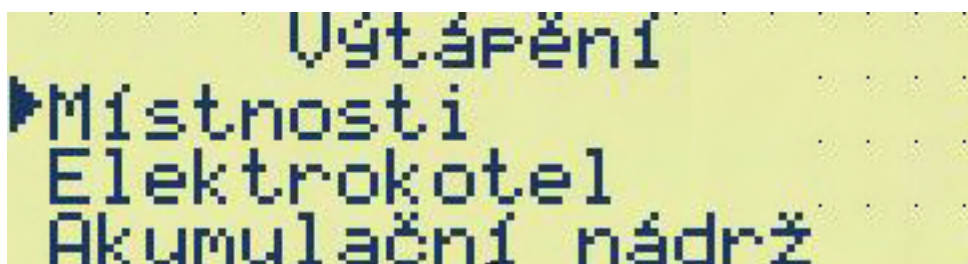
Ovládání obrazovek je intuitivní. Je používáno šipek pro pohyb v menu a změnu hodnot, tlačítka enter pro výběr a tlačítka ESC pro návrat na předchozí obrazovku.

Hlavní nabídka:



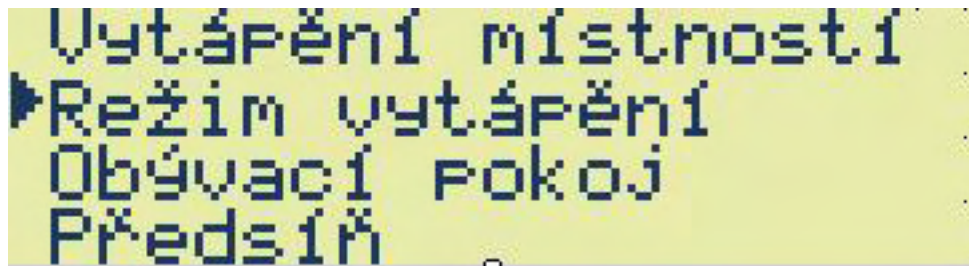
Vytápění: menu nabízející možnosti:

- Místnosti
- Elektro-kotel
- Akumulační nádrž
- Čerpadla



Místnosti: menu nabízející možnosti:

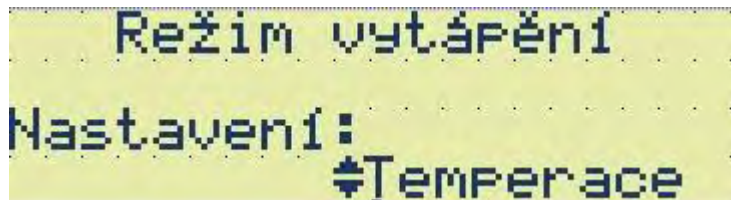
- Režim vytápění
- Obývací pokoj
- Předsíň
- Koupelna
- Pokoj 1
- Pokoj 2



Režim vytápění:

Zde je možno si navolit jaký chceme zapnout režim vytápění domu.

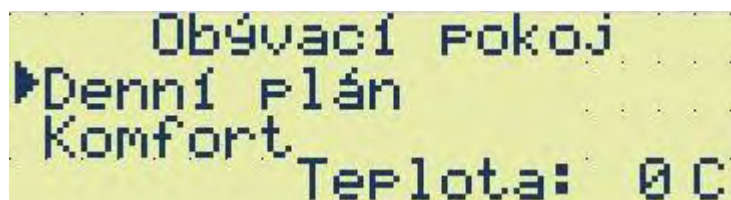
- Temperance
- Útlum
- Komfort
- Denní plán



Jednotlivé místnosti

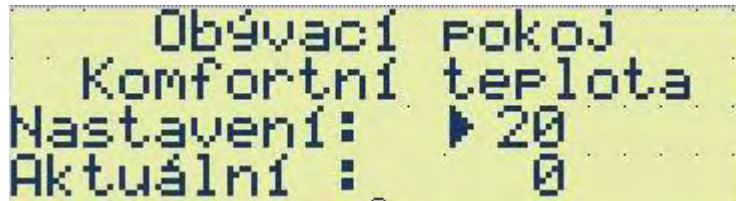
Menu:

Menu jednotlivých místností nabízí možnost výběru obrazovky pro nastavení komfortní teploty, nebo obrazovky s denním plánem. Dále je zde zobrazena aktuální teplota v místnosti.



Komfort místností:

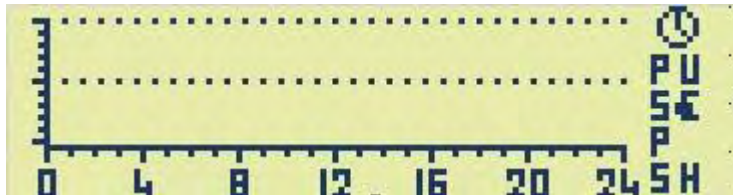
Zde je možné nastavit si teplotu vybraného pokoje.



```
Obývací pokoj
Komfortní teplota
Nastavení: ▶ 20
Aktuální : 0
```

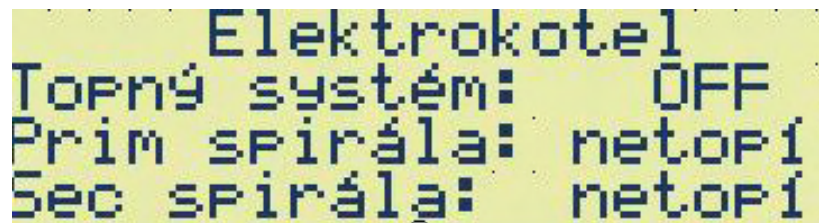
Denní plány místností:

Nastavení teplot v časových zlomech dnů pro vybranou místnost.



Elektrokotel:

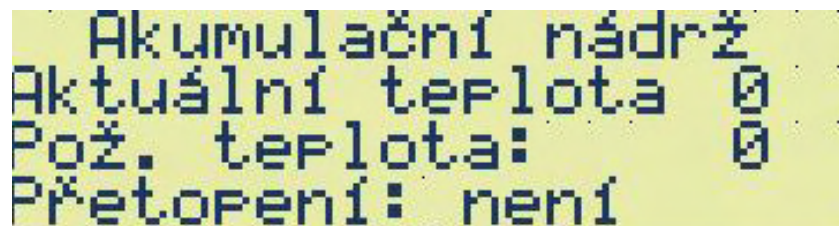
Kontrola stavů elektrokotle



```
Elektrokotel
Topný systém: OFF
Prim spirála: netopí
Sec spirála: netopí
```

Akumulační nádrž:

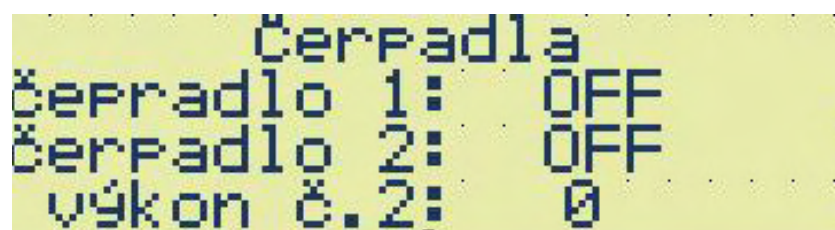
Kontrola teplot akumulční nádrže



```
Akumulační nádrž
Aktuální teplota 0
Pož. teplota: 0
Přetopení: není
```

Čerpadla:

Kontrola chodu čerpadel

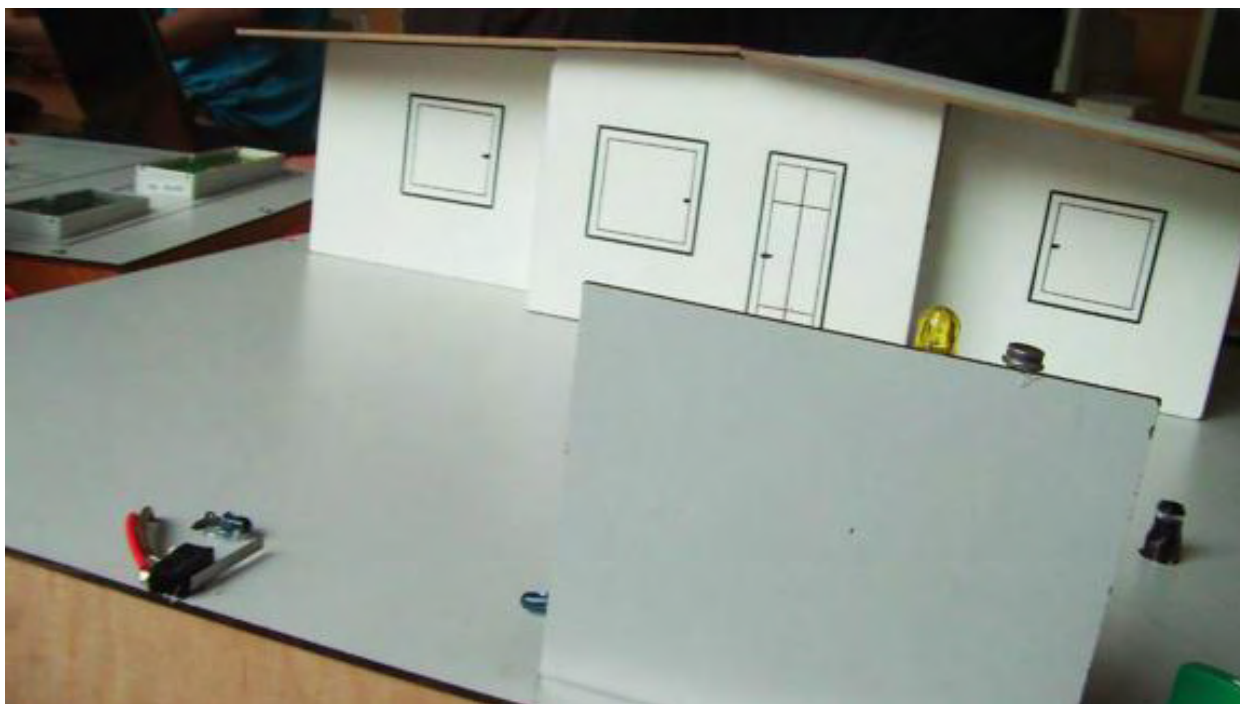


Čerpadla  
čerpadlo 1: OFF  
čerpadlo 2: OFF  
výkon č. 2: 0

## 4 Automatická brána

Automatická brána může být ovládána dálkovým ovladačem, pokud není manuálně zamknutá. Dále se dá ovládat přes obrazovku PLC stanice. Brána je řízena pomocí stavů. Stav se přepíná v závislosti na externích dějích (např. stisknutí tlačítka open). Stav je vázán podmínkami, aby se eliminovaly nežádoucí akce způsobené uživatelem (např. zároveň stisknutí tlačítka open a close).

Je vybavena optickou závorou, která nejenže zabraňuje uzavření, pokud je překážka v cestě, ale dokáže zajistit i automatické zavření brány pokud jí po nějaký čas nic neprojde. Tento interval lze uživatelem nastavit na 15, 30, 45 nebo 60 sekund. Pokud v průběhu odpočítávání vejde překážka, časovač se vynuluje a odečítá se znovu.



Obrázek 6: Celkový pohled na bránu

## **4.1 Programové řešení**

Program se skládá ze dvou navzájem propojených procesů, z rychlého a pomalého. U motoru se ovládají dva výstupní faktory:

- a) Digitální: přepínání relé pro přepólování elektromotoru
- b) Analogový: samotný pohyb elektromotoru

### **4.1.1 Rychlý proces pro ovládání brány**

Brána je ovládána v rychlém procesu, kvůli potřebě rychle reagovat jak na vstupech, tak na výstupech. Jsou zde podmínky pro stavy chodu brány. Stavy vylučují chybné manipulování s ovladačem a jsou nastaveny tak, aby chránily životnost motoru. Např. není možné při otevírání okamžitě začít bránu zavírat.

Je nutno motor nejdříve zastavit a poté ho rozjet na druhou stranu.

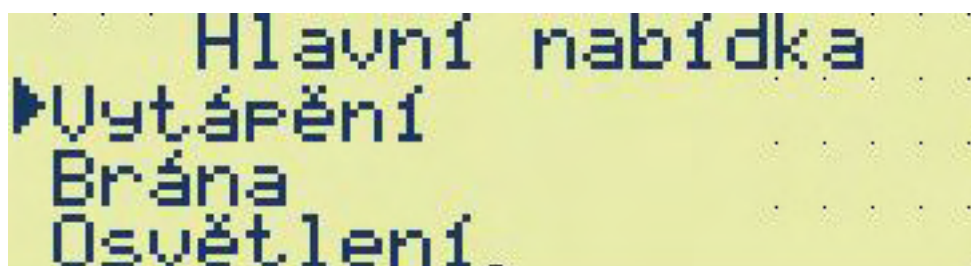
Dále je zde řešena polarizace elektromotoru, aby ho bylo možné otáčet v obou směrech. To je řízeno aliasem `@open_close`, ve kterém se mění boolovská hodnota 0 a 1. Podle toho se přepíná relé přepínač v modelu a polarizuje tak elektromotor. Program viz příloha

### **4.1.2 Pomalý proces (1000ms)**

V pomalém procesu se ovládá signalizace chodu brány (blikání LED diody za pomoci negace výstupní hodnoty) a čítač pro automatické uzavření brány. Proces je nastaven na  $1000\text{ms} = 1\text{sec}$ , takže každá perioda znamená 1sec. Nastaví se podmínky pro přepínání módů časovače 15,30,45,60sec. Pokud je zapnut časovač každý cyklus se inkrementuje. Pokud dosáhne přednastaveného času, brána se začne zavírat, pokud se během čítání intervalu dostane překážka do optické závory, čítač se vynuluje a celý cyklus běží znovu. Program viz příloha

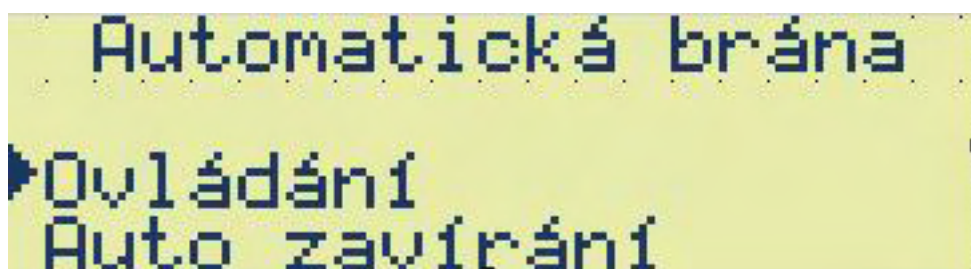
## 4.2 Obrazovky

Hlavní nabídka:



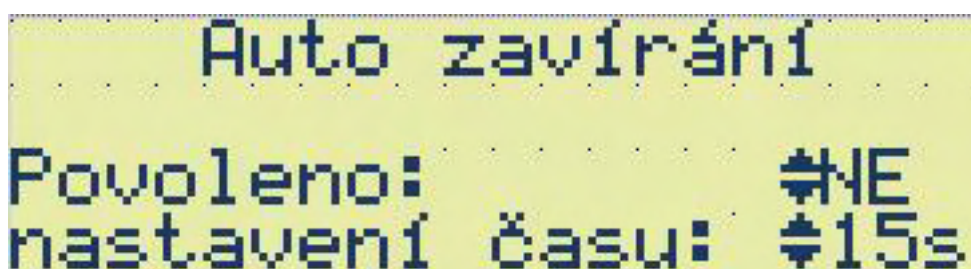
Brána:

Obrazovka nabízí menu pro automatickou bránu.



Automatické uzavírání brány:

Zde je možné povolit automatické zavírání a nastavit si časový interval.



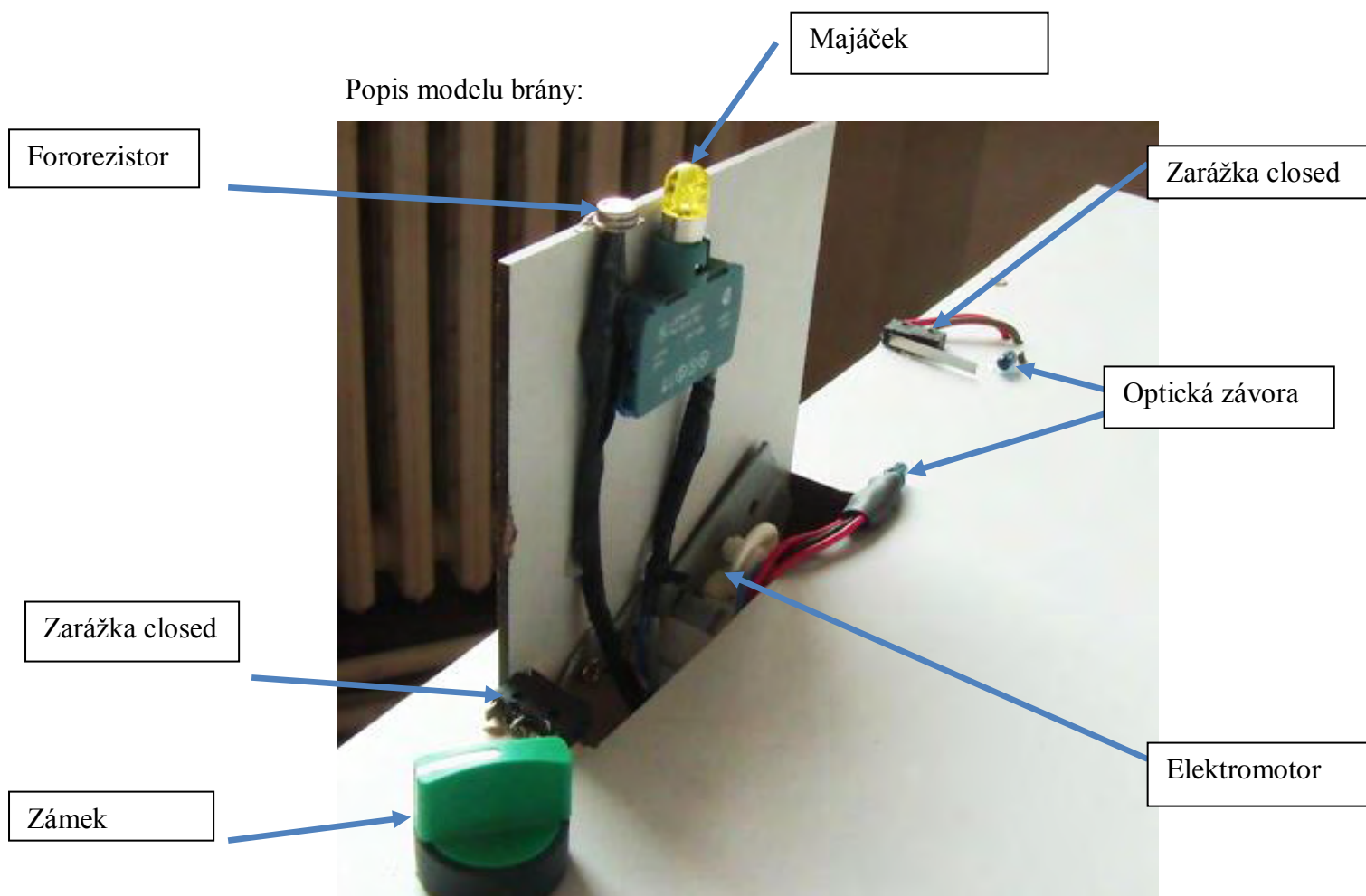
Ovládání brány:

Možnost manuálně ovládat bránu na obrazovce PLC.

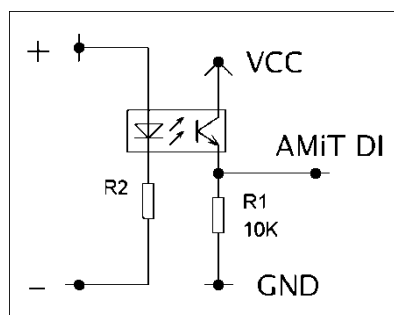


### 4.3 Modelové řešení

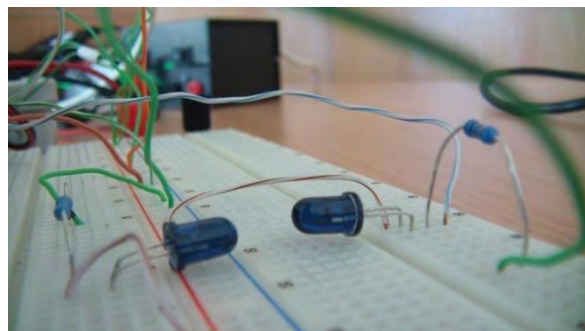
Chod brány je simulován otáčejícím se elektromotorkem. Elektromotorek je nainstalován do periferií modelu domu. Byl použit elektromotor n 5V z CD mechaniky.



Optická závora byla sestavena z fototranzistoru a fotodiody. Zapojeno dle následujícího schéma:



**Obrázek 8: Schéma optozávory**



**Obrázek 7: Zkouška zapojení**

Dálkové ovládání brány je pouze simulováno tlačítky, které mají ale stejný principiální základ jako skutečné dálkové ovládání. Dálkové ovládání po sepnutí tlačítka vyšle kód, který po rozpoznání přijímačem sepne relé, které spustí samotný proces, stejně jako tlačítko přímo.



**Obrázek 9: Dálkové ovládání brány na modelu**

Zámek na uzamčení automatické brány je v modelu nahrazen otočným spínačem.

Dále se nepodařilo spolehlivě rozjet mechanické uzavírání brány, tak je chod brány simulován pouze otáčením elektromotoru a spínání zarážek je nutno udělat ručně.

#### **4.4 Reálné řešení**

Automatická brána v tomto modelu má podstatný nedostatek. Postrádá regulaci rychlosti otáčení pro plynulý rozjezd a dojezd. To by se dalo vyřešit přidáním řízení otáček při rozběhu a doběhu mikroprocesorem zapojeného mezi analogovým výstupem z PLC a elektromotor.

Dále by pak brána fungovala na skutečné dálkové ovládání.

## 5 Automatické osvětlení zahrady

V modelu je zakomponované automatické osvětlení reagující na úpadek denního světla. Pokud hodnota osvětlení padne pod určitou mez, světla se rozsvítí. Osvětlení lze ovládat i manuálně klasickým spínačem.



Obrázek 10: Osvětlení přístupové cesty na modelu

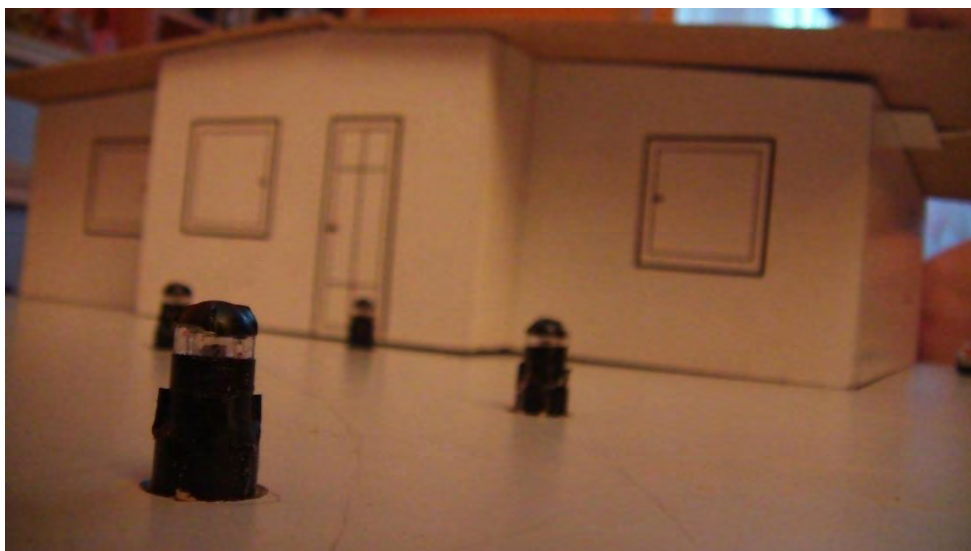
### 5.1 Programové řešení

Program funguje ve dvou režimech, automatickém a manuálním.

Při spuštění je automatický režim vypnutý, spustíme ho na obrazovce PLC. Při poklesu světla pod mez tmy se automaticky spustí osvětlení. Pro definování spodní meze tmy je použit funkční modul limits. V modulu limits je ale paradoxně nastavena horní mez, protože hodnota snímaná z analogového vstupu se s snižujícím se světlem zvyšuje, díky odporovým vlastnostem fotorezistoru. Pokud manipulujeme s osvětlením manuálně, automatický režim se sám vypne. Program viz příloha

### 5.2 Modelové řešení

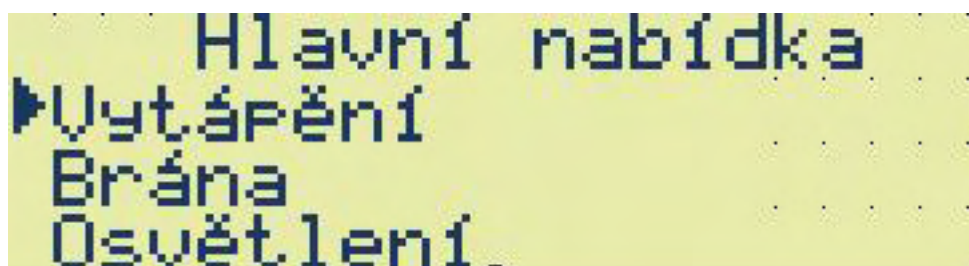
Jako snímač intenzity denního světla je použit klasický fotorezistor, osazený na periférii modelu. Modelová verze zahradních lamp jsou vizuálně upravené bílé LED diody na 24V, které jsou osazený podél zahradní cesty k hlavnímu vchodu. Pro manuální ovládání je v modelu domu kolébkový spínač .



Obrázek 11: Pohled na osvětlení cesty na modelu

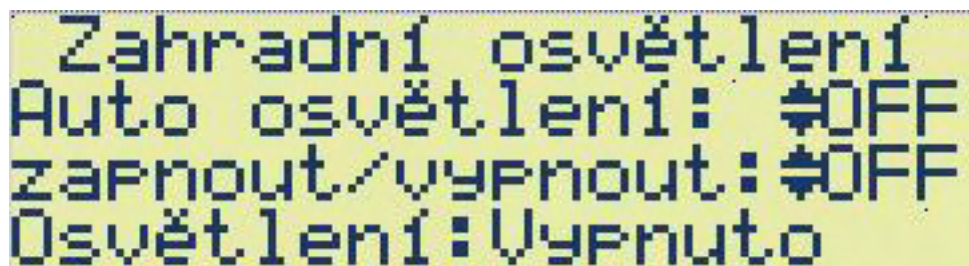
### 5.3 Obrazovky

Hlavní nabídka:



Osvětlení:

Možnost zapnutí automatického režimu i manuálního ovládání



## 6 Elektronický zabezpečovací systém EZS:

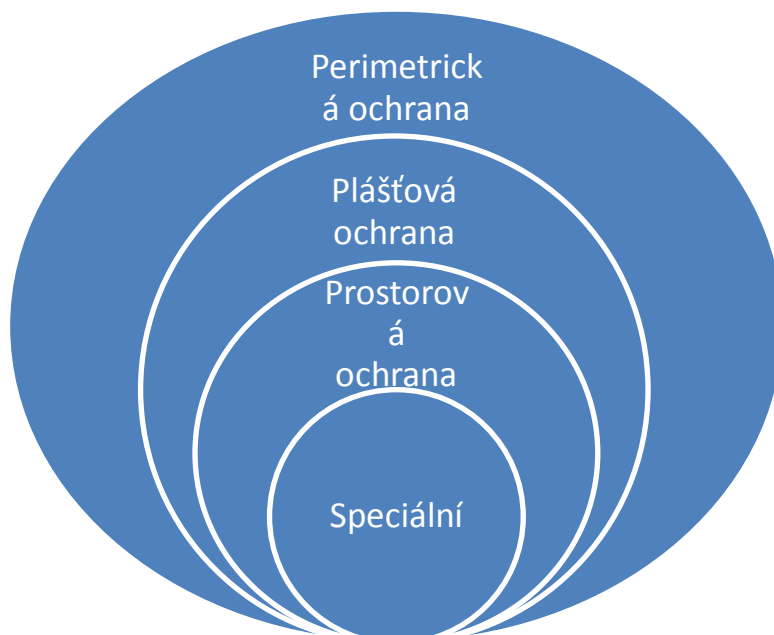
EZS je v dnešní době nepostradatelnou součástí vybavení rodinných domů. Jeho činnost chrání osoby i majetek, spuštěním signalizace při narušení, a automatickým voláním bezpečnostních složek.

V modelu se nachází jen část bezpečnostní techniky oproti skutečnosti. V dnešní době se EZS skládá z mnoha částí. Základ tvoří ústředna, jejímž jádrem je přesně naprogramovaný mikroprocesor. Dále klávesnice, která je komunikačním rozhraním mezi ústřednou a obsluhou. Samozřejmostí jsou detektory, bez kterých by zabezpečovací systém ztrácel smysl.



Obrázek 12: Základní uspořádání EZS

## 6.1 Rozdělení detektorů



### 6.1.1 Venkovní obvodová (perimetrická) ochrana

Předpoklad pro použití venkovní ochrany je, aby byl celý objekt oplocen. Jinak by mohlo docházet k nechtěnému narušení. Venkovní prostory jsou často monitorované kamerovým systémem. Dalšími prvky jsou speciální PIR čidla, tlakové hadice, infračervené závory a otřesové detektory. Nevýhodou jsou vlivy počasí (déšť, mráz, vítr, slunce apod.). Proto tyto detektory musí mít zvýšenou odolnost.

### 6.1.2 Plášťová ochrana

V podstatě se jedná o zabezpečení obvodu budovy (okna a dveře). Používá se nejčastěji v noci, kdy je vnitřek budovy nezabezpečen (pro volný pohyb osob uvnitř). Hlavním prvkem plášťové ochrany jsou magnetické kontakty a detektory tříštění skel.

#### *Magnetické kontakty*

Jedná se o magnet (umístěn na křídle dveří, či okna) a jazýčkový kontakt (umístěn na rámu). Pokud je křídlo dostatečně blízko rámu, je kontakt rozepnut. Při oddálení magnetu se kontakt sepne a dojde k poplachu.

### *GlassBreak (detektor tříštění skel)*

Akustický detektor zachytí specifický zvuk vydaný při tříštění skla. Modernější detektory navíc dokáží detekovat náhlou změnu teploty, či tlaku v místnosti. Tím se zabrání vniknutí zloděje do objektu, pokud by použil např. sadu na vyříznutí otvoru do skla.

### **6.1.3 Prostorová ochrana**

Jedná se o ochranu vnitřního prostoru budovy. Ta lze realizovat různými způsoby. V této kategorii je snad největší výběr detektorů.

#### *PIR detektor*

Nejpoužívanější detektor pro vnitřní prostorovou ochranu. Detekuje změny vyzařování teploty těles v jeho zorném poli. Dnešní PIR jsou navíc vybaveny antimaskingem, což je systém zabraňující „zastínění“ detektoru. Zastínění dříve využívaly zloději pro oblafnutí starších čidel. Spočívalo v pomalém přibližování k čidlu s deskou (nejčastěji polystyrenovou) před sebou a tím zúžení jeho zorného pole.

## **6.2 Ústředna**

Ústředna je nejdůležitějším prvkem zabezpečovacího systému, protože se stará o funkčnost celého systému.

Její funkce jsou příjem a vyhodnocování elektrických signálů z jednotlivých detektorů, ovládání signalizačních zařízení, řízení komunikace s pultem centrální ochrany a také zajištění napájení pro detektory. Jedná se o mozek celého systému.

V našem modelu je jako ústředna použit AMiNi2D, který slouží také jako klávesnice.

## 6.3 Realizace EZS

Při realizaci EZS jsme museli dbát na několik základních věcí. Především na to, aby byl systém co nejvíce podobný tomu skutečnému. Naše ústředna umí:

- Zakódovat a odkódovat (zabezpečit) budovu
- Vyhlásit poplach při narušení
- Nastavit zóny, které se mají zabezpečit
- Pamatuje si zóny narušené při posledním poplachu
- Umožňuje administrátorovi nastavit délku časů příchodu a odchodu
- Umožňuje administrátorovi nastavovat 8 uživatelských kódů (7 uživatelů + 1 administrátor)

Z programového hlediska využíváme pouze dva procesy a jeden podproces. První proces je hlavní (Proc00). Zajišťuje ověřování zadaného kódu (pomocí volání podprocesu), ověřování možnosti zakódování, vyhodnocování poplachů a jejich zaznamenání do historie. Druhý proces zejména zajišťuje vyhodnocování stavů zakódováno/odkódováno a všech podmínek souvisejících s jejich změnou. Podproces je zde pouze pro ověřování zadaného kódu a podmínek pro jeho zadání.

### 6.3.1 Proces Proc00

Tento proces nejdříve načte hodnoty na digitálních vstupech a uloží je do proměnné Tlacitko pod aliasy představujícími jednotlivé zóny. Jedná se tedy o aktuální hodnoty na detektorech (ty jsou realizovány pomocí spínačů – více později). Na konci procesu se tyto hodnoty zapisují na digitální výstupy, které rozsvěcí diody představující detektory.

Dále tento proces testuje prázdnot zón před zakódováním. V zónách které chceme zabezpečit (proměnná Hlidané) nesmí být aktivní detektor.

```
// kontrola pri zakodovani - dotaz na prazdnost zon  
Let Test = (Tlacitko ^ Hlidane)
```

Dále proces volá podprogram pro ověřování zadaného kódu (viz níže) a pracuje s alarmovými stavy. Detekuje alarmy prostým způsobem.

```
Let Poplach = ((Tlacitko & Hlidane) & poplach_pri)
```

Kde Tlacitko jsou aktuální stavy detektorů, Hlidane jsou zóny, které EZS sřeží a poplach\_pri je proměnná nesoucí zóny které nevyhlašují okamžitý poplach (magnet a PIR

v předsíni). Poté poplachy také ruší, při zadání uživatelského kódu a zapisuje do historie poplachů.

### **6.3.2 Proc01**

Tento proces se zabývá problematikou zakódování a odkódování. Hlavním problémem jsou příchodové a odchodové časy, během kterých může uživatel do systému zasahovat. Při odchodu může zakódování zrušit před uplynutím odchodového času, aniž by se budova zabezpečila.

Opačně při příchodu do zabezpečené budovy musí uživatel zadat správný kód, jinak systém vyhlásí poplach. Pomůže mu s tím proměnná `poplach_pri`, která je defaultně nastavena tak, aby nedetekovala příchodové zóny (magnet a PIR předsíň). Po uplynutí příchodového času je tato proměnná nastavena tak, aby i tyto dvě zóny vyhlásily poplach.

### **6.3.3 Podprogram ověření kódu**

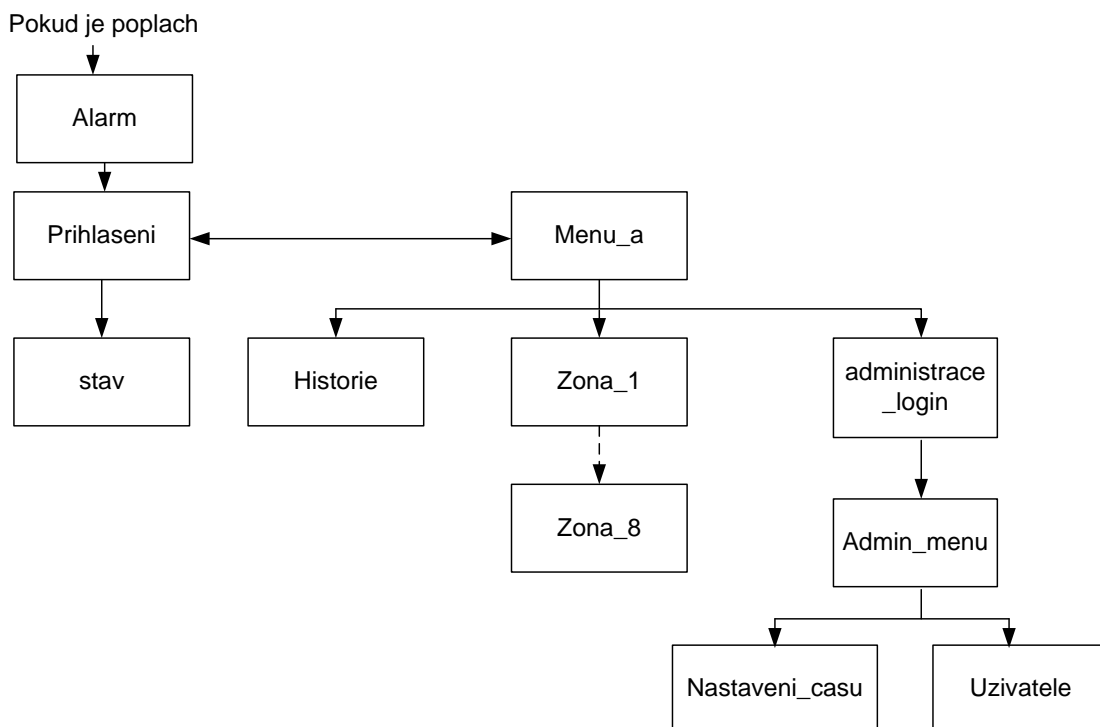
Ověřování kódu je v podprogramu proto, že se z procesů volá na více místech a šetří to tak v nich prostor a paměť PLC.

Pokud jsou splněny náležitě podmínky, je spuštěn algoritmus samotného ověření. Všechny uživatelské kódy jsou uloženy do matice `uzivatele` o velikosti [2,8]. Každý řádek je jeden uživatel s tím, že první je master. Omezení je, že uživatelský kód nesmí být roven 0. Uživatel s kódem master může měnit všechny kódy v administrátorském nastavení.

Samotné ověření probíhá v cyklu `for`, kdy je matice `uzivatele` testována na rovnost se zadaným kódem. Pokud je tato podmínka splněna, tak je to vyhodnoceno jako správný kód.

## 6.4 Obrazovky

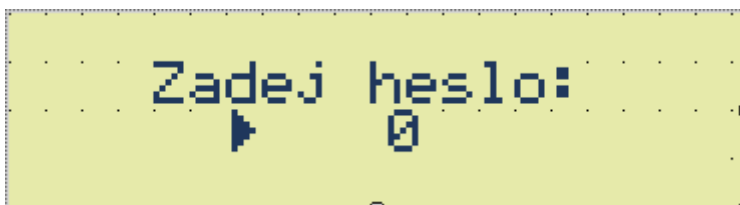
Vizualizace sestává z kolekce obrazovek, ve kterých jsou ovládací prvky. Vzájemné přechody mezi obrazovkami jsou pak řízeny pomocí vlastností některých prvků nebo pomocí skriptů. Obrazovky jsou navrženy tak, aby byly přehledné, uživatelsky příjemné a přitom vystihly svůj význam.



Obrázek 13: Schéma sledu obrazovek

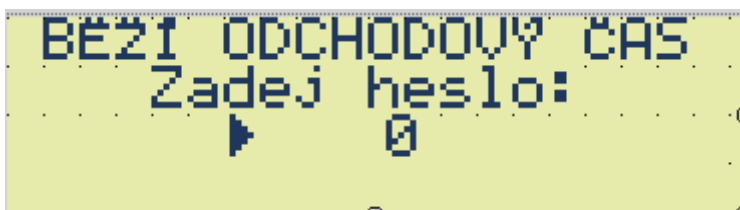
### 6.4.1 Přihlášení:

Obrazovka Přihlášení je nejdůležitější a nejpoužívanější obrazovka. Uživatel na ní zadává svůj kód.



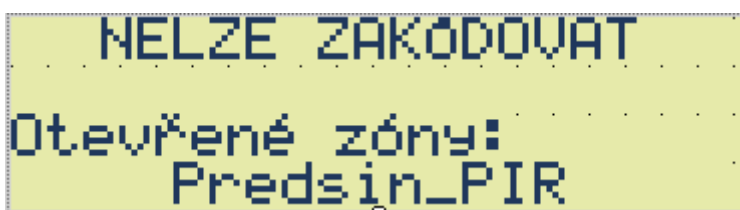
Obrázek 14: Výzva uživateli pro zadání kódu

Po úspěšném zadání, je uživatel informován, že mu běží čas pro odchod. Samotný čas zde zobrazen není kvůli přehlednosti.



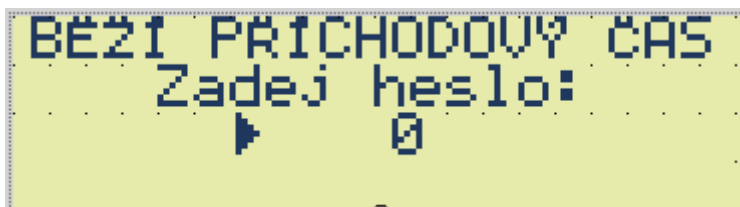
Obrázek 15: Obrázovka po zakódování

Pokud by některá zóna byla otevřená před zakódováním, uživatel je na to upozorněn. Toto upozornění se netýká zón, které uživatel nechce zakódovat.



Obrázek 16: Upozornění na otevřené zóny

Při příchodu do zabezpečeného objektu je uživatel vyzván, aby zadal svůj kód a na obrazovce je informován, že mu běží příchodový čas.



Obrázek 17: Obrázovka po příchodu do zabezpečené budovy

### Skript:

```
event Prihlaseni_OnOpen ()
    Prihlaseni.FocusFirstControl ();
end;

event Prihlaseni_OnRefresh ()

    if @pozadavek == true then
        Label1.Visible = true; //zadej heslo:
        NumericEdit1.Visible = true; // pole pro zadání hesla
        Label2.Visible = false; //NELZE ZAKODOVAT
        Label3.Visible = false; //Otevřené zóny:
        Alarm1.Visible = false; //Seznam zón, které jsou otevřené
    else
        Label1.Visible = false;
```

```

        NumericEdit1.Visible = false;
        Label2.Visible = true;
        Label3.Visible = true;
        Alarm1.Visible = true;
    endif;

    if @prichod == true then
        prihlas_text = 1; //Běží příchodový čas
    endif;

    if @prisel_t == true then
        stav.Show(); //zobrazení obrazovky stav
        prihlas_text = 0;
        @prisel_t = false;
    endif;

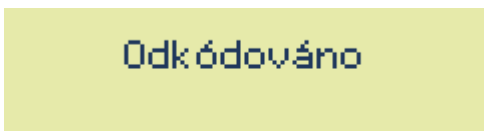
    if @odchod == true then
        prihlas_text = 2; //Běží odchodový čas
    endif;

    if (@odesel_t or @odchod_zrus) == true then
        stav.Show();
        prihlas_text = 0;
        @odesel_t = false;
    endif;

end;

```

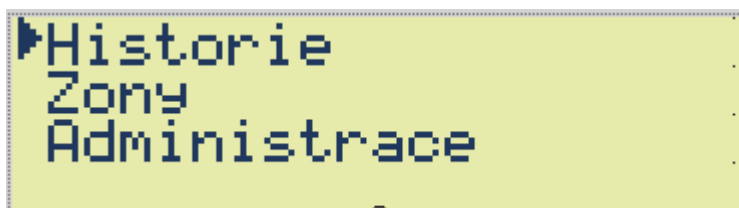
Po uplynutí odchodového času, nebo zadání správného kódu po příchodu se uživateli zobrazí obrazovka se stavem systému. Z té se automaticky systém přesune po 10 vteřinách zpět na obrazovku s přihlášením.



Obrázek 18: Stav systému: Odkódováno

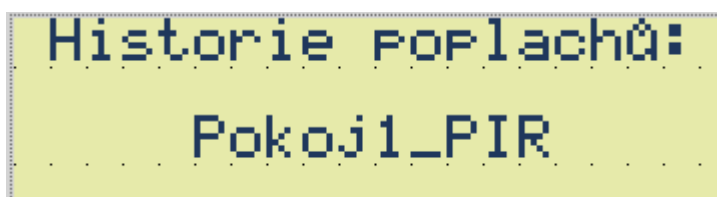
#### 6.4.2 Menu\_a:

Pokud uživatel na obrazovce Přihlášení stiskne ESC, dostane se na obrazovku s nabídkou nastavení, historie poplachů a administrátorským přístupem.



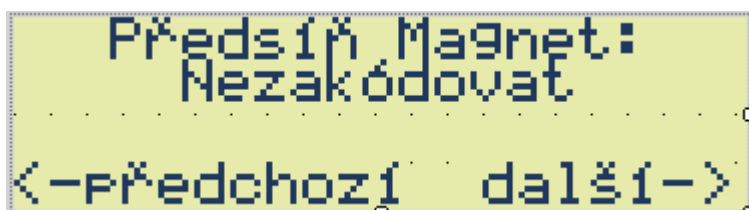
Obrázek 19: Obrazovka Menu\_a

Pokud bude pokračovat na historii, dostane se na obrazovku která zobrazuje zóny narušené při posledním poplachu.



**Obrázek 20: Obrazovka s historií poplachů**

Pokud uživatel vybere volbu Zóny, naskytne se mu příležitost výběru zón, které mají být při dalším zakódování zabezpečeny. Po odkódování se defaultně výběr nastaví na všechny zóny. Mezi jednotlivými zónami se pohybuje pomocí šipek doprava a doleva a zpět na menu se dostane pomocí klávesy ESC. Obrazovky se zónami na sebe navazují v kruhu.



**Obrázek 21: Ukázka obrazovky s výběrem zóny pro zabezpečení**

Pokud uživatel v menu vybere možnost Administrace, dostává se do sekce, kam může vstoupit pouze uživatel s master kódem.

### 6.4.3 Administrace

Administrátor, neboli uživatel s master kódem bývá majitel budovy, nebo její správce. Master kód je v matici uživatelských kódů na pozici: Uzivatele[0,1]. Po vstupu z menu je uživatel vyzván, aby zadal master kód.



**Skript:**

```
event administrace_login_OnOpen()  
  administrace_login.FocusFirstControl();  
  @admin_log = true; //příznak pro podprogram overeni kodu (nejedna se  
o zakodování - neověřuj kód)  
  Admin_log_er = 0; //pomocná proměnná pro identifikaci  
špatnězadaného kódu administrátora  
end;
```

```

event NumericEdit1_OnEditComplete()
    if (Kod_Admin != master) then //pokud se kódy neshodují, přičti 1
        Admin_log_er = Admin_log_er + 1;
        Kod_Admin = 0; // vynulování pole pro zadání kódu
    endif;
end;

event administrace_login_OnRefresh()

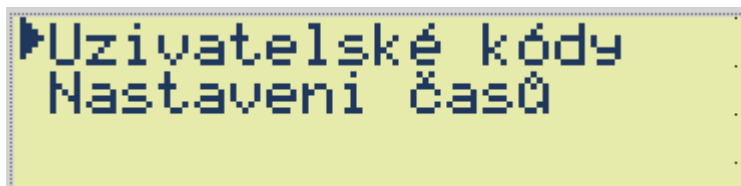
    if (Kod_Admin == master) and (Kod_Admin > 0) then
        admin_menu.Show(); //po zadání správného kódu se zobrazí menu
administrátora
        @admin_log = false;
    else

        if Admin_log_er == 3 then //pokud je kód třikrát špatně
zadán, systém přejde na obrazovku Přihlášení
            Prihlaseni.Show();
            Admin_log_er = 0;
            @admin_log = false;
        endif;
    endif;
end;

```

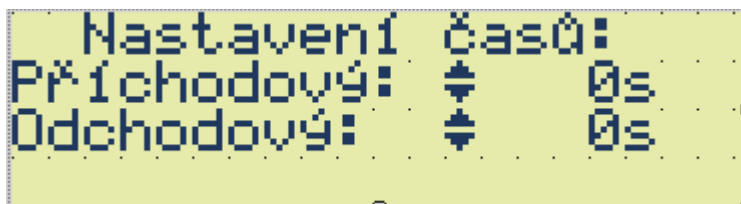
**Obrázek 22: Výzva k zadání master kódu**

Po úspěšném zadání, se dostane do menu administrátora.

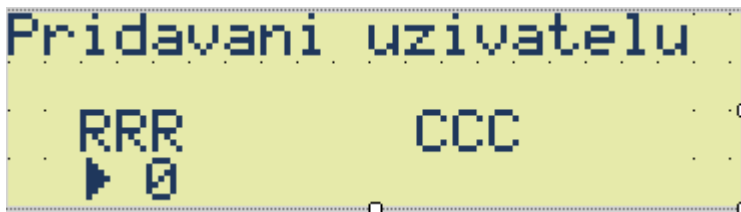


**Obrázek 23: Obrazovka s menu administrátora**

Zde si může vybrat, jestli chce měnit uživatelské kódy, nebo délku příchodových či odchodových časů. Uživatelské kódy se nastavují přímo do matice modulem NumericMtxEdit.



**Obrázek 24: Obrazovka s nastavením příchodových a odchodových časů**



**Obrázek 25: Obrazovka s editací uživatelských kódů**

#### 6.4.4 Alarm

Důležité je také graficky upozornit na poplach. To je zajištěno obrazovkou Alarm. Pokud na ní uživatel stiskne ESC, okamžitě se dostane na obrazovku Přihlášení, aby mohl svým kódem poplachový stav zastavit.



Obrázek 26: Alarmová obrazovka

#### Skript:

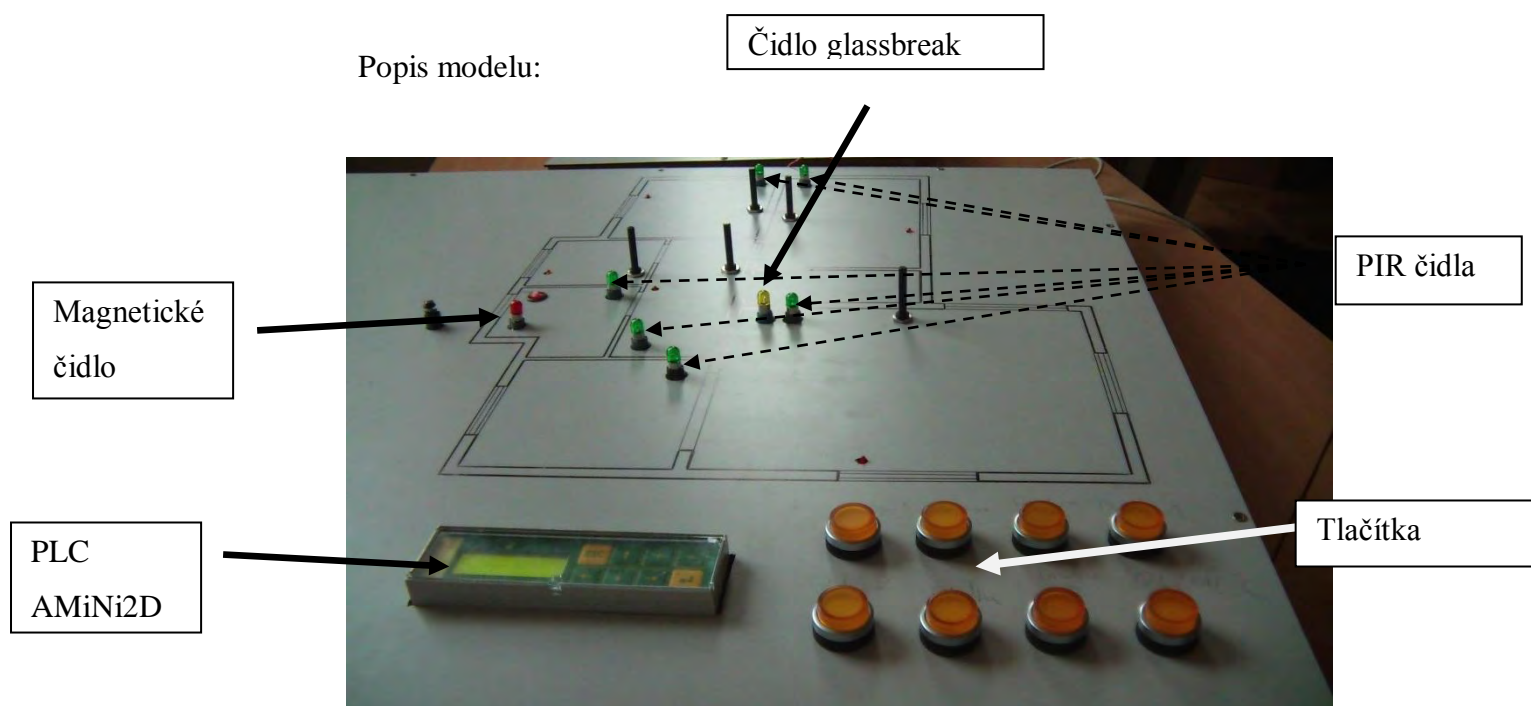
```
event Alarm_OnOpen()  
    Alarm.FocusFirstControl();  
end;  
  
event Key1_OnKeyPress()// při zmačknutí ESC, se poplach vykvituje a  
prepne se na Prihlaseni  
    poplach_det = 0;  
    Prihlaseni.Show();  
end;
```

## 6.5 Modelové řešení

Veškerá čidla jsou nahrazena kombinací tlačítek a LED signalizací, kvůli ekonomické nenáročnosti. V podstatě tak přesně nahrazují detektory používané v realitě. Jejich výstupy totiž jsou bezpotenciálové spínací kontakty. Detektor neustále detekuje a vysílá ústředně pouze logický signál (nulu, nebo jedničku). Je jen na ústředně, jestli vyhlásí poplach, nebo ne.

LED diody jsou osazeny v modelu domu podle rozmístění čidel a tlačítka jsou realizována na periférii domu společně s PLC AMiNi2D.

Popis modelu:



## 7 Závěr

Tato práce se i díky tlaku a metodickému vedení konzultantů nakonec zdárně podařila realizovat. Se spoluřešitelem jsme museli čelit chybám v komunikaci, ale nakonec jsme se naučili spolupracovat. I když jsme řešili každý vlastní problémy, snažili jsme si navzájem pomáhat a vzájemně si vycházet vstříc.

Byl sestaven fungující model, který zůstane ve škole a bude sloužit pro výuku budoucích studentů navštěvující kroužek AMiT.

Když si zpětně vzpomeneme na počáteční představu, jak by měl projekt vypadat, je to až úsměvné. Představovali jsme si to, jak se říká, jako Hurvínek válku. I přes naše nabyté vědomosti a schopnosti, které jsme postupem času získávali a v projektu zúročili, se náš projekt blíží reálným systémům jen velmi vzdáleně. Dnes litujeme toho, že jsme kroužek nenavštěvovali již loňský rok. Ztratili jsme drahocenný čas, čímž i cenné zkušenosti, které jsme mohli letos využít a projekt více rozvinout.

Tento projekt nám dal také určité směrnice, jak přistupovat k problémům a jejich řešení. Dále nás proškolil ve vyhledávání a zjišťování informací a v neposlední řadě také nebát se zeptat, když něco nevíme.

Práce s PLC přístroji, navrhování a realizace automatických systémů nás chytla natolik, že jsme se utvrdili ve správnosti volby oboru automatizace a doufáme, že se v budoucnu budeme moci dále zdokonalit v problematice programování programovatelných automatů.

## 8 Seznam použité literatury

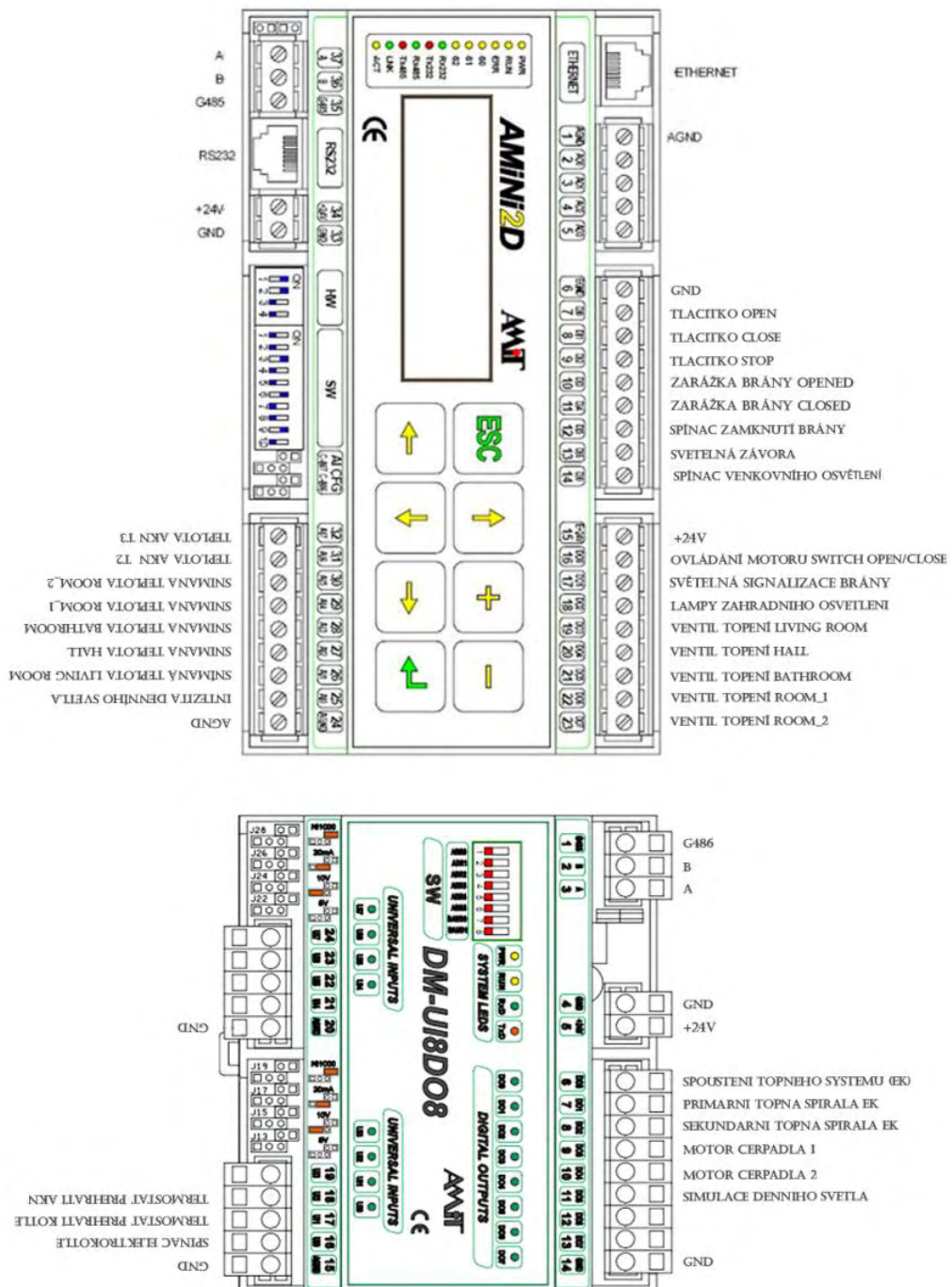
- [1] *Www.itdum.cz* Co je inteligentní dům?  
<http://www.itdum.cz/co-je-inteligentni-dum.html>
- [2] Křeček, P.: *Průručka zabezpečovací techniky*, 4. vydání; Cricetus, 350 stran
- [3] Skřivan Z.a kol.: *Nebojte se zlodějů*, Grada, 216 stran
- [4] Ing Jiří Kindl: *Projektování bezpečnostních systémů 1.díl*, Univerzita Tomáše Bati ve Zlíně, 134 stran
- [5] *Www.jablotron.cz* Zabezpečení objektů Jablotron.  
<http://zabezpeceni-objektu.jablotron.cz/>
- [6] *Www.euroalarm.cz* Zabezpečovací systémy  
<http://www.euroalarm.cz/zabezpecovacitechnika/zabezpecovaci-systemy/>
- [7] *Www.eurosat.cz*
- [8] <http://www.velasot.cz/>
- [9] ] <http://cs.wikipedia.org/> Inteligentní dům  
[http://cs.wikipedia.org/wiki/Inteligentn%C3%AD\\_d%C5%AFm](http://cs.wikipedia.org/wiki/Inteligentn%C3%AD_d%C5%AFm)
- [10] <http://www.inteligentni-byt.cz/>
- [11][http://cs.wikipedia.org/Elektronický zabezpečovací systém](http://cs.wikipedia.org/Elektronick%C3%BD_zabezpe%C4%8Dovac%C3%AD_syst%C3%A9m)  
[http://cs.wikipedia.org/wiki/Elektronick%C3%BD\\_zabezpe%C4%8Dovac%C3%AD\\_syst%C3%A9m](http://cs.wikipedia.org/wiki/Elektronick%C3%BD_zabezpe%C4%8Dovac%C3%AD_syst%C3%A9m)
- [12] Jiří Valter: *Regulace v praxi aneb jak to dělám já, Ben*
- [13] [www.amit.cz](http://www.amit.cz), typová řešení  
<http://www.amit.cz/cz/adoreg/typapl.htm>
- [14] Alpeco: *Hlavice na topení katalog*

## 9 Příloha

### 9.1 Termoregulace, brána, osvětlení

Zapojení vstupů a výstupů pro termoregulaci, automatickou bránu, osvětlení.

#### 9.1.1 Vstupy, výstupy



## 9.1.2 Databázové proměnné

### Seznam databázových proměnných a jejich význam

Jméno	Typ	Řád	Sloupců	WID	Warn	Init	Stanice	Komentář
AK_prom	I			3007	<input type="checkbox"/>		3	pomocná proměnná pro akumulační nádrž
ari_di	I			3016	<input type="checkbox"/>		3	digitální vstupy ze vzdálených vstupů zkrze sít' ARION
ari_out	I			3017	<input type="checkbox"/>		3	bin výstupy na vzdálených výstupech a vstupech
C2vykon	I			3024	<input type="checkbox"/>		3	výkon čerpadla
D_Times	ML	8	5	3028	<input type="checkbox"/>		3	nastavení časových zlomů pro denní plán
EK_prom	I			3005	<input type="checkbox"/>		3	pomocná proměnná pro elektrokotel
Fotorez_An	F			3000	<input type="checkbox"/>		3	intenzita světla z fotorezistoru_analogovy vstup
g_period	F			3002	<input type="checkbox"/>		3	počítání period procesu pro automatické zavření brány
Gate	I			3026	<input type="checkbox"/>		3	Proměnná pro ovládání brány
Gate2	I			3003	<input type="checkbox"/>		3	proměnná pro ovládání brány
LIGHT	I			3001	<input type="checkbox"/>		3	proměnná pro ovládací prvky zahradního osvětlení
PT_bath	F			3011	<input type="checkbox"/>		3	Teplota na prostorovém termostatu v koupelně
PT_hall	F			3010	<input type="checkbox"/>		3	Teplota na prostorovém termostatu v předsíni
PT_liv	F			3009	<input type="checkbox"/>		3	Teplota na prostorovém termostatu v obývacím pokoji
PT_r1	F			3012	<input type="checkbox"/>		3	Teplota na prostorovém termostatu v místnosti 1
PT_r2	F			3013	<input type="checkbox"/>		3	Teplota na prostorovém termostatu v místnosti 2
R_Regprom	I			3008	<input type="checkbox"/>		3	pomocná proměnná pro regulaci místností
Room_reg	I			3018	<input type="checkbox"/>		3	pomocná proměnná pro individuální regulaci místností
sc_g_time	I			3027	<input type="checkbox"/>		3	screen gate time-čas nastavený na obrazovce pro automatické zavření brány
speed	F			3004	<input type="checkbox"/>		3	proměnná pro konfiguraci rychlosti chodu motoru
T_ak_poz	I			3006	<input type="checkbox"/>		3	Teplota požadovaná v akumulační nádrži
T2	F			3014	<input type="checkbox"/>		3	Teplota snímaná u vrchu akumulační nádrže(nádrž jako zdroj tepla)
T3	F			3015	<input type="checkbox"/>		3	teplota snímaná u dna akumulační nádrže(nádrž jako spotřebič)
temrez_var	I			3034	<input type="checkbox"/>		3	proměnná pro režim vytápění
Tlak_reg	I			3025	<input type="checkbox"/>		3	regulace tlaku v topném systému
Tpoz_bath	I			3021	<input type="checkbox"/>		3	požadovaná teplota v koupelně
Tpoz_hall	I			3020	<input type="checkbox"/>		3	požadovaná teplota v předsíni
Tpoz_liv	I			3019	<input type="checkbox"/>		3	požadovaná teplota v obývacím pokoji
Tpoz_r1	I			3022	<input type="checkbox"/>		3	požadovaná teplota v místnosti 1
Values_bath	ML	8	5	3031	<input type="checkbox"/>		3	hodnoty pro dayplan koupelny
Values_hall	ML	8	5	3030	<input type="checkbox"/>		3	hodnoty pro dayplan předsíně
Values_liv	ML	8	5	3029	<input type="checkbox"/>		3	hodnoty pro dayplan obývacího pokoje
Values_r1	ML	8	5	3032	<input type="checkbox"/>		3	hodnoty pro dayplan pokoje 1
Values_r2	ML	8	5	3033	<input type="checkbox"/>		3	hodnoty pro dayplan pokoje 2
Var1	I			3035	<input type="checkbox"/>		3	

## 9.1.3 Aliasy

Seznam použitých aliasů a jejich význam:

Alias	Proměnná	Bit	Komentář
@ac_run	Gate2	3	beh programu pro automatické zavření brány
@auto_close	Gate2	2	automatické zavírání brány
@closed0	Tlak_reg	5	zavřeno 0 ventilů
@closed1	Tlak_reg	4	zavřen 1 ventil
@closed2	Tlak_reg	3	zavřeny 2 ventily
@closed3	Tlak_reg	2	zavřeny 3 ventily
@closed4	Tlak_reg	1	zavřeny 4 ventily
@closed5	Tlak_reg	0	zavřeno 5 ventilů
@dayplan	Room_reg	3	mod vytápění denní plán
@Ek_pswitch	ari_di	0	Spínač elektrokotle
@Ek_run	ari_out	0	chod elektrokotle a topného systému
@eth_1	R_Regprom	0	Elektrottermická hlavice dvoucestného ventilu topení místnost 1
@eth_2	R_Regprom	1	Elektrottermická hlavice dvoucestného ventilu topení místnost 2
@eth_3	R_Regprom	2	Elektrottermická hlavice dvoucestného ventilu topení místnost 3
@eth_4	R_Regprom	3	Elektrottermická hlavice dvoucestného ventilu topení místnost 4
@eth_5	R_Regprom	4	Elektrottermická hlavice dvoucestného ventilu topení místnost 5
@komfort	Room_reg	0	mod vytápění místností
@L_spinac	LIGHT	5	spínač manuálního ovládání osvětlení
@LED	LIGHT	3	výstupní hodnota pro samotné osvětlení zahrady
@LED_sig	Gate2	1	alias pro výstup na majáček světelné signalizace
@mod15	Gate2	4	nastavení automatického zavření brány na 15s
@mod30	Gate2	5	nastavení automatického zavření brány na 30s
@mod45	Gate2	6	nastavení automatického zavření brány na 45s
@mod60	Gate2	7	nastavení automatického zavření brány na 60s
@open_close	Gate	10	přepínač stavů relé (otevírání brány/zavírání brány)
@opz_int	Gate	13	Přerušení světelné závory
@prekazka	Gate	15	nežádoucí objekt v cestě při zavírání brány
@prim_tspir	ari_out	1	Ovládání primární topné spirály EK
@rezim_A	LIGHT	2	automatický režim ovládání osvětlení
@rezim_M	LIGHT	1	manuální režim ovládání osvětlení
@Run_C1	ari_out	3	motor čerpadla C1 osazeného na topné vodě za EK
@Run_C2	ari_out	4	motor čerpadla C2 osazeného na topné vodě za AKN
@sec_tspir	ari_out	2	ovládání sekundární topné spirály EK
@sig	Gate	12	signalizační světlo
@simsun	LIGHT	4	simulace denního světla
@Stav_closed	Gate	6	stav brány zavřeno
@Stav_closin	Gate	4	běh motoru pro zavírání brány
@Stav_opened	Gate	5	stav brány otevřeno
@Stav_openin	Gate	7	běh motoru pro otevírání brány
@stav_Opint	Gate	14	Stav přerušení světelné závory
@Stav_stop	Gate	8	stav nouzového zastavení brány
@stop_state	Gate	11	indikace nouzového stopnutí během
@T_close	Gate	1	Tlačítko close
@T_open	Gate	0	Tlačítko close
@T_stop	Gate	9	tlačítko stop
@T1	ari_di	1	termostat přehřátí kotle
@T4	ari_di	2	termostat přehřátí akumulační nádrže
@temperace	Room_reg	2	mód vytápění místností
@tma	LIGHT	0	mez intenzity světla pro tmu
@unlock	Gate2	0	Manuální uzamčení brány
@utlum	Room_reg	1	mód vytápění místností
@Z_close	Gate	3	Zarážka zavřeno
@Z_open	Gate	2	Zarážka otevřeno

## 9.1.4 Program

### Výpis programu:

Hlavní proces:

Main	ST	Normal_0	1000	0	Hlavní proces
------	----	----------	------	---	---------------

```
//TERMOREGULACE

ARI_DigIn 2, 0, ari_di, 0x0000007 //negované signály 0,1,2
AnIn #AI00_1, PT_liv, 10.000, 0.000, 10.000, 15.000, 25.000
//prostorový termostat obývacího pokoje
AnIn #AI00_2, PT_hall, 10.000, 0.000, 10.000, 15.000, 25.000
//prostorový termostat předsíně
AnIn #AI00_3, PT_bath, 10.000, 0.000, 10.000, 15.000, 25.000
//prostorový termostat koupelny
AnIn #AI00_4, PT_r1, 10.000, 0.000, 10.000, 15.000, 25.000
//prostorový termostat pokoje 1
AnIn #AI00_5, PT_r2, 10.000, 0.000, 10.000, 15.000, 25.000
//prostorový termostat pokoje 2
AnIn #Ni10001_6, T2, 10.000, 0.000, 10.000, 15.000, 95.000
//snímání teploty u vrchu akn
AnIn #Ni10001_7, T3, 10.000, 0.000, 10.000, 15.000, 95.000
//snímání teploty u dna akn

//ELEKTROKOTEL
//elektrokotel se zapne při sepnutí vypínače
//pokud sepne buďto termostat přetopení kotle T1 nebo
termostat přetopení Akumulační nádrže...
//...kotel bude nouzově vypnut
let @Ek_run=if(@Ek_pswitch,true,false)
let @Ek_run=if((@T1) or (@T4)),false,@Ek_run)

if @Ek_run

//primární spirála pojede pokud budeme chtít
dosáhnout požadované
//teploty v akumulaci nádrži
```

```

Let @prim_tspir=if((T_ak_poz>T3),true,false)

//sekundární spirála se bude spínat, pokud bude
rozdíl požadované teploty
//a teploty skutečné větší než 30stupňů-pro
rychlejší vytopení Akn

Let @sec_tspir=if((T_ak_poz-T3)>30),true,false)
else
Let @sec_tspir=false
let @prim_tspir=false
endif

//ČERPADLO PRIMÁRNÍHO(KOTEL) TOPNÉHO OKRUHU (Č1)
//čerpadlo poběží, pokud budeme chtít vytopit akumulární
nádrž na požadovanou hodnotu
//čerpadlo poběží i v příladě, že se nouzově vypne
Elektrokotel kvůli přetopení...
//...snímaném termostatem T1 kvůli vychlazení kotle

If @Ek_run
Let @Run_C1=if((T_ak_poz>T3)or(@T1)),true,false)
Let @Run_C1=if((T_ak_poz<T3),false,@Run_C1)
Else
Let @Run_C1=false
endif

//ČERPADLO SEKUNDÁRNÍHO TOPNÉHO OKRUHU (Č2)

//regulace výkonu v závislosti na počtu zavřených ventilů
Let @closed0=if((R_Regprom==0),true,false)
let
@closed1=if((R_Regprom==1)or(R_Regprom==2)or(R_Regprom==4)or(R_Regprom==
8)or(R_Regprom==16)),true,false)
let
@closed2=if((R_Regprom==3)or(R_Regprom==5)or(R_Regprom==6)or(R_Regprom==
9)or(R_Regprom==10)or(R_Regprom==12)or(R_Regprom==17)or(R_Regprom==18)or(
R_Regprom==20)or(R_Regprom==24)),true,false)
let
@closed3=if((R_Regprom==7)or(R_Regprom==11)or(R_Regprom==13)or(R_Regprom

```

```

==14) or (R_Regprom==19) or (R_Regprom==21) or (R_Regprom==22) or (R_Regprom==25)
or (R_Regprom==26) or (R_Regprom==28) ), true, false)
    let
@closed4=if((R_Regprom==15) or (R_Regprom==23) or (R_Regprom==27) or (R_Regpro
m==29) or (R_Regprom==30) ), true, false)
    let @closed5=if((R_Regprom==31) , true, false)

//výkon čerpadla C2 v závislosti na počtu zavřených ventilů
//všechny otevřené
if @closed0
    let C2vykon=5
endif
//jeden zavřený
if @closed1
    let C2vykon=4
endif
//2 zavřené
if @closed2
    let C2vykon=3
endif
//3 zavřené
if @closed3
    let C2vykon=2
endif
//4 zavřené
if @closed4
    let C2vykon=1
endif

//čerpadlo poběží, pokud bude teplota snímaná odpovídat
teplotě požadované
//pokud bude teplota menší, než požadovaná teplota, čerpadlo
vypne
//pokud budou zavřeny všechny ventily, nebo nebudou tepelné
požadavky, čerpadlo vypne
if @Ek_run

    let @Run_C2=if((T_ak_poz<=T2) , true, @Run_C2)
    let
@Run_C2=if(((T_ak_poz>T2) or (@closed5) ), false, @Run_C2)
else

```

```

        let @Run_C2=false
    endif

//VYTÁPĚNÍ MÍSTNOSTÍ

//denní plán

//obývací pokoj
    DayPlan 0x0001, 1, 0x007F, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, NONE, D_Times, Values_liv, Tpoz_liv

//předsíň
    DayPlan 0x0000, 1, 0x007F, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, NONE, D_Times, Values_hall, Tpoz_hall

//koupelna
    DayPlan 0x0000, 1, 0x007F, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, NONE, D_Times, Values_bath, Tpoz_bath

//pokoj 1
    DayPlan 0x0001, 1, 0x007F, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, NONE, D_Times, Values_r1, Tpoz_r1

//pokoj 2
    DayPlan 0x0000, 1, 0x007F, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, NONE, D_Times, Values_r2, Tpoz_r2

//nastavení režimu vytápění
    Switch termrez_var
        Case 0
            let @temperace = true
        endcase

        Case 1
            let @utlum = true
        endcase

```

```

    Case 2
        let @komfort = true
    endcase

    Case 3
        let @dayplan = true
    endcase

endswitch

//Temperace:
//je požadováno 50 stupňů v akumulární nádrži
//pokud je nastavena temperace budou se místnosti vytápět na 16
stupňů
    if @temperace

        let T_ak_poz = 50
        let @eth_1=if((PT_liv>16), true, false)
        let @eth_2=if((PT_hall>16), true, false)
        let @eth_3=if((PT_bath>16), true, false)
        let @eth_4=if((PT_r1>16), true, false)
        let @eth_5=if((PT_r2>16), true, false)

    endif

//Útlum:
//je požadováno 70 stupňů v akumulární nádrži
//pokud je nastaven režim útlum, budou se místnosti vytápět na 19
stupňů
    if @utlum

        let T_ak_poz = 70
        let @eth_1=if((PT_liv>19), true, false)
        let @eth_2=if((PT_hall>19), true, false)
        let @eth_3=if((PT_bath>19), true, false)
        let @eth_4=if((PT_r1>19), true, false)
        let @eth_5=if((PT_r2>19), true, false)

    endif

```

```

//Komfort:
//je požadováno 90 stupňů v akumulární nádrži
//při režimu komfort si lze zvolit teplotu pro každou místnost od
20 do 25 stupňů

```

```

if @komfort

```

```

    let T_ak_poz = 90
    let @eth_1=if((PT_liv>Tpoz_liv), true, false)
    let @eth_2=if((PT_hall>Tpoz_hall), true, false)
    let @eth_3=if((PT_bath>Tpoz_bath), true, false)
    let @eth_4=if((PT_r1>Tpoz_r1), true, false)
    let @eth_5=if((PT_r2>Tpoz_r2), true, false)

```

```

endif

```

```

//Denní plán:
//je požadováno 90 stupňů v akumulární nádrži
//při režimu denní plán si lze zvolit teplotu pro každou místnost
od 20 do 25 stupňů
//v různých časových zlomech během dne

```

```

if @dayplan

```

```

    let T_ak_poz = 90
    let @eth_1=if((PT_liv>Tpoz_liv), true, false)
    let @eth_2=if((PT_hall>Tpoz_hall), true, false)
    let @eth_3=if((PT_bath>Tpoz_bath), true, false)
    let @eth_4=if((PT_r1>Tpoz_r1), true, false)
    let @eth_5=if((PT_r2>Tpoz_r2), true, false)

```

```

endif

```

```

// VÝSTUPY

```

```

ARI_DigOut 2, 0, 8, ari_out, 0x00000000 //výstupy na
vzdálených výstupech a vstupech

```

```

BinOut @eth_1, 0x0000, #D000_3 //ventil v obývacím
pokoji

```

```

BinOut @eth_2, 0x0000, #D000_4 //ventil v předsíni

```

```

BinOut @eth_3, 0x0000, #D000_5 //ventil v koupelně
BinOut @eth_4, 0x0000, #D000_6 //ventil v pokoji 1
BinOut @eth_5, 0x0000, #D000_7 //ventil v pokoji 2

//AUTOMATICK8 BRÁNA

// proces pro světelnou signalizaci a nastavení časových intervalů
pro automatické zavírání
//světelná signalizace
if @sig

    let @LED_sig = not @LED_sig
    else
    let @LED_sig = false
endif
//nastavení časových intervalů

Let @mod15=if(sc_g_time==15,true,false)
Let @mod30=if(sc_g_time==30,true,false)
Let @mod45=if(sc_g_time==45,true,false)
Let @mod60=if(sc_g_time==60,true,false)

//pokud je povoleno automatické zavírání:
if @ac_run
    //je nastaveno 15vteřin
    If @mod15

        IncDec g_period, @ac_run, 1.000 //inkrement pocítání
periody při každém průchodu procesu
        //pokud 15tá perioda(15sec)
        let @stav_opened=if(g_period==15,false,@stav_opened)
        let @stav_closin=if(g_period==15,true,@stav_closin)

        if @stav_closin
            let g_period = 0 //nulování čítače
        endif

        if @opz_int //nulování čítače pokud se
přeruší světelná závora
            let g_period = 0

```

```

endif

endif
//je nastaveno 30 vteřin viz mod15
If @mod30

IncDec g_period, @ac_run, 1.000

let @stav_opened=if(g_period==30,false,@stav_opened)
let @stav_closin=if(g_period==30,true,@stav_closin)

if @stav_closin
    let g_period = 0
endif

if @opz_int
    let g_period = 0
endif

endif
//je nastaveno 45 vteřin viz. mod 15
If @mod45

IncDec g_period, @ac_run, 1.000

let @stav_opened=if(g_period==45,false,@stav_opened)
let @stav_closin=if(g_period==45,true,@stav_closin)

if @stav_closin
    let g_period = 0
endif

if @opz_int
    let g_period = 0
endif

endif
//je nastaveno 60 vteřin viz. mod 15
If @mod60

IncDec g_period, @ac_run, 1.000

```

```

let @stav_opened=if(g_period==60,false,@stav_opened)
let @stav_closin=if(g_period==60,true,@stav_closin)

if @stav_closin
    let g_period = 0
endif

if @opz_int
    let g_period = 0
endif

endif
endif

//OSVĚTLENÍ

//načtení vstupů
AnIn #AI00_0, Fotorez_An, 10.000, 0.000, 10.000, 0.000, 10.000
//načtení intenzity světla z fotorezistoru
BinIn #DIO0_7, 0x0000, @L_spinac
//spínač venkovního osvětlení

Limits Fotorez_An, @tma, NONE.0, 0.000, 9.900, 0.000, 0x0000
//vymezení hodnoty intenzity denního světla pro tmu

let @rezim_M=if(@L_spinac,true,@rezim_M)
let @rezim_A=if(@L_spinac,false,@rezim_A)
//automatický režim bude vypnutý pokud se zapne spínač
let @simsun = true
//zapnutí simulace denního světla

if @rezim_A                                //automatický režim:
    let @rezim_M=false
        if @tma                                //pokud klesne intenzita
světla pod mez tmy
            let @LED = true                    //zapne se osvětlení
        else                                    //pokud bude nad mezí

```

```

        let @LED = false           //osvětlení bude vypnuté
    endif
endif

if @rezim_M

    if @L_spinac                   //při sepnutí spínače (vypne
se automatický režim):

        let @LED = true           //zapne se osvětlení
    else                           //pokud bude vypnutý

        let @LED = false         //osvětlení bude vypnuté

    endif
endif

// program funguje tak, že při spuštění je automatický režim
vypnutý,
// spustíme ho na obrazovce PLC, při poklesu světla pod mez tmy
námi definované,
// se automaticky spustí osvětlení. Pokud manipulujeme s osvětlením
manuálně,
// automatický režim se sám vypne

    BinOut @LED, 0x0000, #D000_3           //Výstup pro osvětlení
zahrady
    BinOut @simsun, 0x0000, #D000_4       //Výstup pro simulaci
denního světla
                                           //(pro simulaci programu ve
tmavém prostoru)

```

### Rychlý proces:

gatequick	ST	Quick	20	0	rychlý proces pro ovládání automatické brány
-----------	----	-------	----	---	--

```
//načtení vstupů
```

```

    BinIn #DIO0_3, 0x0000, @Z_open       //dorážecí tlačítko pro
otevřenou bránu

```

```

    BinIn #DIO0_4, 0x0000, @Z_close //dorážecí tlačítko pro
zavřenou bránu
    BinIn #DIO0_2, 0x0000, @T_STOP //tlačítko STOP na dálkovém
ovládání brány
    BinIn #DIO0_0, 0x0000, @T_open //tlačítko Otevřít na dálkovém
ovládání brány
    BinIn #DIO0_1, 0x0000, @T_close //tlačítko Zavřít na dálkovém
ovládání brány
    BinIn #DIO0_6, 0x0001, @opz_int //světelná závora (přerušeni)
    BinIn #DIO0_5, 0x0001, @unlock //zámek brány
//podmínky pro povolení stavů brány

    Let @Stav_opened=if(@Z_open, true, false)
//otevřeno-pouze při doražení brány na tlačítko otevřeno
    Let @Stav_openin=if(@T_open, true, @Stav_openin)
//otevřování-při zmáčknutí tlačítka otevřít
    Let @Stav_closed=if(@Z_close, true, false)
//zavřeno při doražení
    Let @Stav_closin=if(@T_close, true, @Stav_closin)
//zavírání
    Let @Stav_stop=if((@T_STOP) or (@prekazka), true, @Stav_stop)
//zastavení chodu

//podmínky pro zakázání stavů brány

    let
@Stav_openin=if((@t_stop) or (@t_close) or (@z_open), false, @stav_openin)
    let @Stav_closin=if((@t_stop) or (@T_open) or (@Z_close) or
(@prekazka), false, @stav_closin)
    let @Stav_stop=if((@T_open) or (@T_close), false, @Stav_stop)
    let @Stav_Opint=if((@T_open) or (@T_close), false, @Stav_Opint)

//pokud bude brána odemčena

    if @unlock

// stav brány: PLNĚ OTEVŘENA
// relé se přepne (přepólování motorku) brána je připravena na
zavírání
// signalizace vypnuta
    if @Stav_opened

```

```

        let @open_close=false
        let speed=0
        let @sig = false

        let @ac_run=if(@auto_close,true,false)
    endif

// stav brány: OTEVÍRÁNÍ
// signalizace běhu
    if @Stav_openin
        //pokud byla brána zastavena
        if @stop_state

                let @open_close=true
                let @stop_state=false
            endif

            let speed = 1
            let @sig = true
        endif

// stav brány: PLNĚ ZAVŘENA
// signalizace vypnuta
    if @Stav_closed

        let @open_close=true
        let speed = 0
        let @sig=false
        let @ac_run = false
    endif

// stav brány: ZAVÍRÁNÍ
// signalizace běhu
    if @Stav_closin

        if @opz_int

                Let @prekazka = true

            endif

```

```

//pokud byla brána zastavena
    if @stop_state

        let @open_close=false
        let @stop_state=false
    endif

    let speed = 1
    let @sig = true
endif

// stav brány: NOUZOVÉ ZASTAVENÍ TLAČÍTKEM
// běh se zastaví nastavením rychlosti do nuly

    if @Stav_stop

        let @prekazka = false
        let speed = 0
        let @sig = false
        let @stop_state = true           //indikace zastavení brány

    endif

// výstupy
endif

    BinOut @open_close, 0x0000, #D000_0           //výstup   na
relé přepínače
    BinOut @LED_sig, 0x0000, #D000_1
//signalizace
    AnOut #A000_1, speed, 10.000, 0.000, 10.000, 0.000, 10.000
//napájení motoru

```

## 9.2 Elektronický zabezpečovací systém

### 9.2.1 Databázové proměnné

Seznam použitých proměnných a jejich význam:

Jméno	▲	Typ	Řádků	Sloupců	WID	Wam	Init hodnota	Stanice	Komentář
Admin_log_er		I			7022	<input type="checkbox"/>		7	čítač špatně zadaného master kódu
cas_odchod		I			7012	<input type="checkbox"/>	15	7	doba odchodového času [ms]
cas_prichod		I			7011	<input type="checkbox"/>	15	7	doba prichodového času [ms]
delta_od		I			7018	<input type="checkbox"/>		7	uplynuly cas odchodu
delta_pri		I			7016	<input type="checkbox"/>		7	uplynuly cas prichodu
Diody		I			7001	<input type="checkbox"/>		7	
Hlidane		I			7002	<input type="checkbox"/>	0xFFFF	7	Výběr zon
i		I			7010	<input type="checkbox"/>		7	ridici promenna cyklu for, který overuje spravnost uzivatelskeho kodu
Kod_Admin		I			7021	<input type="checkbox"/>		7	overeni administratorskeho prava na obrazovce administrace_login
Kod_Zadany		I			7009	<input type="checkbox"/>		7	kod zadany uzivatelem na obrazovce
master		I			7023	<input type="checkbox"/>		7	master kód
Pomocna		I			7007	<input type="checkbox"/>		7	
pomocna2		I			7026	<input type="checkbox"/>		7	
Poplach		I			7005	<input type="checkbox"/>		7	
poplach_det		I			7014	<input type="checkbox"/>		7	poplach detekce pro zapis do historie
poplach_hist		I			7013	<input type="checkbox"/>		7	promenna zaznamenavajici historii poslednich poplachu
poplach_kvity		I			7019	<input type="checkbox"/>		7	kvityce poplachu pro obsluhu obrazovek
poplach_pri		I			7015	<input type="checkbox"/>	0xFFFC	7	
prihlas_text		I			7025	<input type="checkbox"/>		7	pomocná proměnná pro ovládání textů na přihlašovací obrazovce
prichod		I			7017	<input type="checkbox"/>		7	detekce prichodu do zakodovaneho objektu
spravny_t		I			7024	<input type="checkbox"/>		7	pomocná proměnná pro obsluhu obrazovek
Test		I			7003	<input type="checkbox"/>		7	
Tlacitko		I			7000	<input type="checkbox"/>		7	Aktualni
uzivatele		MI	8	2	7008	<input type="checkbox"/>	0,1234,1,9999,...	7	matice uzivatelskych kodu
word		MI	10	1	7020	<input type="checkbox"/>		7	
x		I			7006	<input type="checkbox"/>		7	
Zakodovano		I			7004	<input type="checkbox"/>		7	

## 9.2.2 Aliasy

Seznam použitých aliasů a jejich význam:

Alias	Proměnná	Bit	Komentář
@admin_log	pomocna2	0	rezervace logování pro administrátora (do nastavení)
@cnt_res1	Pomocna	3	vynulovani citace prichodoveho/odchodoveho casu
@cnt_res2	Pomocna	9	reset casovace odchodu
@err_alam	Pomocna	12	bit sinalizujici alam modulu ErrSig
@err_ignore	Pomocna	13	bit pro obsluhu ignorovani modulu ErrSig
@chodba	Hlidane	2	
@chodba_LED	Diody	0	
@chodba_pir	Tlacidko	2	
@kuchyn	Hlidane	5	
@kuchyn_LED	Diody	1	
@kuchyn_pir	Tlacidko	5	
@obavak1_LED	Diody	2	
@obavak_aku	Tlacidko	7	
@obavak_pir	Tlacidko	6	
@obavak1	Hlidane	6	
@obavak2	Hlidane	7	
@obavak2_LED	Diody	3	
@odesel	Pomocna	7	uplynul cas pro odchod
@odesel_t	Pomocna	14	pomocna promenna pro zobrazovani na displayi
@odchod	Pomocna	8	spousteni citace odchodoveho casu
@odchod_zrus	Pomocna	10	zruseni odchodoveho casu zadanim uzivatelskeho kodu
@pokoj1	Hlidane	3	
@pokoj1_LED	Diody	4	
@pokoj1_pir	Tlacidko	3	
@pokoj2	Hlidane	4	
@pokoj2_LED	Diody	5	
@pokoj2_pir	Tlacidko	4	
@poplach_end	Pomocna	4	konec poplachu po zadani uzivatelskeho kodu
@poplach_pri	Pomocna	5	poplach po uplynuti prichodoveho casu
@pozadavek	Pomocna	0	pozadavek na zakodovani
@predsin_mag	Tlacidko	0	
@predsin_pir	Tlacidko	1	
@predsin1	Hlidane	0	
@predsin1_LE	Diody	6	
@predsin2	Hlidane	1	
@predsin2_LE	Diody	7	
@prichod	prichod	2	prichod osoby dvemi do zakodovane budovy
@prichod_d	prichod	0	prichod osoby dvemi do zakodovane budovy
@prichod_res	prichod	1	reset indikace prichodu
@prisel	Pomocna	6	a zadal spravny kod
@prisel_t	Pomocna	15	pomocny bit pro vizualizaci
@rs_od_res	Pomocna	11	reset klopneho obvodu spoustejiciho cas odchodu
@spravny_kod	Pomocna	1	bit znacici ze zadany kod byl spravny
@zakodovano	Pomocna	2	bit znacici stav systemu - zakodovano/odkodovano

## 9.2.3 Program

Výpis programu:

**Proc00 :**

```
Proc00      ST      Normal_0 1000 0      Hlavní proces

// nacteni cidel
BinIn #DIO0_0, 0x0001, @predsin_mag
BinIn #DIO0_1, 0x0000, @predsin_pir
BinIn #DIO0_2, 0x0000, @chodba_pir
BinIn #DIO0_3, 0x0001, @pokoj1_pir
BinIn #DIO0_4, 0x0001, @pokoj2_pir
BinIn #DIO0_5, 0x0001, @kuchyn_pir
BinIn #DIO0_6, 0x0001, @obyvak_pir
BinIn #DIO0_7, 0x0000, @obyvak_aku

// kontrola pri zakodovani - dotaz na prazdnost zon
Let Test = (Tlacitko ^ Hlidane)

if @zakodovano //pokud nebude 1 tak muzu zakodovat
else
    let @pozadavek = if (Test == Hlidane, true, false) //
kdyz nebude narusena zona tak muzu zakodovat
endif

Call overeni_kodu

Let @poplach_end = if((Poplach > 0) and (@spravny_kod ==
true), true, false) // zruseni poplachu pri zadani spravneho kodu

If @poplach_end
    Let Kod_Zadany = 0
    Call overeni_kodu
EndIf

If @zakodovano
    Let Poplach = ((Tlacitko & Hlidane)& poplach_pri) //
zjisteni poplachu (nedetekuje magnet v predsini kvuli prichodovému
času - 0xFFFFE)
```

```

Else
    Let Poplach = 0
EndIf

BinDiff Poplach, poplach_det, 0xFFFF, 0x0000, 0x00000000 //
zaloha poplachu pro zobrazeni na obrazovkach
    let poplach_hist = if (poplach_det > 0, poplach_det,
poplach_hist)

If @spravny_kod //pokud je zadan spravny
kod:
    Let Kod_Zadany = 0
EndIf

Let master = uzivatele[0,1] //prirazeni master kodu z
matice do promenne - kvuli overovani master kodu ve skriptu

// diody vysviceni
BinOut @predsin_mag, 0x0000, #D000_0
BinOut @predsin_pir, 0x0000, #D000_1
BinOut @chodba_pir, 0x0000, #D000_2
BinOut @pokoj1_pir, 0x0000, #D000_3
BinOut @pokoj2_pir, 0x0000, #D000_4
BinOut @kuchyn_pir, 0x0000, #D000_5
BinOut @obyvak_pir, 0x0000, #D000_6
BinOut @obyvak_aku, 0x0000, #D000_7

```

### Proc01:

Proc01	ST	Normal_1	1000	0
--------	----	----------	------	---

```

If @zakodovano //vše co se vykoná po zakódování budovy

Let @odchod = false //ukončuje čítání
Let @cnt_res2 = true //reseet čítače
Let delta_od = 0 //hodnota čítače

If @poplach_pri //jestliže @poplach_pri po uplynuti
prichodoveho casu (po uplinutílynutí času čítače)

```

```

        Let @prichod = false //zakazuje čítání
        Let poplach_pri = 0xFFFF
    Else
        Let poplach_pri = 0xFFFC
        Let @prichod = @predsin_mag and @predsin_pir
//@prichod==true pokud platí @predsin a zároveň @predsin_pir, jinak
@prichod==false
        EndIf

        //jestliže zadáme správný kód a zároveň čítač nenačítal dobu
pro minimální dobu pro zadání kódu
        //tak @prisel==true -> znamená zadal správný kód, jinak
        Let @prisel = if((@spravny_kod == true) and (@poplach_pri ==
false) and (delta_pri > 2), true, false)

        If @prisel //jestliže je zadán správný kód tak vykonej tohle
            Let @zakodovano = false //značí zakódováno/odkódováno
            Let @prisel = false
            Let @prichod = false //zakazuje čítání protože kód je
správný vstup do budovy povolen
            Let @cnt_res1 = true //reset čítače
            Let delta_pri = 0 //hodnota čítače
            Let Hlidane = 0x00FF
            Let @prisel_t = true
        EndIf

        If @poplach_end //jestliže konec poplachu po zadani
uzivatelskeho kodu
            Let @zakodovano = false //značí zakódováno/odkódováno
            Let @poplach_pri = false //nulují poplach po uplynutí
příchodového času
            Let @cnt_res1 = true//reset čítače
        EndIf

        CntUpLv @prichod, @cnt_res1, cas_prichod, @poplach_pri,
delta_pri

        If @cnt_res1 //jestliže je bit resetu čítače nastaven na TRUE
tak
            Let @cnt_res1 = false //nulování resetu
        EndIf

```

```

Else

    If @rs_od_res
        Let @rs_od_res = false //nulování vstupního signálu do
klopného obvodu RS
    EndIf
    RS @spravny_kod, @rs_od_res, @odchod

    If @odchod_zrus
        Let @odchod = false //zakazu/povoluje čítání ->false
zakazuje

        Let @cnt_res2 = true //reset daného čítače
        Let delta_od = 0 //hodnota čítače se nastaví na nulu
    EndIf

    CntUpLv @odchod, @cnt_res2, cas_odchod, @odesel, delta_od

    If @odesel //jestliže odešel
        Let @zakodovano = true //zakóduje se
        Let @odchod = false
        Let @rs_od_res = true //aktivace vstupního signálu do
klopného obvodu RS
        Let @odesel = false
        Let @odesel_t = true
    EndIf

    If @cnt_res2 //reset čítače
        Let @cnt_res2 = false
    EndIf

    //jestliže zadám správný kód a zároveň hodnota čítače je
větší než 2 tak zruseni odchodoveho casu zadanim uzivatelskeho kodu,
jinak false
    Let @odchod_zrus = if((@spravny_kod == true) and (delta_od >
2), true, false)

EndIf

```