

Optimalizace řízení tří topných jednotek v rodinném domku

Optimization of three heating units
in detached house

Martin Krajča

Bakalářská práce
2005



Univerzita Tomáše Bati ve Zlíně
Fakulta technologická

Vložit oficiální zadání diplomové/bakalářské práce

Děkuji vedoucímu bakalářské práce panu Ing. Tomášovi Sysalovi, Ph.D. za odborné vedení, rady a připomínky, které mi při řešení mé práce poskytoval.

Prohlašuji, že jsem na celé diplomové práci pracoval/a samostatně a použitou literaturu jsem citoval/a.

Ve Zlíně, 30. 05. 2005

.....

jméno diplomanta

Abstrakt

Bakalářská práce se zabývá návrhem řízení ohřevu užitkové vody v rodinném domku pomocí tří topných jednotek. Řízení je navrženo pro mikropočítače řady I8051, konkrétně pro model Atmel AT89C2051, tak aby splňovalo požadavky zadané uživatelem. Bakalářská práce obsahuje rozbor konkrétního systému pro ohřev užitkové vody, softwarové a hardwarové řešení řízení daného systému.

Abstract

The bachelor work is dealing with the creation of regulation of three units which is used to water heating in detached house. Regulation is devised for microcomputers series I8051, in particular for type Atmel AT89C2051, so as to satisfy all the demands which were asking for user. The bachelor work contains analysis of specific heater system for heating supply water, software and hardware solution of this heater system.

OBSAH

ÚVOD.....	6
1 PŘEHLED SOUČASNÉHO STAVU PROBLEMATIKY.....	7
1.1 SLUNEČNÍ ABSORBÉRY.....	8
1.2 ELEKTRICKY VYHŘÍVANÝ BOJLER.....	11
1.3 PLYNEM VYHŘÍVANÝ BOJLER.....	13
1.4 SNÍMÁNÍ TEPLoty TEPLÉ UŽITKOVÉ VODY.....	15
1.5 REGULACE TEPLÉ UŽITKOVÉ VODY JEDNOTLIVÝCH OKRUHŮ.....	16
2 PROGRAMOVÉ ŘEŠENÍ DANÉ ÚLOHY.....	18
2.1 PODMÍNKY PRO ŘÍZENÍ DANÉ ÚLOHY.....	18
2.1.1 Podmínky pro serva tří topných jednotek.....	18
2.1.2 Podmínky pro ovládání cirkulačního čerpadla plynového okruhu.....	19
2.1.3 Podmínky pro ovládání čerpadla primárního solárního okruhu.....	19
2.2 NAVRŽENÍ GRAFICKÉ STRUKTURY ŘÍZENÍ.....	19
2.3 PRAVDIVOSTNÍ TABULKY A KARNAUGHOFovy MAPY.....	21
2.3.1 Pro tři zdroje tepla (Solár, Elektro, Plyn), automatické řízení a letní období.....	21
2.3.2 Pro tři zdroje tepla (Solár, Elektro, Plyn), automatické řízení a zimní období.....	22
2.3.3 Pro tři zdroje tepla (Solár, Elektro, Plyn), manuální ovládání a letní období.....	23
2.3.4 Pro tři zdroje tepla (Solár, Elektro, Plyn), manuální ovládání a zimní období.....	24
2.3.5 Pro řízení čerpadla primárního solárního okruhu.....	26
2.3.6 Pro řízení cirkulačního čerpadla plynového okruhu.....	27
2.4 SOFTWARE PRO PROGRAMOVÁNÍ MIKROPOČÍTAČŮ ŘADY I51.....	29
2.5 ŘÍDICÍ PROGRAM URČENÝ PRO MIKROPOČÍTAČE ŘADY I51.....	31
3 HARDWAROVÉ ŘEŠENÍ DANÉ ÚLOHY.....	37
3.1 VÝBĚR IDEÁLNÍHO MIKROPOČÍTAČE PRO ŘEŠENÍ DANÉ ÚLOHY:.....	37
3.1.1 Parametry.....	38
3.1.2 Popis.....	39
3.1.3 Zapojení vývodů mikropočítače.....	39
3.2 SCHÉMA ELEKTRONICKÉHO ZAPOJENÍ A NÁVRH PLOŠNÉHO SPOJE.....	42
3.2.1 Software pro návrh elektronického zapojení a plošného spoje.....	42
3.2.2 Schéma elektronického zapojení.....	43
3.2.3 Návrh plošného spoje.....	46
3.2.4 Seznam použitých elektronických součástek.....	47
3.3 OVLÁDACÍ PANEL.....	48
ZÁVĚR.....	51
SEZNAM POUŽITÝCH ZDROJŮ.....	52
SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....	53
SEZNAM OBRÁZKŮ.....	54
SEZNAM TABULEK.....	55
SEZNAM PŘÍLOH.....	56

ÚVOD

Cílem bakalářské práce je návrh řízení ohřevu teplé užitkové vody. Pro ohřev jsou určeny tři topné jednotky: solární absorbéry, elektricky vyhřívaný boiler a plynem vyhřívaný boiler. Pomocí přesného algoritmu jsou řízeny tři servo pohony, které regulují výstupní ohřátou užitkovou vodu zmiňovaných topných jednotek. Algoritmus řízení vychází především z ekonomického hlediska provozu těchto tří topných jednotek.

Součástí systému pro ohřev užitkové vody jsou i dvě čerpadla, která slouží k cirkulaci vody.

První čerpadlo je určeno k cirkulaci teplé užitkové vody mezi plynovým boilerem a vodovodními bateriemi. Cirkulační čerpadlo nám žene již ohřátou užitkovou vodu zpět do plynového boileru. Touto cirkulací dochází k rychlejšímu prohřátí vody v plynovém boileru a tudíž i k úspoře tepelné energie.

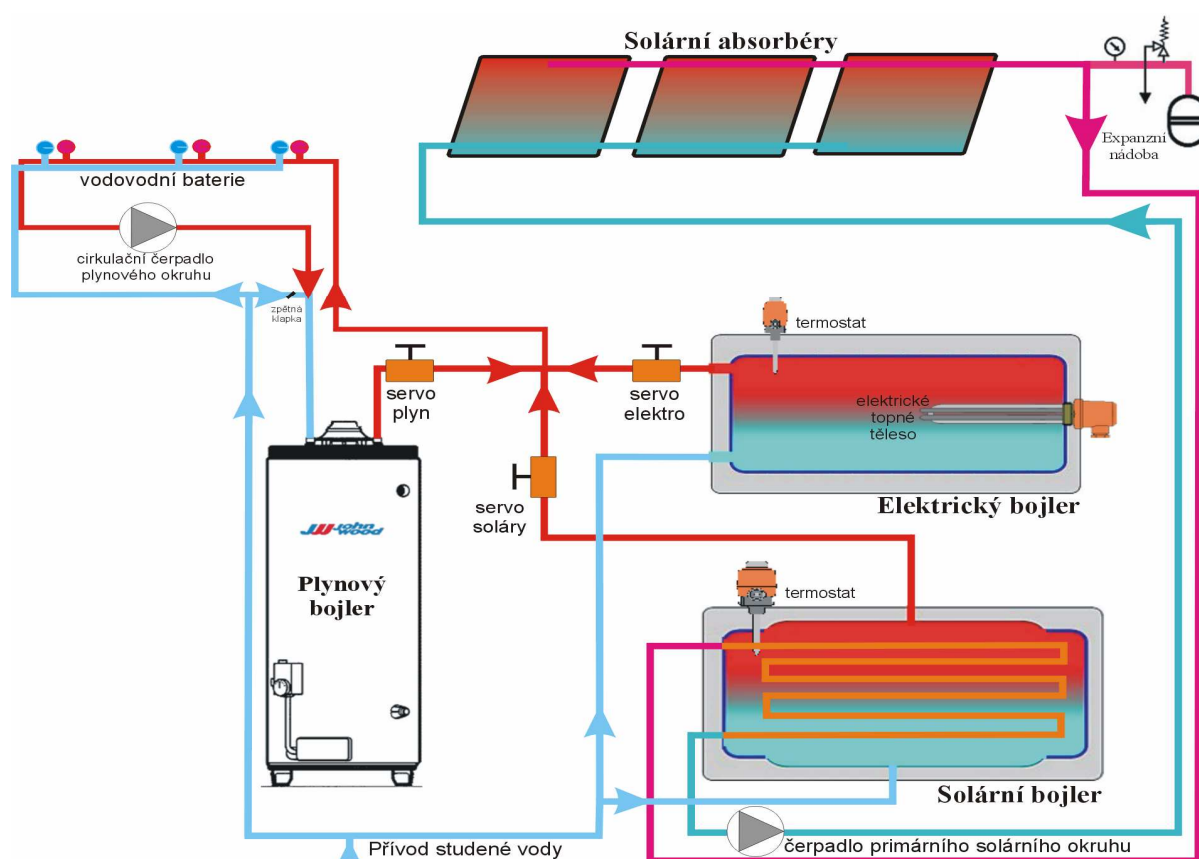
Druhé čerpadlo je určeno k cirkulaci vody mezi solárními absorbéry a boilerem. Čerpadlo žene ohřátou vodu ze solárních absorbérů do tepelného boileru, kde dochází k ohřevu užitkové vody.

Pro realizaci algoritmu řízení tří topných jednotek a dvou cirkulačních čerpadel je dále navržen software určený pro mikropočítače řady I8051 konkrétně pro model firmy Atmel nesoucí označení AT89C2051.

Hardwarová realizace spočívá v návrhu vhodného plošného spoje osazeného mikropočítačem a podpůrnými elektronickými obvody určenými k posílení výstupů mikropočítače. Důvodem použití podpůrných obvodů je, že samostatný mikropočítač není schopný spouštět čerpadla a řídit servo pohony.

1 PŘEHLED SOUČASNÉHO STAVU PROBLEMATIKY

K přípravě teplé užitkové vody jsou použity tři zdroje energie. Sluneční absorbéry, elektricky vyhřívaný bojler a plynem vyhřívaný bojler. Schéma příslušné soustavy určené pro přípravu teplé užitkové vody je zobrazeno v následujícím obrázku.



Obr. 1 Schéma soustavy určené pro přípravu teplé užitkové vody

1.1 Sluneční absorbéry

Sluneční absorbér je energeticky nezávislé zařízení, které mění dopadající sluneční záření (nositele energie) na energii tepelnou, kterou lze dále využít. Tato získaná energie se předává do transportní kapaliny, která získanou energii předává dále. Nejčastější využití slunečních absorbérů je při přípravě teplé užitkové vody, ohřevu vody v bazénu, přitápění, případně jejich vzájemná kombinace.

V našem případě jsou použity tři textilně plastové absorbéry TPA 020 firmy EKOSOLARIS ohřívající užitkovou vodu v boileru o objemu 160l.

Technické parametry absorbérů:

Výška:	1400 mm
Délka:	2000 mm
Tloušťka:	5 mm
Plocha:	2,8 m ²
Aktivní plocha:	2,6 m ²
Hmotnost:	5,3 kg
Hmotnost s vodní náplní:	cca 11 kg
Doporučený pracovní přetlak:	40 kPa
Maximální povolený přetlak:	60 kPa
Specifický objemový průtok:	260 l/hod
Maximální teplota topného média:	90°C
Maximální vnější tepelné namáhání:	110°C
Minimální okolní teplota:	-30°C
Provozování:	Voda 0°C⇒90°C

Příklad užití textilně plastového absorbéru pro ohřev bazénu:

Princip solárního systému je založen na zachytávání slunečních paprsků (nositelů energie) slunečními kolektory a jejich přeměně na tepelnou energii. Pro ohřev bazénu, pro který je textilně plastový absorbér nejvhodnější, lze volit ze dvou systémů:

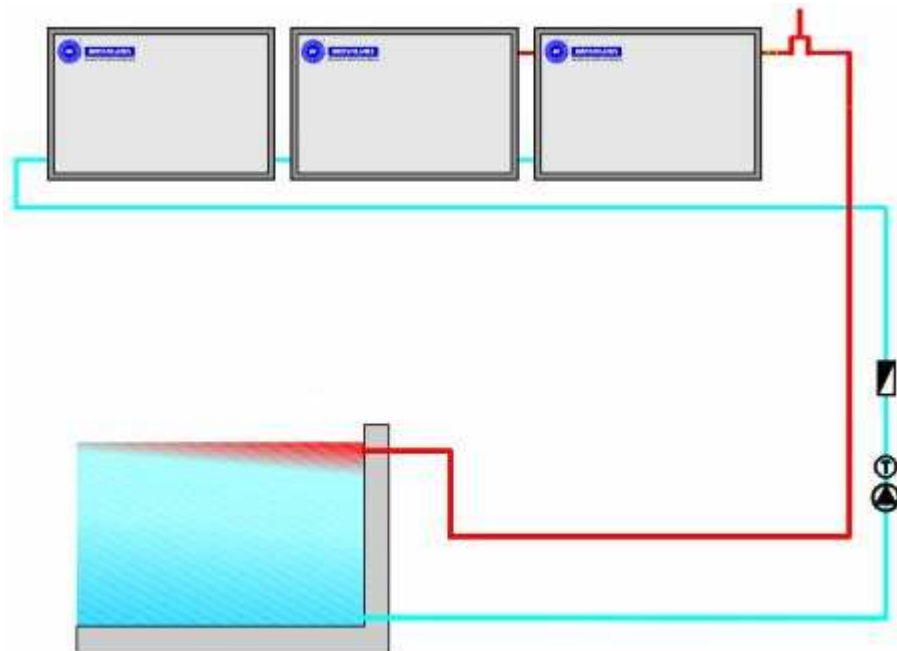
průtočný – ohřívající přímo bazénovou vodu

využívající výměník – je vhodný pro bazény s celoročním provozem

Pro ohřev bazénové vody je potřeba poměrně velké množství energie. Výpočet potřebné energie lze provést tak, že u krytých venkovních bazénů je potřeba cca 60% až 80% vodní hladiny pokrýt plochu kolektorů.

U vnitřních bazénů je pokrytí nižší, cca 40%. U vnitřních bazénů je většinou užíváno dvoukruhového systému s výměníkem a naplněním primárního solárního okruhu nemrznoucí kapalinou.

Vlastní bazén je velkým zásobníkem tepla, takže je zde při správné dimenzaci vyloučeno přehřívání kolektorů a není nutno provádět žádná další technická opatření.



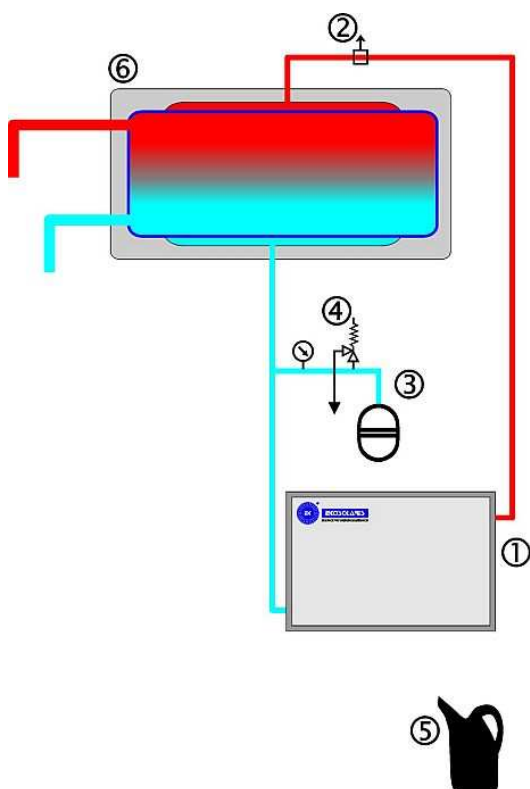
Obr. 2 Užití textilně plastového absorbéru pro ohřev bazénu

Příklad užití textilně plastového absorbéru pro přípravu teplé užitkové vody:

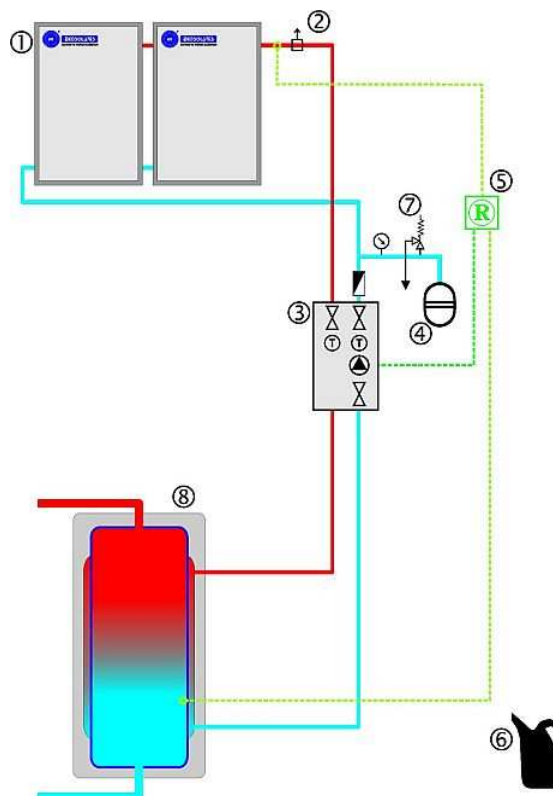
Textilně plastový absorbér lze s určitými omezeními vhodně využít i pro přípravu teplé užitkové vody. Tato omezení (tlak, teplota, sezónnost) jsou dána samotnou technickou podstatou textilně plastových absorbérů.

Při dimenzaci velikosti zásobníku teplé užitkové vody můžeme vycházet z předpokladu, že při použití jednoho textilně plastového absorbéru TPA 020 připravíme v letním období cca 100 litrů teplé vody za den. Díky svým schopnostem je možno textilně plastový absorbér používat jak v samotízných sestavách, tak i v hnaných (náš případ) sestavách pro ohřev vody.

Právě z těchto svých možností vyplývá nejčastější využití při přípravě teplé užitkové vody na chatách, chalupách, zahradních domcích apod., a to především v samotížném provedení. [4]



Obr. 3 Samotížná soustava



Obr. 4 Hnaná sestava

Popis jednotlivých částí (viz Obr. 3):

1. Solární absorbér
2. Automatický odvzdušňovací ventil
3. Expanzní nádoba
4. Pojišťovací ventil
5. Nemrznoucí směs
6. Zásobník s teplou užitkovou vodou

Popis jednotlivých částí (viz Obr. 4):

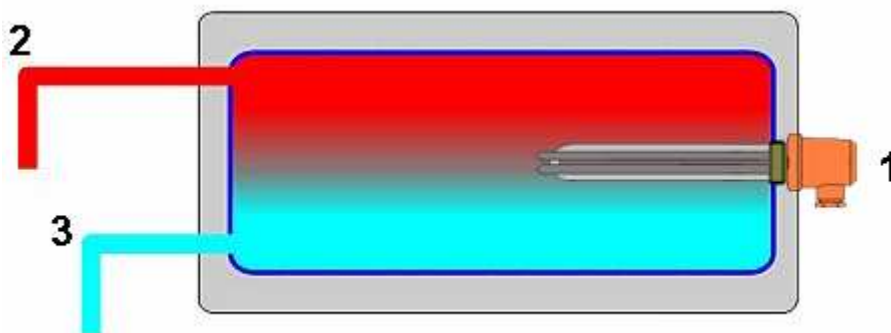
1. Solární absorbér
2. Automatický odvzdušňovací ventil
3. Hnací jednotka
4. Expanzní nádoba
5. Elektronická regulace
6. Nemrznoucí směs
7. Pojišťovací ventil
8. Zásobník s teplou užitkovou vodou

Vzhledem k tomu, že sluneční absorbéry jsou neekonomičtější zdroj energie ze tří použitých zdrojů energie (sluneční kolektory, elektricky vyhřívaný bojler a plynem vyhřívaný bojler), dostávají **nejvyšší prioritu** v ohřevu užitkové vody.

1.2 Elektricky vyhřívaný bojler

Do okruhu pro ohřev užitkové vody je zapojen i elektricky vyhřívaný boiler o objemu 160l. Je to pro případy, kdy dochází k většímu odběru teplé užitkové vody, kdy nestíhají dohřívát vodu sluneční absorbéry.

Elektricky vyhřívaný bojler je řešen samostatným boilerem vyhřívaným elektrickým topným tělesem.



Obr. 5 Elektricky vyhřívaný boiler

Popis jednotlivých částí (viz Obr. 5)

- 1) Elektrické topné těleso
- 2) Výstupní ohřátá užitková voda
- 3) Vstupní studená užitková voda

V našem případě je pro ohřev teplé užitkové vody použito topné těleso firmy *EEtop* typu *14011/90020*. Topné těleso je určeno pro přímý ohřev vody v bojlerech. Těleso se skládá ze tří topných větví tvaru „U” upevněných v šestihranné hlavici typu OK 60 se závitem M 48x2.

Technické parametry elektrického topného tělesa:

Napájení: $U = 3 \times 230 \text{ V}$

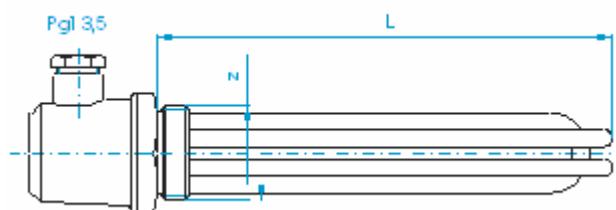
Výkon: $P = 3000 \text{ W}$

Délka: $L = 240 \text{ mm}$

Závit: $z = M48 \times 2$

Materiál topné tyče: Ms-CuP / Ni

Krytí: IP 54



Obr. 6 Elektrické topné těleso

Vzhledem k tomu, že ohřev užitkové vody pomocí elektrického boileru na noční provoz je úspornější než ohřev plynem, má elektrický boiler **střední prioritu**.

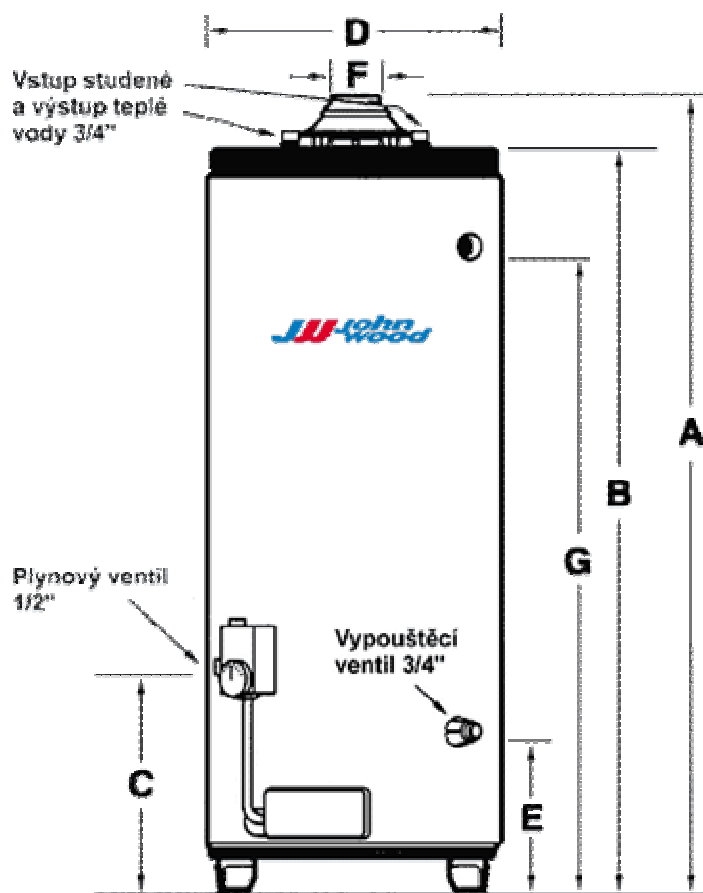
1.3 Plynem vyhříváný bojler

Vzhledem k tomu, že sluneční absorbéry nedokáží neustále ohřívat užitkovou vodu na požadovanou teplotu (absorbéry nelze použít v zimním období, též ve večerních hodinách, při nepříznivém počasí....) a elektricky vyhříváný boiler by též nedokázal pokrýt odběr teplé užitkové vody, je do okruhu zařazen třetí zdroj energie.

V našem případě je použit plynem vyhříváný boiler firmy JJJ John Wood.

Technické parametry plynem vyhříváného boileru:

Typové označení	JW 530 TPC
Kategorie:	I3B
Provedení:	B11
Označení:	3B-G30-30 mbar
Jmenovitý objem zásobníku:	114 l
Jmenovitý příkon:	8,8 kW
Maximální nastavitelná teplota vody:	cca 72°C
Doporučené nastavení teploty vody:	60°C
Maximální přetlak vody:	10,5 bar
Testovací přetlak vody:	21,0 bar
Doporučený připojovací přetlak vody:	3-4 bar
Doba ohřevu užitkové vody o $t=35^{\circ}\text{C}$:	43 minut
Spotřeba propanu:	0,63 kg/hod
Připojovací přetlak propanu maximální:	35 mbar
Teplota spalin:	160°C
Tah komínového průduchu:	3 Pa
Typ hořáku:	Atmosferický
Provedení hořáku:	Litina
Hmotnost prázdného/naplněného ohříváče:	48 kg/164 kg



Obr. 7 Plynem vyhříváný boiler

Rozměry plynem vyhříváného boileru (viz. Obr. 7.):

A = 1524 mm

B = 1440 mm

C = 355 mm

D = 406 mm

E = 241 mm

F = 76 mm

G = 1290 mm

Z předchozích podmínek vyplývá, že plynem vyhříváný boiler má **nejnižší prioritu**.

1.4 Snímání teploty teplé užitkové vody

Abychom mohli regulovat danou soustavu určenou pro ohřev užitkové vody, potřebujeme nějakým způsobem sledovat teplotu u těchto zdrojů tepla. K tomu jsou využity dva dvoupolohové termostaty. Jeden termostat sleduje teplotu vody v boileru solárního okruhu a druhý termostat sleduje teplotu vody v elektricky vyhřívaném boileru.

Termostaty jsou zvoleny proto, že nám stačí znát pouze dva stavy teplé užitkové vody:

- Voda **je nahřátá** na požadovanou teplotu
- Voda **není nahřátá** na požadovanou teplotu

U obou termostatů je nastavena teplota na 55°C a hystereze je nastavená na $\pm 3^\circ\text{C}$.

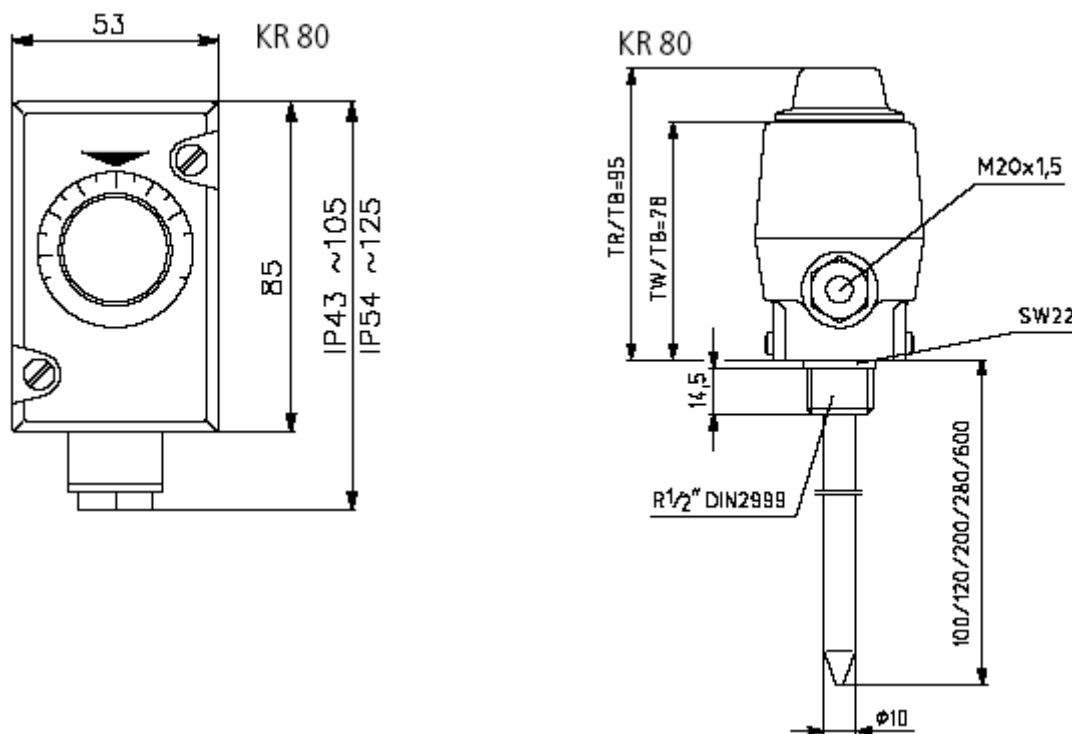
To znamená, že jakmile bude voda ohřátá na 58°C sepne se termostat a tím se otevře regulační klapka daného okruhu. Jakmile teplota poklesne pod 52°C uzavře se regulační klapka daného okruhu a boiler se může znovu nahřát na požadovanou teplotu.

V našem případě používáme termostaty firmy SENTRON typu KR 80.

Technické parametry termostatu KR 80:

Kategorie:	027-5
Provedení:	standard IP 43
Třída ochrany:	I
Přepínací možnosti:	max: 15 A, 250 V
Kontakt:	1 mikropřepínač jako přepínací kontakt
Teplota okolí:	-20...+55°C
Čidlo:	pro tekutá média, materiál Cu
Rozsah teplot:	0...70°C
Přepínací hystereze:	0...5°C
Délka jímky:	100 mm

Přezkoušeno dle TÜV a dle DIN 3440 do 95°C. Reg. Nr. TR/TW 107601



Obr. 8 Termostat KR 80

1.5 Regulace teplé užitkové vody jednotlivých okruhů

K regulaci teplé užitkové vody jsou využity servo pohony, které uzavírají a otevírají klapky a regulují tím tok teplé užitkové vody.

K regulaci používáme servo pohony firmy BELIMO typového označení NM24-SR.

Tento pohon je bez koncového vypínače a je jištěn proti přetížení. Při dosažení zarážky klapky, nebo pohonu se pohon automaticky zastaví.

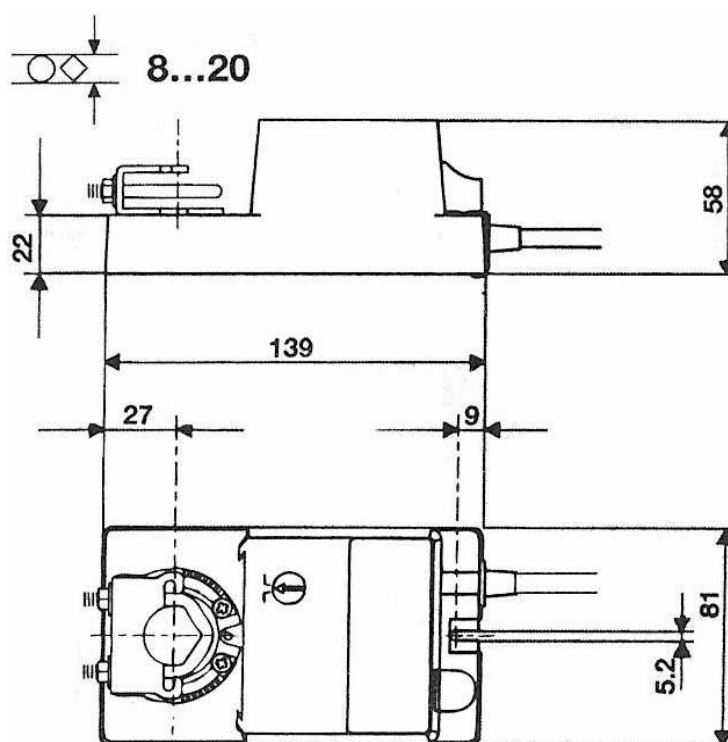
Při zapnutí napájecího napětí, nebo při spuštění tlačítkem provádí NM24-SR automatický test funkce. Pohon samostatně najíždí do obou koncových poloh a přizpůsobuje při tom svůj elektrický pracovní rozsah 2...10V stejnosměrných a dobu působení 150s efektivnímu mechanickému pracovnímu úhlu klapky. Po tomto procesu se pohon nastaví do polohy stanovené řídicím signálem.

Technické parametry termostatu servopohonu BELIMO NM24-SR:

Napájecí napětí: 24V $\pm$ 20%, 50...60Hz

Příkon: 1,3W v pohybu, 0,5W v klidové poloze

Dimenzování:	3VA
Ochranná třída:	III (malé napětí)
Krytí:	IP 54
Připojení:	kabel 1m, 4x0,75 mm ²
Pracovní úhel:	max.95°, přestavitelné mechanické zarážky
Krouticí moment:	min. 8Nm při jmenovitém napětí
Doba přestavení:	150s, nezávisle na mechanickém pracovním úhlu od 0..35° do 0..90°
Smysl otáčení(při Y=0V):	Lze zvolit přepínačem vlevo/vpravo
Řídicí signál Y:	0..10V= (vstupní odpor je 100k Ω)
Pracovní rozsah:	2..10V= pro pracovní úhel 0...100%
Měřicí napětí U:	2..10V= ($\leq 0,7\text{mA}$) pro pracovní úhel 0..100%
Hladina hluku:	max. 35dB
Údržba:	bez údržby
Hmotnost:	900g



Obr. 9 Servo pohon BELIMO NM-24 SR

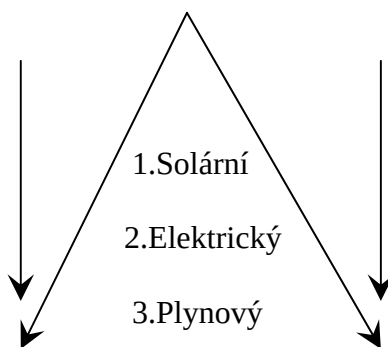
2 PROGRAMOVÉ ŘEŠENÍ DANÉ ÚLOHY

Abychom mohli vytvořit program pro řízení dané úlohy musíme si nejdříve stanovit podmínky řízení. Poté navrhne grafickou strukturu programu pro jednodušší orientaci. Z podmínek si vytvoříme pravdivostní tabulky a za pomoci Karnaughových map vytvoříme rovnice pro řízení.

2.1 Podmínky pro řízení dané úlohy

2.1.1 Podmínky pro serva tří topných jednotek

1. Pokud je nahřátý solární boiler, je klapka solárního okruhu otevřena. Elektrický a plynový okruh je uzavřen a to i v případě, kdy jsou oba bojlerů nahřáté na požadovanou teplotu.
2. Pokud není nahřátý solární boiler na požadovanou teplotu, ale je nahřátý elektrický boiler, potom solární a plynový okruh je uzavřený a klapka elektrického okruhu je otevřená. A to i v případě, kdy je plynový boiler nahřátý na požadovanou teplotu.
3. Pokud není nahřátý solární boiler ani elektrický boiler na požadovanou teplotu a je nahřátý plynový boiler, potom solární a elektrický okruh je uzavřen a klapka plynového okruhu je otevřena.
4. Pokud není nahřátý ani jeden ze tří boilerů, potom solární a elektrický okruh je uzavřený a servo plynového okruhu je otevřené. Ke stavu, že by nebyl ani jeden boiler nahřátý, by nemělo dojít. V případě, že by se tak stalo, sice nepoteče teplá voda, ale v potrubí bude neustále voda, což je žádoucí stav.



Obr. 10 Schéma priority jednotlivých zdrojů tepelné energie

2.1.2 Podmínky pro ovládání cirkulačního čerpadla plynového okruhu

1. Pokus je přepínač sepnutý do polohy AUTOMATICKY, potom je čerpadlo řízeno následující podmínkou: Pokud je nahřátý solární boiler nebo elektrický boiler nebo oba dva zároveň a je přepínač sepnutý do polohy Auto, je čerpadlo plynového okruhu zastaveno.
2. Pokud je přepínač sepnutý do polohy VYPNUTO, tak je cirkulační čerpadlo plynového okruhu vždy zastaveno.
3. Pokud je přepínač sepnutý do polohy MANUALNĚ, je čerpadlo vždy spuštěno.

2.1.3 Podmínky pro ovládání čerpadla primárního solárního okruhu

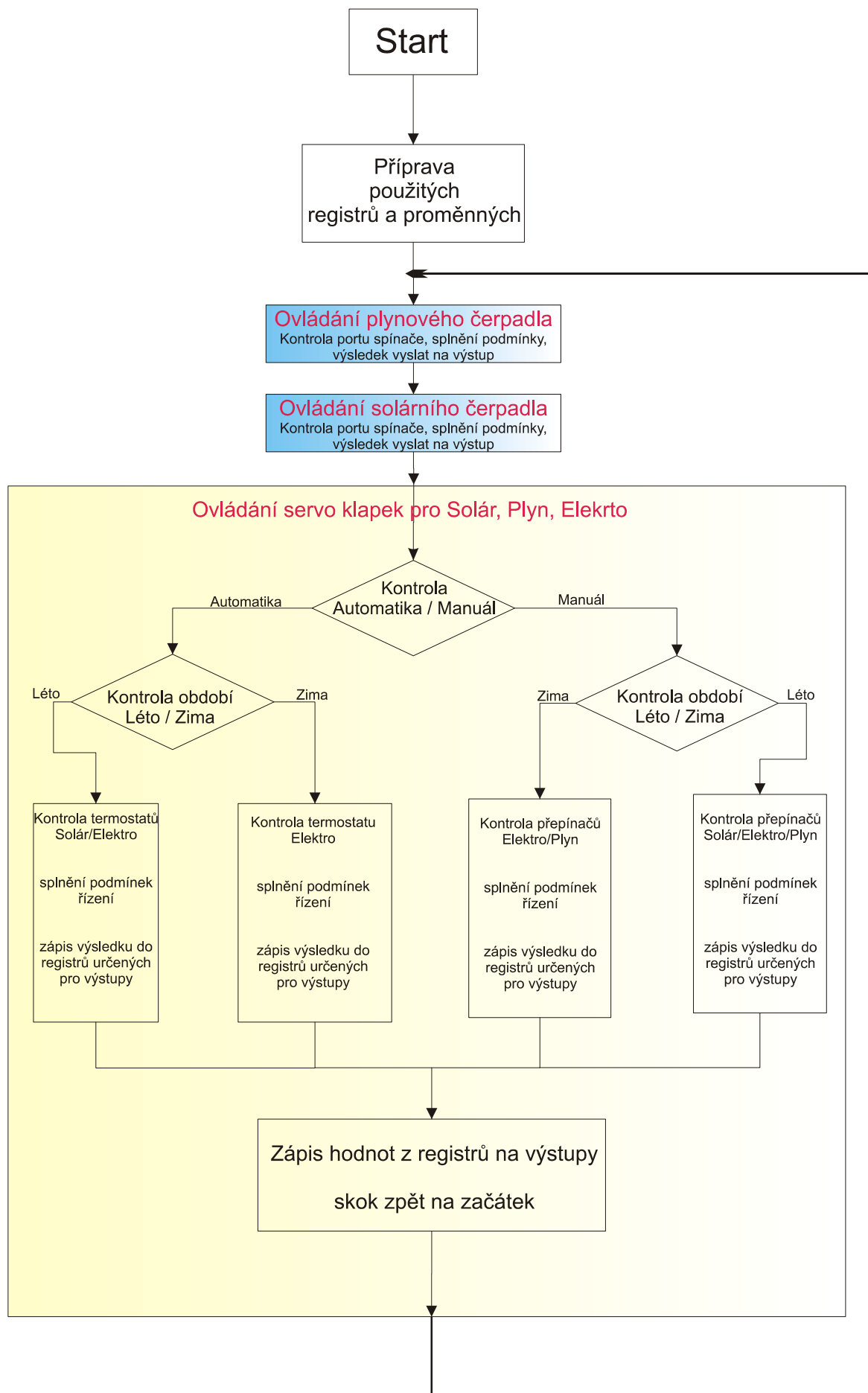
1. Pokud je přepínač sepnutý do polohy AUTO, tak se čerpadlo primárního solárního okruhu spustí podle podmínek daných strukturou elektrického obvodu (viz obr. 10).
2. Pokud je přepínač sepnutý do polohy VYPNUTO, tak je čerpadlo primárního solárního okruhu vždy zastaveno.
3. Pokud je přepínač sepnutý do polohy MANUAL, je čerpadlo vždy spuštěno.

Z podmínek určených pro servo pohony, pro čerpadlo plynového okruhu a pro čerpadlo solárního okruhu, rozdělíme program na tři hlavní části:

- Řízení tří servo pohonů
- Řízení cirkulačního čerpadla plynového okruhu
- Řízení čerpadla primárního solárního okruhu

2.2 Navržení grafické struktury řízení

Podle předchozích podmínek regulace, jsem navrhl schéma programu pro praktické znázornění jeho struktury a též pro jednodušší orientaci při programování mikropočítače.



Abychom mohli vytvořit program pro mikropočítač, je nutné si vytvořit rovnice řízení pro jednotlivé zdroje energie podle daných podmínek.

1. Vytvoříme si pravdivostní tabulky pro jednotlivé zdroje energie a pro obě čerpadla.
2. Tabulky zjednodušíme sestavením Karnaughových map pro jednotlivé proměnné.
3. Z Karnaughovy mapy si vyjádříme algebraické výrazy pro jednotlivé výstupy.

2.3 Pravdivostní tabulky a Karnaughovy mapy

2.3.1 Pro tři zdroje tepla (Solár, Elektro, Plyn), automatické řízení a letní období .

Vstupy: Termostat solárního boileru [T_S], Termostat elektrického boileru [T_E]

Výstupy: Servo plynového boileru [S_P], Servo elektrického boileru [S_E], Servo solárního boileru [S_S]

Tabulka 1. Pravdivostní tabulka pro automatické řízení, letní období.

T_S	T_E	S_P	S_E	S_S
0	0	1	0	0
0	1	0	1	0
1	0	0	0	1
1	1	0	0	1

Karnaughova mapa pro servo solárního boileru:

	T_E	
T_S	0	0
	1	1

$$S_S = T_S \quad (1)$$

Z toho plyne, že servo solárního ohřevu je závislé pouze na výstupu z termostatu solárního boileru.

Karnaughova mapa pro servo elektrického boileru:

	T_E	
T_S	0	1
	0	0

$$S_E = \overline{T_S} + T_E \quad (2)$$

Z toho plyne, že servo elektrického ohřevu je závislé na negovaném výstupu z termostatu solárního boileru a na výstupu z termostatu elektrického boileru.

Karnaughova mapa pro servo plynového boileru:

		T_E
T_S	1	0
0	0	0

$$S_E = \overline{T_S} + \overline{T_E} \quad (3)$$

Z toho plyne, že servo plynového ohřevu je závislé na negovaném výstupu z termostatu solárního boileru a též na negovaném výstupu z termostatu elektrického boileru.

2.3.2 Pro tři zdroje tepla (Solár, Elektro, Plyn), automatické řízení a zimní období

Vstupy: Termostat elektrického boileru [T_E]

Výstupy: Servo plynového boileru [S_P], Servo elektrického boileru [S_E], Servo solárního boileru [S_S]

Tabulka 2. Pravdivostní tabulka pro automatické řízení, zimní období

T_E	S_P	S_E	S_S
0	1	0	0
1	0	1	0

Vzhledem k jednoduchosti pravdivostní tabulky si mohu rovnou vyjádřit algebraické výrazy.

Pro servo solárního ohřevu platí:

$$S_S = 0 \quad (4)$$

Z toho plyne, že servo solárního ohřevu je vždy uzavřené.

Pro servo elektrického ohřevu platí:

$$S_E = T_E \quad (5)$$

Z toho plyne, že servo elektrického ohřevu je závislé pouze na výstupu z termostatu elektrického boileru.

Pro servo plynového ohřevu platí:

$$S_P = \overline{T_E} \quad (6)$$

Z toho plyne, že servo plynového ohřevu je závislé pouze na negovaném výstupu z termostatu elektrického boileru.

2.3.3 Pro tři zdroje tepla (Solár, Elektro, Plyn), manuální ovládání a letní období

Vstupy: Přepínač elektro [M_E], přepínač solár [M_S], přepínač plyn [M_P]

Výstupy: Servo plynového boileru [S_P], Servo elektrického boileru [S_E], Servo solárního boileru [S_S]

Tabulka 3. Pravdivostní tabulka pro manuální ovládání, letní období

M_S	M_E	M_P	S_P	S_E	S_S
0	0	0	0	0	0
0	0	1	1	0	0
0	1	0	0	1	0
0	1	1	0	1	0
1	0	0	0	0	1
1	0	1	0	0	1
1	1	0	0	0	1
1	1	1	0	0	1

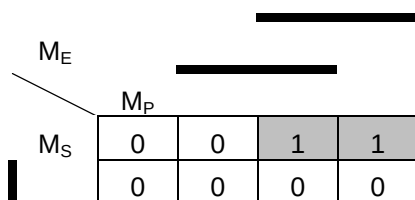
Karnaughova mapa pro servo solárního boileru:

		M_P			
		0	0	0	0
M_S	0	0	0	0	0
	1	1	1	1	1

$$S_S = M_S \quad (7)$$

Z toho plyne, že servo solárního ohřevu je závislé pouze na manuálním přepínači určený pro solární servo.

Karnaughova mapa pro servo elektrického boileru:

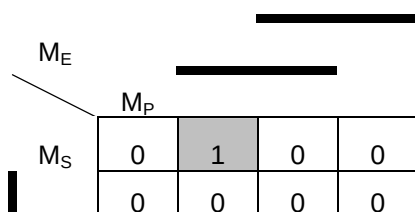


		M_P			
		0	0	1	1
M_S	M_E	0	0	1	1
		0	0	0	0

$$S_E = \overline{M_S} + M_E \quad (8)$$

Z toho plyne, že servo elektrického ohřevu je závislé na negované hodnotě z manuálního přepínače pro solární servo a z manuálního přepínače pro elektro servo.

Karnaughova mapa pro servo plynového boileru:



		M_P			
		0	1	0	0
M_S	M_E	0	1	0	0
		0	0	0	0

$$S_E = \overline{M_S} + \overline{M_E} + M_P \quad (9)$$

Z toho plyne, že servo elektrického ohřevu je závislé na negované hodnotě z manuálního přepínače pro solární servo dále na negované hodnotě z manuálního přepínače pro elektro servo a na manuálním přepínači pro plynové servo.

2.3.4 Pro tři zdroje tepla (Solár, Elektro, Plyn), manuální ovládání a zimní období

Vstupy: Přepínač elektro [M_E], přepínač solár [M_S], přepínač plyn [M_P]

Výstupy: Servo plynového boileru [S_P], Servo elektrického boileru [S_E], Servo solárního boileru [S_S]

Tabulka 4. Pravdivostní tabulka pro manuální ovládání, zimní období

M_S	M_E	M_P	S_P	S_E	S_S
0	0	0	0	0	0
0	0	1	1	0	0
0	1	0	0	1	0
0	1	1	0	1	0
1	0	0	0	0	0
1	0	1	1	0	0
1	1	0	0	1	0
1	1	1	0	1	0

Z pravdivostní tabulky lze vyčíst, že sepnutí nebo vypnutí manuálního přepínače pro solární servo, nemá žádný vliv na zbylé dva servo pohony.

Pro servo solárního ohřevu platí:

$$S_S = 0 \quad (10)$$

Z toho plyne, že servo solárního ohřevu je vždy uzavřené (platí pouze pro zimní období).

Karnaughova mapa pro servo elektrického boileru:

M_E	M_P		
M_S		0	1
		0	1

$$S_E = M_E \quad (11)$$

Z toho plyne, že servo elektrického ohřevu je závislé pouze na manuálním přepínači pro elektro servo.

Karnaughova mapa pro servo plynového boileru:

M_E	M_P		
M_S		0	1
		0	0

$$S_P = \overline{M_E} + M_P \quad (12)$$

Z toho plyne, že servo plynového ohřevu je závislé na negovaném hodnotě z manuálního přepínače pro elektro servo a na manuálním přepínači pro plynové servo.

2.3.5 Pro řízení čerpadla primárního solárního okruhu

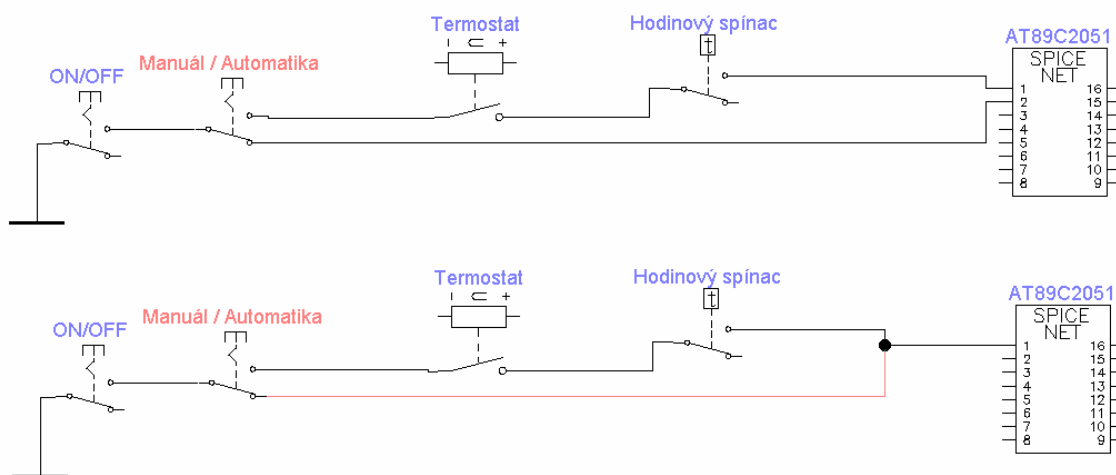
Vstupy: Třípolohový přepínač pro cirkulační čerpadlo primárního solárního oběhu [M_{CS}]

Výstupy: Čerpadlo primárního solárního oběhu [C_S]

Tabulka 5. Pravdivostní tabulka pro čerpadlo primárního solárního okruhu

M_{CS}	C_S
OFF	0
M	1
A	1

Z tabulky je zřejmé, že spínač je tří polohový. Z toho vyplývá, že bude nutné použít dva vstupy mikropočítače. Vzhledem k nedostatku vstupně / výstupních pinů jsem se rozhodl, že tento problém ošetřím hardwarově (viz následující obrázek).



Obr. 11 Hardwarové řešení pro čerpadlo primárního solárního okruhu

Z obrázků je zřejmé hardwarové řešení.

Svorku z Manuální polohy spínače Manuál/Automatika jsem vyvedl na konec sériového zařazení termostatu a hodinového spínače a tím jsem ušetřil jeden vstup mikropočítače.

Podmínka řízení zůstala zachována.

Tabulka 6. Pravdivostní tabulka pro čerpadlo primárního solárního okruhu – po úpravě

$M_{\check{C}S}$	C_S
OFF	0
A/M	1

Vzhledem k jednoduchosti pravdivostní tabulky si mohu rovnou vyjádřit algebraický výraz pro řízení čerpadla primárního solárního okruhu.

$$M_{\check{C}S} = C_S \quad (13)$$

Ovládání čerpadla primárního solárního okruhu je tedy závislé pouze na spínači pro jeho ovládání.

2.3.6 Pro řízení cirkulačního čerpadla plynového okruhu

Vstupy: Manuální přepínač pro čerpadlo plynového okruhu: $[M_{\check{C}PM}]$ – Manuál / $[M_{\check{C}PA}]$ - Automat

Výstupy: Čerpadlo plynového okruhu $[C_P]$

Tabulka 7. Pravdivostní tabulka pro řízení čerpadla plynového okruhu

T_E	T_S	$M_{\check{C}PM}$	$M_{\check{C}PA}$	C_P
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	1
1	1	1	1	0

Karnaughova mapa pro řízení čerpadla plynového oběhu:

		$M_{\check{C}PM}$			
T_E	T_S	$M_{\check{C}PA}$			
		0	1	0	1
		0	0	0	1
		0	0	0	1
		0	0	0	1

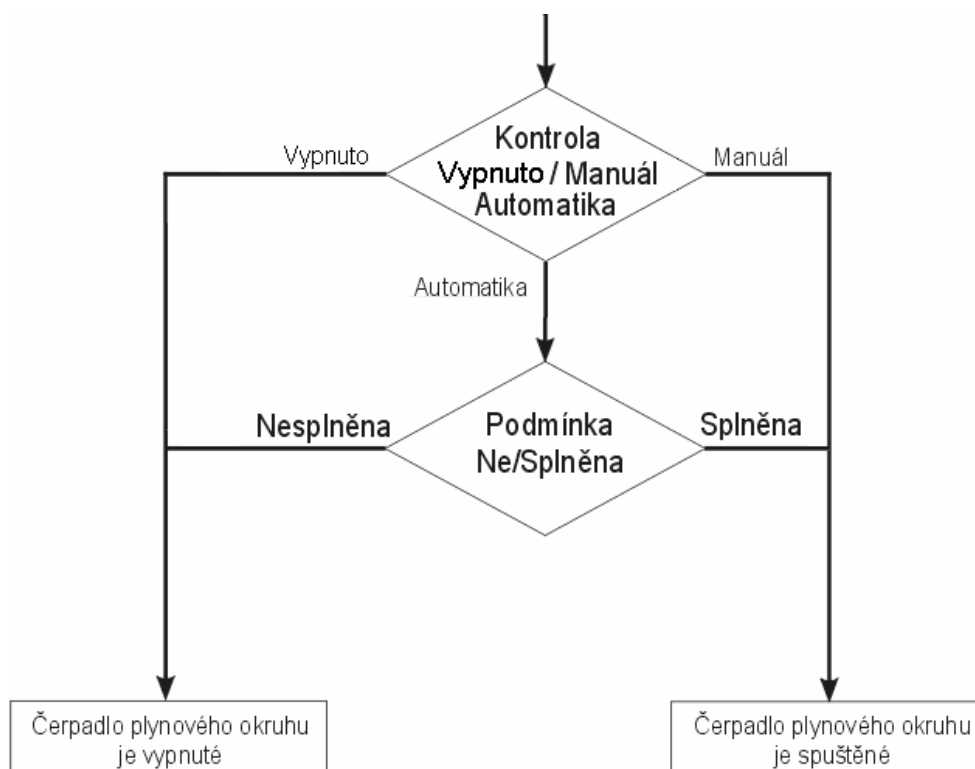
Čerpadlo plynového oběhu je potom řízeno rovnicí:

$$C_P = M_{\check{C}PM} + \overline{M_{\check{C}PM}} \cdot M_{\check{C}PA} \cdot \overline{T_S} \cdot \overline{T_E} \quad (14)$$

Rovnici převedu na logický součin.

$$C_P = \overline{\overline{M_{\check{C}PM}} \cdot \overline{M_{\check{C}PM}} \cdot M_{\check{C}PA} \cdot \overline{T_S} \cdot \overline{T_E}} \quad (15)$$

Realizace, výše uvedené algebraické rovnice, by byla pomocí mikropočítače velmi obtížná a složitá. Proto jsem se rozhodl řešit řízení čerpadla plynového oběhu daleko jednodušším způsobem a to větvením programu (viz následující obrázek).



Obr. 12 Větvení programu pro řízení čerpadla plynového oběhu.

2.4 Software pro programování mikropočítačů řady I51

Pro programové řešení dané úlohy jsem si zvolil software μ Scope51, který je určen pro programování mikropočítačů řady I51.

Vývojové ladící prostředí μ Scope představuje kompletní integrovaný prostředek pro vývoj a ladění projektů na bázi mikropočítačů řady 8051.

Program běží na běžném PC a s laděnou aplikací komunikuje přes běžný sériový port nebo USB/RS232 konvertor.

Prostředek μ Scope podporuje ladění programu na zdrojové úrovni pro jazyky C (Keil, SDCC, Intel) a assembler. Program je možnou spouštět, krokovat po instrukcích nebo funkčních blocích, je možno nastavovat body přerušení, měnit aktuální pozici programového čítače (opakovat průchod blokem programu), prohlížet a měnit obsah paměti, registry a proměnných (typ proměnné je automaticky rozpoznán), disassemblovat bloky programu nebo opravovat kód pomocí in-line assembleru.

Jeho simulační verze umožňuje ladit aplikace bez reálného hardware, pomocí designeru lze vytvořit virtuální aplikaci a tvořit pro ni program jako pro skutečný hardware, což je v mém případě hlavní důvod volby daného softwaru.

Program μ Scope funguje pod běžnými operačními systémy Windows 95/98/Me a Windows NT/2k/XP. [6]

Software se skládá z několika hlavních částí:

Editor zdrojového textu

Editor má všechny standardní funkce včetně přesunu textu, "cut-copy-paste", víceúrovňové "undo/redo" (zpět/vpřed), hledání a náhradu včetně prohledávání souborů a barevné zvýraznění. Je umožněn výběr typu a velikosti fontu. Podporuje rovněž funkci vyhledávání v souborech. [6]

Správa projektu

Projekt se skládá ze zdrojových souborů, pravidel pro překlad a generování cílového kódu a podpůrných nástrojů prostředí μ Scope. Správce projektu umožňuje snadnou konfiguraci

nástrojů, začlenění překladačů a linkerů a vytvoření pravidel pro překlad zdrojového kódu (direktivy, podmíněný překlad atd.). [6]

Assembler/disassembler

Pro rychlou opravu kódu je v prostředí μ Scope implementován jednoduchý in-line assembler. S jeho pomocí je možno rychle testovat změnu cílového kódu bez nutnosti opětovného překladu a zavedení kódu do paměti. To je další významný prvek pro urychlení vývojového cyklu. V okně disassembleru je možno prohlížet cílový kód produkovaný překladačem na úrovni assembleru. [6]

Symbolický browser

Okno symbolického browseru umožňuje dva módy zobrazení; jednak jako strom modulů a jednak jako seznam s podrobnými údaji. Jednoduše poklepáním myši lze zobrazovat a měnit obsah proměnných, nebo vyhledávat jejich deklaraci ve zdrojovém souboru. [6]

Integrovaný assembler

Součástí prostředí je assembler ASM-52. Tento assembler akceptuje Intel MCS-51 kompatibilní zdrojové soubory a vytváří cílový kód ve formátu Intel-HEX vhodný pro programování nebo download. [6]

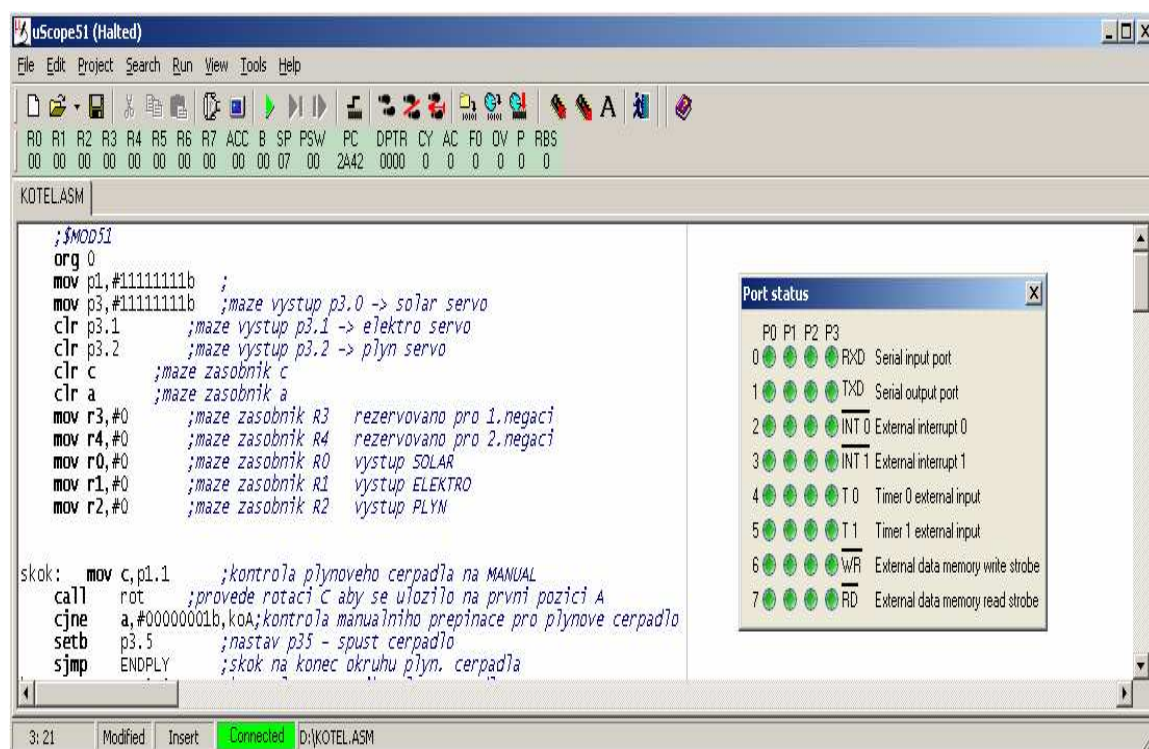
Ladění na úrovni zdrojového kódu

Prostředí μ Scope nabízí podobný komfort ladění jako např. aplikace Delphi nebo Visual C++. V editačním okně zdrojového souboru lze jednoduše nastavovat body přerušení, krokovat, zobrazovat obsah proměnných atd. Pro lepší orientaci v okně jsou jednotlivé ladící elementy barevně odlišeny. Okno zdrojového kódu je tedy současně ladícím oknem. Během ladění programu jsou automaticky monitorovány všechny důležité registry. Obsah proměnných je možno jednoduše zobrazit nebo měnit v dialogu pro zobrazení. [6]

Inspekce paměti, registrů a proměnných

V okně terminálu je možno zobrazovat nebo měnit obsah paměti dat, externích dat, kódu nebo i bitové oblasti paměti a systémových registrů. Je možno volit různé formáty zobrazení (char, byte, word, int, long, pointer, ASCII, float atd.) a různé číselné báze (dvojková, oktálová, dekadická a hexadecimální). Jednotlivé oblasti paměti je možno přesouvat, kopírovat, plnit konstantou atp. [6]

Celkově bych μ Scope hodnotil jako intuitivní software s příjemným prostředím. Výhodou je manuál v češtině.



Obr. 13 Hlavní okno programu μ Scope

2.5 Řídicí program určený pro mikropočítače řady I51

Program je vytvořen v programovacím jazyku assembler. Pro pozdější úpravy programu a pro lepší orientaci, jsem si ke každému příkazu napsal informaci. Dělicí znak „ ; “ (středník) znamená, že při kompilaci nebude uvažován text za tímto znakem a nebude se zavádět ani do jádra programu, který bude nahrán přímo do mikropočítače.

```
org    0                                ;začátek programu

clr    p3.0                            ;maže výstup p3.0 -> solár servo

clr    p3.1                            ;maže výstup p3.1 -> elektro servo

clr    p3.2                            ;maže výstup p3.2 -> plyn servo

clr    p3.7                            ;maže výstup p3.7 -> čerpadlo solárního okruhu

clr    p3.5                            ;maže výstup p3.5 -> čerpadlo plynového okruhu

clr    c                                ;maže zásobník c
```

```

        clr    a                ;maže zásobník a
        mov    r3,#0           ;maže zásobník R3   rezervováno pro 1.negaci
        mov    r4,#0           ;maže zásobník R4   rezervováno pro 2.negaci
        mov    r0,#0           ;maže zásobník R0   vystup SOLAR
        mov    r1,#0           ;maže zásobník R1   vystup ELEKTRO
        mov    r2,#0           ;maže zásobník R2   vystup PLYN
skok:   mov    c,p1.1          ;kontrola plynového čerpadla na MANUAL
        call   rot             ;provede rotaci C aby se uložilo na první pozici A
        cjne   a,#00000001b,koA ;kontrola manuálního přepínače pro plynové čerpadlo
        setb   p3.5            ;nastav p35 - spust' čerpadlo
        sjmp   ENDPLY          ;skok na konec okruhu plyn. čerpadla
koA:    mov    c,p1.0          ;kontrola automatiky plyn.čerpadla
        call   rot             ;skoč na rotaci
        cjne   a,#0,AUTOP      ;skoč na automat plynu
        clr    p3.5            ;vypni čerpadla plynového okruhu
        sjmp   ENDPLY          ;skok na konec okruhu plyn. čerpadla
AUTOP:  call    nega1          ;neguje Te a Ts
        mov    a,r3            ;přesun negovaného R3 do A
        anl    a,r4            ;A=A(negované)*R4(negované)
        clr    c               ;vymaže C
        rrc    a               ;uloží první bit a A do C
        mov    p3.5,c          ;nastaví vystup podle podmínky
ENDPLY: mov    c,p1.7          ;kontrola ovládaní čerpadla soláru
        call   rot             ;vola rotaci
        cjne   a,#0,vyp        ;kontrola manuálního přepínače pro solární čerpadlo
        setb   p3.7            ;spust' čerpadlo soláru

```

```

        sjmp    sko                ;skoč na sko
vyp:    clr     p3.7              ;vypni čerpadlo soláru
sko:    mov     c,p1.3            ;Kontrola AUTOMAT-MANUAL (přesune p1.3 do c)
        call    rot              ;skok na rot:
        cjne    a,#0>manual      ;pokud se a nerovná 0 tak skoč na manuál:
        call    nega1            ;spustit negaci1
        mov     c,p1.2            ;Kontrola LETO-ZIMA (přesune p1.2 do c)
        call    rot              ;skok na rot:
        cjne    a,#0,zimaA       ;pokud se A nerovná 0 tak skoč na zimaA
        mov     c,p3.3            ;Začátek AUTOMAT-LETO
        call    rot              ;skok na rot:
        mov     r0,a              ;přesune A do R0 (připraveno na ON/OFF serva S)
        mov     c,p3.4            ;přesune p3.4 do c
        call    rot              ;skok na rot:
        anl     a,r3              ;A=A(negované)*R3
        mov     r1,a              ;přesune A do R1 (připraveno na ON/OFF serva E)
        mov     a,r3              ;přesune negovanou hodnotu z R3 do A
        anl     a,r4              ;A=A(negované)*R4
        mov     r2,a              ;přesune A do R2 (připraveno na ON/OFF serva P)
        sjmp    nastav           ;skok na nastav:
sko1:    sjmp    skok
zimaA:  mov     r0,#0              ;Servo soláru bude v ZIME vždy zavřené – Začátek
                                           ;AUTOMAT-ZIMA
        mov     c,p3.4            ;přesune p3.4 do c
        call    rot              ;skok na rot:
        mov     r1,a              ;přesune A do R1 (připraveno na ON/OFF serva E)

```

```
    mov    a,r4

    mov    r2,a                ;přesune A do R2 (připraveno na ON/OFF serva P)

    sjmp   nastav             ;skok na nastav

manual: call nega2

    mov    c,p1.2              ;Kontrola LETO-ZIMA (přesune p1.2 do c) –
                                ;MANUAL

    call   rot                 ;skok na rot:

    cjne   a,#0,zimaM          ;pokud se A nerovná 0 tak skoč na zimaM

    mov    c,p1.4              ;Začátek MANUAL-LETO

    call   rot                 ;skok na rot:

    mov    r0,a                ;přesune A do R0 (připraveno na ON/OFF serva S)

    mov    c,p1.5              ;přesune p1.5 do c

    call   rot                 ;skok na rot

    anl    a,r3                ;A=A(negované)*R3

    mov    r1,a                ;přesune A do R1 (připraveno na ON/OFF serva E)

    mov    c,p1.6              ;přesune p1.6 do c

    call   rot                 ;skok na rot:

    anl    a,r3                ;A=A*R3(negované)

    anl    a,r4                ;A=A*R4

    mov    r2,a                ;přesune A do R2 (připraveno na ON/OFF serva P)

    sjmp   nastav             ;skok na nastav:

zimaM: mov    r0,#0            ;Servo soláru bude v ZIME vždy zavřené - Začátek
                                ;MANUAL-ZIMA

    mov    c,p1.5              ;přesune p1.5 do c

    call   rot                 ;skok na rot:

    mov    r1,a                ;přesune A do R1 (připraveno na ON/OFF serva E)
```

```
    mov    c,p1.6          ;přesune p1.6 do c
    call   rot             ;skok na rot:
    anl    a,r4            ;A=A*R4
    mov    r2,a            ;přesune A do R2 (připraveno na ON/OFF serva P)
nastav: mov    a,r0        ;přesune R0 do a
    clr    c              ;vymaže C
    rrc    a              ;přesune 0ty byt z A do C
    mov    p3.0,c         ;vyšle C na výstup uC -> ovládání serva SOLAR
    mov    a,r1           ;přesune R1 do a
    clr    c              ;vymaže C
    rrc    a              ;přesune 0ty byt z A do C
    mov    p3.1,c         ;vyšle C na výstup uC -> ovládání serva ELEKTRO
    mov    a,r2           ;přesune R2 do a
    clr    c              ;vymaže C
    rrc    a              ;přesune 0ty byt z A do C
    mov    p3.2,c         ;vyšle C na výstup uC -> ovládání serva PLYN
    sjmp   sko1
rot:   clr    a           ;maze zásobník a
    rlc    a             ;uloží c do zásobníku a
    ret                    ;návrat na výchozí pozici
nega1: mov    c,p3.3      ;přesune p3.3 do C kvůli negaci
    cpl    c             ;neguje C
    call   rot           ;skok na rot:
    mov    r3,a          ;záloha negovaného C do R3
    mov    c,p3.4        ;přesune p3.4 do C kvůli negaci
    cpl    c             ;neguje C
```

```
call    rot                ;skok na rot:
mov     r4,a                ;záloha negovaného C do R4
ret
nega2:  mov     c,p1.4       ;přesune p1.4 do C kvůli negaci
cpl     c                   ;neguje C
call    rot                ;skok na rot:
mov     r3,a                ;záloha negovaného C do R3
mov     c,p1.5             ;přesune p1.5 do C kvůli negaci
cpl     c                   ;neguje C
call    rot                ;skok na rot:
mov     r4,a                ;záloha negovaného C do R4
ret
end
```

Program obsahuje 123 řádků instrukcí. Po kompilaci a následujícím převedení do HEX souboru je výsledná velikost programu 635 bajtů.

3 HARDWAROVÉ ŘEŠENÍ DANÉ ÚLOHY

3.1 Výběr ideálního mikropočítače pro řešení dané úlohy:

Rozhodl jsem se pro mikropočítač řady I51. Důvodem mé volby je jeho široká nabídka na trhu, softwarová podpora (assembler), pořizovací cena a v neposlední řadě mé zkušenosti s programováním této řady mikropočítačů.

Řešení daného řízení je možné provést též například pomocí kompaktního programovatelného automatu (dále jen PLC). Srovnáme-li PLC s mikropočítačem, tak převažují výhody PLC. PLC můžeme přímo zapojit do reálné soustavy bez toho, aniž bychom museli řešit podpůrné obvody pro vstupy a výstupy. Kompaktní PLC je možno rozšířit o přídatné zásuvné moduly a tím ho přizpůsobit řízené soustavě. Ale vzhledem k pořizovacím nákladům, kdy cena PLC vyhovující danému řízení převyšuje částku 10000 Kč, byla zvolena levnější varianta řízení a to výše zmiňovaným mikropočítačem od firmy Atmel. Tento mikropočítač zcela vyhovuje dané situaci a celková cena soustavy nepřesáhne částku 1000 Kč.

Jakmile známe druh použitého mikropočítače, je nutné zvolit si přesný typ. Hlavním kritériem pro volbu typu mikropočítače byl počet vstupně výstupních linek (dále už jen I/O z anglického Input/Output).

Pro danou úlohu je nutné zvolit mikropočítač, který bude mít minimálně 15 I/O linek (počet linek vyplývá z následující tabulky).

Tabulka 8. Rozpis jednotlivých pinů mikropočítače Atmel AT89C2051.

Typ I/O	Popis	Označení	Číslo pinu
Vstupy:	Termostat elektrického boileru	T_E	p3.4
	Termostat solárního boileru	T_S	p3.3
	Přepínač období ZIMA/LÉTO	Z/L	p1.2
	Přepínač Automatické/Manuální řízení	A/M	p1.3
	Přepínač Zapnuto/Vypnuto servo pro PLYN	M_P	p1.6
	Přepínač Zapnuto/Vypnuto servo pro ELEKTRO	M_E	p1.5
	Přepínač Zapnuto/Vypnuto servo pro SOLÁR	M_S	p1.4
	Přepínač Automatika čerpadla plynového okruhu	$M_{\text{čpA}}$	p1.0
	Přepínač Manual čerpadla plynového okruhu	$M_{\text{čpM}}$	p1.1
	Přepínač čerpadla hlavního solárního okruhu	$M_{\text{čs}}$	p1.7
Výstupy:	Servo pro Plynový boiler	S_P	p3.2
	Servo pro Elektrický boiler	S_E	p3.1
	Servo pro Solární boiler	S_S	p3.0
	Čerpadlo pro hlavní solární okruh	$\check{C}_S$	p3.7
	Čerpadlo pro oběh vody plynového okruhu	$\check{C}_P$	p3.5

Z velké nabídky mikropočítačů jsem si zvolil typ od firmy ATMEL. Kódové označení tohoto mikropočítače je AT89C2051.

3.1.1 Parametry

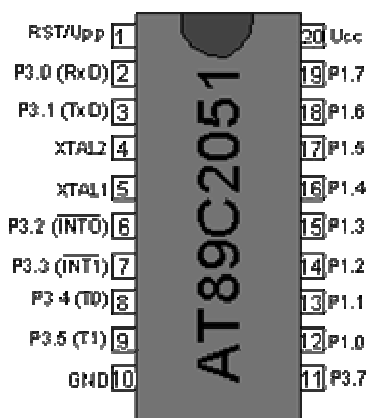
- Kompatibilita s mikropočítači řady MCS-51TM
- Programová paměť Flash velikosti 2 KB, zaručený počet přeprogramování je 1000 cyklů.
- Rozsah napětí: 2,7 – 6V
- Oscilátor 0Hz – 24MHz
- 128 x 8-Bitová vnitřní paměť RAM
- Mikrokontroler může ovládat 15 vstupně/výstupních linek, které mohou přímo budit LED (zkratový proud jedné linky je 20 mA, součet proudů všech výstupů nesmí překročit 80 mA).
- Dva 16-Bit Čítače/Časovače

- Šest zdrojů přerušení
- Plně duplexní programovatelný sériový kanál
- Interní analogový komparátor
- Dvouúrovňová ochrana dat
- 2 úsporné režimy napájení

3.1.2 Popis

Procesor AT89C2051 je určen pro jednoduché aplikace, u kterých není potřeba mnoho I/O pinů a paměti programu a dat. Oproti standardní řadě x51 obsahuje navíc analogový komparátor se vstupy na P1.0 a P1.1. Jeho výstup je softwarově přístupný na bitu P3.6. Velice dobře je propracován systém ochrany dat. Každý I/O pin může přímo budit LED diodu (až 20mA), takže nejsou potřeba žádné okolní posilovače. Připojením krystalu a dvou kondenzátorů získáme plně funkční mikropočítač.

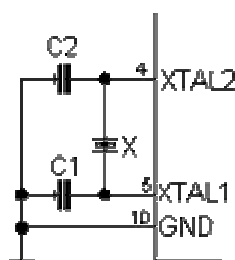
3.1.3 Zapojení vývodů mikropočítače



Obr. 14 Zapojení vývodů mikropočítače AT89C2051 v pouzdru DIP20.

V_{cc} a **GND** slouží pro připojení napájecího napětí (2,7 až 6 V).

XTAL1 a **XTAL2** slouží pro připojení krystalu. Pokud místo krystalu použijeme vnější hodinový signál, ponecháme XTAL2 nezapojený a hodinový signál připojíme na vývod XTAL1.



Obr. 15 Zapojení krystalu.

Hodnoty kondenzátorů použité pro dva druhy krystalů: $C1 = C2 = 30 \pm 10 \text{ pF}$ pro krystaly, $C1 = C2 = 30 \pm 10 \text{ pF}$ pro keramické rezonátory

Port 1 (P1.0 až P1.7) obsahuje 8 vstupně/výstupních linek. Vývody P1.2 až P1.7 jsou opatřeny vnitřními zdvihacími rezistory (pull-up), které zajišťují definovanou logickou úroveň (log. 1) těchto vývodů i v případě, že nejsou nikam připojeny. Vývody P1.0 a P1.1 nejsou zdvihacími rezistory opatřeny. Je-li třeba zajistit definovanou logickou úroveň těchto vstupů, musí být zdvihací rezistory připojeny z vnějšku. Vývod P1.0 (AIN0) je neinvertujícím vstupem a P1.1 (AIN1) je invertujícím vstupem vnitřního analogového komparátoru (výstup komparátoru je k dispozici na z vnějšku nedostupném vývodu P3.6).

Port 3 (P3.0 až P3.5 a P3.7) obsahuje 7 vstupně/výstupních linek. Všechny jsou opatřeny zdvihacím rezistory. Linka P3.6 není dostupná z vnějšku a je připojena na výstup analogového komparátoru. Port 3 sdružuje také funkce spojené s vnitřními periferiemi (viz následující tabulka).

Tabulka 9. Vnitřní periferie portu 3.

Vývod	Druhá funkce
P3.0	RxD (sériový vstup)
P3.1	TxD (sériový vstup)
P3.2	INT0non (vstup vnějšího přerušení číslo 0)
P3.3	INT1non (vstup vnějšího přerušení číslo 1)
P3.4	T0 (vnější vstup čítače/časovače číslo 0)
P3.5	T1 (vnější vstup čítače/časovače číslo 1)
Alternativní význam vývodů portu P3	

Sériový kanál

Mikropočítač AT89C2051 obsahuje plně duplexní sériový kanál (příjem i vysílání může probíhat současně).

Pro práci se sériovým kanálem jsou určeny registry:

- Řídicí registr SCON (konfiguruje vlastnosti sériového kanálu).
- Datový registr SBUF (slouží pro příjem/vysílání znaku).
- Bit SMOD v registru PCON ovlivňuje přenosovou rychlost.
- Registr SBUF slouží pro příjem/vysílání znaku. Zápis znaku do SBUF způsobí jeho vyslání (při správné konfiguraci sériového kanálu). Podobně čtením SBUF získáme přečtený znak (pokud je příjem znaku povolen).

Úlohu bitů řídicího registru SCON vysvětluje následující tabulka.

Tabulka 10. Byty řídicího registru SCON.

7	6	5	4	3	2	1	0
SM0	SM1	SM2	REN	TB8	RB8	TI	RI

Tabulka 11. SM0, SM1 - režim sériového kanálu

mód	SM0	SM1	režim	přenosová rychlost
0	0	0	8bitový posuvný registr	OSC/12
1	0	1	8bitový asynchronní přenos	čítač/časovač 1
2	1	0	9bitový asynchronní přenos	OSC/64 nebo OSC/32
3	1	1	9bitový asynchronní přenos	čítač/časovač 1

SM2 - povolení

tzv. víceprocesorové komunikace

REN - povolení příjmu

TB8 - vysílaný 9. bit (v režimech 2,3)

RB8 - přijatý 9. bit (v režimech 2,3)

TI - indikace vyprázdnění vysílacího registru, v režimu 0 je aktivován na konci vysílání 8. bitu v režimech 1,2 a 3 je aktivován na začátku stop-bitu tento bit se nastaví hardwarově, nuluje se programově.

RI - indikace naplnění vysílacího registru, v režimu 0 je aktivován po přijetí 8. bitu.

Nejčastěji se používá režim (mód) 1. Jedná se o osmibitový asynchronní přenos dat. Bity se vysílají na TxD (P3.1) a přijímají na RxD (P3.0). Přenos začíná start-bitem ("log. 0"), následuje 8 datových bitů (v pořadí od nejméně významného) a poslední je stop-bit ("log. 1"). Přenosová rychlost je dána přetečením časovače 1.

Pro časovač 1 nastavený do režimu 2 je přenosová rychlost PR:

$$PR = \left(\frac{2^{SMOD}}{32} \right) \left[\frac{f_0}{12 \cdot (256 - TH1)} \right] \quad (16)$$

kde:

f_0 je hodinový kmitočet mikropočítače (až 24 MHz), TH1 je obsah registru TH1 časovače 1.

Bit **SMOD** v registru PCON umožňuje zdvojnásobit přenosovou rychlost v režimech 1,2,3.

Přenosové rychlosti vysílače a přijímače při asynchronním přenosu (režimy 1,2 a 3) nemusí být shodné, nesmí se však vzájemně lišit o více než $\pm 5\%$ [7]

3.2 Schéma elektronického zapojení a návrh plošného spoje

3.2.1 Software pro návrh elektronického zapojení a plošného spoje

Pro návrh elektronického zapojení a návrh plošného spoje jsem si zvolil software EAGLE firmy Cadsoft Computer GmbH.

EAGLE je program určený pro kreslení schémat a návrh desek plošných spojů, který se skládá ze tří modulů:

- **Layout Editor:** je základním modulem programu Eagle a je současně kompletním programem pro interaktivní návrh plošných spojů, který vyniká jednoduchostí obsluhy. Maximálně 16 vrstev desky, palcové i metrické jednotky, návrh z i bez netlistu (Eagle, Orcad, Pads). Uživatelsky definovaný DRC pro kontrolu dodržení návrhových pravidel. Knihovna součástek s možností doplňování a modifikování. Podporuje SMT a montáž na obou stranách desky. Automatické vyplňování měděných ploch. Výstupy na tiskárny, plotry, fotoplotry a vrtačku.

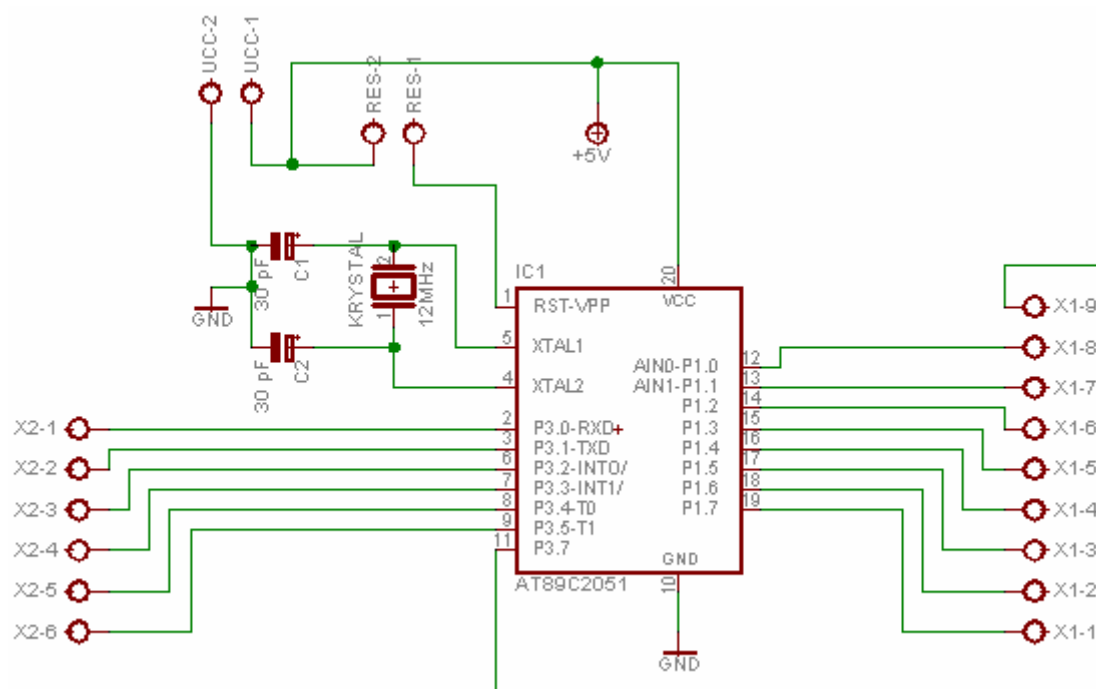
- **Schema:** modul pro kreslení schematických zapojení. Elektrická kontrola správnosti nakresleného schématu. Jedním повеlem se přechází do návrhu desky plošných spojů, přičemž si program automaticky generuje potřebné informace o schématu (netlist a partlist), zpětné anotace schéma-deska a naopak. Ke své činnosti vyžaduje modul Layout Editor!
- **Autrouter:** propoje automaticky vzdušné spoje podle nastavených parametrů. Přemísťuje nevhodně položené spoje (rip-up, re-route). Pracuje na celé desce, či jenom na jednom spoji. Ke své činnosti vyžaduje modul Layout Editor!

EAGLE pracuje pod operačními systémy Windows, Linux a Mac OS a komunikuje v angličtině nebo v němčině. [8]

3.2.2 Schéma elektronického zapojení

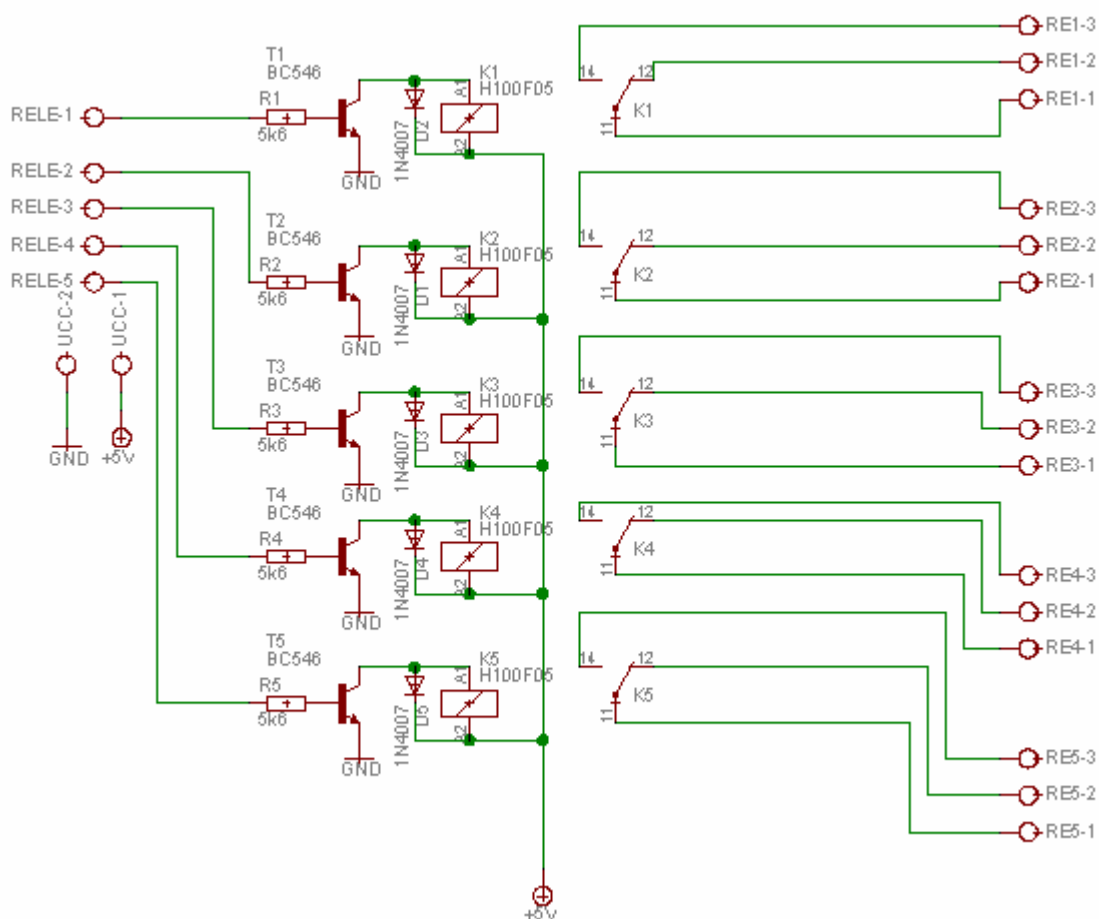
Zapojení elektronického obvodu jsem si rozdělil na tři části, abych oddělil mikropočítač od relé a abych omezil co nejvíce křížení elektronických spojů.

Obvod s mikropočítačem – ve schématu je zapojen mikropočítač AT89C2051, dále je do obvodu zapojen krystal s frekvencí 12 MHz, který spolu s dvěma kondenzátory 30pF udávají takt mikropočítače.



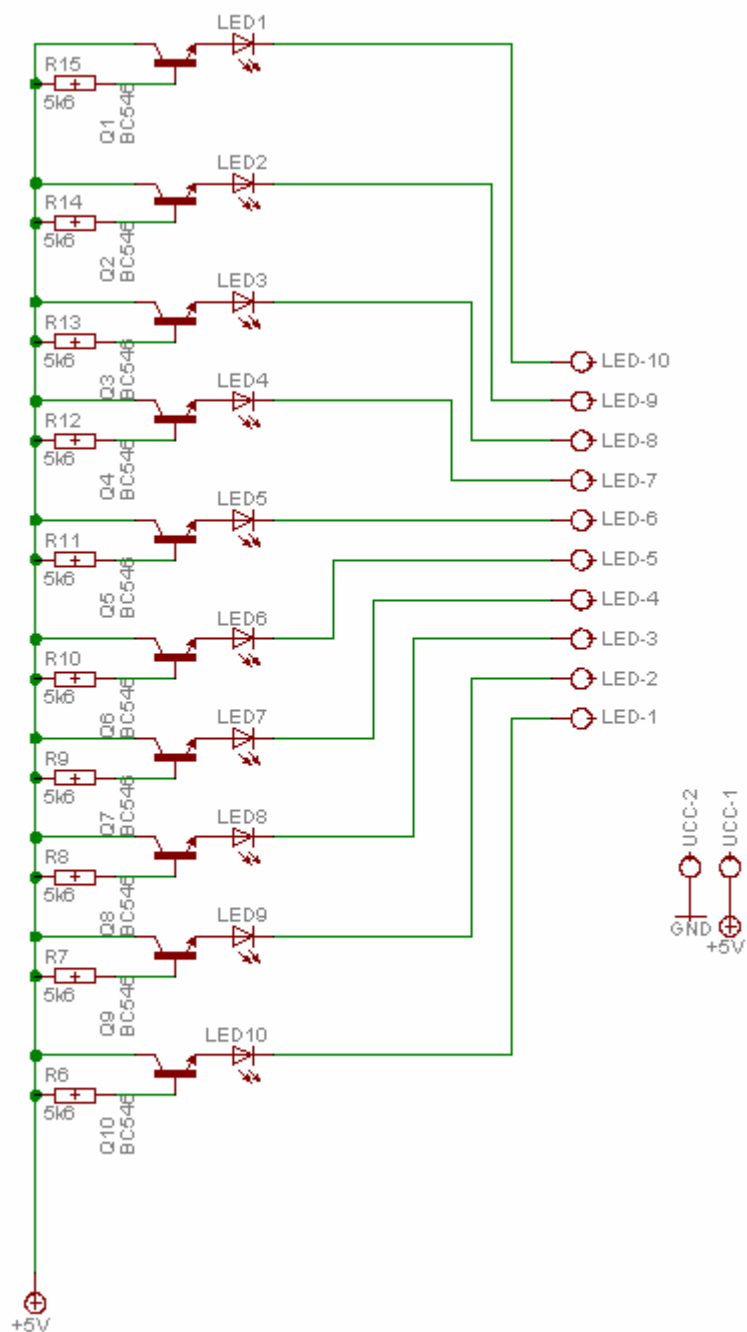
Obr. 16 Schéma elektronického zapojení s mikropočítačem.

Obvod s relé – V obvodu jsou použity NPN tranzistory BC546, aby se posílily výstupy mikropočítače pro sepínání relé. Před každý tranzistorem jsem musel zařadit rezistor o hodnotě $R=5600\ \Omega$, aby nedošlo ke zničení tranzistoru vlivem budícího napětí z výstupu mikropočítače. Dále je mezi kontakty cívky relé paralelně vřazena dioda, aby nedošlo opět ke zničení tranzistoru vlivem indukovaného napětí na svorkách cívky relé.



Obr. 17 Schéma elektronického zapojení s relátký.

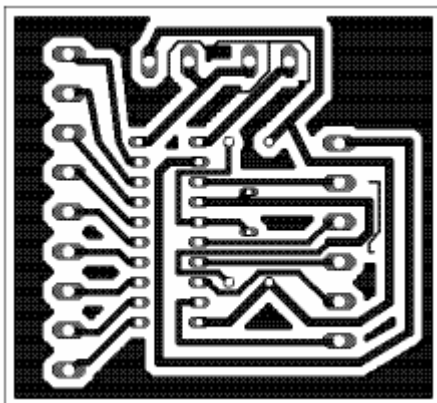
Obvod se světelnými diodami (LED) – V obvodu jsou opět použity NPN tranzistory BC546, aby se posílily výstupy mikropočítače pro rozsvícení LED. Před každý tranzistorem jsem musel zařadit odpor o hodnotě $R=5600\ \Omega$, aby nedošlo ke zničení tranzistoru vlivem budícího napětí z výstupu mikropočítače.



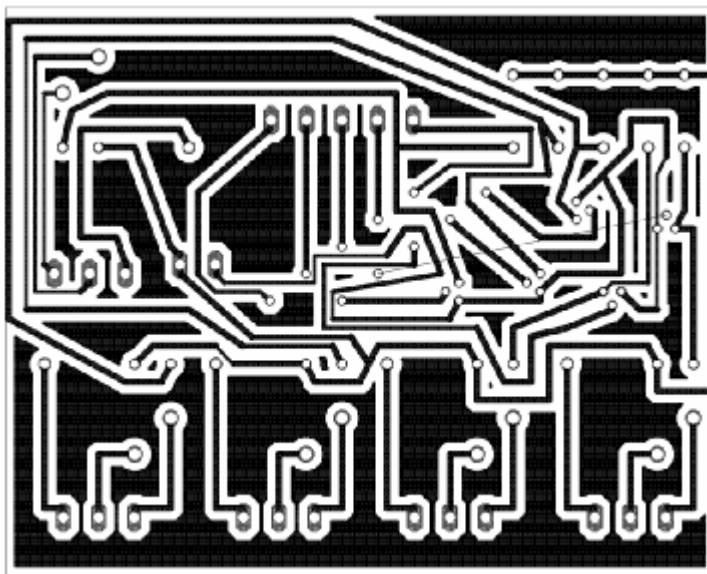
Obr. 18 Schéma elektronického zapojení s LED.

3.2.3 Návrh plošného spoje

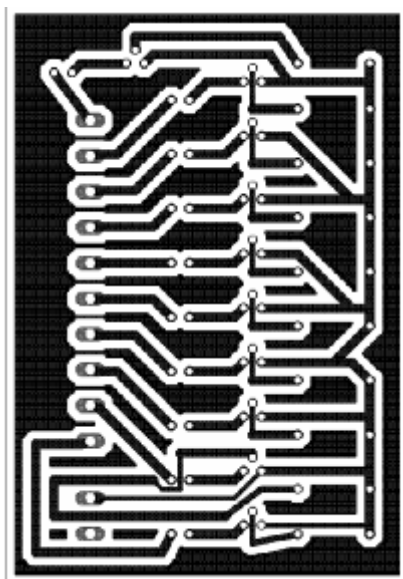
Při návrhu plošných spojů jsem vycházel z výše uvedených elektronických schémat (viz. Obr. 15,16,17). K návrhu jsem použil *autorouter*, což je modul programu Eagle. Modulem *autorouter* jsem nechal automaticky vykreslit pole plošného spoje. Dále jsem použil funkci *polygon* k vyplnění nevyužitých míst mědí (viz. Obr. 18,19,20 – velké černé plochy)



Obr. 19 Plošný spoj pro mikropočítač.



Obr. 20 Plošný spoj pro relátka.



Obr. 21 Plošný spoj pro LED.

3.2.4 Seznam použitých elektronických součástek

V elektronickém zapojení (viz Obrázek č. 19 a 21) jsem použil následující elektronické součástky:

Tabulka 12. Seznam použitých elektronických součástek pro plošné spoje.

Označení ve schematu	Hodnota (označení)	Popis	Cena/kus s DPH [kč]	Počet kusů
IC1	AT89C2051	mikrokontroler ATMEL, pouzdro DIL20	29,5	1
Q1	12 MHz	krystal HC-49/U, 30ppm, -10/+60 °C	12,5	1
C1,C2	33 pF	kondenzátory, 50V 5%	1,2	2
R1-R15	5k6	odpor, metalický, 0,4W 1%	0,8	15
T1-T5 Q1-Q10	BC546	tranzistor NPN; 65V; 0,1A; 0,5W; B:110-220	1,2	15
D1, D2, D3 D4, D5	1N4007	univerzální dioda 1A/1000V	1,2	5
K1, K2, K3 K4, K5	RELEH100F05	Relé 1U, 6A, 250 V AC, 5V, 89R	36	5
LED1-LED10	LED	LED 5mm, červené a zelené	2,5	10

Celková cena: 285,4 Kč

Tabulka 13. Seznam použitých elektronických součástek pro ovládací panel.

Označení	Popis	Cena/kus s DPH [Kč]	Počet kusů
LED	LED 5mm, červené a zelené	2,5	12
P-KNX1	dvoupolohový páčkový přepínač	9,5	5
P-B100G	dvoupolohový kolébkový přepínač	11	1
P-B100G	třípolohový kolébkový přepínač	11	2
P-S1317BF	červený spínací knoflík	6,5	1
5k6	odpor, metalický, 0,4W 1%	0,8	22

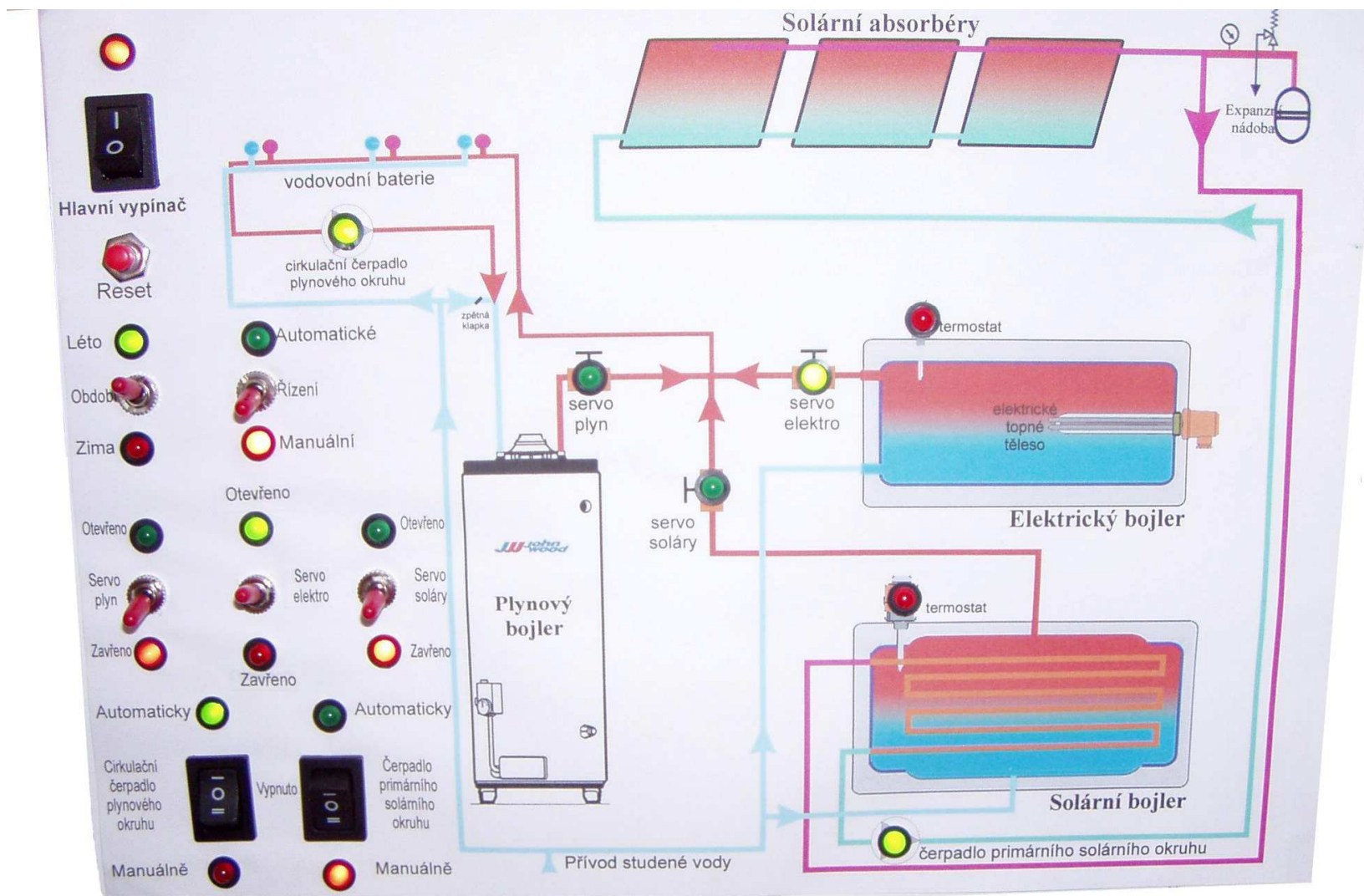
Celková cena: 134,6 Kč

Celková cena použitých elektronických součástek je 420 Kč.

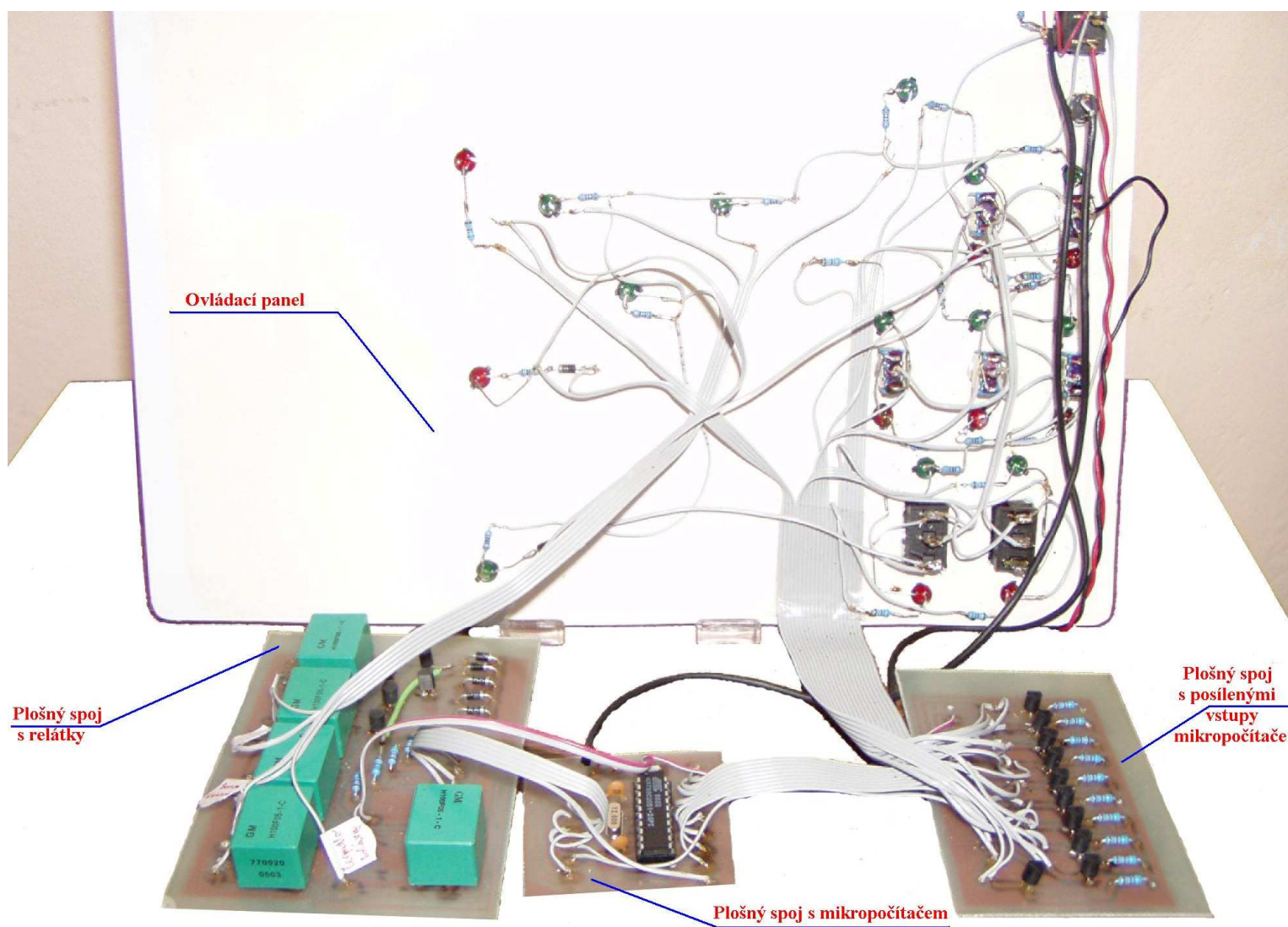
Započítáme-li do celkové ceny i náklady na plošné spoje (1dm² jednovrstvého plošného spoje stojí 125 Kč), stabilizovaný napájecí zdroj 5V (180 Kč), cín, kalafuna a propojovací vodiče (cca 80 Kč), potom celkové náklady činí zhruba 990 Kč.

3.3 Ovládací panel

Ovládací panel je vytvořen z PVC o rozměrech 21x30cm. Kvůli snadnému ovládání a přehlednosti je na přední straně ovládacího panelu umístěno schéma řízené soustavy (viz. Obr.1) , do kterého jsou vsazeny kontrolní diody a přepínače (viz Obr. 22). Na zadní straně ovládacího panelu jsou umístěny propojovací vodiče a ke každé kontrolní LED je připojen odpor (viz. Obr. 23). Důvodem je, že použité LED jsou konstruovány na napětí 5V a při maximálním napětí $U_{\max} = 6V$ by mohlo dojít k destrukci diody.



Obr. 22 Ovládací panel – přední strana



Obr. 23 Ovládací panel – zadní strana

ZÁVĚR

Výsledkem práce je funkční řídicí systém pro řízení tří servo pohonů a dvou cirkulačních čerpadel. Systém se skládá ze tří plošných spojů, které jsou propojeny s ovládacím panelem. První plošný spoj je osazen mikropočítačem jenž je řízen programem, který je naprogramován podle uživatelem zadaného algoritmu. Druhý plošný spoj je osazen odpory a tranzistory, které slouží k posílení vstupů mikropočítače. Třetí plošný spoj je osazen též tranzistory a odpory k posílení výstupů mikropočítače a pěti relé, které spínají napájecí napětí servo pohonů a čerpadel. Plošné spoje jsou navzájem propojeny mezi sebou a zároveň s ovládacím panelem. Ovládací panel je osazen šesti dvupolohovými přepínači, dvěma třípolohovými přepínači, jedním spínačem a dvaceti dvěma červenými a zelenými kontrolními světelnými diodami (LED). Ovládací panel bude zasazen do závěsné skříně, která obsahuje 5V stabilizovaný napájecí zdroj a mikropočítač, který je zde propojen s reálnou soustavou. Ovládací panel je zobrazen na obrázcích č.22 a č.23.

Správnou funkci programu a tím i celé soustavy si lze ověřit pomocí pravdivostních tabulek (viz. příloha č.2).

Celkové pořizovací náklady řídicího systému dané soustavy činí cca 990 Kč.

SEZNAM POUŽITÝCH ZDROJŮ

- 1) HEINZ, Ladener. FRANK, Späte. Solární zařízení, 1. vyd., Praha : Grada, 2003. 267 s., ISBN 80-247-0362-9
- 2) TRUXA, Jan. BERANOVSKÝ, Jiří.: Alternativní zdroje pro váš dům. 2. vyd., ERA Group, 2003. 125 s., ISBN 80-86517-89-6
- 3) MATOUŠEK, David. Práce s mikrokontroléry ATMEL AT89C2051, 1. vyd., BEN – technická literatura, 2002. 248 s., ISBN: 80-7300-048-2
- 4) Ekosolaris: solární zařízení [online]. [cit. 2005-3-12]. Dostupný z URL: <<http://www.ekosolaris.cz>>.
- 5) Automatizační technika [online]. [cit. 2005-4-23]. Dostupný z URL: <<http://www.spszl.cz>>.
- 6) Software určený pro programování mikropočítačů řady I51 [online]. [cit. 2005-3-18] Dostupný z URL: <www.promislbc.cz>.
- 7) Popis mikrokontroléru ATMEL AT89C2051 – anglický manuál [online]. [cit. 2005-3-22]. Dostupný z URL: <<http://www.atmel.com/dyn/resources/>>.
- 8) Software určený pro návrh plošných spojů Eagle [online]. [cit. 2005-3-18]. Dostupný z URL: <<http://www.eagle.cz>>.

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

LED – Light Emiting Diode – dioda emitující světlo

I/O – input / output – vstupně / výstupní

SEZNAM OBRÁZKŮ

Obr. 1 Schéma soustavy určené pro přípravu teplé užitkové vody	7
Obr. 2 Užití textilně plastového absorbéru pro ohřev bazénu	9
Obr. 3 Samotížná soustava.....	10
Obr. 4 Hnaná sestava	10
Obr. 5 Elektricky vyhřívaný boiler	11
Obr. 6 Elektrické topné těleso	12
Obr. 7 Plynem vyhřívaný boiler	14
Obr. 8 Termostat KR 80	16
Obr. 9 Servo pohon BELIMO NM-24 SR.....	17
Obr. 10 Schéma priority jednotlivých zdrojů tepelné energie.....	18
Obr. 11 Hardwarové řešení pro čerpadlo primárního solárního okruhu.....	26
Obr. 12 Větvení programu pro řízení čerpadla plynového oběhu.	28
Obr. 13 Hlavní okno programu μ Scope.....	31
Obr. 14 Zapojení vývodů mikropočítače AT89C2051 v pouzdru DIP20.....	39
Obr. 15 Zapojení krystalu.	40
Obr. 16 Schéma elektronického zapojení s mikropočítačem.....	43
Obr. 17 Schéma elektronického zapojení s relátkou.	44
Obr. 18 Schéma elektronického zapojení s LED.....	45
Obr. 19 Plošný spoj pro mikropočítač.	46
Obr. 20 Plošný spoj pro relátka.	46
Obr. 21 Plošný spoj pro LED.....	47
Obr. 22 Ovládací panel – přední strana	49
Obr. 23 Ovládací panel – zadní strana.....	50

SEZNAM TABULEK

Tabulka 1. Pravdivostní tabulka pro automatické řízení, letní období.....	21
Tabulka 2. Pravdivostní tabulka pro automatické řízení, zimní období.....	22
Tabulka 3. Pravdivostní tabulka pro manuální ovládání, letní období.....	23
Tabulka 4. Pravdivostní tabulka pro manuální ovládání, zimní období.....	25
Tabulka 5. Pravdivostní tabulka pro čerpadlo primárního solárního okruhu.....	26
Tabulka 6. Pravdivostní tabulka pro čerpadlo primárního solárního okruhu – po úpravě ..	27
Tabulka 7. Pravdivostní tabulka pro řízení čerpadla plynového okruhu.....	27
Tabulka 8. Rozpis jednotlivých pinů mikropočítače Atmel AT89C2051.	38
Tabulka 9. Vnitřní periferie portu 3.....	40
Tabulka 10. Byty řídicího registru SCON.	41
Tabulka 11. SM0, SM1 - režim sériového kanálu.....	41
Tabulka 12. Seznam použitých elektronických součástek pro plošné spoje.	47
Tabulka 13. Seznam použitých elektronických součástek pro ovládací panel.....	48

SEZNAM PŘÍLOH

Příloha 1: Soubor instrukcí pro mikropočítač řady I51

Příloha 2: Tabulky pro ověření funkce řídicího systému

Příloha 1

Soubor instrukcí pro mikropočítač řady I51

ACALL adr11 $PC(10\div 0) \leftarrow \text{adr11}$

Volání podprogramu uvnitř 2kB adresového prostoru (adresa je 11bitová). Instrukce uloží návratovou adresu do zásobníku s tím, že nejprve uloží nižší a potom vyšší byte. Volaný podprogram musí ležet uvnitř 2kB stránky, v které leží instrukce následující po instrukci ACALL.

ADD A, <zdrojová hodnota>

Sečtení střadače se zdrojovou hodnotou.

ADD A, Rn $(A) \leftarrow (A) + (Rn)$

ADD A, @Ri $(A) \leftarrow (A) + ((Ri)), i=0,1$

ADD A, adresa $(A) \leftarrow (A) + (\text{adresa})$

ADD A, #data $(A) \leftarrow (A) + \text{data}$

Instrukce přičte obsah adresované hodnoty ke střadači a výsledek v něm ponechá. Adresovanou hodnotou může být registr aktivní banky registrů, přímo uvedená adresa (8 bitů), nepřímo adresované paměťové místo registrem R0 nebo R1 (obsah registru určuje adresu místa jehož obsah se bude přičítat) nebo přímo uvedená hodnota data.

ADDC A, <zdrojová hodnota >

Sečtení střadače s hodnotou a příznakem přenosu.

ADDC A, Rn $(A) \leftarrow (A) + (Rn) + (C)$

ADDC A, @Ri $(A) \leftarrow (A) + ((Ri)) + (C), i=0,1$

ADDC A, adresa $(A) \leftarrow (A) + (\text{adresa}) + (C)$

ADDC A, #data $(A) \leftarrow (A) + \text{data} + (C)$

Instrukce přičte obsah adresované hodnoty a příznakový bit přenosu C ke střadači a výsledek v něm ponechá.

AJMP adr11 $PC(10\div 0) \leftarrow \text{adr11}$

Krátký nepodmíněný skok na adresu uvnitř 2kB stránky.

ANL <cílová hodnota >, <zdrojová hodnota >

Operace logického součinu mezi cílovou a zdrojovou hodnotou.

ANL A, Rn $(A) \leftarrow (A) \text{ AND } (Rn), n=0,1,\dots,7$

ANL A, @Ri $(A) \leftarrow (A) \text{ AND } ((Ri)), i=0,1$

ANL A, adresa $(A) \leftarrow (A) \text{ AND } (\text{adresa})$

ANL A, #data $(A) \leftarrow (A) \text{ AND } \text{data}$

ANL adresa, A $(\text{adresa}) \leftarrow (\text{adresa}) \text{ AND } (A)$

ANL adresa, #data $(\text{adresa}) \leftarrow (\text{adresa}) \text{ AND } \text{data}$

Instrukce ANL provede logický součin mezi odpovídajícími bity cílového a zdrojového bytu a výsledek uloží do cílové hodnoty. Je-li cílovým bytem výstupní brána (přímá adresa), pak se operace provede mezi výstupním registrem a zdrojovým bytem (nikoli vstupními signály).

ANL C, <zdrojový bit>

Logický součin bitů.

ANL C, bit $(C) \leftarrow (C) \text{ AND } (\text{bit})$

ANL C, /bit $(C) \leftarrow (C) \text{ AND } (\text{negace bit})$

Logický součin příznaku přenosu C s přímo adresovaným bitem. Výsledek operace se uloží do příznaku C. Je-li před adresou bitu lomítko, potom hodnota bitu bude před operací negována. Adresovaný bit lze adresovat jenom přímou adresou.

CJNE <cílová hodnota >, <zdrojová hodnota >, relativní adresa

Porovnej a skoč, když se operandy nerovnájí.

CJNE A, adresa, relativní adresa $(PC) \leftarrow (PC) + 3$ Je-li $(A) \neq (\text{adresa})$,

pak $(PC) \leftarrow (PC) + \text{relativní adresa}$

jinak $(PC) \leftarrow (PC) + 2$

CJNE A, #data, relativní adresa $(PC) \leftarrow (PC) + 3$ Je-li $(A) \neq \text{data}$

pak $(PC) \leftarrow (PC) + \text{relativní adresa}$

jinak $(PC) \leftarrow (PC) + 2$

CJNE Rn, #data, relativní adresa $(PC) \leftarrow (PC) + 3$ Je-li $(Rn) \neq \text{data}, n=0,1,\dots,7$

pak $(PC) \leftarrow (PC) + \text{relativní adresa}$

jinak (PC) $\leftarrow$ (PC) + 2

CJNE @Ri, #data, relativní adresa (PC) $\leftarrow$ (PC) + 3 Je-li ((Ri)) $\neq$ data, i=0,1

pak (PC) $\leftarrow$ (PC) + relativní adresa

jinak (PC) $\leftarrow$ (PC) + 2

Instrukce porovná střadač nebo registr nebo hodnotu adresovanou registry R0, R1 se zdrojovou hodnotou (adresa, data). V případě jejich nerovnosti provede relativní skok (maximálně o +127 nebo -128byťů) na požadovanou adresu. Adresa skoku se vypočte přičtením relativního posunu k čítači instrukcí, který byl nejprve třikrát inkrementován v důsledku čtení prováděné instrukce. Je-li cílová hodnota menší než zdrojová je zároveň nastaven příznak přenosu C=1, jinak ho vynuluje C=0.

CLR A (A) $\leftarrow$ 0

Instrukce vynuluje obsah střadače.

CLR C (C) $\leftarrow$ 0

CLR bit (bit) $\leftarrow$ 0

Instrukce vynuluje adresovaný bit.

CPL A (A) $\leftarrow$ $\neg$ (A) = $\neg$ (A)

Instrukce neguje každý bit střadače a vytváří tak jeho jednotkový doplněk.

CPL C (C) $\leftarrow$ NOT (C) = $\neg$ (C)

CPL bit (bit) $\leftarrow$ NOT (bit) = $\neg$ (bit)

Instrukce neguje (invertuje) adresovaný bit.

DA A Dekadická korekce střadače.

Instrukce koriguje obsah střadače po binárním sčítání dvou dekadických čísel vyjádřených v BCD kódu tak, aby výsledek opět vytvořil dvě čtyřbitová BCD čísla. Je-li hodnota na nižších čtyřech bitech >9 nebo AC = 1, potom se ke střadači přičte hodnota 6. Je-li hodnota na vyšších čtyřech bitech >9 nebo C = 1, potom se ke střadači přičte hodnota 60H.

DEC hodnota Dekrementace registru nebo paměťového místa

DEC A (A) $\leftarrow$ (A) - 1

DEC Rn (Rn) $\leftarrow$ (Rn) - 1, n = 0,1,...,7

DEC adresa (adresa) $\leftarrow$ (adresa) - 1

DEC @Ri ((Ri)) $\leftarrow$ ((Ri)) - 1, i = 0,1

Instrukce odečte od obsahu adresovaného paměťového místa hodnotu jedna. Po zmenšení hodnoty 00H dojde k podtečení na hodnotu FFH. Dekrementace obsahu výstupní brány zmenšuje obsah přečtený z registru brány a nikoliv ze vstupně/výstupních vodičů.

DIV AB (A) $\leftarrow$ celá část (A) / (B)

(B) $\leftarrow$ zbytek (A) / (B)

Instrukce provádí celočíselné dělení obsahu střadače s obsahem registru B. Celá část podílu zůstává ve střadači, zbytek (nikoli desetinná část) zůstává v registru B. Při dělení nulou se nastaví příznak přetečení OV = 1.

DJNZ Rn, relativní adresa (Rn) $\leftarrow$ (Rn) - 1, n = 0,1,...,7

Je-li (Rn) $\neq$ 0, pak (PC) $\leftarrow$ (PC) + relativní adresa

jinak (PC) $\leftarrow$ (PC) + 2

Instrukce odečte od adresovaného registru jedničku a zjistí zda výsledek je nulový. Je-li výsledek nenulový provede skok na definovanou adresu.

DJNZ adresa, relativní adresa (adresa) $\leftarrow$ (adresa) - 1

Je-li (adresa) $\neq$ 0, pak (PC) $\leftarrow$ (PC) + relativní adresa

jinak (PC) $\leftarrow$ (PC) + 2

Instrukce odečte od adresovaného paměťového místa jedničku a zjistí zda je výsledek nulový. Je-li výsledek nenulový provede skok na definovanou adresu.

INC hodnota Inkrementace registru nebo paměťového místa

INC A (A) $\leftarrow$ (A) + 1

INC Rn (Rn) $\leftarrow$ (Rn) + 1, n = 0,1,...,7

INC adresa (adresa) $\leftarrow$ (adresa) + 1

INC @Ri ((Ri)) $\leftarrow$ ((Ri)) + 1, i = 0,1

Instrukce přičte k obsahu adresovaného paměťového místa hodnotu jedna. Po zvětšení hodnoty FFH dojde k přetečení na hodnotu 00H. Inkrementace obsahu výstupní brány zvětšuje obsah přečtený z registru brány a nikoli ze vstupně/výstupních vodičů.

INC DPTR (DPTR) $\leftarrow$ (DPTR) + 1

Instrukce přičte jedničku k registrovému páru DPH a DPL, které vytváří 16-bitový ukazatel datové paměti DPTR. Dojde-li při přičtení k přetečení u registru DPL ($FFH \rightarrow 00H$) potom je přičtena jednička k registru DPH. Přičítání jedničky k DPTR probíhá modulo 2^{16} ($FFFFH + 1 \rightarrow 0000H$). Instrukce INC DPTR je jedinou 16-bitovou instrukcí v instrukčním souboru procesoru 8051.

JB bit, relativní adresa Je-li bit = 1, potom skoč

Instrukce testuje adresovaný bit a v případě jeho nastavení (log. 1) provede skok na adresu kterou vypočte jako součet čítače instrukcí a relativní adresy.

JBC bit, relativní adresa Je-li bit = 1, potom skoč a nuluj bit ($bit \leftarrow 0$)

Instrukce testuje adresovaný bit a v případě jeho nastavení (log. 1) provede skok na adresu, kterou vypočte jako součet čítače instrukcí a relativní adresy a vynuluje testovaný bit.

JC relativní adresa Je-li C = 1, potom skoč

Instrukce testuje příznak přenosu a v případě jeho nastavení (log. 1) provede skok na adresu, kterou vypočte jako součet čítače instrukcí a relativní adresy.

JMP @A + DPTR $(PC) \leftarrow (A) + (DPTR)$

Instrukce nepřímého nepodmíněného skoku na adresu určenou 16-bitovým součtem obsahu střadače (8 bitů bez znaménka) s obsahem ukazatele datové paměti DPTR.

JNB bit, relativní adresa Je-li bit = 0, potom $(PC) \leftarrow (PC) + \text{relativní adresa}$

Instrukce testuje adresovaný bit a v případě jeho nulovosti (log. 0) provede skok na adresu, kterou vypočte jako součet čítače instrukcí a relativní adresy.

JNC relativní adresa Je-li C = 0, potom $(PC) \leftarrow (PC) + \text{relativní adresa}$

Instrukce testuje příznak přenosu a v případě jeho nulovosti (log. 0) provede skok na adresu, kterou vypočte jako součet čítače instrukcí a relativní adresy.

JNZ relativní adresa Je-li $(A) \neq 0$, potom $(PC) \leftarrow (PC) + \text{relativní adresa}$

Instrukce testuje obsah střadače a v případě jeho nenulovosti provede skok na adresu, kterou vypočte jako součet čítače instrukcí a relativní adresy.

JZ relativní adresa Je-li $(A) = 0$, potom $(PC) \leftarrow (PC) + \text{relativní adresa}$

Instrukce testuje obsah střadače a v případě jeho nulovosti provede skok na adresu, kterou vypočte jako součet čítače instrukcí a relativní adresy.

LCALL adr16 $PC(15 \div 0) \leftarrow \text{adr } 16$

Instrukce vykoná nepodmíněné volání podprogramu z přímo uvedené adresy ve svém druhém (vyšší byte) bytu. Před uložením přechtené adresy do čítače instrukcí (PC) se uloží návratová adresa (současný stav (PC) = adresa následující instrukce) do zásobníku.

LJMP adr16 $PC(15 \div 0) \leftarrow \text{adr } 16$

Instrukce vykoná nepodmíněný skok na adresu přímo uvedenou ve svém druhém (vyšší byte) a třetím (nižší byte) bytu. Adresa může ležet kdekoli v 64 kB adresovém prostoru.

MOV <cílová hodnota>, <zdrojová hodnota>

Přesun bytu z paměťového místa na jiné paměťové místo ve vnitřní datové paměti.

MOV A, Rn $(A) \leftarrow (Rn)$, kde $n = 0, 1, \dots, 7$

MOV A, @Ri $(A) \leftarrow ((Ri))$, kde $i = 0, 1$

MOV Rn, A $(Rn) \leftarrow (A)$, kde $n = 0, 1, \dots, 7$

MOV @Ri, A $((Ri)) \leftarrow (A)$, kde $i = 0, 1$

MOV A, adresa $(A) \leftarrow (\text{adresa})$

MOV A, #data $(A) \leftarrow (\text{data})$

MOV Rn, #data $(Rn) \leftarrow (\text{data})$, kde $n = 0, 1, \dots, 7$

MOV @Ri, #data $((Ri)) \leftarrow (\text{data})$, kde $i = 0, 1$

MOV adresa, A $(\text{adresa}) \leftarrow (A)$

MOV Rn, adresa $(Rn) \leftarrow (\text{adresa})$, kde $n = 0, 1, \dots, 7$

MOV adresa, Rn $(\text{adresa}) \leftarrow (Rn)$, kde $n = 0, 1, \dots, 7$

MOV @Ri, adresa $((Ri)) \leftarrow (\text{adresa})$, kde $i = 0, 1$

MOV adresa, @Ri $(\text{adresa}) \leftarrow ((Ri))$, kde $i = 0, 1$

MOV adresa1, adresa2 $(\text{adresa1}) \leftarrow (\text{adresa2})$

MOV adresa, #data $(\text{adresa}) \leftarrow (\text{data})$

Instrukce MOV přesune obsah zdrojové hodnoty do cílové hodnoty bez ovlivnění jakýchkoliv příznaků.

MOV <cílový bit>, <zdrojový bit>

Přesune hodnoty mezi přenosem C a daným bitem.

MOV C, bit $(C) \leftarrow (\text{bit})$

MOVC A, @A + <bázový registr>

Přesuň byte z paměti programu

MOVC A, @A + DPTR $(A) \leftarrow ((A) + (DPTR))$

MOVC A, @A + PC $(A) \leftarrow ((A) + (PC))$

Instrukce přesune byte z programové paměti (operační kód nebo konstantu) do střadače. Adresa místa, jehož obsah se přesouvá, získáme jako 16-bitový součet obsahu střadače (8 bitů) a ukazatele dat DPTR nebo čítače instrukcí PC. Pro případ čítače instrukcí je jeho obsah před provedením instrukce inkrementován (ukazuje následující instrukci).

MOVX <cílová hodnota>, <zdrojová hodnota>

Přesun byte z/do vnější datové paměti

MOVX A, @DPTR $(A) \leftarrow ((DPTR))$

MOVX @DPTR, A $((DPTR)) \leftarrow (A)$

MOVX A, @Ri $(A) \leftarrow ((Ri))$, kde $i = 0,1$

MOVX @Ri, A $((Ri)) \leftarrow (A)$, kde $i = 0,1$

Instrukce přesune byte ze/do střadače do/z vnější paměti dat. Instrukce mohou využívat 16-bitovou nebo 8-bitovou nepřímou adresu. V prvním případě se vysílá adresa uložená v DPTR na bránu P2 (DPH) a bránu P0 (DPL). V případě druhém se vysílá na bránu P0 adresa uložená v registru R0 nebo R1 a na bráně P2 zůstává hodnota naposledy zapsaná.

MUL AB $(A) \leftarrow$ nižší byte $(A)*(B)$, $(B) \leftarrow$ vyšší byte $(A)*(B)$

Instrukce vynásobí dvě osmibitová čísla bez znaménka uložená ve střadači a registru B. Je-li součin větší než hodnota 255 (FFH), nastaví se příznakový bit přetečení OV = 1. V opačném případě jej vynuluje.

NOP

Prázdná operace

Kromě čítače instrukcí neovlivňuje instrukce žádné registry a příznaky.

ORL <cílová hodnota>, <zdrojová hodnota>

Operace logického součtu mezi cílovou a zdrojovou hodnotou.

ORL A, Rn $(A) \leftarrow (A) \text{ OR } (Rn)$, kde $n = 0,1,...,7$

ORL A, @Ri $(A) \leftarrow (A) \text{ OR } ((Ri))$, kde $i = 0,1$

ORL A, adresa $(A) \leftarrow (A) \text{ OR } (\text{adresa})$

ORL A, #data $(A) \leftarrow (A) \text{ OR } \text{data}$

ORL adresa, A $(\text{adresa}) \leftarrow (\text{adresa}) \text{ OR } (A)$

ORL adresa, #data $(\text{adresa}) \leftarrow (\text{adresa}) \text{ OR } (\text{data})$

Instrukce ORL provede logický součet mezi odpovídajícími bity cílového a zdrojového bytu a výsledek uloží do cílové hodnoty. Pro operaci s výstupní bránou (přímá adresa) se operace provede mezi výstupním registrem a zdrojovým bytem a nikoliv vstupními signály.

ORL C, <zdrojový bit>

Logický součet bitů

ORL C, bit $(C) \leftarrow (C) \text{ OR } (\text{bit})$

ORL C, /bit $(C) \leftarrow (C) \text{ OR } /(\text{bit})$

Logický součet příznaku přenosu C s přímo adresovaným bitem. Výsledek operace se uloží do příznaku C. Je-li před adresou bitu lomítko, potom hodnota bitu bude před operací negována. Adresovaný bit lze adresovat jenom přímou adresou.

POP adresa $(\text{adresa}) \leftarrow ((SP))$, $(SP) \leftarrow (SP) - 1$

Instrukce vyzvedne obsah vrcholu zásobníku a uloží jej na adresované paměťové místo. Pak odečte od ukazatele zásobníku jedničku.

PUSH adresa $(SP) \leftarrow (SP) + 1$, $((SP)) \leftarrow (\text{adresa})$

Instrukce přičte k ukazateli zásobníku jedničku a potom uloží obsah adresovaného místa do vrcholu zásobníku (zásobník je vždy ve vnitřní datové paměti).

RET $(PC(15 \div 8)) \leftarrow ((SP))$, $(SP) \leftarrow (SP) - 1$

$(PC(7 \div 0)) \leftarrow ((SP))$, $(SP) \leftarrow (SP) - 1$

Instrukce návratu z podprogramu vyzvedne ze zásobníku dva byty a uloží je do čítače instrukcí (PC).

Nejprve vyjme vyšší byte a potom nižší byte a odečte od ukazatele zásobníku hodnotu dvě.

RETI $(PC(15 \div 8)) \leftarrow ((SP))$, $(SP) \leftarrow (SP) - 1$

$(PC(7 \div 0)) \leftarrow ((SP))$, $(SP) \leftarrow (SP) - 1$

Instrukce návratu z obslužného podprogramu přerušení vyzvedne ze zásobníku dva byty a uloží je do čítače instrukcí (PC). Nejprve vyjme vyšší byte a potom nižší byte a odečte od ukazatele zásobníku hodnotu dvě.

Nakonec se povolí přijetí žádosti o přerušení se stejnou nebo nižší úrovní priority. Stavové slovo (jako 8048) se automaticky neobnovuje.

RL A $(A(x+1)) \leftarrow (A(x)), \text{ pro } x = 0,1,\dots,6$

$(A(0)) \leftarrow (A(7))$

Instrukce osmibitové logické rotace střadače o jednu pozici vlevo.

RLC A $(A(x+1)) \leftarrow (A(x)), \text{ pro } x = 0,1,\dots,6$

$(A(0)) \leftarrow (C), (C) \leftarrow (A(7))$

Instrukce devítibitové logické rotace střadače a příznaku přenosu o jednu pozici vlevo.

RR A $(A(x)) \leftarrow (A(x+1)), \text{ pro } x = 0,1,\dots,6$

$(A(7)) \leftarrow (A(0))$

Instrukce osmibitové logické rotace střadače o jednu pozici vpravo.

RRC A $(A(x)) \leftarrow (A(x+1)), \text{ pro } x = 0,1,\dots,6$

$(A(7)) \leftarrow (C), (C) \leftarrow (A(0))$

Instrukce devítibitové logické rotace střadače a příznaku přenosu o jednu pozici vpravo.

SETB <bit>

Nastav bit

SETB C $(C) \leftarrow 1$

SETB bit $(\text{bit}) \leftarrow 1$

Instrukce nastaví přímo adresovaný bit na log. 1.

SJMP relativní adresa $(PC) \leftarrow (PC) + 2, (PC) \leftarrow (PC) + \text{relativní adresa}$

Instrukce realizuje krátký nepodmíněný skok na definovanou adresu, která se vypočítá jako součet čítače instrukcí ukazujícího na adresu následující instrukce a posunu (-128; 127) určeného relativní adresou.

SUBB A, <zdrojová hodnota >

Odečítání hodnoty a příznaku přenosu od střadače.

SUBB A, Rn $(A) \leftarrow (A) - (Rn) - (C), \text{ kde } n = 0,1,\dots,7$

SUBB A, @Ri $(A) \leftarrow (A) - ((Ri)) - (C), \text{ kde } i = 0,1$

SUBB A, adresa $(A) \leftarrow (A) - (\text{adresa}) - (C)$

SUBB A, #data $(A) \leftarrow (A) - \text{data} - (C)$

Instrukce odečte obsah adresované hodnoty od střadače včetně příznaku přenosu (výpůjčky) a výsledek v něm ponechá. Adresovanou hodnotou může být registr aktivní banky, přímo uvedená adresa, nepřímo adresované paměťové místo registrem R0 nebo R1 (obsah registru určuje adresu místa jehož obsah budu odečítat) nebo přímo uvedená hodnota data. Je-li při výpočtu vyžadována v bitu b_7 výpůjčka, je příznakový bit C nastaven, v opačném případě je vynulován.

SWAP A $(A) \leftarrow (A(3 \div 0) * 16 - A(7 \div 4))$

Instrukce prohodí navzájem obsah nižšího a vyššího „půlbytu“ střadače. Instrukce je shodná s osmibitovou rotací o čtyři bity vpravo nebo vlevo.

XCH A, < hodnota >

Zaměň obsah střadače a adresové hodnoty.

XCH A, Rn $(A) \leftrightarrow (Rn), n = 0,1,\dots,7$

XCH A, @Ri $(A) \leftrightarrow ((Ri)), i = 0,1$

XCH A, adresa $(A) \leftrightarrow (\text{adresa})$

Instrukce vymění obsah střadače a určeného registru nebo adresovaného paměťového místa.

XCHD A, @Ri $A(3 \div 0) \leftrightarrow (Ri(3 \div 0)), i = 0,1$

Instrukce prohodí navzájem obsah nižší „půlbyte“ střadače s nepřímo adresovaným paměťovým místem vnitřní paměti RAM.

XRL <cílová hodnota >, <zdrojová hodnota >

Operace neekvivalence (EX – OR) mezi cílovou a zdrojovou hodnotou.

XRL A, Rn $(A) \leftarrow (A) \oplus (Rn), n = 0,1,\dots,7$

XRL A, @Ri $(A) \leftarrow (A) \oplus ((Ri)), i = 0,1$

XRL A, adresa $(A) \leftarrow (A) \oplus (\text{adresa})$

XRL A, #data $(A) \leftarrow (A) \oplus \text{data}$

XRL adresa, A $(\text{adresa}) \leftarrow (\text{adresa}) \oplus (A)$

XRL adresa, #data $(\text{adresa}) \leftarrow (\text{adresa}) \oplus \text{data}$

Instrukce Xrl provede operaci EX – OR odpovídajícími hodnotami cílového a zdrojového bytu a výsledek uloží do cílové hodnoty. Je-li cílovým bytem výstupní brána (přímá adresa), potom se operace provede mezi výstupním registrem a zdrojovým bytem (nikoliv vstupními signály). [5]

Příloha 2

Tabulky pro ověření funkce řídicího systému

Tabulka 1 - Manuální řízení servo pohonů.

Období	Manuální přepínače			Servo		
	Plyn	Elektro	Soláry	Plyn	Elektro	Soláry
Léto	0	0	0	0	0	0
	0	0	1	0	0	1
	0	1	0	0	1	0
	0	1	1	0	0	1
	1	0	0	1	0	0
	1	0	1	0	0	1
	1	1	0	0	1	0
	1	1	1	0	0	1
Zima	0	0	0	0	0	0
	0	0	1	0	0	0
	0	1	0	0	1	0
	0	1	1	0	1	0
	1	0	0	1	0	0
	1	0	1	1	0	0
	1	1	0	0	1	0
	1	1	1	0	1	0

Tabulka 2 - Automatické řízení servo pohonů.

Období	Termostat		Servo		
	Elektro	Soláry	Plyn	Elektro	Soláry
Léto	0	0	1	0	0
	0	1	0	0	1
	1	0	0	1	0
	1	1	0	0	1
Zima	0	0	1	0	0
	0	1	1	0	0
	1	0	0	1	0
	1	1	0	1	0

Tabulka 3 – Řízení čerpadla solárního okruhu

Režim	Čerpadlo solárního okruhu
Automatický	Zapnuto
Vypnuto	Vypnuto
Manuální	Zapnuto

Tabulka 4 – Řízení čerpadla plynového okruhu

Režim	Termostat		Čerpadlo plynového okruhu
	Elektro	Soláry	
Automatický	0	0	Zapnuto
	0	1	Vypnuto
	1	0	Vypnuto
	1	1	Vypnuto
Vypnuto	x	x	Vypnuto
Manuální	x	x	Zapnuto

(x – nemá vliv)