

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA STROJNÍHO INŽENÝRSTVÍ
ÚSTAV AUTOMATIZACE A INFORMATIKY

FACULTY OF MECHANICAL ENGINEERING
INSTITUTE OF AUTOMATION AND COMPUTER SCIENCE

ŘÍZENÍ CHODU KOSTELNÍCH ZVONŮ PROGRAMOVATELNÝM AUTOMATEM

SACRISTIAN BELL OPERATION CONTROL BY PROGRAMMABLE LOGIC CONTROLLER

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

LADISLAV KRÍŽEK

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. ZDENĚK NĚMEC, CSc.

BRNO 2008

Vysoké učení technické v Brně, Fakulta strojního inženýrství

Ústav automatizace a informatiky

Akademický rok: 2007/08

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

student(ka): Křížek Ladislav

který/která studuje v **bakalářském studijním programu**

obor: **Aplikovaná informatika a řízení (3902R001)**

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma bakalářské práce:

Řízení chodu kostelních zvonů programovatelným automatem

v anglickém jazyce:

Sacristian bell operation control by programmable logic controller

Stručná charakteristika problematiky úkolu:

Úkol řeší praktickou potřebu automatického řízení chodu specifické regulované soustavy s výrazným rezonančním charakterem pohybu.

Cíle bakalářské práce:

1. Seznamte se s problematikou automatického řízení chodu kostelního zvonu a dvojice zvonů.
2. Pro zadanou kostelní věž zpracujte a odzkoušejte programové vybavení k řízení dvojice zvonů pomocí programovatelného automatu Simatic S7-212.
3. Vycházejte z dosavadního provizorního stavu, zejména pohonů zvonů, z daného umístění snímačů a navrhňte eventuální varianty změn ke zdokonalení funkcí.
4. Navrhňte a realizujte fyzikální nebo počítačový model soustavy pro účely zkoušení řídicího programu, protože zkoušky na reálném objektu jsou možné jen omezeně.
5. Zhodnoťte vlastnosti řízení a kvalitu zvonění před úpravami a po navržených změnách.

Seznam odborné literatury:

Šmejkal, L.-Martinásková, M. PLC a automatizace. Praha: BEN, 1999.

Vedoucí bakalářské práce: Ing. Zdeněk Němec, CSc.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2007/08.

V Brně, dne **07. 05. 2008**

L.S.

^c
Seda



doc. RNDr. Ing. Miloš Šeda, Ph.D.
Ředitel ústavu

v. r. M. Doupovec

doc. RNDr. Miroslav Doupovec, CSc.
Děkan fakulty

LICENČNÍ SMLOUVA POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO

uzavřená mezi smluvními stranami:

1. Pan

Jméno a příjmení: **Ladislav Křížek**
Bytem: **398 11 Protivín, Ve Školce 724**
Narozen (datum a místo): **14.5.1979, Prachatice**
(dále jen „autor“)

a

2. Vysoké učení technické v Brně

Fakulta strojního inženýrství
se sídlem Technická 2896/2, 616 69 Brno
jejímž jménem jedná na základě písemného pověření děkanem fakulty:
Doc. RNDr. Ing. Miloš Šeda, Ph.D., ředitel ÚAI
(dále jen „nabyvatel“)

Čl. 1

Specifikace školního díla

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):
- ☐ disertační práce
 - ☐ diplomová práce
 - ☒ bakalářská práce
 - ☐ jiná práce, jejíž druh je specifikován jako
- (dále jen VŠKP nebo dílo)

Název VŠKP: Řízení chodu kostelních zvonů programovatelným automatem

Vedoucí/ školitel VŠKP: Ing. Zdeněk Němec, CSc.

Ústav: Ústav automatizace a informatiky

Datum obhajoby VŠKP: _____

VŠKP odevzdal autor nabyvateli v*:

- ☒ tištěné formě – počet exemplářů2.....
- ☒ elektronické formě – počet exemplářů2.....

* hodící se zaškrtněte

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.
3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.
4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

Článek 2

Udělení licenčního oprávnění

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užít, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizování výpisů, opisů a rozmnoženin.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti
 - ☒ ihned po uzavření této smlouvy
 - ☐ 1 rok po uzavření této smlouvy
 - ☐ 3 roky po uzavření této smlouvy
 - ☐ 5 let po uzavření této smlouvy
 - ☐ 10 let po uzavření této smlouvy(z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/ 1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

Článek 3

Závěrečná ustanovení

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísní a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne:23.5.2008.....

.....
Nabyvatel

.....
Autor

ANOTACE

V této práci je popsán zvon, jeho součásti, konstrukční předpoklady pro jeho správnou funkci, řízení chodu kostelních zvonů s ručním pohonem a řízení s pohonem rotačního elektromotoru. Dále je v této práci popsán programovatelný automat Siemens Simatic S7 – 200 CPU 221, který je součástí zvonícího stroje v kostele obce Bukovinka. Tento zvonící stroj a zvláště pak chod obou zvonů vykazují určité nedostatky, pro které jsou, na základě přání a podmínek provozovatele, připraveny dvě varianty programů pro zlepšení jejich správné funkce. Z těchto dvou variant byla jedna odzkoušena a odlazena přímo na zvonícím stroji. Tato varianta, na přání provozovatele, dodnes řídí chod dvojice zvonů v kostele obce Bukovinka.

ANOTATION

The bell is described, its parts and constructive suppositions for the right function in this work, sacristian bell operation control with hand-working and with the rotating electromotor. Programmable logic controller machine SIEMENS Simatic S7-200 CPU 221 is also described, which is the part of ringing machine in Bukovinka's church. This ringing machine and especially the mode of both bells have specific defect. According to owner's wishes and instructions were two variants prepared for this ringing machine to improve the working of couple bells. One of these variants was tested and just tuned at the ringing machine. This one variant follows the owner's wish, works in Bukovinka's church till now.

KLÍČOVÁ SLOVA

Kostelní zvon, automatické řízení, pohon, elektromotor, programovatelný automat, zvonící stroj, program

KEY WORDS

Sacristian bell, automatic operate, working, electromotor, programmable logic controller, ringing machine, program

Prohlášení:

Prohlašuji, že předložená bakalářská práce je mým původním autorským dílem, které jsem vypracoval samostatně za použití literatury a podkladů, které jsou v seznamu uvedeny.

V Protivíně 14. května 2008

Podpis:

Poděkování:

Na tomto místě bych chtěl poděkovat vedoucímu mé bakalářské práce Ing. Zdeňku Němcovi CSc. za pomoc a cenné rady v průběhu zpracování. Dále patří můj dík Ing. Petru Ševčíkovi za odbornou pomoc při řešení praktických úkolů automatického řízení.

OBSAH

ZADÁNÍ BAKALÁŘSKÉ PRÁCE.....	3
LICENČNÍ SMLOUVA.....	5
ANOTACE.....	7
OBSAH.....	13
SEZNAM POUŽITÝCH ZKRATEK A PROMĚNNÝCH.....	17
1 ÚVOD.....	19
2 ZVON A JEHO SOUČÁSTI.....	21
2.1 Zvon a jeho součásti.....	21
2.2 Výroba a zvuk zvonu.....	22
2.3 Srdce zvonu a jeho zavěšení.....	22
2.4 Teorie pohybu zvonu.....	23
2.4.1 Zvon jako matematické kyvadlo.....	23
2.4.2 Zvon jako fyzikální kyvadlo.....	25
2.5 Konstrukční předpoklady zvonů a jejich součástí.....	26
3 ŘÍZENÍ CHODU KOSTELNÍCH ZVONŮ A DVOJICE ZVONŮ.....	27
3.1 Ruční zvonění.....	27
3.2 Zvonící stroje.....	27
3.2.1 Zvonící stroj poháněný rotačním elektromotorem.....	27
3.3 Řízení chodu dvojice zvonů.....	28
4 PROGRAMOVATELNÝ AUTOMAT SIEMENS SIMATIC S7-200.....	29
4.1 Popis automatu.....	29
4.2 Pracovní režimy a způsob provádění programu.....	30
4.3 Přístup k datům.....	31
4.4 Řídící program.....	32
4.5 Vývojové prostředí STE7-Micro/WIN.....	32
4.6 Vybrané instrukce Simatic.....	34
5 ZVONÍCÍ STROJ V OBCI BUKOVINKA	37
5.1 Kostel v obci Bukovina.....	37
5.2 Zvonící stroj v objektu.....	38
6 ROZBOR PROBLÉMU.....	41
6.1 Nedostatky zvonícího stroje.....	41
6.1.1 Nesprávné vyvážení zvonů.....	41
6.1.2 Nevhodný tvar srdce zvonu.....	42
6.2 Chod zvonů s použitím původního programu.....	42
7 ŘEŠENÍ PROBLÉMU – ŘÍZENÍ CHODU ZVONŮ.....	45
7.1 Model pro zkoušení řídicího programu.....	45

7.2 Podmínky provozovatele.....	46
7.3 Řešení variantou 1.....	46
7.3.1 Vývojový diagram.....	46
7.3.2 Rozbor programu.....	50
7.3.3 Chod zvonů s použitím programu varianta 1.....	51
7.4 Řešení variantou 2.....	52
7.4.1 Vývojový diagram.....	52
7.4.2 Rozbor programu.....	56
8 POZNATKY ZÍSKANÉ PŘI ŘEŠENÍ, NAVRŽENÉ ZMĚNY.....	57
8.1 Simulace řízení chodu zvonu s proměnnou délkou impulzu.....	58
9 ZÁVĚR.....	61
SEZNAM POUŽITÉ LITERATURY.....	63
PŘÍLOHY.....	65

SEZNAM POUŽITÝCH PROMĚNNÝCH A ZKRATEK

B	-	Byte
CPU	-	Central Processing Unit
HW	-	Hardware
DW	-	Double Word
IMP11	-	Potřeba většího pracovního impulzu, zvon 1
IMP12	-	Potřeba menšího pracovního impulzu, zvon 1
IMP21	-	Potřeba většího pracovního impulzu, zvon 2
IMP22	-	Potřeba menšího pracovního impulzu, zvon 2
IS	-	Indukční snímač
IS11	-	1. zvon 1. indukční snímač
IS12	-	1. zvon 2. indukční snímač
IS21	-	2. zvon 1. indukční snímač
IS22	-	2. zvon 2. indukční snímač
KP	-	Krajní poloha
M1	-	Stykač motor 1
M2	-	Stykač motor 2
NT	-	Network
PLC	-	Programmable Logic Controller
RP	-	Rovnovážná poloha
ŘJ	-	Řídící jednotka
START1	-	Zvonu 1 byl udělen 1. impulz
START2	-	Zvonu 2 byl udělen 1. impulz
Stav1A	-	Zvon 1 je v klidu
Stav1B	-	Požadavek na startovací impulz, zvon 1
Stav1C	-	Požadavek na větší pracovní impulz, zvon 1
Stav1D	-	Požadavek na menší pracovní impulz, zvon 1
Stav2A	-	Zvon 2 je v klidu
Stav2B	-	Požadavek na startovací impulz, zvon 2
Stav2C	-	Požadavek na větší pracovní impulz, zvon 2
Stav2D	-	Požadavek na menší pracovní impulz, zvon 2
STRANA1A	-	Určení směru pohybu, zvon 1
STRANA2A	-	Určení směru pohybu, zvon 2
SW	-	Software
W	-	Word
ZV1	-	požadavek na zvonění zvon 1
ZV2	-	požadavek na zvonění zvon 2

1 ÚVOD

Snaha zvýšit produktivitu práce, snižovat jak spotřebu času, tak i náklady a nahrazovat lidskou sílu technikou je patrná ve všech odvětvích lidské činnosti již od nepaměti.

Podíváme-li se do historie, zjistíme, že v průběhu vývoje společnosti člověk nejprve začal zjednodušovat práce fyzicky namáhavé. Vymyslel, vyrobil a začal používat stroje. Tento proces nazýváme mechanizací.

Postupem času, s rozvojem techniky a vzrůstajícími nároky na řídicí činnosti se snaží si i tuto činnost zjednodušit, přenechat jí technice. Z pohledu industrializace jde o krok, který následuje po mechanizaci. Zatímco mechanizace poskytuje lidem k práci řízení, stroje, jež jim usnadňují práci, automatizace snižuje nutnost, aby byl člověk při vykonávání dané činnosti.

Modernizace pracovních postupů, vzrůstající podmínky kvality, přesnosti a efektivity zvyšují význam automatizaci pracovních procesů. Lidský faktor je postupně nahrazován technikou. Jelikož v porovnání s automaty, počítači a umělou inteligencí je člověk nepřesný a selhává. Samotná automatizace využívá široké spektrum technických prostředků. S jejich využitím probíhají operace pomocí člověkem vytvořených programů. Zároveň bývají tyto operace zpětně kontrolovány, a to opět technikou.

V našem případě se jedná o řízení chodu kostelních zvonů na zadaném objektu v obci Bukovinka. Ve věži kostela se nachází zvonící stroj s dvojicí zvonů. Jejich chod je řízen programovatelným automatem Siemens Simatic S7 řady 200 CPU 221. Zvukový projev každého zvonu je neharmonický a proto je třeba jej zlepšit.

2 ZVON A JEHO SOUČÁSTI

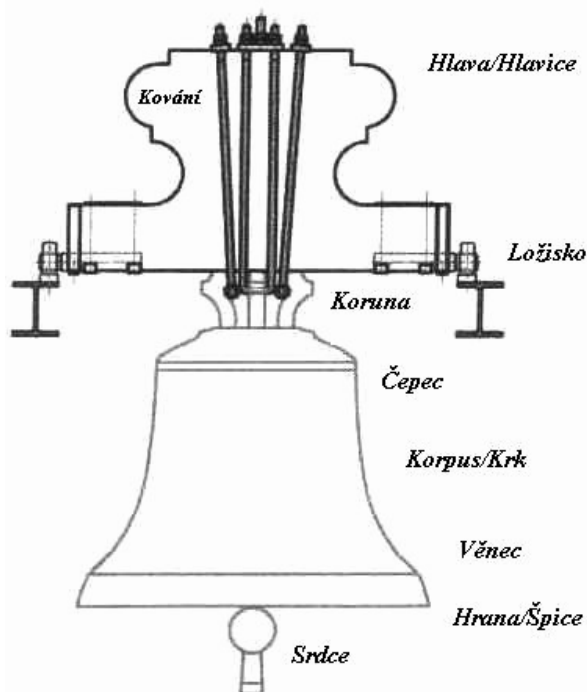
2.1 Zvon a jeho součásti

„Zvon je specifický hudební nástroj z kovu, širokého významu a uplatnění. Kromě své základní funkce zvukového nástroje je také technickým výtvořem, ztvárněním vnější podoby výtvarným dílem a pro svou značnou dokumentační hodnotu také historickou památkou..“

Petr Rudolf Manoušek, zvonář

Pokud bychom si samotný zvon představili jako obrovskou kovovou desku, tak poté základní tón desky je tím vyšší, čím je tato deska hrubší a menší. Tenké desky mají základní tóny poměrně hluboké. Mohou se chvět různým způsobem a tím reprodukovat různé zvuky. Toho se využívá v mikrofonech a reproduktorech. Velké desky, které jsou dost hrubé, mohou vydávat poměrně vysoký zvuk. Tento zvuk naroste do mohutnosti spolu s velikostí desky. Takovouto desku bychom mohli přirovnat ke kostelnímu zvonu. Tuto desku rozezná úder srdce (kladiva), zavěšeného v jeho středu. Vhodnou volbou tvaru zvonu lze dosáhnout konstantních tónů, které jsou příjemné a lahodné uchu. [1]

Samotný zvon je složen z několika základních částí, které jsou znázorněny na obrázku 1.



Obr. 1 Zvon a jeho součásti [2]

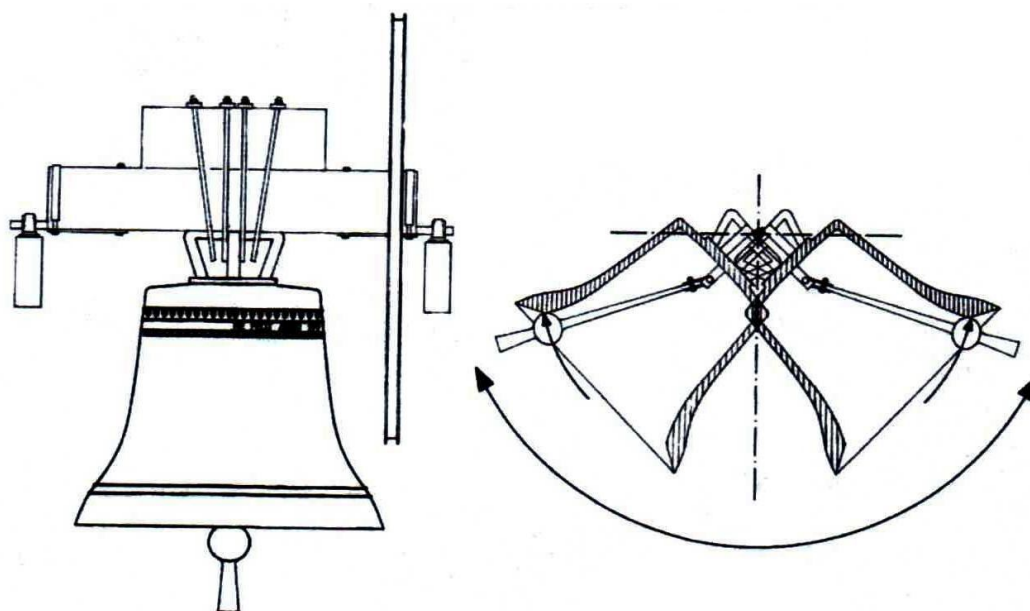
2.2 Výroba a zvuk zvonu

Výroba a technologie odlévání zvonu je dosti stará, zůstává ve svém principu stejná již po mnoho století. Každý zvon musí vydávat hlavní tón a všechny alikvotní tóny v přesném harmonickém vzestupu, což se docílí změnou rozměrů a síly zvonového žebra podle přesně stanovených propočtů. Každý zvonař musel svoji teorii propočtu zvonového žebra podrobit dlouholetým zkouškám, pokusům a změnám, než dosáhl s jistotou žádané kvality tónu. Výkresy zvonových žebířů jsou největším bohatstvím a duševním majetkem zvonaře, které si předávají z generace na generaci. Další faktor výrazně ovlivňující zvuk zvonu je použitý materiál, tzv. zvonovina. Slitina mědi a cínu v poměru 78% mědi a 22% cínu. Čím je poměr přesnější a čistota kovu obou kovů vyšší, tím je i samotný zvuk zvonu jasnější. [2][3]

2.3 Srdce zvonu a jeho zavěšení

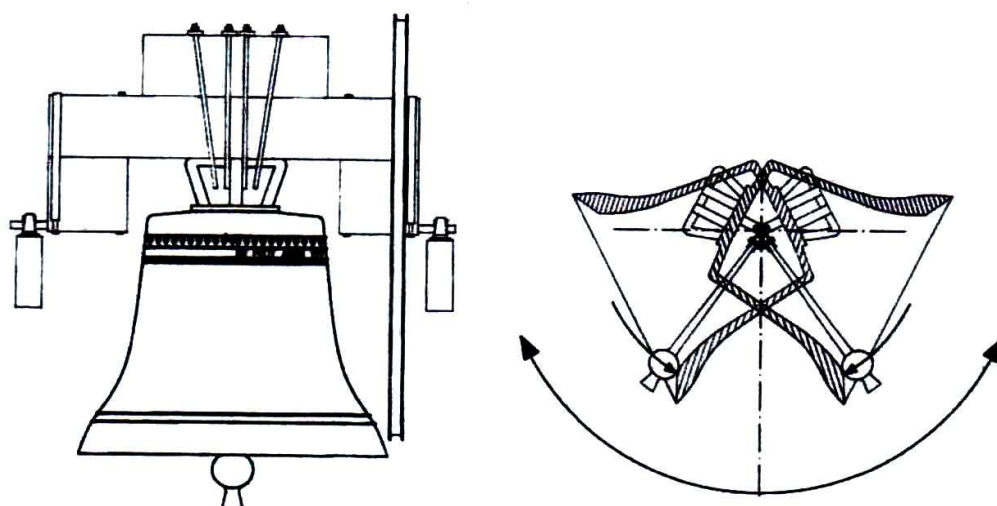
Srdce podstatně ovlivňuje funkci a životnost zvonu, a proto je po zvonu druhým nejdůležitějším prvkem. Zvon a srdce spolu tvoří z fyzikálního hlediska soustavu, pár dvou sprzęžených kyvadel, která spolu fungují za předpokladu dokonalého vyvážení a vzájemného sladění.

Existují dva základní způsoby zavěšení srdcí. Prvním způsobem je tzv. „srdce létající“. Tato varianta je pravděpodobně nejrozšířenější, protože při správném provedení jedině tento způsob umožňuje zvonu žádané vyznění po úderu. Při pohybu zvonu do krajní polohy se srdce pohybuje spolu se zvonem a udeří na horní straně zvonu v okamžiku, kdy zvon dosáhne krajní polohy. [2]



Obr. 2 Zavěšení srdce zvonu – „srdce létající“ [2]

Druhým způsobem zavěšení srdce je tzv. „srdce na padání“. Tato varianta není tak výrobně složitá, avšak i při velmi malém překročení potřebného úhlu výkyvu má srdce sklon k dvojúderům, nebo naopak k tlumení zvuku. Přesná regulace síly na zvon je velice obtížná, ne-li nemožná. A z toho plyne i vysoké riziko poškození zvonu. Toto srdce se nepohybuje se zvonem, pouze visí stále vertikálně a udeří v krajní poloze na spodní část zvonu. [2]



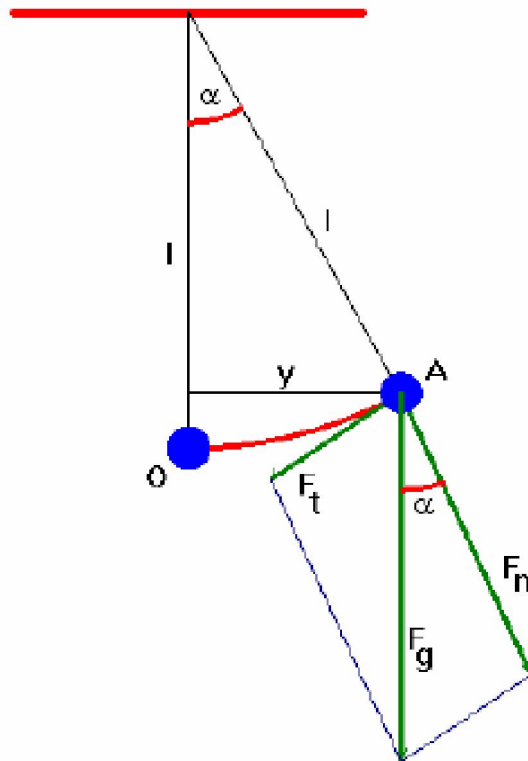
Obr. 3 Zavěšení srdce zvonu – srdce „na padání“ [2]

2.4 Teorie pohybu zvonu

2.4.1 Zvon jako matematické kyvadlo [1]

Pokud bychom zanedbali vliv srdce na kyv zvonu, tak zvon koná periodicky vratný pohyb, tedy pohyb kyvadla. Kmitavý pohyb kyvadel je velice složitý, a proto zavedeme jednoduchý model – matematické kyvadlo, které koná periodický vratný pohyb.

Matematické kyvadlo si představíme jako hmotný bod zavěšený na nehmotném a nedeformujícím se závěsu.



Obr. 4 Matematické kyvadlo

Na obrázku 4 je matematické kyvadlo zakresleno ve dvou polohách – v rovnovážné, kdy je hmotný bod po uvolnění v klidu a v krajní poloze, kdy došlo k uvedení kyvadla do pohybu.

Jestliže je hodnota $\alpha < 5^\circ$ platí:

$$\tau = \pi \sqrt{\frac{g}{l}} \quad (3.1)$$

Pokud platí, že doba kyvu τ je rovna polovině doby kmitu T , platí:

$$T = 2\pi \sqrt{\frac{l}{g}} \quad (3.2)$$

Frekvenci kmitání lze vyjádřit:

$$f = \frac{1}{2\pi} \sqrt{\frac{g}{l}} \quad (3.3)$$

Z uvedených rovnic je zřejmé, že frekvence a perioda harmonického pohybu závisí na délce závěsu a tíhovém zrychlení v daném místě. Pokud úhel $\alpha < 5^\circ$, tak pohyb kyvadla nezávisí na hmotnosti tělesa zavěšeného na kyvadle, a ani na rozkyvu kyvadla.

Kyvadlo se dá z hlediska energií popsat takto. Kinetická energie je maximální v krajní poloze. To je poloha, kdy se kyvadlo při jeho pohybu na malý okamžik zastaví a vychýlení tělesa je maximální. V tomto okamžiku je jeho rychlost rovna nule, tím i jeho kinetická energie je nulová. Potenciální energie je maximální. Poté se kyvadlo začne opět pohybovat vlivem gravitační síly směrem k rovnovážné poloze. To je poloha, kdy potenciální energie je nulová a kinetická energie je maximální. Těleso má v tomto bodě maximální rychlost. Poté se těleso začne pohybovat k druhé krajní poloze a začne tím snižovat svoji rychlost a tím i kinetickou energii. Ta se začne vlivem gravitační síly měnit v potenciální. To vše až do okamžiku, kdy těleso dosáhne druhé krajní polohy a potenciální energie je opět maximální a kinetická nulová. Poté se opět celý cyklus začne opakovat.

2.4.2 Zvon jako fyzikální kyvadlo[1]

Skutečná kyvadla se většinou velice odlišují od kyvadel matematických. Ta si lze představit jako množinu velkého počtu kyvadel matematických různých délek, tvořících jeden celek. Fyzikálním kyvadlem může být jakékoli tuhé těleso upevněné tak, aby se mohlo otáčet kolem vodorovné osy neprocházející těžištěm. Pohyb samotného zvonu je bližší pohybu fyzikálního kyvadla.

Pro dobu kmitu fyzikálního kyvadla platí:

$$T = 2\pi \sqrt{\frac{J}{mgh}} \quad (3.4)$$

Kde J je krouticí moment a m hmotnost kyvadla.

Pro frekvenci kmitání platí:

$$f = \frac{1}{2\pi} \sqrt{\frac{mgh}{J}} \quad (3.5)$$

Tyto vztahy platí pro nekonečně malé výchylky kyvadla, kde $\sin \alpha \approx \alpha$

2.5 Konstrukční předpoklady zvonů a jejich součástí

K tomu aby zvon správně zněl, jeho chod byl harmonický a pravidelný je potřeba, aby splňoval některá kritéria. Často se stávalo, že neodborným odléváním nebo zavěšením byla tato kritéria nedodržena. Proto zde uvedu nejdůležitější předpoklady pro správnou funkci zvonových soustav.[2]

- *Tvar a konstrukce zvonu* – tvar a konstrukce zvonu musí být navrženy vzhledem k akustickým požadavkům. Musí být dodržena technologie odlévání s použitím zvonoviny.
- *Tvar a konstrukce srdce* – v tomto případě musí být splněna podmínka hmotnostní, kdy hmotnost srdce zvonu odpovídá 3 – 5 % hmotnosti zvonu. Použitým materiálem je ocel, jejíž tvrdost je nižší než tvrdost zvonoviny. Technologie výroby je kování.
- *Správné vyvážení zvonu* – správné vyvážení zvonu je důležité pro harmonický chod zvonu. Nesprávné vyvážení zvonu se projevuje nevyváženým hlasovým projevem zvonu v jedné z krajních poloh – respektive zvon „bije na jednu stranu“ více. To je způsobeno menším výkyvem v této krajní poloze. Špatné vyvážení lze eliminovat vyvažovací pákou, na kterou se upevňují příslušná závaží. Existují dva nejčastější důvody špatného vyvážení zvonu.
 - Přesazení koruny zvonu – koruna zvonu je přesazena vůči ose zvonu
 - Přesazení zvonu vůči hlavici – osa zvonu je přesazen vůči ose hlavice
- *Správné zavěšení srdce vůči ose zvonu* - srdce zvonu musí být zavěšeno v ose zvonu. Pokud by bylo vůči ose zvonu přesazeno, projevovalo by se to opět nevyváženým hlasovým projevem zvonu v jedné krajní poloze. A tím by docházelo k tzv. „vytloukání“ zvonu. [2]

3 ŘÍZENÍ CHODU KOSTELNÍCH ZVONŮ A DVOJICE ZVONŮ

3.1 Ruční zvonění

Důležitým faktorem určující zvuk zvonu je úhel výkyvu zvonu, který musí být dostatečně velký na to, aby srdce udeřilo optimální rychlostí do věnce zvonu. Nesmí být příliš velký, aby se tento náraz následně uskutečnil, a zároveň nesmí být příliš malý, aby rázy zvonu a srdce nebyly příliš vysoké, a tím nedocházelo k poškození celé soustavy. Nelze jednoznačně určit, jak velký musí být tento úhel, který závisí na celé konstrukci zvonu a optimálních rozložení hmotností. Ale jednoznačně lze říci, že je závislý na zvukovém projevu celého zvonu. Zkušený zvoník podle citu reguluje celou soustavu pouze podle tohoto projevu a následně uděluje sílu na zvon. Jinou sílu (impulz) používá k rozhoupání zvonu, jinou k udržení zvonu v chodu. K udělení momentu zvonu dochází vždy, když se zvon vrací z krajní polohy ve směru pohybu zvonu, který je shodný se směrem pohybu zvoníka a to po úderu srdce do zvonu. [2]

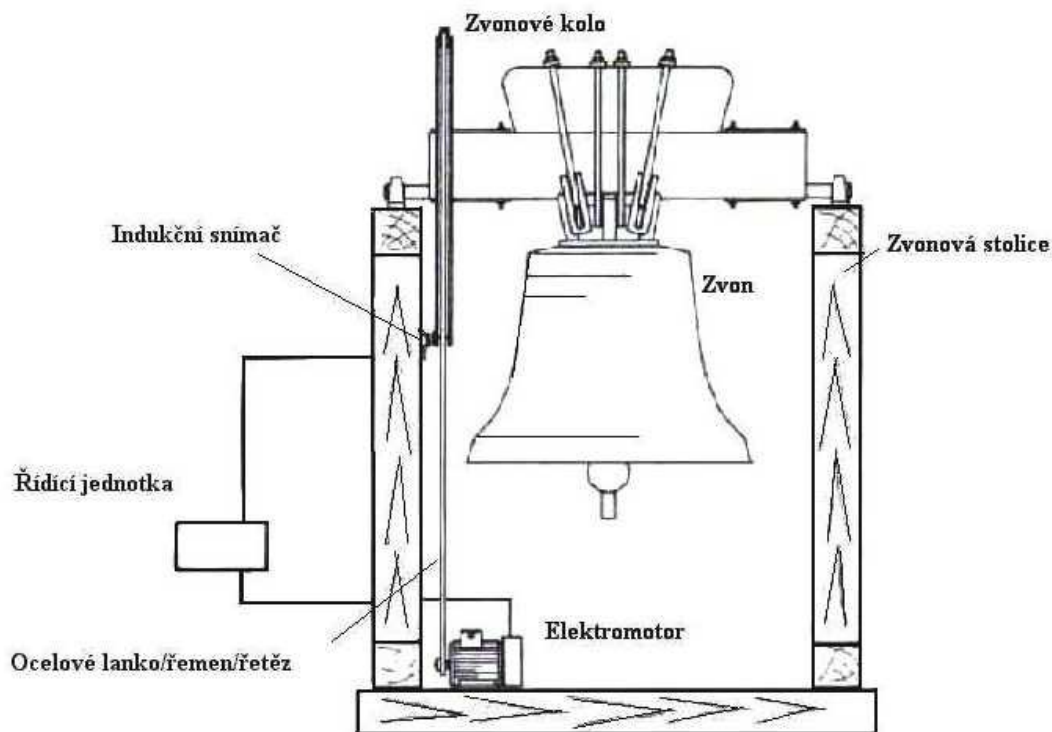
3.2 Zvonící stroje

Jak už jsem již zmínil, zvonící stroje mají za úkol nahradit ve své činnosti živého zvoníka. V současné době se používají dva typy zvonících strojů. Zvonící stroj s rotačním elektromotorem a zvonící stroj s lineárním motorem. Pokud bychom měli oba pohony porovnávat, tak pohon lineárním motorem je bezesporu šetrnější než pohon rotačním elektromotorem a lépe nahrazuje ruční zvonění. Přesnost s jakou je možno regulovat, však není tak vysoká jako u rotačního elektromotoru. Další nevýhodou je velikost síly, kterou lineární motor vyvodí. Ta je menší než u rotačních elektromotorů. To jsou důvody, proč se tento systém používá u malých zvonů do velikosti přibližně 200 kg. Větší zvony vyžadují přesnější regulaci a větší síly působící na zvon. Nevýhodou pohonu pracujícím na principu rotačního elektromotoru jsou větší rázy, které vznikají při udělení impulzu. U větších zvonů není však tento problém tak zásadní.[2][6]

3.2.1 Zvonící stroj poháněný rotačním elektromotorem

Řízení chodu kostelních zvonů pomocí rotačního elektromotoru a zvonového kola je dnes pravděpodobně nejrozšířenější. Jeho základem jsou rotační elektromotor a zvonové kolo spojené navzájem převodem. Ve většině případů se jedná o reverzní motor s možností pohonu na obě strany. Převod je realizován ocelovým lankem, řetězem či ozubeným řemenem. K vyhodnocení polohy zvonu se používají indukční snímače, které bývají umístěny tak, aby zaznamenaly okamžik průchodu zvonu rovnovážnou polohou.

Samozřejmě nejdůležitější součástí celého stroje je programovatelný automat. Ten bývá většinou umístěn mimo věž, aby nedošlo vlivem počasí k poškození samotného přístroje. Tento automat sleduje všechny naprogramované veličiny, které vyhodnocuje a následně spíná napájení elektromotoru. Takto řídí chod zvonu. [2]



Obr. 5 Výkres zvonícího stroje poháněného rotačním elektromotorem[2]

3.3 Řízení chodu dvojice zvonů

Samozřejmě pokud je ve zvonícím stroji umístěna dvojice zvonů s rozdílnými parametry a vzniká nám požadavek na zvonění obou zvonů současně, musíme regulovat chod každého zvonu nezávisle na sobě. Tato nezávislost plyne z velikosti zvonu a parametrů zvonícího stroje každého zvonu. Zvony s větší hmotností mají dobu periody kyvu větší, než zvony menší. A to je hlavní důvod, proč zvony jsou regulovány nezávisle a není tak možnost nijak sladit zvukový projev. Pokud bychom se pokoušeli sladit zvukový projev jednotlivých zvonů, museli bychom narušit periody kyvu jednotlivých zvonů. To znamená u menších zvonů prodlužovat periody kyvu a u větších bychom tuto periodu zmenšovali. Tím bychom narušili pravidelnost a velikost úderů v jednotlivých krajních polohách a mohlo by dojít k poškození jednotlivých zvonů, či celých zvonících strojů. Dalšími parametry ovlivňující chod kostelních zvonů jsou - způsoby zavěšení či tvar srdce, které ovlivňují periodu kyvu zvonu.

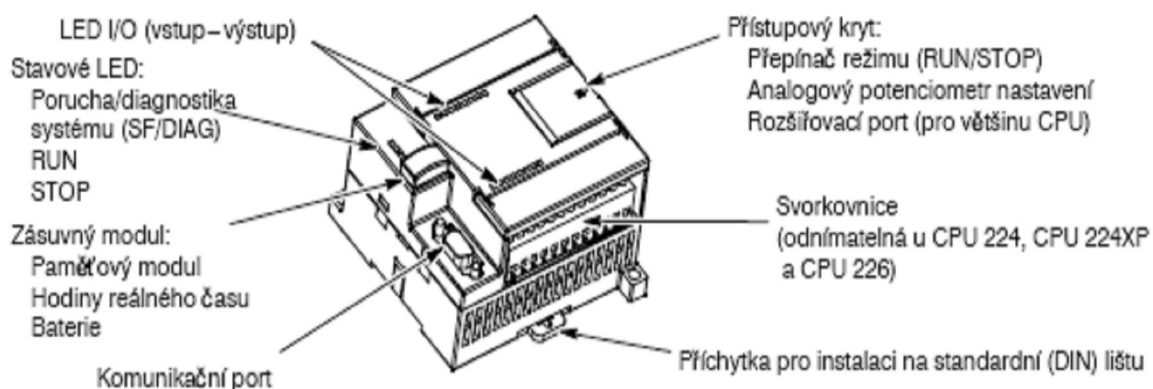
4 PROGRAMOVATELNÝ AUTOMAT SIEMENS SIMATIC S7-200 [4]

Firma Siemens řadí automaty S7-200 do skupiny malých řídicích systémů – mikrosystémů. Tyto automaty jsou určeny pro řízení malých aplikací. Vývojovým prostředím pro tyto automaty je software STEP7 - Micro/WIN. Tyto přístroje lze rozšířit o další moduly: digitální vstupy a výstupy, analogové vstupy a výstupy, komunikační procesory, speciální moduly a operátorské moduly.

4.1 Popis automatu S7-200

Samotný přístroj se skládá z několika základních částí:

- *Svorkovnice* – každý automat S7-200 má dvě svorkovnice – pro vstupní a pro výstupní hodnoty. Svorkovnice slouží k připojení zařízení, které nám budou dodávat informace – vstupní hodnoty a které budeme ovládat – výstupní hodnoty. Pro modely CPU 224, CPU 226 a CPU 226 XM je svorkovnice odnímatelná.
- *Indikační LED diody* – tyto diody určují stav jednotlivých vstupů/výstupů, a nabývají hodnot log.1 nebo 0.
- *Stavové LED diody* – tyto diody určují, v jakém režimu se automat nachází – RUN/STOP.
- *Zásuvný modul* – výměnný paměťový modul, do kterého lze ukládat informace
- *Komunikační port* – pomocí tohoto portu automat komunikuje s ostatními zařízeními
- *Přepínač režimů* – pomocí tohoto přepínače je možno manuálně přepnout automat do stavu RUN/STOP/TERM
- *Analogový potenciometr nastavení* – tuto funkci lze použít v programu, a tím manuálně regulovat další hodnoty
- *Rozšiřovací port* – slouží k připojení dalších rozšiřujících zařízení



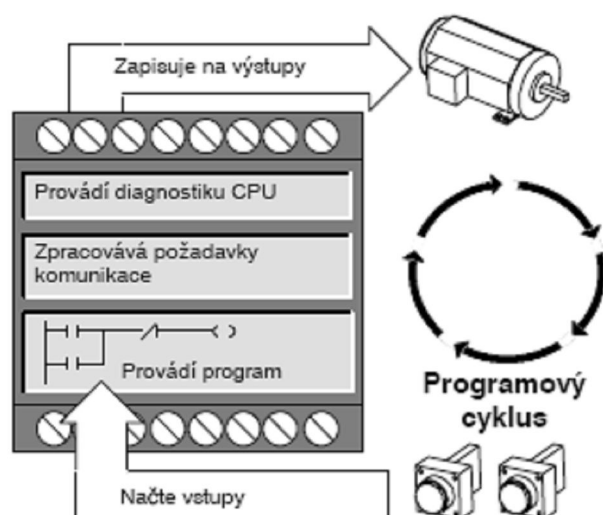
Obr. 6 Popis automatu S7-200

4.2 Pracovní režimy a způsob provádění programu

Automat S7-200 pracuje ve dvou pracovních režimech a to RUN a STOP. V režimu STOP můžeme nahrávat program nebo konfigurovat CPU. Při zapnutém režimu RUN se provádí příslušný program. Mezi těmito režimy lze přepínat manuálně pomocí přepínače, umístěného na přístroji, nebo pomocí STEP7-Microwin. K tomu abychom mohli přepínat z režimu RUN do režimu STOP softwarově, musí být přepínač v poloze RUN, nebo TERM.

Programovatelný automat vykonává cyklicky jednotlivý program, uložený v automatu a cyklicky provádí i řadu úloh s tím spojených. Toto cyklické provádění jednotlivých úloh se nazývá programový cyklus. Celý cyklus se skládá z následujících kroků:

- *Čtení aktuálních hodnot na vstupech* – aktuální hodnoty automat přečte a zapíše do registru obrazů vstupů
- *Provádění programu* – automat provádí instrukce programu a to od první až do poslední. Instrukce bezprostředních vstupů a výstupů poskytují v době běhu programu bezprostřední přístup na vstupy a výstupy.
- *Zpracování požadavků komunikace* – automat během této fáze zpracovává všechny informace přijaté přes komunikační rozhraní nebo vstupních a výstupních modulů.
- *Provádění autodiagnostiky* - během této fáze automat zkontroluje správný provoz CPU, paměťových oblastí a stav všech rozšiřovacích modulů.
- *Zapísování na výstupy* – automat zapisuje hodnoty, uložené v registru obrazů výstupu na výstupy.



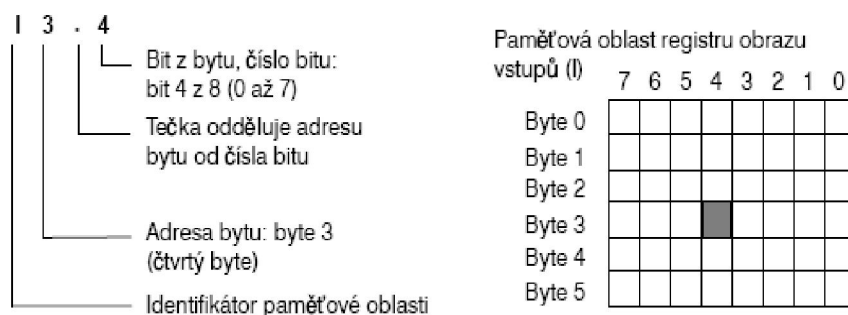
Obr. 7 Schéma programového cyklu

4.3 Přístup k datům

Programovatelný automat S7-200 uchovává informace v různých paměťových oblastech. Každá tato oblast má jedinečnou adresu, což umožňuje přímý přístup k informacím. Jsou to oblasti typu *V, I, Q, S, L, SM, T, C, HC*. Pro nás nejdůležitějšími jsou typy *I, Q, L a V*, se kterými budeme dále pracovat.

Přístup do těchto oblastí je pomocí tzv. bytového adresního formátu, jehož princip je znázorněn na obrázku 22. Tento formát se skládá z identifikátoru oblasti (*L, Q, M*), adresy bytu a čísla, nebo pořadí, jednotlivého bitu v tomto bytu. Mezi adresou bytu a číslem jednotlivého bytu je tečka, která tyto dva odděluje.

Informace jsou řazeny podle velikosti do tří základních skupin – Byte (B), Word (W), Double Word (D).



Obr. 8 Příklad bytového adresního formátu

Reprezentace	Byte (B)	Word (W)	Double word (D)
Unsigned integer	0 až 255 0 až FF	0 až 65 535 0 až FFFF	0 až 4 294 967 295 0 až FFFF FFFF
Signed integer	–128 až +127 80 až 7F	–32 768 až +32 767 8000 až 7FFF	–2 147 483 648 až +2 147 483 647 8000 0000 až 7FFF FFFF
Real IEEE 32bitová pohyblivá řádová čárka	Nelze	Nelze	+1.175495E–38 až +3.402823E+38 (kladné) –1.175495E–38 až –3.402823E+38 (záporné)

Obr. 9 Desítkové a šestnáctkové rozsahy

- *Registr obrazu vstupů - I* – v této paměťové oblasti jsou zapsány hodnoty jednotlivých vstupních bodů. Automat na počátku každého programového cyklu načte aktuální hodnoty na vstupních bodech, které potom do toho registru zapíše.
- *Registr obrazu výstupů - Q* – v této paměťové oblasti jsou zapsány hodnoty jednotlivých výstupních bodů. Automat na konci každého programového cyklu zapisuje aktuální hodnoty, uložené v tomto registru, na výstupy.

- *Oblast lokální paměti – L* – tato paměť je určena pouze pro lokální část programu a nelze ji adresovat z jiné části programu. Například, pokud máme paměť *L* přidělenou hlavnímu programu, nelze ji adresovat z podprogramu nebo přerušení.

4.4 Řídící program

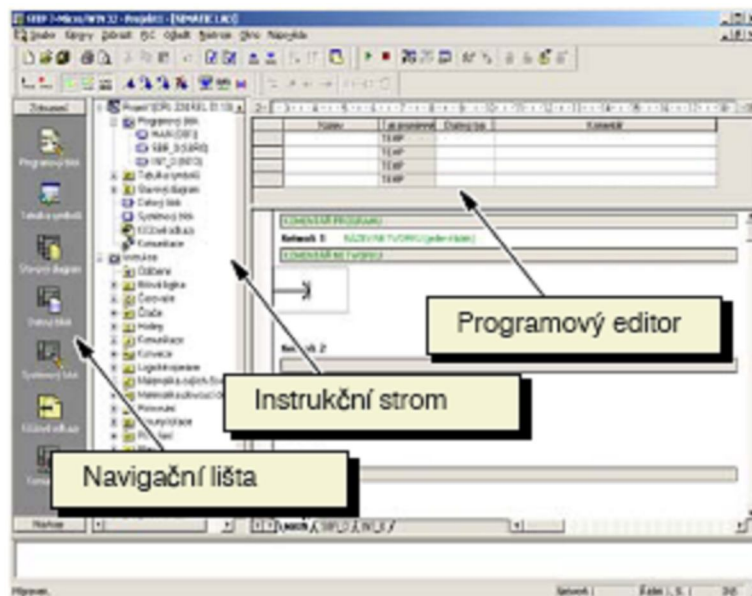
Programový blok, vytvořený ve vývojovém prostředí STEP7-Micro/WIN se skládá ze spustitelného kódu a komentáře tohoto kódu. Kód lze zkompileovat ve STEP7 a posléze nahrát do automatu. Je složen z následujících částí:

- *Hlavní program* – hlavní program řídí celou instrukci, obsahuje aplikace a instrukce, které vykonává všechny postupně za sebou jednou za programový cyklus.
- *Podprogram* – je to program, který je volitelný a provádí se pouze v případě, pokud je zavolán: hlavním programem, podprogramem nebo programem přerušení. Používají se v případě, že chceme nějakou část programu vykonávat opakovaně, čímž nám zmenšuje hlavní program. Pokud podprogram není volán, vždy dochází i k zrychlení celého programového cyklu.
- *Podprogram přerušení* – Tento podprogram reaguje na konkrétní událost, která vyvolá přerušení. Kdykoli nastane tato událost, automat tento program provede.
- *Ostatní prvky programu* – tyto bloky se provádí při downloadu uživatelského programu a obsahují informace pro automat.

4.5 Vývojové prostředí STEP7-Micro/WIN

Toto prostředí slouží mimo jiné pro vytváření editací a prohlížení řídicích programů. Samotný řídicí program může být zobrazen ve třech základních editorech – *LAD*, *FBD*, *STL*. Samotné vývojové prostředí STEP7-Micro/WIN lze rozdělit na několik základních částí:

- *Nástrojové lišty* – tato lišta obsahuje zkratky k často používaným příkazům Tyto lišty je možno skrýt.
- *Navigační lišta* – tato lišta nabízí přístup k různým programovacím prvkům STEP7-Micro/WIN
- *Instrukční strom* – zobrazuje všechny objekty a instrukce potřebné pro tvorbu programu
- *Programový editor* – skládá se z programu a tabulky lokálních proměnných, která slouží pro přiřazení názvu dočasných lokálních proměnných. Další podprogramy a přerušení jsou zobrazeny jako záložky ve spodní části okna editoru.

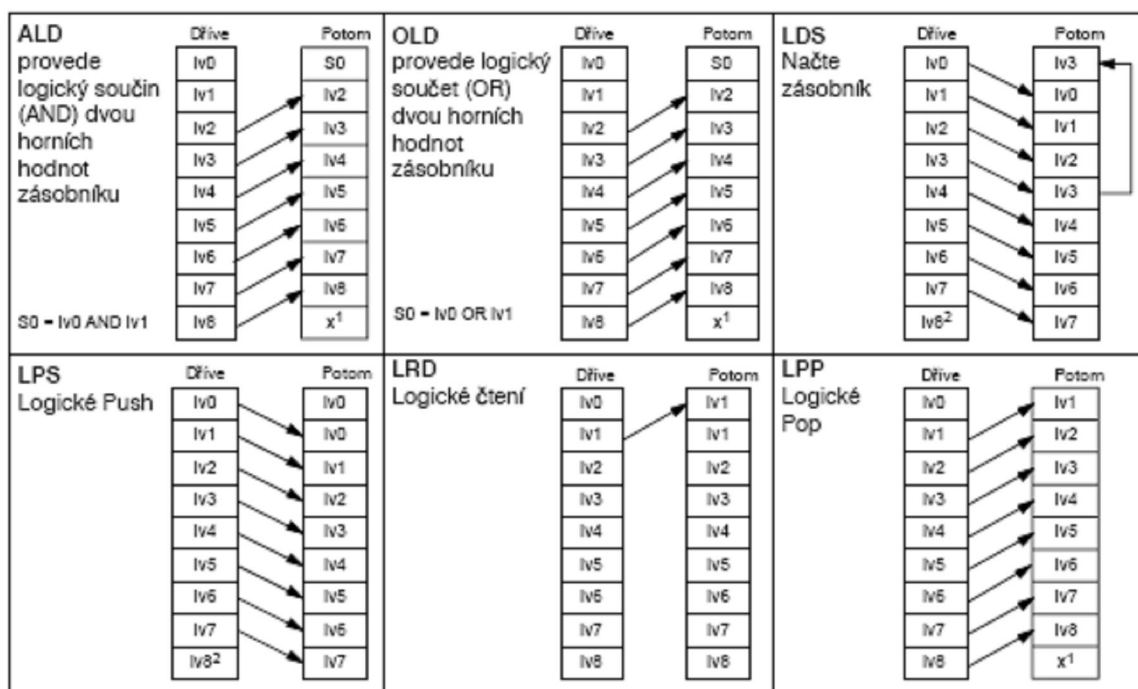


Obr. 10 Vývojové prostředí STEP7-Micro/WIN

- *Editor STL* – tento editor zobrazuje řídicí program jako znakové orientovaný programovací jazyk. Pomocí tohoto programu lze vytvářet programy, které by v ostatních editorech vytvořit nešly. Editor STL lze použít na prohlížení, editaci programu, který byl vytvořen pomocí editorů LAD nebo FBD, opačně to však možné není. Tento editor lze použít pouze s instrukčním souborem SIMATIC. Editor STL používá pro řešení řídicí logiky logický zásobník a do programu vkládáme instrukce (ALD, OLD, LPS, LRD, LPP), které s tímto zásobníkem manipulují. Princip jednotlivých instrukcí je popsán na obrázku 29, kde „iv0“ až „iv7“ jsou počáteční hodnoty zásobníku, „nv“ označuje novou hodnotu a „S0“ označuje vypočítanou hodnotu, která je uložena do zásobníku.

LD	I0.0	//Načtení jednoho vstupu
A	I0.1	//Logický součin s jiným vstupem
=	Q1.0	//Zapsání hodnoty na výstup 1

Obr. 11 Příklad programu STL



¹ Hodnota je neznámá (může to být 0 nebo 1).

² Po provedení instrukcí "Logické Push" nebo "Načíst zásobník" se hodnota Iv8 ztratí.

Obr. 12 Vysvětlení principu instrukcí zásobníku

4.6 Vybrané instrukce SIMATIC

V této kapitole se zmíním pouze o instrukcích důležitých pro náš projekt. Systémový manuál Siemens SIMATIC S7-200 tyto instrukce popisuje na 138 stranách formátu A4. Nám však bude stačit popis těch nejdůležitějších.

- **Kontakty** – instrukce standardního kontaktu získávají hodnotu z paměti, nebo z registru obrazu procesu, a to při každém programovém cyklu. Na rozdíl od instrukce okamžité kontakty, které se aktualizují okamžitě bez závislosti na programovém cyklu.
- **Negace** - tato instrukce mění logiku vstupního signálu. Například logickou 1 na vstupu změní na logickou 0.
- **Náběžná a sestupná hrana** – obě instrukce aktivují signálový tok na dobu jednoho programového cyklu. Náběžná hrana aktivuje signálový tok v případě, pokud se signál na vstupu změní z logické 0 na logickou 1. Sestupná hrana aktivuje tento tok, jestliže se signál na vstupu změní z logické 1 na logickou 0.
- **Cívky (výstupy)** – instrukce standardního výstupu přepíše novou hodnotu výstupního bitu do registru obrazu procesu, a to při každém cyklu. U instrukce okamžitého výstupu se nové hodnoty zapíší do registru obrazu procesu okamžitě, nezávisle na programovém cyklu.

- *Set/Reset* – tyto instrukce nastaví/vynulují stanovený počet bitů výstupu. Tyto instrukce se provádí jednou za programový cyklus. Narozdíl od instrukcí set/reset – okamžité, které okamžitě nastaví/vynulují požadovaný počet bitů.
- *Časovač zapnutí/Remanentní časovač zapnutí* – tyto instrukce odpočítávají čas od sepnutí vstupu a samotný bit časovače nabude hodnoty logické 1, pokud časovač dopočítá do určené hodnoty. Číslo časovače určuje jednotlivé časovače a jeho časové rozlišení. Na obrázku 30 jsou uvedena rozlišení jednotlivých časovačů. Rozdíl mezi časovačem zapnutí a remanentním časovačem zapnutí spočívá v nulování odpočítávání – časovač remanentního zapnutí se musí nulovat pomocí instrukce *reset*, na rozdíl od časovače zapnutí, který se vynuluje při každém rozepnutí vstupu.
- *Časovač zpožděného vypnutí* – tato instrukce se používá na zpoždění rozepnutí výstupu. Tento čas je hodnota časovače logická 1 a po ukončení změny svoji hodnotu na logickou 0.

Typ časovače	Odstranění	Maximální hodnota	Číslo časovače
TONR (remanentní)	1 ms	32,767 s (0,546 min)	T0, T64
	10 ms	327,67 s (5,46 min)	T1 až T4, T65 až T68
	100 ms	3276,7 s (54,6 min)	T5 až T31, T69 až T95
TON, TOF (bez remanence)	1 ms	32,767 s (0,546 min)	T32, T96
	10 ms	327,67 s (5,46 min)	T33 až T36, T97 až T100
	100 ms	3276,7 s (54,6 min)	T37 až T63, T101 až T255

Obr. 13 Typy a rozlišení jednotlivých časovačů

Rozlišení jednotlivých časovačů ovlivňuje činnost jednotlivých časovačů. Při použití časovače s rozlišením 1ms je bit časovače a aktuální hodnota tohoto časovače aktualizována asynchronně s programovým cyklem. U časovače s rozlišením 10ms dochází k aktualizaci bitu a hodnoty časovače vždy na počátku každého programového cyklu. Během programového cyklu se hodnota časovače a bit nemění. Časovač s rozlišením 100ms aktualizuje svůj bit a hodnotu časovače vždy při provedení instrukce.

5 ZVONÍČÍ STROJ V OBCI BUKOVINKA

5.1 Kostel v obci Bukovinka

Objekt – kostel, ve kterém se nachází zvonící stroj, stojí v obci Bukovinka. Tuto obec se 358 obyvateli najdeme v okrese Blansko - severovýchodně od Brna. Zvony, které se objektu nachází, byly vysvěceny a vsazeny do 37 metrové věže roku 1949, kde jsou dodnes. O pohon zvonu se staral zvoník. Samozřejmě časem bylo nutno řešit otázku automatického pohonu zvonu, a proto byl nainstalován zvonící stroj s rotačním elektromotorem. Celé zařízení bylo postupně během posledních šesti let modernizováno až do dnešní podoby, kdy je osazeno programovatelným automatem Siemens, o kterém bude zmínka níže. [5]



Obr. 14 Mapa okresu Blansko[7]



Obr. 15 Kostel v obci Bukovinka

5.2. Zvonící stroj v objektu

Zvonící stroj je umístěn ve věži kostela. Samotný zvonící stroj se skládá:

- *Zvonová stolice* – zvonová stolice v objektu je sestrojena z ocelové konstrukce, a to pravděpodobně z důvodu malého půdorysu samotné věže. Celá konstrukce je upevněna na dřevěných patkách, snižující vibrace celé soustavy. Samotná stolice je sestavena z ocelových L profilů, navzájem k sobě snýtovaných.
- *Zvony* – ve zvonícím stroji jsou zavěšeny tři zvony, prvních z nich, která má ϕ hrany 380 mm je zavěšen na přímé ocelové hlavici, přichycen je pomocí pěti šroubů, z nichž jeden slouží k přichycení srdce. Mezi korunou zvonu a hlavici je umístěna distanční dřevěná podložka. Druhý, menší z dvojice zvonů má ϕ 310 mm, je též zavěšen na přímé ocelové hlavici mezi níž a zvonem je umístěná distanční podložka. Tento zvon je přichycen sedmi šrouby, z nichž jeden slouží k uchycení srdce zvonu. Oba zvony mají srdce typu „létající“.



Obr. 16 Větší zvon

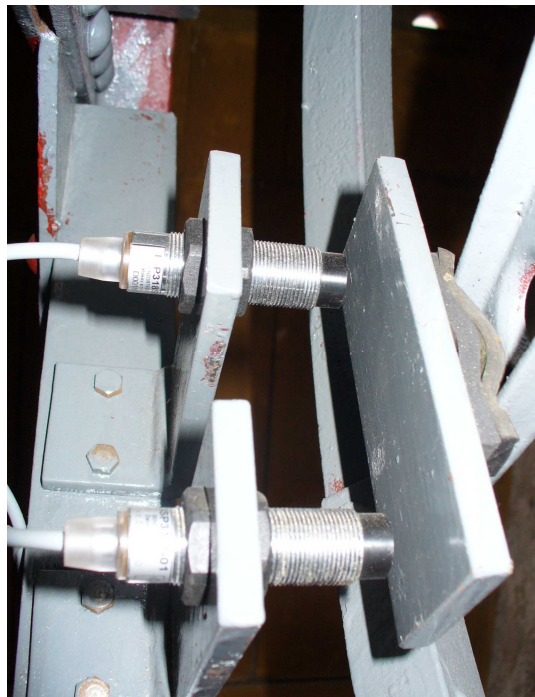


Obr. 17 Menší zvon

- *Pohon zvonů* – oba zvony jsou poháněny jedním asynchronním elektromotorem, který je dostatečně dimenzován pro tento pohon. Oba motory jsou stejného typu. Hřídel motoru je osazena ozubeným kolem, který přes řetěz přenáší krouticí moment na zvonové kolo. Samotný řetěz je nastaven lanem tak, aby i při maximálním výkyvu zvonu byl krouticí moment přenášen přes řetěz. V místě spoje řetězu a lanka, pouze na odlehčené větvi, je umístěna pružina. Ta slouží k napínání této větve. Aby nedocházelo k prokluzu mezi lankem a zvonovým kolem, je lanko přichyceno v nejnižším místě ke kolu.
- *Řídící prvky* – polohu zvonů vyhodnocují dva snímače, které jsou připevněny na zvonové stolici. Tyto snímače reagují na ocelovou destičku připevněnou na zvonovém kole. Ta je umístěna tak, aby oba snímače byly zapnuty, pokud se zvon nachází v rovnovážné poloze. Chod obou zvonů je řízen programovatelným automatem Siemens Simatic S7 – 200 CPU 221, který je umístěn v prvním patře věže v prachotěsné skříni.



Obr. 18 Pohon zvonu, zvonová stolice



Obr. 19 Indukční snímače



Obr. 20 Pohled na hlavici zvonu, zvonové kolo

6 ROZBOR PROBLÉMU

6.1. Nedostatky zvonícího stroje

6.1.1 Nesprávné vyvážení zvonů

Oba zvony vykazují nesprávné vyvážení. Toto nesprávné vyvážení se projevuje při zapnutém zvonění nevyrovnaným akustickým projevem u obou zvonů. Zvon bije v jedné krajní poloze silněji než v druhé. To znamená, že výchylka zvonu v jedné krajní poloze je větší než v druhé. Jak jsem již zmínil výše, tak tento problém může být způsoben několika faktory, jako je přesazení matice zvonu, nesprávné zavěšení atd. (Vzhledem k časovému omezení nebylo možno přesnou příčinu špatného vyvážení zjistit a mým úkolem bylo toto vyvážení odstranit softwarově.) Na samotném zvonu se tento problém projevuje poškozením zvonu v místě doteku srdce, jde o tzv. vytloukání zvonu, které může vést až k jeho prasknutí.



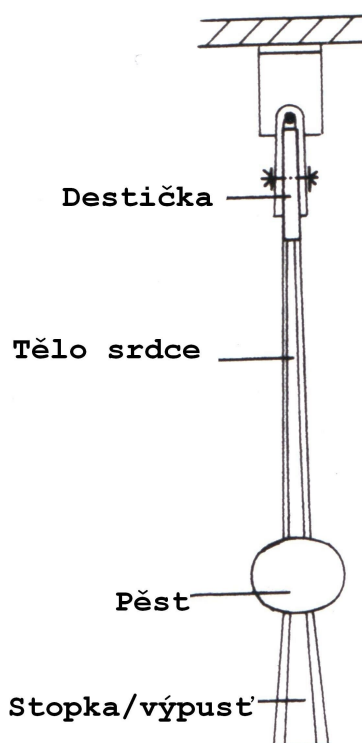
Obr. 21 Poškození zvonu nesprávným vyvážením

6.1.2 Nevhodný tvar srdce zvonů

Pokud srovnáme tvar srdce v obou zvonech s optimálním tvarem používaných srdcí, lze se domnívat, že i masivnost obou použitých srdcí má vliv na zvukový projev jednotlivých zvonů. Pravděpodobně není dodržena podmínka hmotnostní, kdy hmotnost srdce by měla odpovídat 3 – 5 % hmotnosti zvonu. Samozřejmě se jedná z mého pohledu pouze o odhad, neznám hmotnosti jednotlivých zvonů, ani hmotnost srdcí. Detaily, jako je hmotnost zvonu a srdce, či jaký materiál je použit, nezná ani provozovatel. Ale pokud porovnáme velikost a tvar jednotlivých srdcí s běžně používanými zvony, lze se domnívat, že to tak skutečně je.



Obr. 22 Tvar srdce v Bukovince



Obr. 23 Optimální tvar srdce[2]

6.2 Chod zvonů s použitím původního programu

Kód původního programu je napsán v editoru STL. Program se skládá z hlavního programu a devíti podprogramů. Podprogramy SBR_3, SBR_7 nejsou využívány. Hlavní program je rozdělen na jedenáct obvodů. Obvod dva až šest reguluje chod většího zvonu a je principiálně shodný s obvodem sedm až jedenáct, který reguluje chod menšího zvonu. Liší se pouze v označení proměnných pro menší zvon a v hodnotě nastavení časovačů.

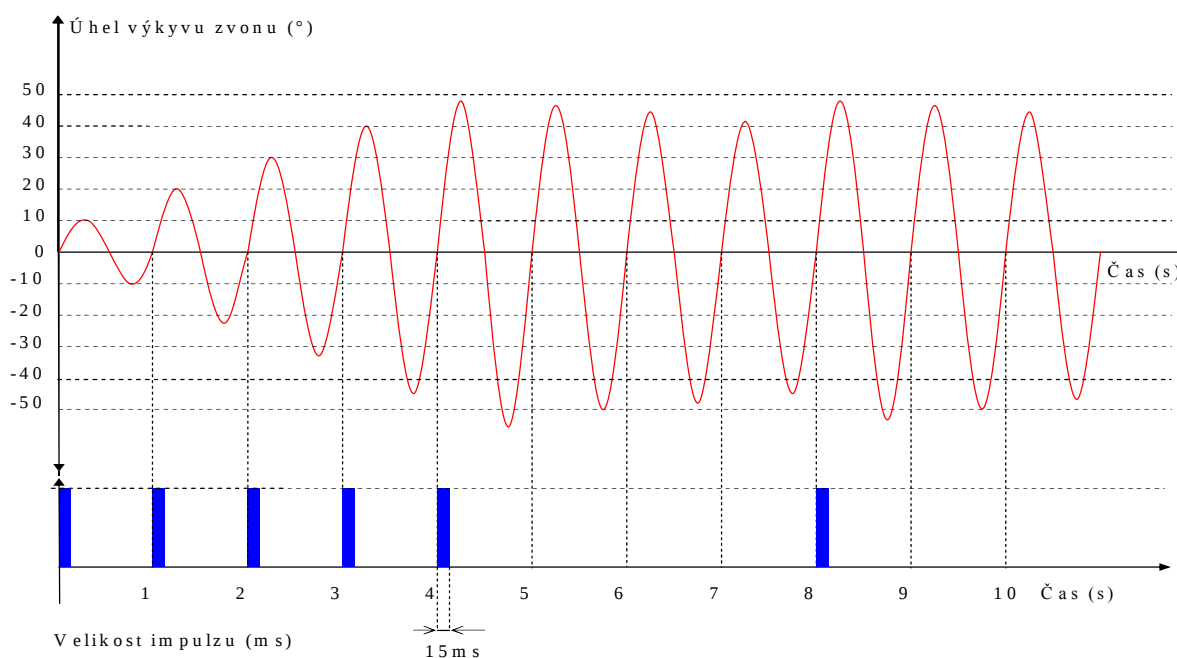
Program je naprogramován jako regulace s konstantní délkou impulzu. Oba zvony mají stejnou délku impulzu, sepnutí motoru T34 a T98 = 0.15s. To zda bude zvonu udělen impulz, rozhoduje časový okamžik průchodu zvonu rovnovážnou polohou. Jinak řečeno,

pokud se zvon pohybuje od jednoho senzoru k druhému pomaleji, než je nastaven časový okamžik, dochází k udělení tohoto impulsu. Tento nastavený čas je u obou zvonů rozdílný. Pro větší zvon je hodnota časovače $T_{32} = 60\text{ms}$ a pro menší zvon $T_{96} = 55\text{ms}$. Zvonu je udělován pracovní impuls při průchodu rovnovážnou polohou a to ve směru shodným se smyslem otáčení motoru. K udělení impulsu dochází v momentě průchodu zvonu rovnovážnou polohou.

Tento program používá, jak jsem již zmínil jeden impuls pro rozhoupání zvonu, tak i k jeho regulaci, to má za následek dlouhé rozhoupávání zvonu. A druhým nedostatkem je délka tohoto impulsu, která je vyšší, než je potřebná pro samotnou regulaci. Tím pádem dochází k udělování tohoto impulsu jedenkrát za tři až čtyři pracovní cykly. Údery srdce do zvonu jsou nestejněměrné a zvukový projev zvonu není harmonický.

Dalším problémem, se kterým program nepočítá je špatné vyvážení, které vede opět nestejněměrnému zvukovému projevu, tak i nadměrnému opotřebení zvonu. V jedné krajní poloze jsou údery příliš velké a zvon se vytlouká. V druhé krajní poloze jsou údery velmi malé a často se stávalo, že zvon neudeřil vůbec.

Vzhledem k tomu že zvony vykazují stejné problémy, přikládám zde pouze graf výchylky velkého zvonu. Graf malého zvonu se liší pouze v časových hodnotách.



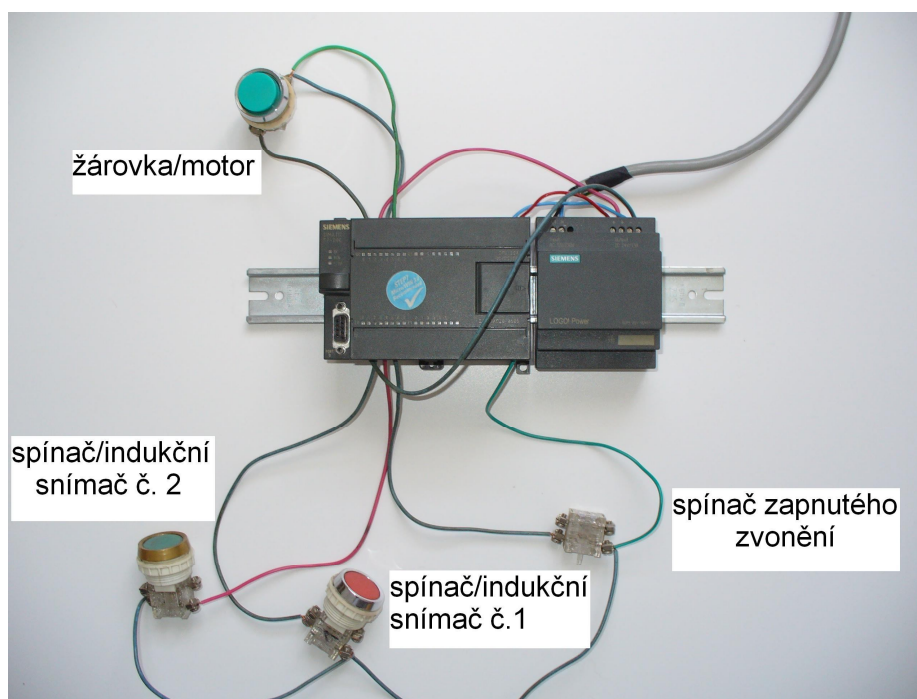
Obr. 24 Graf výchylky velkého zvonu, udělování impulsu – (původní program)

7 ŘEŠENÍ PROBLÉMU – ŘÍZENÍ CHODU ZVONŮ

7.1. Model pro zkoušení řídicího programu

Vzhledem k tomu, že zkoušky na reálném objektu jsou velmi omezené, bylo třeba vyzkoušet funkčnost řídicího programu, tak aby na reálném objektu proběhlo pouze doladění celého programu, tzn. pouze úprava hodnot jednotlivých časovačů, dle zvukového projevu jednotlivých zvonů.

Proto, abych vyzkoušel funkčnost jednotlivých programů, jsem vytvořil malý model. Zde jsou nahrazeny jednotlivé snímače spínači a motor žárovkou. Následným ručním spínáním jsem simuloval průchod zvonu rovnovážnou polohou a tím otestoval všechny stavy, které mohou v praxi nastat. Při nastavování časovačů jsem vycházel z jejich nastavení v původním programu. Pro řešení problému jsem vypracoval dvě varianty optimalizace chodu kostelních zvonů v Bukovince.



Obr. 25 Model pro zkoušení funkčnosti programu

7.2 Podmínky provozovatele

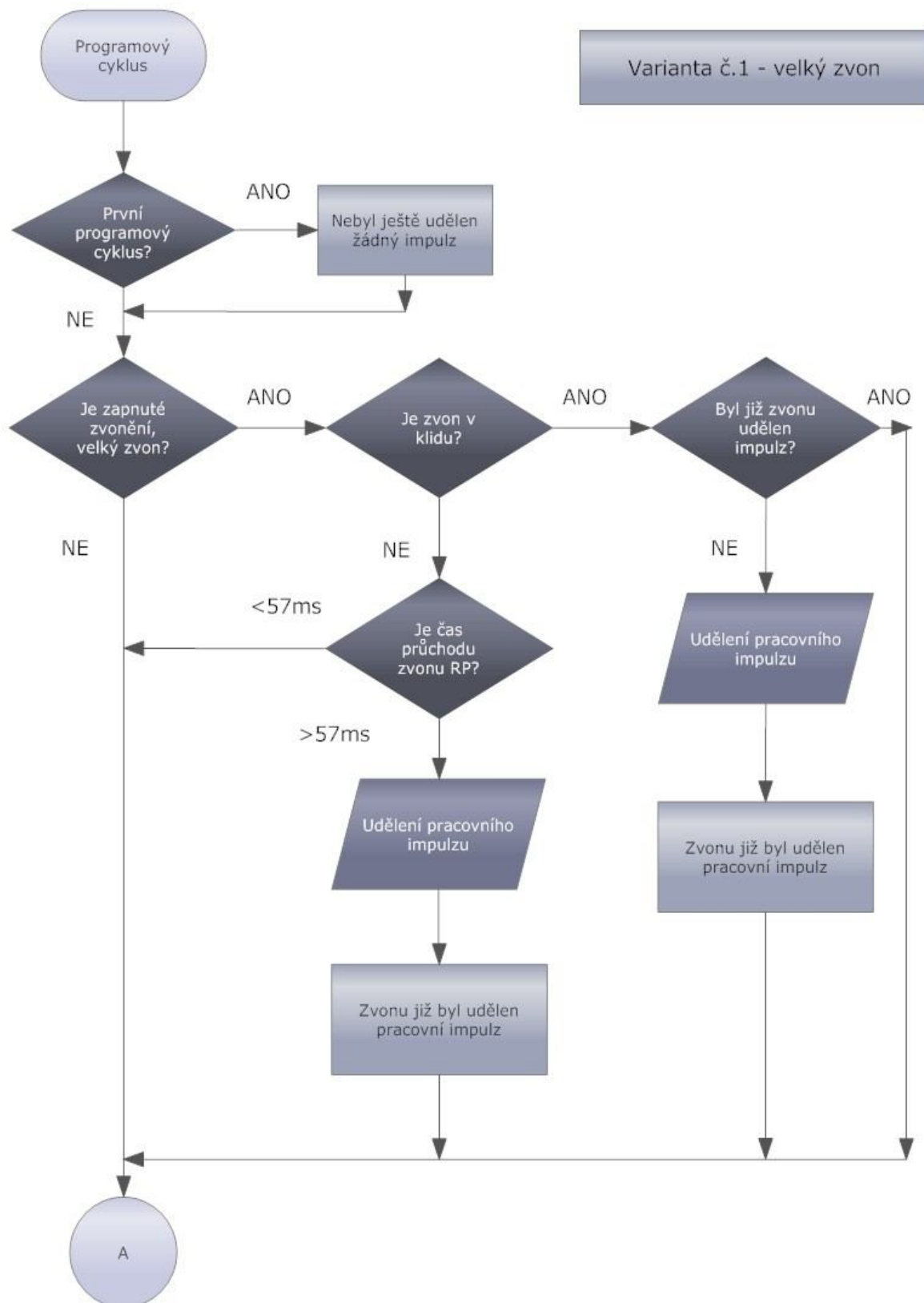
Provozovatel si stanovil podmínky, které bylo nutno při řešení daných problémů dodržet, a sám stanovil nedostatky, které je bylo potřeba odstranit.

- *Podmínka č.1* - zachovat dosavadní fyzický stav zvonícího stroje, to znamená ponechat současný pohon zvonů, umístění snímačů, či špatné vyvážení zvonů a nijak mechanicky nezasahovat do tohoto zvonícího stroje.
- *Podmínka č.2* - omezit zkoušky na reálném objektu na minimální míru. Vzhledem k tomu, že akustický projev obou zvonů je velmi výrazný a zvuk obou zvonů se rozléhá do velké vzdálenosti, bylo přáním provozovatele tyto zkoušky omezit na úplné minimum.
- *Podmínka č.3* - řešit všechny nedostatky chodu kostelních zvonů pouze programově.
- *Úkol č.1* – odstranit časově náročné uvedení zvonu do harmonického pohybu.
- *Úkol č.2* – odstranit neharmonický zvukový projev zvonu způsobený nesprávným vyvážením zvonu a současným programovým vybavením.

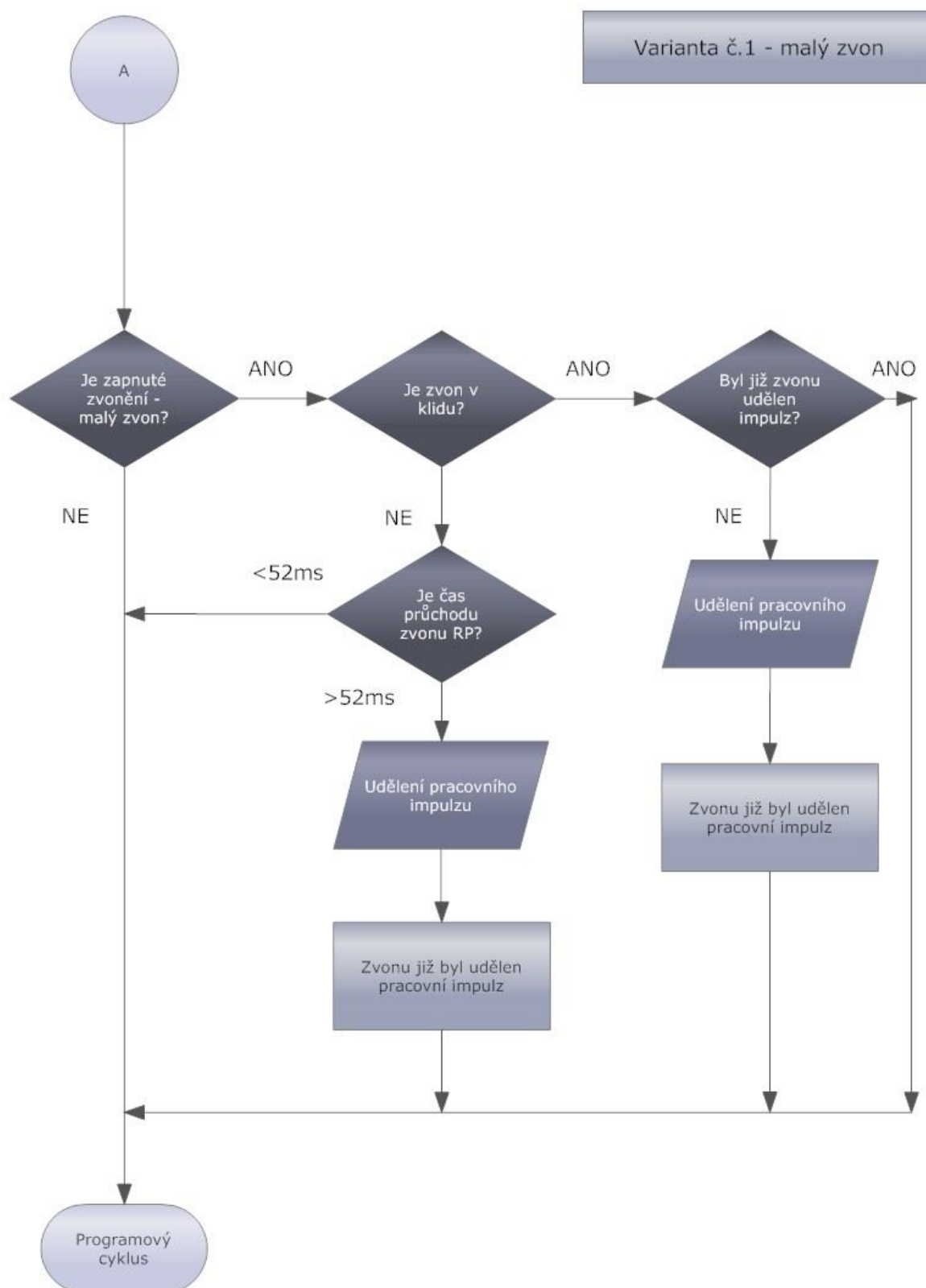
7.3 Řešení variantou 1

Program, který jsem označil jako první je ze všech variant nejjednodušší a využívá pro regulaci zvonu dvou délek impulsů.

7.3.1 Vývojový diagram, kód programu



Obr. 26 Vývojový diagram varianta 1 – velký zvon



Obr. 27 Vývojový diagram varianta 1 – malý zvon

Program Varianta 1									
Network 1			Network 7			Podprogram SBR_0			
LD	SM0.1		LDN	ZV2		LDN	SM0.0		
CALL	SBR_0		CALL	SBR_6		=	START1		
Network 2			NOT			=	START2		
LDN	ZV1		AN	START2		Podprogram SBR_1			
CALL	SBR_2		A	IS21		LD	SM0.0		
NOT			A	IS22		S	START1, 2		
AN	START1		=	stav2A		R	IMP11, 2		
A	IS11		LD	stav2A		R	stav1B, 2		
A	IS12		TON	T39, +10		Podprogram SBR_2			
=	stav1A		Network 8			LD	SM0.0		
LD	stav1A		LDI	IS22		R	START1, 2		
TON	T37, +10		TON	T96, +52		Podprogram SBR_5			
Network 3			Network 9			LD	SM0.0		
LDI	IS12		LDI	IS21		S	START2, 2		
TON	T32, +57		EU			R	IMP21, 2		
Network 4			A	T96		R	stav2B, 2		
LDI	IS11		S	IMP21, 2		Podprogram SBR_6			
EU			Network 10			LD	SM0.0		
A	T32		LD	ZV2		R	START2, 2		
S	IMP11, 2		LPS						
Network 5			A	T39					
LD	ZV1		AN	START2					
LPS			S	stav2B, 2					
A	T37		LPP						
AN	START1		A	IMP21					
S	stav1B, 2		=	stav2C					
LPP			Network 11						
A	IMP11		LD	stav2B					
=	stav1C		TON	T99, +35					
Network 6			LD	stav2C					
LD	stav1B		TON	T97, +22					
TON	T35, +40		LD	T97					
LD	stav1C		TON	T98, +15					
TON	T33, +24		LDN	T99					
LD	T33		A	stav2B					
TON	T34, +15		LDN	T98					
LDN	T35		A	T97					
A	stav1B		OLD						
LDN	T34		=I	M2					
A	T33		LD	T99					
OLD			O	T98					
=I	M1		CALL	SBR_5					
LD	T34								
O	T35								
CALL	SBR_1								

7.3.2 Rozbor programu

Kód programu je psán v editoru STL. Hlavní program je doplněn tentokrát 5 podprogramy, které jsou plně využívány hlavním programem. Hlavní program se skládá ze stejného počtu obvodů jako původní, tedy 11. Obvod 2 až 6 reguluje chod většího zvonu a je principiálně shodný s obvodem 7 až 11, který reguluje chod menšího zvonu. Liší se pouze v označení proměnných pro menší zvon a v hodnotě nastavení časovačů. První network je pouze inicializační a jeho funkce jsou prováděny pouze v prvním programovém cyklu.

Z vývojového diagramu vyplývá, že diagnostika obou zvonů a jejich následná regulace je prováděna nezávisle na sobě. To je samozřejmě dáno rozdílnou charakteristikou jednotlivých zvonů.

Program využívá pro regulaci zvonu dvou délek impulzů. Hodnota délky startovacího impulzu je $T35 = 40\text{ms}$ u většího zvonu a $T99 = 35\text{ms}$ u menšího. Původní hodnota tohoto impulzu však byla nastavena u obou zvonů nižší a to $T35 = 30\text{ms}$. Tato délka však nebyla dostatečná pro potřebné počáteční vychýlení zvonu z rovnovážné polohy, a proto byla zkoušením zvýšena až na optimální hodnotu 40 a 35 ms.

K tomu, aby program udělil zvonu startovací impulz, musí být splněny následující podmínky. První podmínka – musí být zapnuto zvonění. To znamená hodnota proměnné ZV1 pro velký a hodnota ZV2 pro malý zvon je kladná neboli log.1.

Druhá podmínka - pokud již došlo k udělení jakéhokoli impulzu na zvon (během jednoho zapnutí zvonění), tak zvonu již nebude udělen startovací impulz. Tato podmínka je v programu pro případ poškození zvonícího stroje. Pokud dojde například k přetržení lanka zvonícího stroje, došlo by v případě absence této podmínky k nepřetržitému udělení startovacího impulzu. Hodnota proměnné START1 pro větší zvon a START2 pro menší, nám zaručuje, že pokud již byl udělen během jednoho zvonění jakýkoli impulz, nedojde k dalšímu udělení start impulzu.

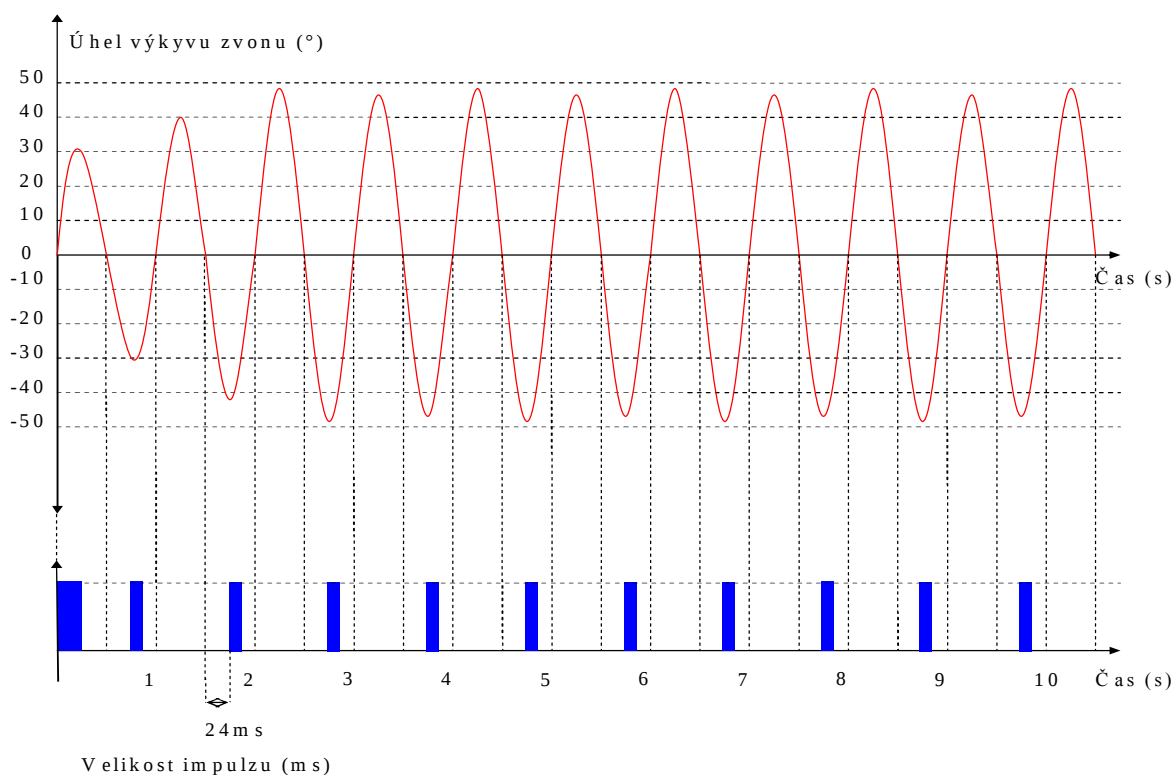
Třetí podmínka - zvon musí setrvávat v rovnovážné poloze alespoň po dobu jedné sekundy. Tato podmínka je důležitá v případě vypnutí zvonění a následného zapnutí. Pokud by byl časový úsek mezi těmito dvěma událostmi menší, než je ustálení zvonu v rovnovážné poloze a program by toto ustálení nemonitoroval, mohlo by dojít k udělení startovacího impulzu v momentě, kdy by se zvon pohyboval proti směru otáčení elektromotoru. Tím by došlo k poškození celého zvonícího stroje. Proto program testuje, zda se zvon nachází po dobu jedné sekundy v rovnovážné poloze.

Velikost impulzu, který udržuje zvon v pohybu je $T34$ a $T98 = 15\text{ ms}$. Tato hodnota je pro oba zvony shodná a je převzata z původního programu. K tomu aby zvon udělil tento impulz, musí být splněny následující podmínky. První podmínka – musí být zapnuto zvonění pro příslušný zvon. To znamená hodnota proměnné ZV1 pro velký a hodnota ZV2 pro malý zvon je kladná, neboli log.1. Druhou podmínkou je malá rychlost pohybu zvonu při průchodu rovnovážnou polohou. Pokud při pohybu zvonu od snímače IS11 (IS21) ke snímači IS12 (IS22) uplyne časový okamžik větší než je nastavená hodnota časovače $T32$ ($T96$), program vyhodnotí pohyb zvonu jako malý. Hodnota časovače pro malý zvon byla $T32 = 60\text{ms}$ a pro malý $T96 = 55\text{ms}$. Tyto hodnoty byly následně během zkoušení programu upraveny na $T32 = 57\text{ms}$ a $T96 = 52\text{ms}$. Zároveň je v tomto obvodu vložena podmínka náběžné hrany, která nám zabezpečuje testování rychlosti zvonu při

správném směru pohybu zvonu. V tomto programu musí být směr pohybu zvonu opačný než směr otáčení motoru. A to proto, abychom mohli regulovat velký výkyv zvonu jeho brzděním v krajní poloze. Pokud je tato podmínka splněna může být zvonu udělen impuls, který udržuje zvon v harmonickém pohybu.

Mezi tím, než program vyhodnotí tuto podmínku a samotným udělením impulsu je v programu nastavena časová prodleva. Velikost této prodlevy byla nastavena u velkého zvonu $T_{33} = 35\text{ms}$ a u malého zvonu $T_{97} = 30\text{ms}$. Během zkoušení byly tyto hodnoty upraveny na $T_{33} = 24\text{ms}$ a $T_{97} = 22\text{ms}$. Toto zpoždění slouží k posunutí udělení impulsu do krajní polohy. Při správném načasování tím odstraníme nesprávné vyvážení zvonu. Jak jsem již uvedl, tak zvon je špatně vyvážen a tím i vychýlení zvonu je v jedné krajní poloze větší než v druhé. Program je naprogramován tak, aby byl impuls udělen těsně před zastavením zvonu v rovnovážné poloze, a tím částečně snížil setrvačnou sílu zvonu a tím i velikost výchylky. Celková energie, která je zvonu udělena slouží k brzdění zvonu v krajní poloze a zároveň mu uděluje impuls potřebný k setrvání zvonu v harmonickém pohybu.

7.3.3. Chod zvonů s použitím programu varianta 1



Obr. 28 Graf výchylky velkého zvonu, udělování impulsu (Varianta 1)

Vzhledem k tomu že zvony vykazují stejné problémy, jejich odladění je stejné, a proto zde přikládám pouze graf výchylky velkého zvonu. Graf malého zvonu se liší pouze v časových hodnotách.

Z následujícího grafu vyplývá, že posunutím udělení impulzu do krajní polohy se podařilo odstranit nesprávné vyvážení zvonu. Tím se odstranilo i nepravidelné udělování impulzu na zvon, které v původním programu bylo jedenkrát za tři až čtyři programové cykly. Tento program uděluje impulz každý pracovní cyklus. To je způsobeno zmenšením hodnot časovačů ovlivňující rozhodování o tom, zda bude impulz udělen, tím došlo k častější potřebě tento impulz udělit, i když velikost impulzu zůstala stejná jako u původního programu. Druhým důvod, proč program uděluje impulz každý programový cyklus, je rozdělení energie impulzu na energii pro brždění zvonu v krajní poloze a energii udržující zvon v harmonickém pohybu.

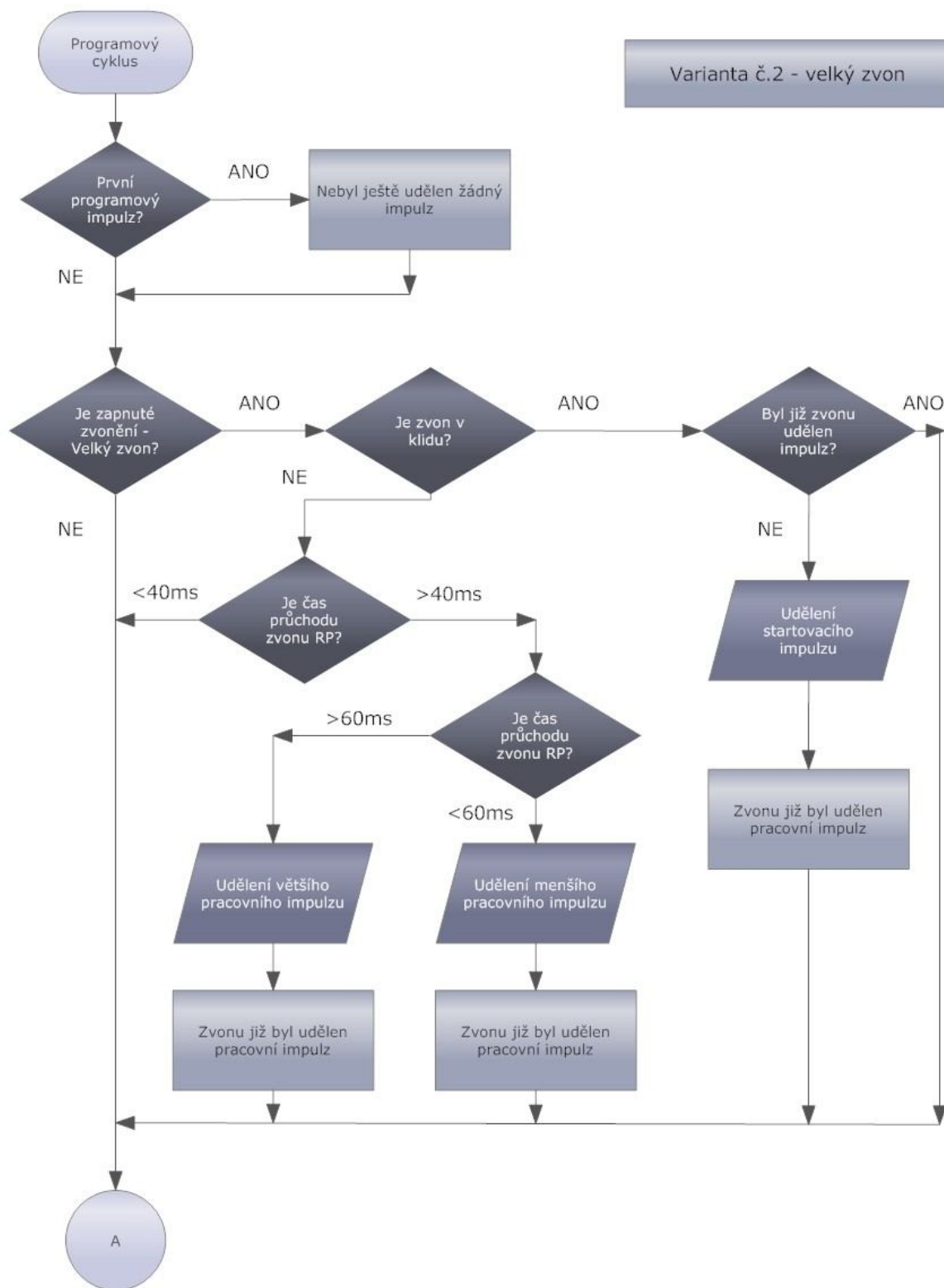
Druhý problém, který řeší tento program je samotný rozběh obou zvonů. Pokud porovnáme oba digramy, myslím tím diagram původního programu a diagram tohoto programu vidíme, že došlo k výraznému zkrácení časů potřebných pro uvedení zvonů do harmonického pohybu.

Tento program jsem spolu s provozovatelem úspěšně odzkoušel a dle zvukového projevu odladil. Program odstraňuje dva základní nedostatky, které provozovatel požadoval doladit. První je odstranění dlouhého rozběhu zvonu, respektive uvedení zvonu do rovnovážné polohy. Tento nedostatek se vložením startovacího impulzu podařilo eliminovat. Druhým nedostatkem, který provozovatel požadoval odstranit, byl neharmonický pohyb. Respektive špatné vyvážení zvonů. Tento problém se podařilo odstranit vhodným načasováním udělování impulzu zvonu. Jak jsem již zmínil výše, tak zvonu je udělován impulz každý programový cyklus. Energie tohoto impulzu je rozdělována pro brždění zvonu v krajní poloze a část energie slouží k udržení zvonu v harmonickém pohybu. Tím byla odstraněna neharmoničnost obou soustav. Proto mohu říci, že vzhledem k jednoduchosti je tento program vhodný pro regulaci nesprávně vyvážených zvonů. Jedinou nevýhodou je přesné naladění všech časovačů na optimální hodnoty, které jsem prováděl přímo na reálném zvonu.

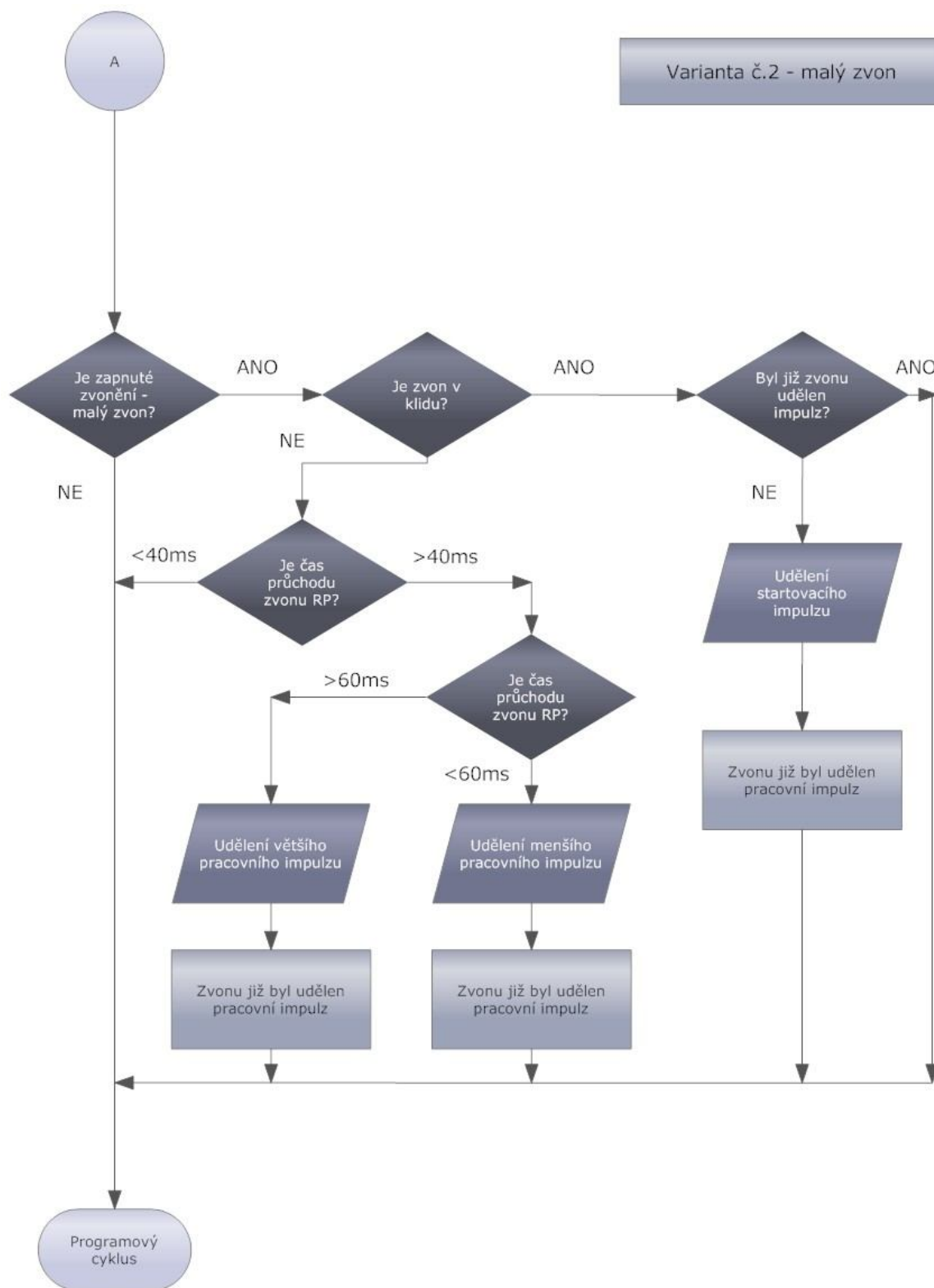
7.4 Řešení variantou 2

Program, který jsem označil jako druhý, využívá pro regulaci zvonu tří délek impulzů.

7.4.1. Vývojový diagram, kód programu



Obr. 29 Vývojový diagram varianta 2 – velký zvon



Obr. 30 Vývojový diagram varianta 2 – malý zvon

Program Varianta 2				
Network 1	A	STRANA1A	TONR T64, +40	TON T98, +18
LD SM0.1	LPS		NOT	LD stav2D
CALL SBR_0	A	IMP11	R T0, 1	TON T99, +30
Network 2	=	stav1C	R T64, 1	LD T99
LDN ZV1	LPP		Network 12	TON T100, +15
CALL SBR_2	A	IMP12	LDNI IS21	LDN T40
NOT	=	stav1D	LPS	A stav2B
AN START1	Network 8		A T0	LDN T98
A IS11	LD stav1B		S IMP21, 2	A T97
A IS12	TON T38, +3		R IMP22, 2	LDN T100
= stav1A	LD stav1C		LPP	A T99
LD stav1A	TON T33, +30		A T64	OLD
TON T37, +10	LD T33		AN T0	OLD
Network 3	TON T34, +18		S IMP22, 2	=I M2
LDI IS12	LD stav1D		R IMP21, 2	Network 17
TON T32, +60	TON T35, +30		Network 13	LD T40
TON T96, +40	LD T35		LDI IS22	O T98
Network 4	TON T36, +15		ED	O T100
LDNI IS11	LDN T38		AI IS21	CALL SBR_5
LPS	A stav1B		S STRANA2A, 2	
A T32	LDN T34		Network 14	Podprogram SBR_0
S IMP11, 2	A T33		LDI IS21	LDN SM0.0
R IMP12, 2	LDN T36		ED	= START1
LPP	A T35		AI IS22	= START2
A T96	OLD		R STRANA2A, 2	Podprogram SBR_1
AN T32	OLD		Network 15	LD SM0.0
S IMP12, 2	=I M1		LD ZV2	S START1, 2
R IMP11, 2	Network 9		LPS	R IMP11, 2
Network 5	LD T34		A T39	R IMP12, 2
LDI IS12	O T38		AN START2	R stav1B, 2
ED	O T36		S stav2B, 2	Podprogram SBR_2
AI IS11	CALL SBR_1		LPP	LD SM0.0
S STRANA1A, 2	Network 10		A STRANA2A	R START1, 2
Network 6	LDN ZV2		LPS	Podprogram SBR_5
LDI IS11	CALL SBR_6		A IMP21	LD SM0.0
ED	NOT		= stav2C	S START2, 2
AI IS12	AN START2		LPP	R IMP21, 2
R STRANA1A, 2	A IS21		A IMP22	R IMP22, 2
Network 7	A IS22		= stav2D	R stav2B, 2
LD ZV1	= stav2A		Network 16	Podprogram SBR_6
LPS	LD stav2A		LD stav2B	LD SM0.0
A T37	TON T39, +10		TON T40, +3	R START2, 2
AN START1	Network 11		LD stav2C	
S stav1B, 2	LDI IS22		TON T97, +30	
LPP	TONR T0, +60		LD T97	

7.4.2 Rozbor programu

Kód programu je psán také v editoru STL a hlavní program je doplněn, stejně jako v předchozím, 5 podprogramy, které jsou využívány hlavním programem. Hlavní program se skládá ze 17 obvodů. Obvod 2 až 9 reguluje chod většího zvonu a je principiálně shodný s obvodem 10 až 17, který reguluje chod menšího zvonu. Liší se pouze v hodnotě nastavení časovačů a označení proměnných pro menší zvon. První network je pouze inicializační a jeho funkce jsou prováděny pouze v prvním programovém cyklu.

Stejně jako u předchozího programu, tak diagnostika obou zvonů a jejich následná regulace je prováděna nezávisle na sobě. To samozřejmě vyplývá z rozdílné charakteristiky jednotlivých zvonících strojů.

Program využívá pro regulaci zvonu tentokrát tři délek impulzů. Hodnota délky startovacího impulzu je $T38 = 30\text{ms}$ u většího zvonu a $T40 = 30\text{ms}$ u menšího. Tyto hodnoty zůstaly nezměněny a jsou přednastaveny pro další testování.

Stejně jako je tomu u varianty 1, tak k tomu aby program udělil zvonu startovací impulz, musí být splněny následující podmínky. První podmínka – musí být zapnuto zvonění. Druhá podmínka – zvonu nesměl být udělen jakýkoli impulz. A třetí podmínka – zvon musí setrvávat v rovnovážné poloze alespoň po dobu jedné sekundy. Tyto podmínky jsou stejné jako u předchozího programu, tedy varianty 1.

Tento program používá dvě velikosti impulzu, pro větší zvon to jsou hodnoty $T34 = 18\text{ms}$ a $T36 = 15\text{ms}$ a pro menší zvon $T98 = 18\text{ms}$ a $T100 = 15\text{ms}$. Tyto hodnoty jsou přednastaveny pro další testování. K tomu, aby zvon udělil jeden z impulzů, musí být splněny následující podmínky.

První podmínka – musí být zapnuté zvonění pro příslušný zvon. To znamená hodnota ZV1 pro velký zvon a hodnota ZV2 pro malý, musí nabývat hodnoty log.1.

Druhá podmínka – zvon se musí pohybovat ve správném směru, to znamená proti smyslu otáčení elektromotoru. Tento směr je z důvodu udělení impulzu v krajní poloze tak, aby energie impulzu byla rozdělena částečně pro brždění zvonu v této poloze, tak sloužila k pohonu zvonu. K tomu, zda dojde udělení menšího nebo většího impulzu rozhoduje okamžik průchodu zvonu rovnovážnou polohou. Pokud při pohybu zvonu od snímače IS11 (IS21) ke snímači IS12 (IS22) uplyne časový okamžik větší než je nastavená hodnota časovače $T96$ a $(T64) = 40\text{ms}$ a zároveň je tento okamžik menší než je hodnota časovačů $T0$ a $32 = 60\text{ms}$ zvonu je udělen menší pracovní impulz – 15ms . Pokud při pohybu zvonu od snímače IS11 (IS21) ke snímači IS12 (IS22) je větší než 60ms , tak je zvonu udělen větší pracovní impulz, tedy 18ms . Samozřejmě tyto hodnoty nejsou odzkoušeny přímo na zvonícím stroji a jistě by došlo k jejich korekcím.

I v tomto programu je udělení impulzu zvonu posunuto do krajní polohy. Je to opět z důvodu špatného vyvážení zvonu a tak část energie je spotřebována bržděním zvonu v krajní poloze a část slouží pro pohyb zvonu. Tím korigujeme špatné vyvážení. Princip, který používá tento program je shodný s variantou jedna. Hodnoty nastavení obou časovačů zůstávají původní.

Vzhledem k tomu že varianta č.1 byla úspěšně odzkoušena a na návrh provozovatele ponechána jako program pro chod zvonů v Bukovince, nedošlo ke zkoušení této programové verze. Samozřejmě to bylo způsobeno také časovým omezením, které při náročnosti obou programů na správné odladění je velké a zkoušky na reálném objektu jsou jinak velice omezené, nebyl tento program přímo v reálu odzkoušen.

8 POZNATKY ZÍSKANÉ PŘI ŘEŠENÍ, NAVRŽENÉ ZMĚNY

Dovolím si tvrdit, že navržená řešení v programu Varianta 1 odstraňují všechny nedostatky zvonícího stroje (vyjma nevhodného tvaru srdce zvonu). Tento program odstranil nejen časově náročné uvedení zvonu do harmonického pohybu, tak nesprávné vyvážení obou zvonů a neharmoničnost chodu zvonů. Tyto problémy byly odstraněny zejména načasováním udělení impulzu před krajní polohu zvonu. Tím došlo k rozložení energie impulzu na brždění zvonu před rovnovážnou polohou a na energii, která udržuje zvon v rovnovážném pohybu. Samotná velikost impulzu zůstala stejná jako v původním programu, pouze došlo k jejímu rozdělení. Do programu byl přidán ještě startovací impulz, který nám uvádí zvon do rovnovážného pohybu.

Samotný program řeší všechny nedostatky a to jak programové, tak nedostatky zvonícího stroje (vyjma nevhodného tvaru srdce zvonu), je však velice náročný na samotné odladění a jeho vznik byl ovlivněn podmínkami, které stanovil provozovatel. A to zejména podmínka řešení nesprávného vyvážení pomocí softwaru. Dalším limitujícím faktorem byla nemožnost zaznamenávat velikost úhlu výkyvu zvonu, a to hlavně z těchto důvodů:

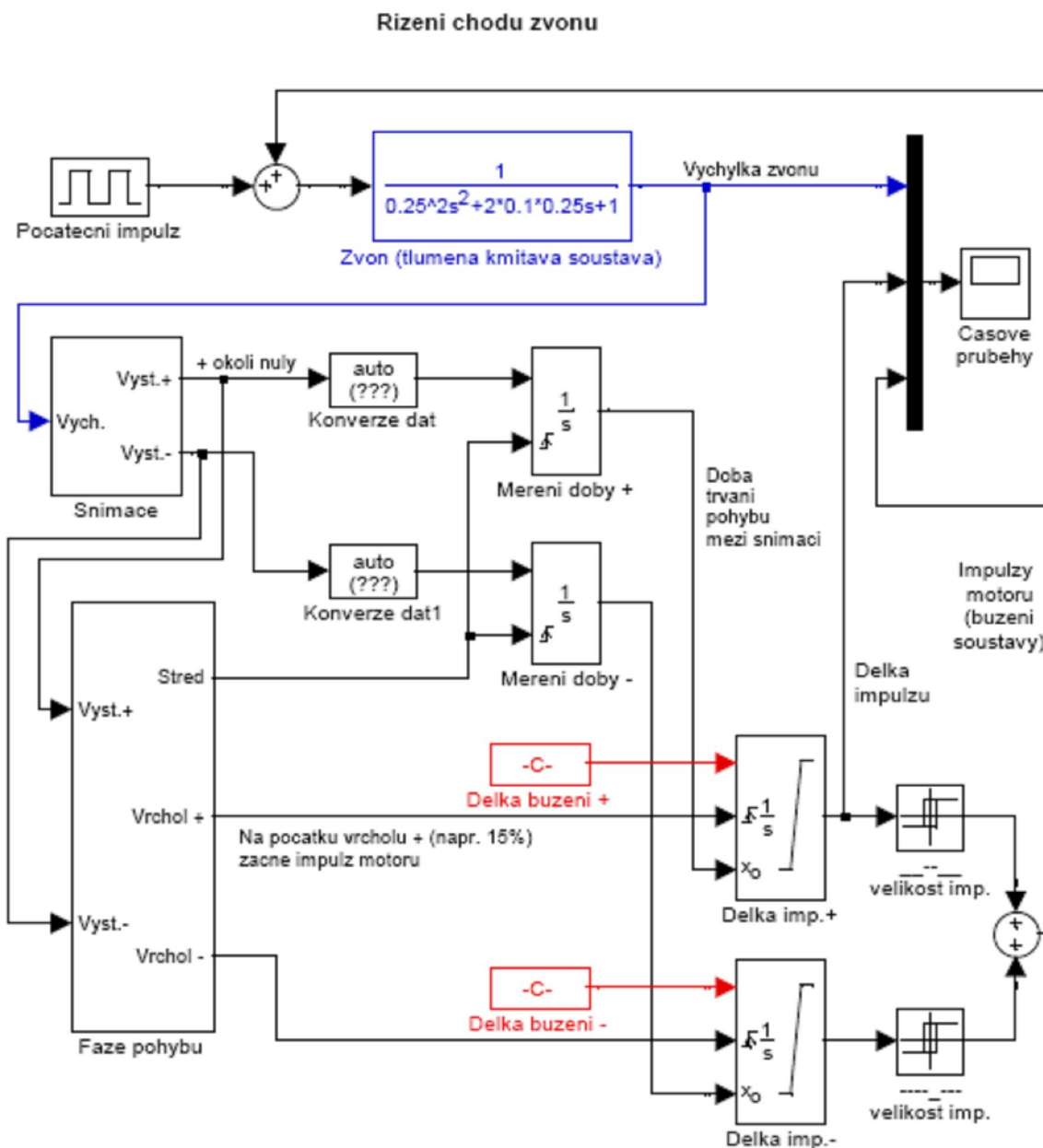
- *Časová omezenost* – vzhledem k tomu že zkoušky byly na reálném objektu omezeny na minimální hodnotu, bylo by další měření časově náročné, tím byl prostor pro další experimentování velice omezen.
- *Ztísněné podmínky pro měření* – vzhledem k tomu, že prostor ve věži kostela je velice stísněn, samotná výška, ve které se nachází zvony, je více jak tři metry od podlahy, bylo by velice náročné instalovat zařízení pro měření výkyvu a tato měření během zkoušení vyhodnocovat.
- *Nedostupnost přístroje* – vhodný přístroj pro měření výkyvu nebyl dostupný

Na základě získaných poznatků si dovoluji navrhnout další změny vedoucí ke zlepšení stavu zvonících strojů, tak ke zdokonalení chodu kostelních zvonů.

- *Dovyvážení zvonu* – pomocí protizávaží vyvážit zvon tak, aby byl úhel výkyvu v obou krajních polohách stejný
- *Instalace reverzního elektromotoru* – tento motor by umožnil udělovat zvonu impulz v obou směrech pohybu zvonu – tím by se zpřesnila regulace chodu kostelních zvonů
- *Řízení chodu zvonu s proměnnou délkou impulzu* – toto řízení využívá možnosti reverzního elektromotoru a délka jeho impulzu je závislá na rychlosti průchodu zvonu rovnovážnou polohou.

8.1 Simulace řízení chodu zvonu s proměnnou délkou impulsu [8]

Tento model byl vytvořen v prostředí Matlab-Simulink a simuluje pohyb zvonu a jeho následnou regulaci.

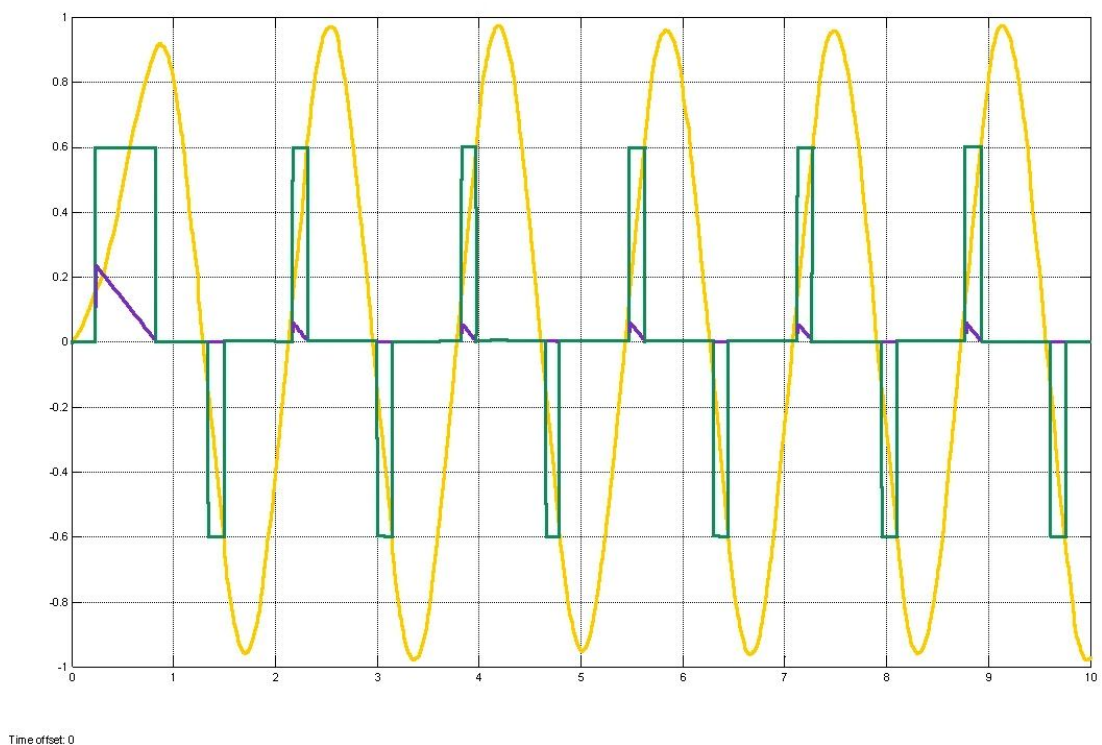


Obr. 31 Model řízení chodu zvonu s proměnnou délkou impulsu

Tento model simuluje pohyb zvonu, který pak vyhodnocuje a uděluje zvonu impuls. Velikost impulsu je závislá na době průchodu zvonu rovnovážnou polohou. Oba snímače umístěné v této poloze měří dobu průchodu ocelové destičky touto polohou a na základě této hodnoty dochází k udělování impulsu. Impulz je tím větší, čím je tato změřená doba větší. Proto je velikost impulsu proměnná a závisí vždy na rychlosti průchodu zvonu rovnovážnou polohou. Velikost těchto impulsů je možno regulovat pomocí parametru C, který nám zvětšuje nebo zmenšuje velikosti impulsu.

Model využívá možnosti reverzního elektromotoru, to znamená udělování impulsu motoru v obou směrech otáčení. To, jakým způsobem dochází k udělování impulsu je zřejmé i následujícího grafu, kde je znázorněno udělování impulsu a samotné vychýlení zvonu. Hodnota 1 = 50° výchylky zvonu.

Samozřejmě v tomto modelu nejsou započítány další vlivy působící na soustavu, ale jednoznačně lze říci, že takového řízení by bylo nejpřesnější, protože velikost impulsu je přímo závislá na rychlosti a k regulaci chodu zvonu dochází 2 x za jeden pracovní cyklus zvonu. Samotný program je pak přiložen na CD nosiči.



Obr. 32 Graf výchylky zvonu

9 ZÁVĚR

V této práci jsem se naučil pracovat s programovatelným automatem Siemens Simatic S7-200 CPU 221, který patří dnes k nejpoužívanějším zařízením u nás. Dále jsem se seznámil s problematikou automatického chodu řízení kostelních zvonů, dvojice zvonů a problematikou, která nepřímo souvisí s tímto tématem, jako jsou teorie zvonů, zvonící stroje atd. a to přesto, že literatura k této problematice není příliš rozšířena.

Mým úkolem v této práci bylo pomocí programového vybavení odladit chod kostelních zvonů na základě přání a podmínek, určených provozovatelem objektu, ve kterém se nachází zvonící stroj s dvojicí zvonů.

Vzhledem k tomu že zvonící stroj použitý na reálném objektu vykazuje určité netypické vlastnosti, jako je například nesprávné vyvážení zvonů. Také chod obou zvonů byl neharmonický a ve svém původním stavu poškozoval oba zvony. Na základě těchto poznatků jsem vypracoval dvě varianty programů, které tyto netypické vlastnosti odstraňují. Oba programy jsou si svou charakteristikou podobné, liší se v počtu impulzů udělujících zvonu. Varianta 1 používá pro řízení chodu zvonů dva a Varianta 2 používá tři délky impulzů. Během zkoušení se ukázalo, že Varianta 1 je vhodná pro řízení chodu kostelních zvonů na zvonícím stroji v obci Bukovinka. Tuto Variantu jsem úspěšně odzkoušel a odladil na tomto zvonícím stroji.

Na závěr bych chtěl poznamenat, že obě Varianty programů jsou vhodné pro řízení takto netypických zvonících strojů. A na základě poznatků, které jsem při řešení získal, jsem navrhl další změny vedoucí ke zlepšení chodu kostelních zvonů v kostele obce Bukovinka.

SEZNAM POUŽITÉ LITERATURY

- [1] VYPLAŠIL J. *Řízení zvonu inteligentním relé Easy*. Brno, 38 s. Bakalářská práce na Fakultě strojního inženýrství na Ústavu automatizace a informatiky. Vedoucí bakalářské práce Ing. Zdeněk Němec , CSc.
- [2] MANOUŠEK P. *Zvonařství*. 1. vyd. Praha : GRADA, 2006.187 s. ISBN80-247-1294-6
- [3] WIKIPEDIE. *Zvon* [online].2002, 3. května 2002, [cit. 18. ledna 2008] Dostupné z: <http://cs.wikipedia.org/wiki/Zvon>
- [4] SIEMENS, Programovatelný automat S7-200: Systémový manuál. vyd. 06/2004, Nürnberg : Siemens 2004. 534 s. 6ES7298-8FA24-8BH0
- [5] Oficiální stránky obce Bukovinka
<http://www.bukovinka.cz>
- [6] Oficiální stránky firmy Impuls – B
<http://www.impulsb.com/>
- [7] www.mapy.cz
- [8] NĚMEC, Z. *Model soustavy v prostředí MATLAB-SIMULINK*. Pracovní materiál FSI VUT v Brně z roku 2008.

SEZNAM PŘÍLOH

	Programové prostředí	Operační systém	Hardware	Strana
Program Varianta 1	Step7 Microwin v.4.0. a vyšší	Microsoft Windows 2000 a vyšší	min. 100 MB volného prostoru na HD	67 až 69
Program Varianta 2				71 až 75

Program Varianta 1		
Network 1		
LD	SM0.1	// příznak zapnutí
CALL	SBR_0	// inicializační subrutina
Network 2		
LDN	ZV1	// načtení požadavku ZV1
CALL	SBR_2	// pokud zvon vypnut, nastavení START1=0
NOT		// stav ZV1
AN	START1	// pokud nebyl 1. impuls
A	IS11	// 1. snímač = 1 ?
A	IS12	// 2. snímač = 1 ?
=	stav1A	// rutina - je zvon v klidu ?
LD	stav1A	// načtení požadavku stav1A
TON	T37, +10	// zpoždění 1 s
Network 3		
LDI	IS12	// stav snímače IS12
TON	T32, +57	// čekání na hranu IS11 - 57 ms
Network 4		
LDI	IS11	// stav snímače IS11
EU		// pokud je náběžná hrana
A	T32	// prodleva je větší jak hodnota T32
S	IMP11, 2	// rychlost pohybu zvonu je malá
Network 5		
LD	ZV1	// načtení požadavku ZV1
LPS		// duplikace horní hodnoty zásobníku
A	T37	// zvon je po dobu 1s v klidu
AN	START1	// nebyl udělen žádný impuls
S	stav1B, 2	// požadavek na startovací impuls
LPP		// druhá hodnota na 1 místo
A	IMP11	// je rychlost pohybu zvonu malá
=	stav1C	// požadavek na pracovní impuls
Network 6		
LD	stav1B	// požadavek startovací impuls
TON	T35, +40	// délka startovacího impulsu
LD	stav1C	// požadavek na pracovní impuls
TON	T33, +24	// prodleva do udělení impulsu
LD	T33	// načtení časovače T33
TON	T34, +15	// délka pracovního impulsu
LDN	T35	// pokud je časovač T35 negativní
A	stav1B	// a je požadavek na startovací impuls
LDN	T34	// pokud je časovač T34 negativní
A	T33	// a uběhla prodleva do udělení impulsu
OLD		// součet předchozích hodnot zásobníku
=I	M1	// výstup na motor 1
LD	T34	// pokud byl udělen pracovní impuls
O	T35	// nebo byl udělen startovací impuls
CALL	SBR_1	// volání podprogramu SBR_1

Program Varianta 1		
Network 7		
LDN	ZV2	// načtení požadavku ZV2
CALL	SBR_6	// pokud zvon vypnut, nastavení START2=0
NOT		// stav ZV2
AN	START2	// pokud nebyl 1. impuls
A	IS21	// 1. snímač = 1 ?
A	IS22	// 2. snímač = 1 ?
=	stav2A	// rutina - je zvon v klidu ?
LD	stav2A	// načtení požadavku stav2A
TON	T39, +10	// zpoždění 1 s
Network 8		
LDI	IS22	// stav snímače IS22
TON	T96, +52	// čekání na hranu IS21 - 52 ms
Network 9		
LDI	IS21	// stav snímače IS21
EU		// pokud je náběžná hrana
A	T96	// prodleva je větší jak hodnota T96
S	IMP21, 2	// rychlost pohybu zvonu je malá
Network 10		
LD	ZV2	// načtení požadavku ZV2
LPS		// duplikace horní hodnoty zásobníku
A	T39	// zvon je po dobu 1s v klidu
AN	START2	// nebyl udělen žádný impuls
S	stav2B, 2	// požadavek na startovací impuls
LPP		// druhá hodnota na 1 místo
A	IMP21	// je rychlost pohybu zvonu malá
=	stav2C	// požadavek na pracovní impuls
Network 11		
LD	stav2B	// požadavek startovací impuls
TON	T99, +35	// délka startovacího impulsu
LD	stav2C	// požadavek na pracovní impuls
TON	T97, +22	// prodleva do udělení impulsu
LD	T97	// načtení časovače T97
TON	T98, +15	// délka pracovního impulsu
LDN	T99	// pokud je časovač T99 negativní
A	stav2B	// a je požadavek na startovací impuls
LDN	T98	// pokud je časovač T98 negativní
A	T97	// a uběhla prodleva do udělení impulsu
OLD		// součet předchozích hodnot zásobníku
=I	M2	// výstup na motor 2
LD	T99	// pokud byl udělen startovací impuls
O	T98	// nebo byl udělen pracovní impuls
CALL	SBR_5	// volání podprogramu SBR_5

Varianta 1 - Podprogramy		
Podprogram SBR_0		
LDN	SM0.0	
=	START1	// 1. impuls 1. zvon = 0
=	START2	// 1. impuls 2. zvon = 0
Podprogram SBR_1		
LD	SM0.0	
S	START1, 2	// zvonu již byl udělen první impuls
R	IMP11, 2	// konec - (rychlost pohybu zvonu je malá)
R	stav1B, 2	// konec požadavku na start impuls
Podprogram SBR_2		
LD	SM0.0	
R	START1, 2	// zvonu již byl udělen první impuls
Podprogram SBR_5		
LD	SM0.0	
S	START2, 2	// zvonu již byl udělen první impuls
R	IMP21, 2	// konec - (rychlost pohybu zvonu je malá)
R	stav2B, 2	// konec požadavku na start impuls
Podprogram SBR_6		
LD	SM0.0	
R	START2, 2	// zvonu již byl udělen první impuls

Program Varianta 2		
Network 1		
LD	SM0.1	// příznak zapnutí
CALL	SBR_0	// inicializační subrutina
Network 2		
LDN	ZV1	// načtení požadavku ZV1
CALL	SBR_2	// pokud zvon vypnut, nastavení START1=0
NOT		// stav ZV1
AN	START1	// pokud nebyl 1. impuls
A	IS11	// 1. snímač = 1 ?
A	IS12	// 2. snímač = 1 ?
=	stav1A	// rutina - je zvon v klidu ?
LD	stav1A	// načtení požadavku stav1A
TON	T37, +10	// zpoždění 1 s
Network 3		
LDI	IS12	// stav snímače IS12
TON	T32, +60	// čekání na hranu IS11 - 60ms
TON	T96, +40	// čekání na hranu IS11 - 40ms
Network 4		
LDNI	IS11	// stav snímače IS11
LPS		// duplikace horní hodnoty zásobníku
A	T32	// prodleva je větší jak hodnota T32
S	IMP11, 2	// je potřeba větší pracovní impuls
R	IMP12, 2	// a zároveň není potřeba malého impulsu
LPP		// druhá hodnota na 1 místo
A	T96	// prodleva je větší jak hodnota T96
AN	T32	// a zároveň prodleva je menší jak hodnota T32
S	IMP12, 2	// je potřeba malého pracovní impulsu
R	IMP11, 2	// a zároveň není potřeba velkého prac. impulsu
Network 5		
LDI	IS12	// stav snímače IS12
ED		// pokud je sestupná hrana
AI	IS11	// stav snímače IS11
S	STRANA1A, 2	// sepne trvale STRANA1A
Network 6		
LDI	IS11	// stav snímače IS11
ED		// pokud je sestupná hrana
AI	IS12	// stav snímače IS12
R	STRANA1A, 2	// rozepne trvale STRANA1A

Program Varianta 2		
Network 7		
LD	ZV1	// načtení požadavku ZV1
LPS		// duplikace horní hodnoty zásobníku
A	T37	// zvon je po dobu 1s v klidu
AN	START1	// nebyl udělen žádný impuls
S	stav1B, 2	// požadavek na startovací impuls
LPP		// druhá hodnota na 1 místo
A	STRANA1A	// zvon je ve správném směru pohybu
LPS		// duplikace horní hodnoty zásobníku
A	IMP11	// je potřeba větší pracovní impuls
=	stav1C	// požadavek na větší pracovní impuls
LPP		// druhá hodnota na 1 místo
A	IMP12	// je potřeba menší pracovní impuls
=	stav1D	// požadavek na menší pracovní impuls
Network 8		
LD	stav1B	// požadavek na startovací impuls
TON	T38, +3	// délka startovacího impulsu
LD	stav1C	// požadavek na větší pracovní impuls
TON	T33, +30	// prodleva do udělení většího impulsu
LD	T33	// načtení časovače T33
TON	T34, +18	// délka většího pracovního impulsu
LD	stav1D	// požadavek na menší pracovní impuls
TON	T35, +30	// prodleva do udělení menšího pracovního impulsu
LD	T35	// načtení časovače T35
TON	T36, +15	// délka malého pracovního impulsu
LDN	T38	// pokud je časovač T38 negativní
A	stav1B	// a je požadavek na udělení startovacího impulsu
LDN	T34	// pokud je časovač T34 negativní
A	T33	// a zároveň uběhla prodleva pro udělení většího imp.
LDN	T36	// pokud je časovač T36 negativní
A	T35	// a zároveň uběhla prodleva pro udělení menšího imp.
OLD		// součet předchozích hodnot zásobníku
OLD		// součet předchozích hodnot zásobníku
=I	M1	// sepnutí motoru 1
Network 9		
LD	T34	// pokud byl udělen větší impuls
O	T38	// nebo startovací impuls
O	T36	// nebo menší pracovní impuls
CALL	SBR_1	// volání podprogramu SBR_1

Program Varianta 2		
Network 10		
LDN	ZV2	// načtení požadavku ZV2
CALL	SBR_6	// pokud zvon vypnut, nastavení START2=0
NOT		// stav ZV2
AN	START2	// pokud nebyl 1. impuls
A	IS21	// 1. snímač = 1 ?
A	IS22	// 2. snímač = 1 ?
=	stav2A	// rutina - je zvon v klidu ?
LD	stav2A	// načtení požadavku stav2A
TON	T39, +10	// zpoždění 1 s
Network 11		
LDI	IS22	// stav snímače IS22
TONR	T0, +60	// čekání na hranu IS21 - 60ms
TONR	T64, +40	// čekání na hranu IS21 - 40ms
NOT		// negace
R	T0, 1	// pokud je hodnota snímače IS22 = 0, nuluje čítač
R	T64, 1	// pokud je hodnota snímače IS22 = 0, nuluje čítač
Network 12		
LDNI	IS21	// stav snímače IS21
LPS		// duplikace horní hodnoty zásobníku
A	T0	// prodleva je větší jak hodnota T0
S	IMP21, 2	// je potřeba větší pracovní impuls
R	IMP22, 2	// a zároveň není potřeba malého impulsu
LPP		// druhá hodnota na 1 místo
A	T64	// prodleva je větší jak hodnota T64
AN	T0	// a zároveň prodleva je menší jak hodnota T0
S	IMP22, 2	// je potřeba malého pracovní impulsu
R	IMP21, 2	// a zároveň není potřeba většího Impulzu
Network 13		
LDI	IS22	// stav snímače IS22
ED		// pokud je sestupná hrana
AI	IS21	// stav snímače IS21
S	STRANA2A, 2	// sepne trvale STRANA2A
Network 14		
LDI	IS21	// stav snímače IS21
ED		// pokud je sestupná hrana
AI	IS22	// stav snímače IS22
R	STRANA2A, 2	// rozezne trvale STRANA2A

Program Varianta 2		
Network 15		
LD	ZV2	// načtení požadavku ZV2
LPS		// duplikace horní hodnoty zásobníku
A	T39	// zvon je po dobu 1s v klidu
AN	START2	// nebyl udělen žádný impuls
S	stav2B, 2	// požadavek na startovací impuls
LPP		// druhá hodnota na 1 místo
A	STRANA2A	// zvon je ve správném směru pohybu
LPS		// duplikace horní hodnoty zásobníku
A	IMP21	// je potřeba větší pracovní impuls
=	stav2C	// požadavek na větší pracovní impuls
LPP		// druhá hodnota na 1 místo
A	IMP22	// je potřeba menší pracovní impuls
=	stav2D	// požadavek na menší pracovní impuls
Network 16		
LD	stav2B	// požadavek na startovací impuls
TON	T40, +3	// délka startovacího impulsu
LD	stav2C	// požadavek na větší pracovní impuls
TON	T97, +30	// prodleva do udělení většího impulsu
LD	T97	// načtení časovače T97
TON	T98, +18	// délka většího pracovního impulsu
LD	stav2D	// požadavek na menší pracovní impuls
TON	T99, +30	// prodleva do udělení menšího pracovního impulsu
LD	T99	// načtení časovače T99
TON	T100, +15	// délka malého pracovního impulsu
LDN	T40	// pokud je časovač T40 negativní
A	stav2B	// a je požadavek na udělení startovacího impulsu
LDN	T98	// pokud je časovač T98 negativní
A	T97	// a zároveň uběhla prodleva pro udělení většího. imp.
LDN	T100	// pokud je časovač T100 negativní
A	T99	// a zároveň uběhla prodleva pro udělení menšího imp.
OLD		// součet předchozích hodnot zásobníku
OLD		// součet předchozích hodnot zásobníku
=I	M2	// sepnutí motoru 2
Network 17		
LD	T40	// pokud byl udělen startovací impuls
O	T98	// nebo větší pracovní impuls
O	T100	// nebo menší pracovní impuls
CALL	SBR_5	// volání podprogramu SBR_5

Varianta 2 - Podprogramy		
Podprogram SBR_0		
LDN	SM0.0	
=	START1	// 1. impuls 1. zvon = 0
=	START2	// 1. impuls 2. zvon = 0
Podprogram SBR_1		
LD	SM0.0	
S	START1, 2	// zvonu již byl udělen první impuls
R	IMP11, 2	// konec (je třeba udělit větší pracovní Impulz)
R	IMP12, 2	// konec (je třeba udělit menší pracovní Impulz)
R	stav1B, 2	// konec požadavku na start impuls
Podprogram SBR_2		
LD	SM0.0	
R	START1, 2	// zvonu již byl udělen první impuls
Podprogram SBR_5		
LD	SM0.0	
S	START2, 2	// zvonu již byl udělen první impuls
R	IMP21, 2	// konec (je třeba udělit větší pracovní. Impulz)
R	IMP22, 2	// konec (je třeba udělit menší pracovní. Impulz)
R	stav2B, 2	// konec požadavku na start impuls
Podprogram SBR_6		
LD	SM0.0	
R	START2, 2	// zvonu již byl udělen první impuls