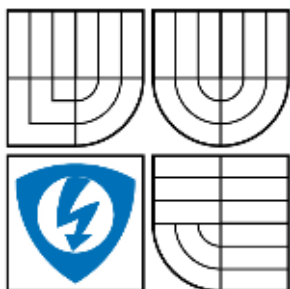


VYSOKÉ UČENÍ TECHNICKÉ V  
BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A  
KOMUNIKAČNÍCH TECHNOLOGIÍ  
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND  
COMMUNICATION DEPARTMENT OF TELECOMMUNICATIONS

## 10 PORTOVÝ GSM RELÉOVÝ SPÍNAČ S GUI

10 PORT GSM RELAY SWITCH WITH GUI

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. TOMASZ BARTULEC

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. FILIP JANOVIČ

BRNO 2008

# LICENČNÍ SMLOUVA

## POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO

uzavřená mezi smluvními stranami:

### 1. Pan/paní

Jméno a příjmení: Bc. Tomasz Bartulec  
Bytem: K Zimovůdce 262, 73514, Orlová - Lutyně  
Narozen/a (datum a místo): 12.4.1984, ve Starém Bohumíně

(dále jen "autor")

a

### 2. Vysoké učení technické v Brně

Fakulta elektrotechniky a komunikačních technologií  
se sídlem Údolní 244/53, 60200 Brno 2  
jejímž jménem jedná na základě písemného pověření děkanem fakulty:  
prof. Ing. Kamil Vrba, CSc.

(dále jen "nabyvatel")

## Článek 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: 10 portový GSM reléový spínač s GUI

Vedoucí/školitel VŠKP: Ing. Filip Janovič

Ústav: Ústav telekomunikací

Datum obhajoby VŠKP: .....

VŠKP odevzdal autor nabyvateli v:

- tištěné formě - počet exemplářů 2
- elektronické formě - počet exemplářů 2

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: .....

.....

Nabyvatel

.....

Autor

## **ANOTACE**

Návrh hardware i software pro GSM spínač s deseti spínanými výstupy a ovládáním přes SMS. Popis použitého GSM modulu a jeho vlastností, možnosti a požadavků, návrh a popis pomocných obvodů pro převedení výstupu z modulu na releové výstupy, elektrická realizace výstupu. Tvorba aplikačního software pro GSM modul, realizace požadavku na vzdálenou správu přes SMS, kontrola telefonního čísla odesílatele a informování o provedení povelu prozvoněním. Přidání AT příkazu pro nastavení zařízení. Vytvoření webového serveru zajišťujícího ovládání spínače přes internetový prohlížeč, vyřešení správy dat a komunikace serveru s zařízením. Otestování vytvořeného SW a zhodnocení výsledného návrhu z hlediska použitelnosti.

**Klíčová slova:** reset, spínač, resetátor, restart, GSM, SMS, vzdálená správa, spínání výstupu, síťové aplikace, GSM modul, Wavecom, OpenAT, GUI, webový server

## **ABSTRACT**

System hardware and software design for a GSM switch with ten switching outputs, controlled via SMS. Description of used GSM module and its properties, feasibility and requirements, design and description of supply circuits for module outputs to relay outputs decoding, plug-in realisation of outputs. GSM module application software creation, remote SMS control requests implementation, sender's phone number check and information about completed requests by call-backs. Adding AT command to set up device. Development of web server providing control of switch through Internet browser, solving communication between server and device, and data management. Testing of created SW and evaluation of final design from aspects of usage.

**Keywords:** reset, resetator, restart, GSM, SMS, remote control, switching outputs, network application, GSM module, Wavecom, OpenAT, GUI, web server

## **Prohlášení**

Prohlašuji, že svůj semestrální projekt na téma 10 portový reléový GSM spínač jsem vypracoval samostatně pod vedením vedoucího semestrálního projektu a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedeného semestrálního projektu dále prohlašuji, že v souvislosti s vytvořením tohoto projektu jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestně právních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

V Brně dne .....

..... podpis autora

## **Poděkování**

Děkuji vedoucímu diplomové práce Ing. Filipovi Janovičovi za užitečnou pomoc a cenné rady při zpracování diplomové práce.

V Brně dne .....

..... podpis autora

## Abecední seznam použitých zkratk:

API	-	Application Programming Interface
ASCII	-	American Standard Code for Information Interchange
BCD	-	Binary Coded Decimal
CMOS	-	Complementary Metal – Oxide Semiconductor
CSD	-	Circuit Switched Data
CSS	-	Cascading Style Sheet
DC	-	Direct Current
DCE	-	Data Communications Equipment
DNS	-	Domain Name System
DPS	-	Deska Plošných Spojů
FTP	-	File Transfer Protocol
GPC	-	General Purpose Connector
GPI	-	General Purpose Input
GPIO	-	General Purpose Input Output
GPO	-	General Purpose Output
GPRS	-	General Packet Radio Service
GSM	-	Global System for Mobile
GUI	-	Graphic User Interface
HTTP	-	HyperText Transfer Protocol
ICMP	-	Internet Control Message Protocol
IP	-	Internet Protocol
LED	-	Light Emitting Diode
LSB	-	Least Significant Bit
MD5	-	Message Digest algorithm 5
MSB	-	Most Significant Bit
ODBC	-	Open Database Connectivity Overwiev
PC	-	Personal Computer
PHP	-	Personal Home Page
PIN	-	Personal identification Number
POP3	-	Post Office Protocol
PPP	-	Point to Point Protocol
RAM	-	Random Access Memory
RF	-	Radio Frequency
SIM	-	Subscriber Identity Module
SMD	-	Surface Mount Device
SMS	-	Short Message Service
SMTP	-	Simple Mail Transfer Protocol
SQL	-	Structured Query Language
SW	-	Software
TCP	-	Transportt Control Protocol
UART	-	Universal Asynchronous Reciever Transmitter
UDP	-	User Datagram Protocol
URL	-	Uniform Resource Locator
WIP	-	Wavecom Internet Protocol
WWW	-	World Wide Web

## **Seznam obrázků:**

Obr. 1: Logický diagram obvodu MAX232

Obr. 2: Vnitřní zapojení obvodu 4028.

Obr. 3: Vnitřní zapojení obvodu 4011.

Obr. 4: Vnitřní zapojení obvodu 4013.

## Obsah:

Abecední seznam použitých zkratk:	- 6 -
Seznam obrázků:	- 7 -
Obsah	- 7 -
1. Úvod	- 9 -
2. Popis „resetátoru“	- 10 -
2.1. Návrh konstrukce	- 10 -
2.2. Popis funkce	- 11 -
2.3. Technická specifikace	- 13 -
3. Konstrukční část	- 13 -
3.1. Schéma zapojení	- 13 -
3.2. Použité obvody	- 14 -
3.2.1. GSM modul	- 14 -
3.2.1.1. Základní popis	- 14 -
3.2.1.2. Napájení modulu	- 15 -
3.2.1.3. Digitální vstupy a výstupy	- 15 -
3.2.1.4. Rozhraní SIM karty	- 17 -
3.2.1.5. Vstupy ON/~OFF a BOOT	- 18 -
3.2.1.6. Sériový port	- 18 -
3.2.2. Budič sériového portu	- 19 -
3.2.3. Dekadický dekodér BCD	- 20 -
3.2.4. Globální povolení zápisu na výstup	- 22 -
3.2.5. Udržování výstupu v nastaveném stavu	- 23 -
3.2.6. Ostatní pomocné součástky	- 24 -
3.3. Návrh DPS	- 25 -
4. Software modulu	- 26 -
4.1. Popis prostředí OpenAT <sup>®</sup>	- 26 -
4.1.1. Vývojové prostředí	- 26 -
4.1.2. SW architektura modulu	- 26 -
4.2. Popis hlavních funkcí	- 27 -
4.2.1. Globální úroveň	- 27 -
4.2.2. Hlavní funkce	- 28 -
4.2.3. Paměť FLASH	- 29 -
4.2.4. Rozšiřující AT příkazy	- 29 -
4.2.4.1. AT příkazy obecně	- 29 -
4.2.4.2. +RPIN	- 30 -
4.2.4.3. +RTEL	- 30 -
4.2.4.4. +RFOR	- 31 -
4.2.4.5. +RINT	- 31 -
4.2.4.6. +ROUT	- 31 -
4.2.4.7. +RAPP	- 32 -
4.2.4.8. +RWEB	- 32 -
4.2.4.9. +RSVR	- 32 -
4.2.4.10. +RNAZ	- 32 -
4.2.4.11. +RHES	- 33 -
4.2.5. Příjem SMS	- 33 -
4.2.6. Uskutečnění vybrané akce	- 34 -
4.2.7. Nastavení výstupu	- 34 -
4.3. Webové rozhraní modulu	- 35 -

4.3.1.	Popis WIP plug-inu.....	- 35 -
4.3.2.	Inicializace .....	- 36 -
4.3.3.	Zpracování povelů .....	- 36 -
4.4.	Nahrání aplikace do modulu .....	- 37 -
5.	Webové rozhraní.....	- 38 -
5.1.	Použité webové technologie .....	- 38 -
5.1.1.	Apache .....	- 38 -
5.1.2.	PHP .....	- 39 -
5.1.3.	JavaScript.....	- 40 -
5.1.4.	MySQL .....	- 41 -
5.2.	Uživatelské rozhraní .....	- 42 -
5.2.1.	Struktura stránek .....	- 42 -
5.2.2.	Přihlášení a udržování relace .....	- 43 -
5.2.3.	Hlavní stránka .....	- 43 -
5.2.4.	Ovládání.....	- 44 -
5.2.5.	Nastavení .....	- 44 -
5.2.6.	Administrace .....	- 45 -
5.2.7.	Info.....	- 45 -
5.3.	Databáze.....	- 46 -
5.3.1.	Tabulky .....	- 46 -
5.3.1.1.	Tabulka „gsm“ .....	- 46 -
5.3.1.2.	Tabulka „aktualne“ .....	- 46 -
5.3.1.3.	Tabulka „uzivatele“ .....	- 47 -
5.3.1.4.	Tabulka „telciska“ .....	- 47 -
5.3.1.5.	Tabulka „vystupy“ .....	- 47 -
5.4.	Rozhraní modulu.....	- 47 -
5.4.1.	Obecný popis .....	- 47 -
5.4.2.	Inicializace .....	- 48 -
5.4.3.	Zápis hodnot .....	- 49 -
6.	Závěr .....	- 50 -
7.	Použitá literatura .....	- 51 -
8.	Seznam příloh .....	- 52 -
A.	Příloha 1 – Schéma zapojení GSM modulu.....	- 53 -
B.	Příloha 2 – Schéma zapojení pomocných obvodů .....	- 54 -
C.	Příloha 3 – Výsledky simulace chování výstupu .....	- 55 -
a.	Vstupní úroveň 0V, tranzistor Q1 uzavřen, relé vypnuto, závěrný proud Q1. ....	- 55 -
b.	Vstupní úroveň 3,3V, tranzistor Q1 otevřen, relé zapnuto, proud cca 100mA. ...	- 55 -
D.	Příloha 4 – Návrh DPS pro pomocné obvody.....	- 56 -
a.	Pohled shora.....	- 56 -
b.	Pohled zespodu .....	- 56 -
E.	Příloha 5 – Návrh DPS pro GSM modul .....	- 57 -
a.	Pohled shora.....	- 57 -
b.	Pohled zespodu .....	- 57 -
F.	Příloha 6 – Rozložení součástek na DPS .....	- 58 -
a.	DPS GSM modulu .....	- 58 -
b.	Pohled zespodu .....	- 58 -
G.	Příloha 7 – Úvodní stránka webového rozhraní.....	- 59 -
H.	Příloha 6 – Nastavení spínače skrze webové rozhraní.....	- 60 -
I.	Příloha 6 – Ovládání spínače skrze webové rozhraní .....	- 61 -

# 1. Úvod

V dnešní době jsou počítačové sítě již nezbytnou součástí každodenního života, využívají se všude, počínaje malými sítěmi umístěnými v jednom objektu, až po velké distribuované sítě, nezřídka rozmístěné v různých městech nebo různých státech. Správa takovýchto sítí se pak realizuje vzdáleným dohledem z jednoho nebo více dohledových center, která nemusí být nutně umístěna tam, kde se nachází samotný síťový hardware. Toto řešení je ideální z hlediska ekonomické náročnosti, protože dokáže efektivně využívat techniky správy sítí. Bohužel jej lze aplikovat pouze v případě softwarového problému, kdy je třeba změnit nastavení běžících aplikací, přidat či odebrat uživatele, stáhnout či naopak obnovit data ze záložního serveru atp. Problém nastává, pokud se závada týká samotného síťového hardware.

Porucha síťového hardware, zvláště pak prvků jako jsou routery, switche nebo modemy, znemožní technikům z dohledového centra jakýkoliv přístup k ostatnímu hardware. Poruchy těchto zařízení často nebývají nijak vážné, může se jednat pouze o zaseknutí, ztrátu synchronizace apod. Často selže také automatické přepnutí na záložní komunikační kanál, který je pak třeba zprovoznit ručně. V takovém případě musí někdo v různou denní dobu vážit cestu dlouhou mnohdy několik desítek až stovek kilometrů daleko, aby zařízení třeba jen restartoval, nebo jej vypnul a připojil místo něj záložní, což je zbytečně nákladné.

Tomuto problému se snaží předejít zařízení fungující jako samočinný nebo vzdáleně ovládaný spínač obecně nazývané „resetátor“. Ten se stará o to, aby k restartu daného switche či modemu došlo bez nutnosti kamkoliv jezdit, buď automaticky, anebo vzdáleným zásahem administrátora. Restart zařízení může být vyvolán buď přerušením napájení, nebo pokud má zařízení vyvedený reset například na tlačítko, mohou využít jej. Podle způsobu, jakého se využívá k vyvolání restartu, můžeme „resetátory“ rozdělit do několika skupin. Nejjednodušší jsou časované. Ty připojené zařízení resetují periodicky v zadaném intervalu. Nevýhodou je, že nezjišťují, zda zařízení běží nebo ne, a tak k restartům dochází většinou zbytečně. U složitějších zařízení, která mají delší dobu náběhu, tak dochází ke znatelným výpadkům komunikace. U zařízení obsahujících diskové jednotky dochází také k jejich zbytečnému opotřebením. Další variantou jsou „resetátory“ fungující na principu sledování zadaného signálu, např. komunikační linky. Pokud je na lince dlouho klid, předpokládají problém a vyvolají reset. U tohoto typu může být problémem schopnost rozlišit, zda došlo skutečně k výpadku zařízení, nebo se linka momentálně nepoužívá. Poslední významnou skupinou jsou „resetátory“ využívající služeb sítě GSM.

GSM „resetátory“ využívají služeb mobilních sítí, jejichž pokrytí je dnes již takřka stoprocentní. Lze je tedy umístit prakticky kamkoliv a ovládat prakticky odkudkoliv. Nejčastěji používaným způsobem ovládání je pomocí SMS zpráv. Z důvodu nutnosti použít silnější výpočetní jednotku pro ovládání GSM části „resetátoru“ mívají tato zařízení často přídatné rozšiřující funkce, jako ovládání přes web, alarm, hlídání teploty apod., kterých je dosaženo minimálními náklady, pouze využitím řídicí jednotky v době klidu.

Tato práce popisuje 10 portový GSM „resetátor“; od návrhu, přes vlastní realizaci, programování až k doladění a uvedení do provozu. V závěru je zhodnocený průběh realizace a použitelnost navrženého zařízení pro běžný provoz.

## **2. Popis „resetátoru“**

### **2.1. Návrh konstrukce**

Základ „resetátoru“ tvoří GSM/GPRS modul Wavecom<sup>®</sup> Q2406B. Tento modul je volně programovatelný ve vývojovém prostředí OpenAT<sup>®</sup>. To umožňuje naprogramovat rozpoznávání SMS, na základě kterých pak bude docházet k resetování výstupu. Modul má vyvedené programovatelné výstupy v úrovni CMOS.

Modul obsahuje mikroprocesor, paměť RAM i paměť FLASH, takže není třeba použít žádné další výpočetní nebo paměťové obvody. Jelikož počet programovatelných výstupů modulu je pro tento „resetátor“ nedostatečný, součástí zařízení musí být i dekodér, který signály z modulu převede na požadovaný počet výstupů „resetátoru“. Ten pracuje s šesti vstupními signály, podle kterých vybírá a nastavuje hodnotu na výstupech. Silovou část „resetátoru“ tvoří relé, která jsou buzena výkonovými tranzistory. Stav na výstupech se udržuje pomocí klopných obvodů typu D.

Modul se programuje skrze standardní rozhraní RS232, které má ale vyvedené pouze v úrovních CMOS. Proto potřebuje budič portu RS232, integrovaný obvod MAX232, který zajistí přizpůsobení na úroveň portu v PC.

Napájení zařízení zajišťují dva stabilizátory, jeden s výstupním napětím 3,3V, pro napájení GSM modulu a integrovaných obvodů dekodéru, a druhý s výstupním napětím 5V pro napájení budiče sériové linky a silové části „resetátoru“. Nutnost použít dvě úrovně napájecího napětí plyne z faktu, že maximální přípustné napětí GSM modulu je 4,5V a to je příliš málo jak pro budič sériového portu, tak pro spínání relátek.

## 2.2. Popis funkce

Navržený „resetátor“ lze ovládat třemi způsoby, každý z nich má svou oblast použití a rozsah působnosti.

Základní ovládání resetátoru probíhá skrze sériovou linku pomocí AT příkazu. AT příkazy jsou podrobněji popsány v kapitole 4.2.2. Ačkoliv jejich použitelnost je omezena na nutnost kabelového spojení s řídicím PC, jsou tím nejrozsáhlejším způsobem, který umožňuje ovládat veškeré aspekty funkce „resetátoru“. Modul obsahuje standardní sadu modemových AT příkazů k ovládání samotného modulu a jeho nižších funkčních vrstev (více kapitola 4.1), plus několik dalších, přidaných pro potřeby tohoto zařízení, k ovládání samotné aplikace. Umožňují tak nastavit telefonní čísla, ze kterých budou SMS akceptovány, ručně zapnout či vypnout libovolný výstup, uložit PIN pro SIM kartu tak, aby v případě výpadku došlo k automatickému obnovení činnosti, nastavit formát zpráv a nastavit časový interval pro resetování. Ovládání je možné z jakéhokoliv programu typu Hyperterminál, umožňujícího přenášet ASCII znaky po sériové lince RS232. Vzhledem k nutnosti fyzického přístupu k zařízení není ovládání AT příkazy nijak zabezpečené. Pokud by bylo třeba zamezit nepovolanému přístupu k zařízení, předpokládá se jeho uzamčení v rozváděči spolu s dalšími citlivými přístroji.

Další možností jak „resetátor“ ovládat je pomocí SMS zpráv, které specifikují výstup a typ akce, která se má provést. Typy akce jsou tři. Buď výstup zapne, vypne, nebo resetuje. Zapnutí a vypnutí jsou trvalé, stav výstupů se ukládá do FLASH paměti, takže i po výpadku napájení se neztratí ale obnoví se do stavu před výpadkem. Reset znamená vypnutí daného výstupu na nastavitelný časový interval a následné zapnutí. Reset lze provést pouze u zapnutého výstupu, požadavek na reset výstupu, který je momentálně vypnutý, se ignoruje. Stav výstupu během resetu se neukládá, tj. pokud dojde k výpadku napájení během resetu, po obnovení napájení bude výstup zapnutý.

Zabezpečení zde je provedeno dvěma způsoby. Prvním je kontrola čísla odesílatele. V paměti FLASH je uložen seznam akceptovaných telefonních čísel a po přijetí SMS se tento seznam porovná s číslem odesílatel. Pokud je v něm toto číslo nalezeno, SMS se dále zpracuje, pokud ne, zahodí se. Druhým stupněm je kontrola úvodního slova. Obsluha „resetátoru“ má možnost zadat libovolné souvislé slovo, kterým by SMS měla začínat. Pokud se pak úvodní slovo ve vlastní SMS liší od tohoto přednastaveného, SMS se zahodí. V opačném případě se zpracuje a vykoná se povel, který zpráva obsahuje. Po úspěšném

provedení příkazu „resetátor“ krátce prozvoní číslo, ze kterého SMS přišla, aby odesílatel věděl, že vše proběhlo správně. Formát SMS zprávy lze částečně uživatelsky definovat.

Přes SMS zprávy lze tedy zařízení jen používat, není zde možnost nijak měnit jeho nastavení nebo kontrolovat stav. Na druhou stranu ale toto rozhraní umožňuje dostupnost prakticky odkudkoliv za použití jen mobilního telefonu. Přesný popis mechanismu příjmu SMS zpráv je v kapitole 4.2.3.

Posledním způsobem ovládní „resetátoru“ je webové rozhraní. Toto musí běžet na samostatném webovém serveru, jelikož je nad technické možnosti modulu, aby tak obsáhlý systém obsluhoval, a také pro klasické GPRS připojení není možné se na modul připojit, jelikož přidělované IP adresy jsou neveřejné. Pořízení statické veřejné IP adresy by pro takovouto službu neúměrně zvedalo náklady na provoz zařízení.

Webová aplikace je velmi nenáročná na systémové prostředky, může tedy běžet na téměř libovolném serveru podporujícím PHP a MySQL i jako vedlejší služba. Díky použití databáze MySQL je možné pomocí jedné aplikace na jednom serveru ovládat více „resetátorů“ rozmístěných libovolně na území republiky.

Skrze web tak lze pohodlně a intuitivně měnit stav výstupu a nastavovat část parametrů. Přesný popis je v kapitole 4.4.

Webová aplikace běží na serveru Apache s podporou PHP. Data, mezi něž patří aktuální stav výstupu a nastavení modulu, se ukládají do MySQL databáze. Data a ovládací rozhraní jsou zabezpečena před zneužitím pomocí autentizace heslem, které se do databáze ukládá šifrovaně.

IP vrstva modulu obsahuje plně funkční specifikaci protokolů TCP i HTTP, komunikace mezi serverem a GSM modulem také probíhá na principu webových formulářů. Je to řešení výhodné jak z hlediska jednoduchosti konfigurace webového serveru, tak z hlediska vývoje případných rozšíření, protože odpadá nutnost navrhovat vlastní komunikační protokol a tím i riziko, že by se tento časem stal nedostačujícím.

## **2.3. Technická specifikace**

Napájecí napětí: 5-24V  
Proudový odběr trvalý: max 1.5A  
Proudový odběr špičkový: max 4A  
Rozměry:  
Váha:  
GSM vysílaný výkon: 2W  
Výstupy: 10x relé  
Maximální spínané napětí: 250V  
Maximální spínaný proud na jeden výstup: 12A  
Operační rozsah teplot: -20°C až +55°C  
Skladovací rozsah teplot: -30°C až +85°C  
GSM pásmo: 900/1800MHz  
Paměť FLASH: 2MB  
Paměť SRAM: 256kB  
Uživatelské rozhraní: RS232  
Přednastavené parametry RS232: 115200, 8N1, RTS/CTS

## **3. Konstrukční část**

### **3.1. Schéma zapojení**

Z praktických důvodů je GSM modul spolu se SIM kartou umístěn zvlášť na vlastní desce, která se zapojuje do hlavní desky, na níž jsou umístěny zbylé obvody. Hlavními důvody pro toto uspořádání je zajištění co nejjednoduššího přístupu ke SIM kartě, prostorová úspora a taky možnost dalšího rozšíření v budoucnu. Mezi obě desky lze totiž pak vložit ještě další obvody, aniž by se do nich muselo zasahovat, nebo lze upravit jen jednu desku a druhou ponechat beze změn. Schémata byla nakreslena v programu Eagle. Protože zapojení obsahuje rozměrné výkonové součástky, rozměry zařízení jsou víceméně jimi určeny. Z tohoto důvodu jsou použity klasické součástky, které jsou sice větší než SMD součástky, ale jsou oproti nim levnější, jednodušeji se s nimi pracuje a jsou lehce vyměnitelné v případě, že dojde k jejich poškození.

Samotná schémata zapojení jsou uvedena jako přílohy. Příloha 1 obsahuje schéma zapojení části s GSM modulem, Příloha 2 pak zapojení dekodéru, silové části, budiče 232 a napájecích obvodů.

## 3.2. Použité obvody

### 3.2.1. GSM modul

#### 3.2.1.1. Základní popis

GSM modul tvořící základ „resetátoru“ je WISMO<sup>®</sup> Quik Q2406B od firmy Wavecom<sup>®</sup>. Je to kompaktní GSM modul podporující GPRS třídy 10, pracující v pásmu 900/1800MHz. Modul obsahuje mikroprocesor a paměť RAM o velikosti 256kB. Také obsahuje 2MB paměti FLASH použitelné jak pro aplikační software, tak pro vlastní data.

Modul má dvě externí rozhraní. RF rozhraní pro připojení antény a 60 pinové rozhraní značené GPC (General Purpose Connector), na který jsou vyvedeny všechny ostatní vstupy a výstupy modulu.

RF rozhraní tvoří pájecí ploška na vrchní straně modulu, ke které lze anténu připevnit buď přímo a nebo prostřednictvím koaxiálního kabelu. V druhém případě musí být stínění kabelu připojeno na zemnicí plošky, které jsou zde k tomuto účelu vyvedené. Nominální impedance je 50Ω, DC impedance je pak 0Ω. Za použití koaxiálního kabelu a externího konektoru pro připojení antény, jako je to v našem případě, by nemělo dojít ke ztrátám mezi modulem a tímto konektorem vyšším než 0,5dB. Všechny RF funkce odpovídají standardům Phase II E-GSM 900/DCS 1800. Používané frekvence jsou následující:

- Rx (E-GSM 900): 925 až 960 MHz
- Rx (DCS 1800): 1805 až 1880 MHz
- Tx (E-GSM 900): 880 až 915 MHz
- Tx (DCS 1800): 1710 až 1785 MHz

Hlavní konektor modulu označovaný jako GPC připojuje modul k externí desce obsahující externí obvody. Současně se přes něj připojuje i napájení modulu. Má 60 pinů v 0,5 mm rozestupech. Konektor dodává firma KYOCERA/AVX pod označením 14 5087 060 931 861. Protikus připojovaný na externí desku má označení 24 5087 060 X00 861.

V následujících kapitolách budou popsána pouze ta rozhraní vyvedená na GPC, která jsou v tomto projektu použita. Ostatní je možno najít v literatuře [2].

### **3.2.1.2. Napájení modulu**

Napájení modulu je velmi důležitou částí, protože nároky kladené na napájecí zdroj jsou vysoké. Modul má tři různé napájecí vstupy.  $V_{BATT}$ ,  $V_{DD}$ , a  $V_{CC\_RTC}$ . Vstup  $V_{CC\_RTC}$  slouží k udržování běhu hodin reálného času, když je odpojeno hlavní napájení. V současném stadiu nemá pro „resetátor“ význam.

Vstup  $V_{DD}$  slouží k napájení řídicího jádra modulu. Přípustný napětíový rozsah se pohybuje mezi 3.1 a 4.5 volty. Maximální zvlnění na tomto vstupu by nemělo přesáhnout  $100mV_{pp}$ .

Vstup  $V_{BATT}$  slouží k napájení RF části modulu. Optimální napájecí napětí je 3,6V, jako minimum se uvádí 3,3V a maximum je opět 4,5V. Maximální zvlnění je 50mV pro frekvence menší než 200kHz, a 50 $\mu$ V pro frekvence vyšší než 200kHz. Důležité je, aby napájecí napětí na tomto vstupu nekleslo pod minimální hodnotu během vysílání GSM. Během něj dochází k odesílání krátkých shluků dat velkým výkonem, který vyvolá proudovou špičku až 2A. Při komunikaci GSM nebo GPRS třídy 2 dochází k těmto špičkám každých 4,615 ms, a každá špička trvá 577 $\mu$ s, při komunikaci GPRS třídy 10 dochází ke špičkám opět každých 4,615ms, ale každá trvá 1154 $\mu$ s [2].

Napájecí vstupy  $V_{BATT}$  a  $V_{DD}$  mohou být spojeny a napájeny z jednoho zdroje. Pokud se k jejich napájení používá rozdílných zdrojů, musí být oba zapínané a vypínané shodně. Průměrný proudový odběr vstupů  $V_{BATT}$  a  $V_{DD}$  dohromady během GSM komunikace na 900MHz je 250mA při výkonu RF vysílače 2W a 190mA při výkonu 0,5W. Pro komunikaci v pásmu 1800 jsou tyto hodnoty nepatrně nižší, 190mA pro výkon RF vysílače 1W a 150mA pro 0,25W. Klidový odběr je pro pásmo 900MHz cca 3mA, maximálně 7mA, pro pásmo 1800MHz jsou tyto hodnoty ekvivalentně 2mA a 6,5mA [2].

Zem tvoří stínění modulu a je vhodné, aby byla připojena rovnoměrně na všechny čtyři montážní piny modulu.

### **3.2.1.3. Digitální vstupy a výstupy**

Digitální vstupy a výstupy modulu se dělí na tři typy. Jsou to vstupy, označované jako GPI (General Purpose Input), dále výstupy, označované jako GPO (General Purpose Output) a nakonec kombinované označované jako GPIO (General Purpose Input Output), u kterých se to, zda budou vstupy nebo výstupy, volí programově. Tyto vstupy pracují v napětíových úrovních jako 3V CMOS [2]. V tabulce 1 jsou uvedeny hodnoty krajních napětí pro oba směry a obě logické úrovně.

Tab. 1

Parametr	Typ I/O	Min [V]	Max [V]	Poznámka
$V_{IL}$	CMOS	-0,5	0,8	
$V_{IH}$	CMOS	2,1	3	
$V_{OL}$	1X		0,2	$I_{OL} = -1\text{mA}$
	2X		0,2	$I_{OL} = -2\text{mA}$
	3X		0,2	$I_{OL} = -3\text{mA}$
$V_{OH}$	1X	2,6		$I_{OH} = 1\text{mA}$
	2X	2,6		$I_{OH} = 2\text{mA}$
	3X	2,6		$I_{OH} = 3\text{mA}$

Některé GPIO jsou multiplexovány s jinými signály, z důvodu omezeného počtu pinů u GPC. Pro potřeby „resetátoru“ jsou tři z nich nedostupné. Dva z nich se využívají pro signály sériové linky, na poslední je připojená LED indikující stav modulu. Popis jednotlivých GPIO, které jsou k dispozici, je v Tabulce 2. Tabulka 3 pak zobrazuje výstup pro stavovou LED a v Tabulce 4 jsou popsány jednotlivé stavy modulů, které tato indikuje.

Pro ovládání 10 výstupů „resetátoru“ je použito 6 signálů. První 4 jsou použity pro výběr aktivního výstupu a jsou označeny ABCD, kde A značí LSB a D je MSB. Další je označena EN a je to globální povolení zápisu na výstup, poslední s označením DATA reprezentuje, zda budeme daný výstup zapínat nebo vypínat. Pro vlastnosti dekodéru se číslo znázorňující daný výstup na signálech ABCD zobrazuje negovaně. Mapování jednotlivých logických signálů na skutečné výstupy modulu znázorňuje Tabulka 5.

Tab. 2

Signál	Číslo pinu	I/O	Typ I/O	Popis	Multiplexováno s
GPIO0	24	I/O	CMOS/2X	General Purpose I/O	CT106/CTS2
GPIO4	53	I/O	CMOS/2X	General Purpose I/O	
GPIO5	35	I/O	CMOS/2X	General Purpose I/O	CT105/RTS2
GPO0	26	O	3X	General Purpose O	SPI_AUX
GPO1	22	O	3X	General Purpose O	
GPO2	20	O	1X	General Purpose O	CT104/RXD2
GPI	18	I	CMOS	General Purpose I	CT103/TX2
GPO3	28	O	CMOS	General Purpose O	SPI_EN

Tab. 3

Signál	Číslo pinu	I/O	Typ I/O	Popis	Multiplexováno s
FLASH LED	52	I/O	CMOS/2X	Stavová LED	GPIO 1

Tab. 4

LED status	WISMO Quik Q2406 status	
OFF	Vypnuto nebo v nahrávacím módu	
ON	Trvale	Modul zapnutý, není přihlášen k síti
	Pomalé blikání ON = 200ms OFF = 2s	Modul zapnutý, přihlášen k síti
	Rychlé blikání ON = 200ms OFF = 600ms	Modul zapnutý, přihlášen k síti probíhá komunikace

Tab. 5

Logický signál	Název výstupu	Pin	Typ
A	GPIO 5	35	CMOS/2X
B	GPO 1	22	3X
C	GPIO 0	24	CMOS/2X
D	GPO 0	26	3X
EN	GPO 3	28	CMOS
DATA	GPO 2	20	1x

Jelikož všechny výstupy budou přivedeny v dekodéru na logické obvody CMOS, proudový odběr bude minimální; nezáleží tedy na typu výstupu. Proto je jejich mapování na logické signály přizpůsobeno rozvržení DPS tak, aby signálové cesty byly co nejjednodušší.

#### 3.2.1.4. Rozhraní SIM karty

Rozhraní SIM karty je schopno přímo připojit SIM karty používající napětí 3V, což jsou všechny dnešní SIM karty. Pro starší SIM karty pracující na napětí 5V je třeba použít pomocného napětí, avšak v tomto případě to nebude třeba [2].

Rozhraní je plně kompatibilní s normou pro GSM 11.11 zahrnující funkcionalitu SIM karty. Obsahuje 5 vývodů, jejichž účel a typ popisuje Tabulka 6. Pro signál SIM\_PRES platí, že pro vloženou SIM kartu je jeho logická úroveň High, pro vyjmutou je Low [2]. Pokud není signál používán, což je případ „resetátoru“, je trvale připojen k úrovni High.

Tab. 6

Signál	Pin	I/O	Typ I/O	Popis
SIM_CLK	3	O	2X	Hodinové impulsy
SIM_RST	5	O	2X	Reset
SIM_DATA	7	I/O	CMOS/3X	Data
SIM_VCC	9	O		Napájení SIM karty
SIM_PRES	50	I	CMOS	Detekce vložení SIM karty

### 3.2.1.5. Vstupy ON/~OFF a BOOT

Vstupy ON/~OFF a BOOT jsou speciální vstupy k řízení funkcionality modulu. Jejich zapojení a typ popisuje Tabulka 7.

Vstup ON/~OFF se používá pro zapínání a vypínání modulu signálem v úrovni CMOS. Pro zapnutí modulu musí být v logickém stavu High minimálně 1s. V našem případě je signál ON/~OFF přímo připojen k napájecímu napětí, k zapnutí modulu dojde tedy rovnou po zapnutí celého „resetátoru“ [2].

Vstup BOOT aktivuje zvláštní režim pro nahrávání software přímo do FLASH paměti. Pro aplikace využívající AT příkazu je to jediná záložní cesta, jak přepsat FLASH při zaseknutí modulu. Boot režim se aktivuje připojením na logickou úroveň Low skrze 1kΩ rezistor a následným restartem modulu [2]. V tomto režimu nelze modul normálně používat. V normálním režimu je tento vstup ponechán volný. V “resetátoru“ je přiveden na jumper, který ho v případě potřeby spojí přes rezistor na zem, čímž docílíme přepnutí do Boot režimu pro případ potřeby.

Tab. 7

Signál	Pin	I/O	Typ I/O	Popis
ON/~OFF	6	I	CMOS	Zapnutí/Vypnutí modulu
BOOT	12	I	CMOS	Nahrávání do FLASH

### 3.2.1.6. Sériový port

Modul má vyvedené dva sériové porty, UART1 a UART2. Port UART2 je pomocný a v “resetátoru“ není použitý. Port UART1 obsahuje všech osm modemových signálů a taky umožňuje nahrávání aplikace. Jelikož modul nebude využíván jako modem, stačí

k ovládání modulu a nahrávání aplikace pouze čtyři z nich, a to RxD, TxD, RTS a CTS. Popis těchto signálů najdeme v Tabulce 8. Tyto signály jsou plně kompatibilní se signálním protokolem V24. Bohužel nejsou jejich napěťové úrovně kompatibilní se specifikací V28, ale používají 2,8V logiku [2], proto před připojením k PC musí být jejich úrovně upraveny převodníkem, který je popsán v kapitole 3.2.2. Sériový port byl navržen pro použití všech signálů pro datové přenosy, a proto je doporučeno používat hardwarové řízení toku (RTS/CTS), aby se zabránilo ztrátě dat. Modul je z pohledu řízení linky brán jako DCE.

Tab. 8

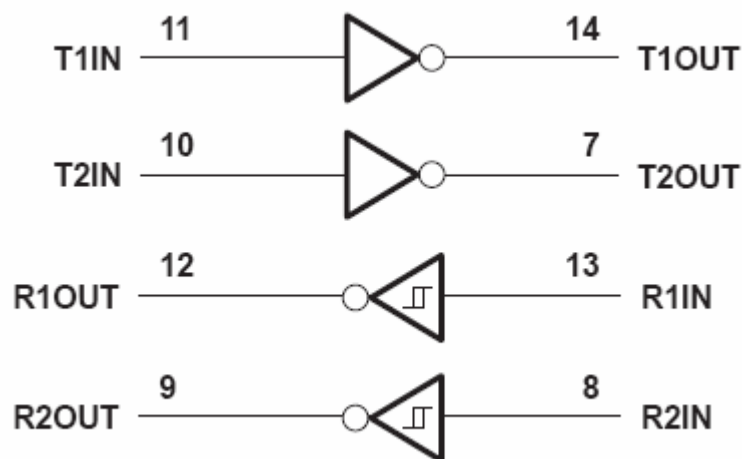
Signál	Pin	I/O	Typ I/O	Popis
CT103/TXD1	39	I	CMOS	Transmit data
CT104/RXD1	32	O	1X	Recieve data
CT105/RTS1	30	I	CMOS	Request to send
CT106/CTS1	37	O	1X	Clear to send

### 3.2.2. Budič sériového portu

Jelikož výstupy sériového rozhraní GSM modulu pracují v úrovních CMOS, nejsou kompatibilní s napěťovými úrovněmi standardu RS232. K realizaci sériového rozhraní RS232 připojitelného přímo k PC je nutno použít externí budící obvody.

Budícím obvodem sériové linky je v „resetátoru“ integrovaný obvod firmy Maxim® MAX232 [3]. Tento obvod na rozdíl od ostatních používá pouze napájecí napětí 5V, zbylé napětí je schopen si pomoci externích kondenzátorů vytvořit sám. Obsahuje dva budiče a dva přijímače, celkem je tedy schopen zpracovat čtyři signály, dva vstupní a dva výstupní. Pro potřeby „resetátoru“, který používá právě dva vstupní (TxD,RTS) a právě dva výstupní (RxD,CTS) signály, je tedy ideální.

Obvod MAX232 je schopen pracovat na přenosové rychlosti až 120 kbit/s, což je pro modul používající standardně rychlost 115200bit/s zcela postačující. Napájecí napětí se pohybuje mezi 4,5 – 5,5V, přičemž se doporučuje pro správnou funkci použít napětí alespoň 5V. Proudový odběr obvodu je velmi nízký, pohybuje se okolo 8mA. Teplotní rozsah obvodu v použité verzi je 0 – 70 °C. Na Obr. 1 je uveden logický diagram obvodu, a Tabulka 9 znázorňuje logiku obvodu. Ke své činnosti potřebuje obvod čtyři elektrolytické kondenzátory o kapacitě 1μF.



Obr. 1: Logický diagram obvodu MAX232.

Tab. 9

Vstup TIN	Výstup TOUT	Vstup RIN	Výstup ROUIT
1	0	1	0
0	1	0	1

### 3.2.3. Dekadický dekodér BCD

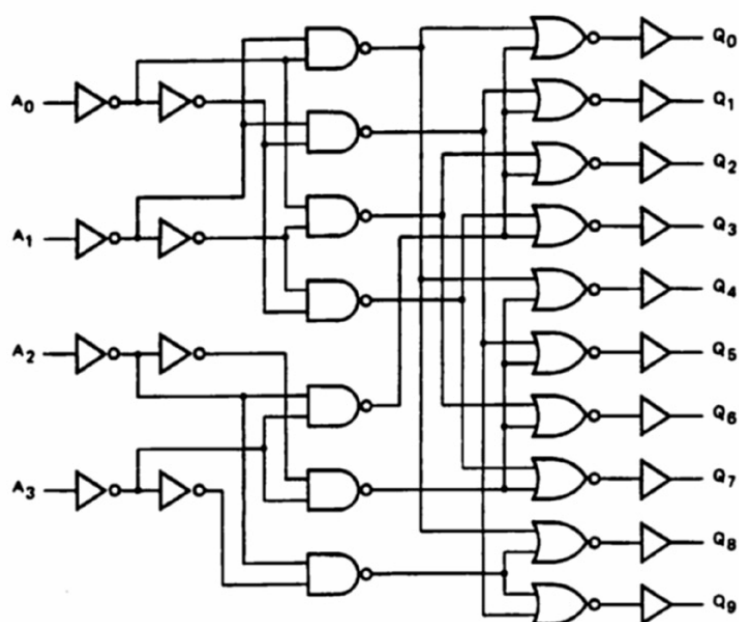
Dekadický dekodér slouží k nastavení požadovaného výstupu podle čísla přivedeného na vstup. Je tvořen obvodem CMOS 4028 [10]. Na vstup dekodéru jsou tedy přivedené signály z GSM modulu označené A, B, C, D, kde A značí LSB a D značí MSB. Odpovídající výstup má logickou hodnotu 1, všechny ostatní 0. Z praktických důvodů umístění na DPS neodpovídá pořadí silových výstupu „resetátoru“ binárním číslům přiváděným na vstup. Mapování vstupního slova na jednotlivé výstupy znázorňuje Tabulka 10. Hodnoty signálu A, B, C, D uvedené v Tabulce 10 jsou hodnoty nastavované na výstup GSM modulu. Skutečné hodnoty těchto signálů na vstupu dekodéru jsou invertované kvůli použití hradla NAND, jehož účel je popsán v následujícím odstavci.

V případě, že jako vstupní slovo přivedeme čísla 1010b až 1111b, není aktivní žádný z výstupů. Tohoto se využívá k zajištění, aby se na vstupu obvodu objevilo vždy správné vstupní slovo. Při nastavování jednotlivých signálů A, B, C nebo D může dojít v GSM modulu k vnitřnímu přerušení a modul může nastavení správné jednoho či více těchto signálů opozdit tak, že by došlo k aktivování nesprávného výstupu. A toto by, byť na

krátkou dobu, mohlo mít za následek změnu jeho stavu, která by ovšem nebyla modulem zaznamenána. Aby se tomuto předešlo, jsou tyto signály z modulu nejprve přivedeny na hradlo NAND popsané v kapitole 3.2.4. Pak, dokud není povolen zápis na výstup, je na všech vstupech dekodéru logická hodnota 1 a žádný z výstupů není aktivní. Na Obrázku 2 je zobrazeno vnitřní zapojení dekodéru 4028.

Tab. 10

Výstup „resetátoru“	A	B	C	D	EN	Výstup dekodéru
1	1	0	1	1	1	Q <sub>2</sub>
2	0	0	1	1	1	Q <sub>3</sub>
3	1	1	1	1	1	Q <sub>0</sub>
4	0	1	1	1	1	Q <sub>1</sub>
5	1	0	0	1	1	Q <sub>6</sub>
6	0	0	0	1	1	Q <sub>7</sub>
7	1	1	1	0	1	Q <sub>8</sub>
8	0	1	1	0	1	Q <sub>9</sub>
9	1	1	0	1	1	Q <sub>4</sub>
10	0	1	0	1	1	Q <sub>5</sub>



Obr. 2: Vnitřní zapojení obvodu 4028.

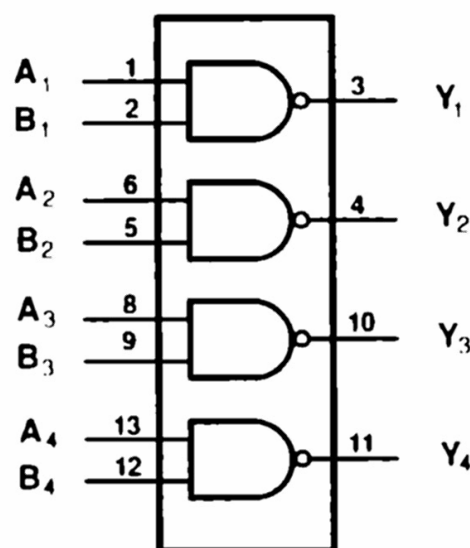
### 3.2.4. Globální povolení zápisu na výstup

Aby nedošlo k nechtěnému zápisu na náhodný výstup při změně vstupního slova do dekodéru, jsou signály A, B, C a D připojeny k vstupům obvodu 4028 přes hradla NAND, která zajistí, že dokud nebudou všechny nastaveny na platné hodnoty, bude na vstupech obvodu 4028 hodnota 1111b a všechny jeho výstupy budou tím pádem neaktivní. Signálem, který povoluje zápis na výstup, je signál EN. Ten je přiveden na druhý vstup každého hradla. Podle pravdivostní tabulky hradla NAND, zobrazené v Tabulce 11, pokud alespoň na jednom ze vstupů hradla bude logická hodnota 0, na výstupu bude 1. Pokud tedy signál EN nastavíme na hodnotu 0, budou na výstupu všech hradel logické hodnoty 0. Pokud jej nastavíme na hodnotu 1, bude na každém výstupu hradla invertovaná hodnota druhého vstupu hradla.

Hradla jsou realizována obvodem CMOS 4011 [10], který obsahuje čtyři dvouvstupová hradla NAND. Vnitřní zapojení obvodu je znázorněno na Obrázku 3.

Tab. 11

A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

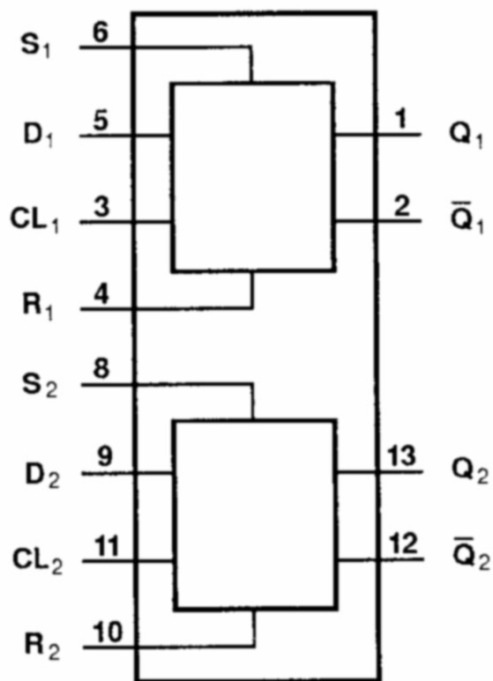


Obr. 3: Vnitřní zapojení obvodu 4011.

### 3.2.5. Udržování výstupu v nastaveném stavu

K tomu, aby po deaktivaci výstupu dekodéru zůstala nastavena hodnota na příslušném silovém výstupu „resetátoru“, je použit klopný obvod typu D. Tento obvod má na vstup D přivedený signál Data z GSM modulu, který určuje, jestli daný výstup rozpojit nebo zapojit. Na vstup CL je pak přiveden výstup z dekodéru. V okamžiku, kdy se na vstupu dekodéru objeví platné vstupní slovo, vybere se příslušný výstup a jeho stav se změní z 0 na 1. Jelikož vstup klopného obvodu CL reaguje právě na tuto změnu, dojde k zapamatování aktuální hodnoty vstupu D až do další náběžné hrany na CL. Výstup KO je pak přímo přiveden na bázi výkonového tranzistoru pracujícího jako spínač, který sepne nebo rozezne výstupní relé a tento stav na něm bude udržovat.

Klopný obvod je realizován integrovaným obvodem CMOS 4013 [10], který obsahuje dva KO typu D. Pro deset výstupů je jich tedy 5. Obvod má kromě vstupu D a CL ještě vstupy R a S, ale ty se zde nepoužijí, proto jsou trvale připojeny na 0. Aby po zapnutí „resetátoru“ nedocházelo byt' ke krátkodobému sepnutí všech výstupů, je výkonový tranzistor připojen k negovanému výstupu. Signál Data je tedy oproti stavu výstupu invertován. Zapojení obvodu je na Obrázku 4.



Obr. 4: Vnitřní zapojení obvodu 4013.

### 3.2.6. Ostatní pomocné součástky.

Mezi ostatní součástky, které stojí za zmínku, se řadí napěťové stabilizátory, výkonové tranzistory a spínací relé.

Napěťovým stabilizátorem pro napětí 5V je integrovaný obvod 7805CV [1]. Obvod má pevně nastavenou hodnotu výstupního napětí. Jedinými součástkami, které ke své činnosti potřebuje, jsou vstupní a výstupní vyhlazovací kondenzátory. Obvod je schopen dodat maximální proud 1,5A. Je schopen pracovat při napětí na vstupu až 35V, ale pak vykazuje velké tepelné ztráty, proto se doporučuje maximální napětí 24V. I při tomto napájecím napětí se obvod dosti zahřívá, je proto opatřen hliníkovým chladičem.

Napěťovým stabilizátorem pro napětí 3,3V je obvod LM1084 [5]. Stejně jako v předchozím případě obvod nepotřebuje žádné nastavovací prvky, pouze externí vyhlazovací kondenzátory. Obvod je možno připojit až na napětí 30V, ale opět platí, že při vyšších hodnotách se obvod nadměrně zahřívá, je tedy také osazen chladičem a nedoporučuje se používat vyšší napětí než 24V. Obvod dokáže dodat proud až 5A, což je postačující i s rezervou pro GSM modul [4], který si při GSM vysílání bere až 2A.

Hodnoty vyhlazovacích kondenzátorů jsou v obou případech stejné, 100nF na vstupu a 2,2μF na výstupu.

Výkonové tranzistory pro spínání výstupních relé jsou BD 137-10. Tyto tranzistory mají maximální hodnotu kolektorového proudu až 1,5A, a snesou výkon až 12,5W. Tyto hodnoty jsou daleko vyšší, než s jakými budou tranzistory pracovat, nedochází tedy k přílišnému zahřívání tranzistorů a není tak třeba pro ně žádných chladičů. Jejich funkce při předpokládaných hodnotách  $U_{BE}$  a  $I_C$  byla úspěšně simulována v programu MicroCap. Výsledky simulace jsou v Příloze 3.

Jako výstupní spínací prvek jsou použity relé H810F05C. Relé je spínané napětím 5V a odpor cívky je 47Ω. Maximální spínaný proud, který jím může protékat, je 12A při napětí 250V. Rozsah pracovních teplot je -30 až 70 °C. Rozměry jsou 29x21x13mm. Testovací napětí mezi kontakty bylo 1kV, mezi kontakty a cívkou 5kV. Nikdy se na spínací kontakty nesmí dostat napětí vyšší, jinak by mohlo dojít k poškození celého „resetátoru“. Mechanická životnost je udávána na deset miliónu spínacích cyklů.

### 3.3. Návrh DPS

Z důvodů jak praktických, tak prostorových je zařízení rozděleno na dvě DPS. Jedna obsahuje GSM modul a několik pomocných součástek, včetně držáku pro SIM kartu. Na druhé, hlavní, jsou všechny ostatní obvody, včetně výstupních a napájecích svorek a konektoru rozhraní RS232. Návrh DPS byl vytvořen v programu Eagle, verze 4.16.

Rozměry desek jsou voleny podle rozměrů krabičky, v níž bude zařízení umístěno. Rozměry hlavní desky jsou tedy 157 x 86mm, rozměry desky GSM modulu jsou 33 x 63.

Návrh hlavní DPS je uveden v Příloze 4, návrh DPS pro GSM modul je v Příloze 5. Rozmístění součástek na obou deskách je uvedeno v Příloze 6.

DPS pro umístění GSM modulu je oboustranná. Datové spoje jsou vedeny cestami o tloušťce 0.25 mm. Důvodem pro použití této tloušťky jsou malé rozestupy mezi jednotlivými kontakty konektoru GSM modulu. Tento je umístěn na horní straně desky a zabírá téměř celou její plochu. Z druhé strany desky je pak držák pro SIM kartu, stavová LED modulu a jumper umožňující přehrávání firmware modulu v případě jakýchkoliv problémů. Toto rozdělení bylo motivováno minimalizací počtu pinů nutných ke spojení se základní deskou. Napájení desky je vedeno cestami o tloušťce 1mm z důvodu nárazového vysokého proudového odběru modulu. Pro správné připojení modulu bylo nutné propojit všechny čtyři piny modulu [2].

Hlavní DPS je také oboustranná, datové spoje jsou opět vedeny v tloušťce 0.25 mm, tentokrát z nutnosti procházení mezi piny integrovaných obvodů. U širších spojů by mohly vznikat problémy při výrobě a následná nežádoucí, těžce odhalitelná propojení. Napájecí spoje jsou vedeny v tloušťkách 1 mm a 0,65 mm, v závislosti na předpokládaném zatížení a prostorových možnostech desky. Proto se doporučuje spoje vedoucí napětí 3,3V od stabilizátoru ke konektoru pro desku GSM modulu potáhnout tenkou vrstvou cínu pro lepší vodivost. Ve spodní části desky jsou umístěna výkonová relé, na jejichž spínací straně se předpokládá napětí až 250V při maximálním proudu 12A [6]. Proto jsou spoje mezi nimi a výstupními svorkami tloušťky 1,3 mm a také se doporučuje je potáhnout tenkou vrstvou cínu. Napěťové stabilizátory jsou z prostorových důvodů umístěny v levém horním rohu. Potřebují totiž alespoň menší chladič k odvodu ztrátového tepla, které, zvláště při použití vyššího napájecího napětí, není zanedbatelné. LED dioda indikující funkčnost a stavové LED diody jednotlivých výstupů mohou být umístěny přímo na desce, anebo v závislosti na krabičce, do které se zařízení umístí, vyvedeny na zvláštní indikační panel.

## 4. Software modulu

### 4.1. Popis prostředí OpenAT<sup>®</sup>

#### 4.1.1. Vývojové prostředí

OpenAT<sup>®</sup> je vývojové prostředí od firmy Wavecom<sup>®</sup> vytvořené k zjednodušenému programování modemů a modemových modulů na vyšší úrovni. Jeho základem je jazyk C. OpenAT<sup>®</sup> používá dvě různé knihovny, ADL a BASIC, k přístupu k vrstvě rozhraní API pro jednotlivé funkce modemu. Knihovna ALD je jednodušší, ale pro základní použití modulu postačující. Knihovna BASIC je komplexnější a složitější, obsahuje ale více možností, jak k jednotlivým API přistupovat. Na potřeby „resetátoru“ je knihovna ADL postačující, bude proto použita jako hlavní [7].

Jako překladač používá dva linuxové kompilátory, gcc a arm. Proto pro použití v operačních systémech Windows je třeba použít emulátor Cygwin. Součástí instalačního balíčku je i program OpenAT<sup>®</sup> Project Wizard, který umožňuje vygenerovat potřebné soubory pro nový projekt a nastavit tyto kompilátory dle potřeb uživatele. Dokáže také nový projekt přímo asociovat s programy sady Microsoft Visual Studio, tzn. je možné využít jejich prostředí k pohodlnější práci. Zmíněné kompilátory se automaticky nastaví jako výchozí.

Ačkoliv je OpenAT<sup>®</sup> založené na C, několik věcí funguje trochu jinak. Funkce main má zde spíše úlohu úvodní funkce, která definuje služby, které se budou dále vykonávat. Po jejím ukončení tak nekončí celá aplikace, jako je tomu u klasického C, ale modul čeká na definované události. Až některá z nich nastane, pokračuje příslušnou funkcí obsluhy události. Dalším omezením je nemožnost používat nekonečné smyčky. Modul má zabudovaný watchdog, který při detekci takovéto smyčky navodí automatický restart po 8 sekundách [7].

#### 4.1.2. SW architektura modulu

Uživatelská aplikace běží v modulu na nejvyšší úrovni, tj. dokáže ovládat a zachytávat události všech nižších vrstev. Dokáže tak zachytávat a zpracovávat události na všech rozhráních modulu a nastavovat na nich potřebné parametry. V případě „resetátoru“ jako hlavní rozhraní budou tyto čtyři, UART1 (RS232), SMS, GPRS a GPIO (vstupy a výstupy). Jako pomocné rozhraní se bude využívat hlasového rozhraní pro prozvonění telefonu odesílatele SMS po úspěšném provedení příkazu a také služeb pro práci s pamětmi FLASH a

RAM Kromě funkcí přístupujících k těmto rozhraním pak jsou definovány funkce nezávislé na nižší vrstvě zajišťující pomocné výpočty či pomocné operace s daty, řízení těchto rozhraní a jejich aktivaci. Všechny důležité funkce budou popsány v kapitole 4.2.

## **4.2. Popis hlavních funkcí**

### **4.2.1. Globální úroveň**

Na globální úrovni se definují globální proměnné, tj. proměnné trvale obsazené v paměti anebo globální pointery, kterým lze paměť přiřazovat libovolně za běhu programu, ale jsou viditelné ze všech míst programu. Umožňují tak uchovávat hodnoty, které jsou používány ve více funkcích.

Na rozdíl od klasického programování, kde převládá trend snižování počtu globálních proměnných na nutné minimum, je zde jejich použití nejen nutností, ale i akceptovatelným pohodlím. Některé funkce, které proměnnou potřebují, nevolají sebe navzájem, nýbrž jsou náhodně vyvolávány externími nebo interními událostmi. Hodnoty pro tyto funkce společně tehdy, pokud se zrovna nevykonává žádná z nich, musí být uloženy do globální proměnné, jinak by byly ztraceny. Druhým aspektem, který umožňuje zvýšit pohodlí programování využitím globálních proměnných, je fakt, že programovaná aplikace je jediná, která na modulu běží, může tedy využívat všechny jeho dostupné prostředky. Lze tedy určit, kolik paměti lze obsadit globálními proměnnými bez rizika, že by jí bylo třeba pro něco jiného.

Na začátku je třeba definovat část paměti pro zásobník, kde se budou ukládat návratové adresy při vnořených voláních funkcí a ostatní pomocná data např. pro náročnější výpočty. Velikost zásobníku je třeba zvolit podle uvážení, malý zásobník může přetéct, velký znamená plýtvání paměti. V naší aplikaci je velikost zásobníku 512 B, což by mělo být zcela dostačující.

Globálních proměnných je v aplikaci jen pár. Z důvodu šetření paměti jsou rozměrnější datové typy definovány pouze jako pointery, jež si paměť alokují až před samotným použitím a pak jí zase uvolní. Jako globální proměnné jsou definovány tyto:

- Tabulka stavu výstupu – je pole deseti čísel, kde indexy pole odpovídají jednotlivým výstupům a číselná hodnota odpovídá aktuálnímu stavu pro daný výstup.
- Aktuální číslo výstupu pro nastavení po startu – při nastavování výstupu do posledního zaznamenaného stavu po startu dochází k výše uvedenému problému uchování hodnoty pro dvě různé a nezávisle volané funkce.

- Aktuálně zpracovávaný výstup – pro restartování výstupu je třeba jej prvně vypnout, pak zapnout. Tato proměnná uchovává číslo výstupu, který byl vypnut a má být po uplynutí časového intervalu zapnut.
- Telefonní číslo, ze kterého přišel poslední požadavek – zde se ukládá telefonní číslo, ze kterého přišel poslední požadavek tak, aby po úspěšném dokončení požadavku mohlo být prozvoněno.

Další globální proměnné jsou poněkud specifické. Nazývají se „handlers“ a obsahují vždy číslo instance služby, pro kterou jsou definovány, např. časovače. Díky nim je lze pak v programu jednoznačně identifikovat, např. zastavit jeden konkrétní časovač bez zásahu do ostatních. Popis těchto proměnných je uveden v [7].

#### 4.2.2. Hlavní funkce

Funkce main zde na rozdíl od klasického jazyka C slouží jako inicializace konkrétních služeb, které se budou používat. Jejím ukončením se tedy neukončuje běh celé aplikace. Ta po jejím skončení běží dál ve vyčkávací smyčce, tj. čeká, až nastane některá z definovaných událostí a následně spustí příslušnou obslužnou funkci. Konkrétní popisy jednotlivých funkčních prototypů jsou uvedeny v literatuře [7].

Jako první se inicializují vstupy a výstupy. To proto, že u nich se současně nastavují i inicializační hodnoty, tj. minimalizuje se doba, kdy jsou výstupy po zapnutí v neurčitěm stavu. Následně se inicializuje obsluha FLASH paměti a volání, pro pozdější potvrzování úspěšného provedení příkazu.

Dále se definují rozšiřující AT příkazy pro obsluhu funkcí „resetátoru“. Příkazy jsou podrobněji popsány v kapitole 4.2.2. U každého příkazu se definuje jeho obslužný podprogram, který je volán vždy po jeho zachycení na libovolném portu. Dále se definuje, které formy AT příkazu jsou přípustné, a minimální a maximální počet zadávaných parametrů.

Následně se inicializuje SMS služba, a poté se nastaví výstupy do posledních stavů před vypnutím nebo restartem „resetátoru“.

Nakonec funkce ověří, zda je v paměti FLASH uložen PIN pro SIM kartu a pokud ano, pokusí se jej zadat. Po úspěšném zadání pokračuje v inicializaci, po neúspěšném vypíše chybu na sériový port UART1.

### 4.2.3. Paměť FLASH

GSM modul Q2406B obsahuje 2MB sdílené FLASH paměti. FLASH paměť obsahuje vlastní firmware modulu, rozhraní API pro příslušnou verzi OpenAT<sup>®</sup>, uživatelskou aplikaci a uživatelská data.

Existují dvě základní metody pro přístup k ní. První je pomocí služby virtuální jednotky, kdy lze do paměti nahrát datové soubory nebo soubory obsahující Wavecom<sup>®</sup> firmware k pozdější instalaci [7]. Druhým je přímý přístup do FLASH pomocí předdefinované struktury značek, které identifikují jednotlivé uložené položky. Lze tak jednoduše ukládat jak jednoduché proměnné, tak i pole nebo řetězce. Těto metody je využito i v naší aplikaci, kde je potřeba některá data zachovat i v případě výpadku napájení „resetátoru“.

### 4.2.4. Rozšiřující AT příkazy

#### 4.2.4.1. AT příkazy obecně

AT příkazy jsou specifickou formou komunikace používané pro modemy a modemové moduly [9]. Důvodem jejich používání je jejich univerzálnost a srozumitelnost pro člověka, protože jsou posílány v textové podobě. Většina AT příkazů je u všech výrobců stejná, liší se pouze ve speciálních sadách definovaných pro konkrétní zařízení a jeho konkrétní speciální vlastnosti. Toto umožňuje vzájemně komunikovat se zařízeními od různých výrobců bez nutnosti jakéhokoliv přizpůsobení. Charakteristické jsou tím, že začínají sekvencí znaků „AT“, za kterou následuje samotné znění příkazu. Sekvence „AT“ je tedy značka, která dává modemu najevo, že následující znaky jsou platným příkazem.

V použitém modulu je nainstalována základní sada AT příkazů podle specifikace GSM 07.05 a GSM 07.07. Přesný seznam norem a doporučení, z nichž AT příkazy vycházejí, i jejich kompletní seznam je uveden v [8].

Podle formátu, ve kterém se příkazy zadávají, můžeme rozdělit AT příkazy do 5 typů. Všechny tyto typy i s příklady použití jsou uvedeny v [7]. V případě „resetátoru“ se používají pouze tyto tři z nich: PARA, TEST a READ. Rozlišení a definice těchto typů jsou důležité pro následné definování vlastních AT příkazů, a jejich naprogramování.

Příkaz typu PARA je charakterizován tím, že po vlastní syntaxi následuje znak „=“ a za ním pak následuje jeden a více parametrů. Jedná se tedy o typ příkazu umožňující zaslání hodnot do modemu. Počet a typ parametrů závisí na konkrétním AT příkazu.

Příkaz typu TEST je charakterizován tím, že po vlastní syntaxi následuje sekvence znaků „=?“. V tomto případě se od modemu očekává výpis korektní syntaxe daného příkazu s uvedením typu a rozsahu parametrů, se kterými tento operuje. Má tedy pouze informativní charakter, neposkytuje informace o nastavených hodnotách, nýbrž jen o jejich formátu.

Příkaz typu READ je charakterizován tím, že po vlastní syntaxi následuje znak „?“ . Po odeslání příkazu se pak jako odpověď vypíše aktuálně nastavené hodnoty všech parametrů, které příkaz obsahuje.

Celkově existuje ještě několik dalších specifik, jež jsou pro pochopení problematiky AT příkazů důležité, avšak nejsou nutné pro pochopení cílů této práce a svým rozsahem překračují její rámec. Lze je najít v literatuře [7], [8] a [9].

#### **4.2.4.2. +RPIN**

Tento příkaz slouží k uložení PIN pro SIM kartu tak, aby po výpadku napájení mohl „resetátor“ automaticky obnovit činnost bez nutnosti vnějšího zásahu. Má definované dva typy: TEST a PARA.

Typ READ je u tohoto příkazu vynechán s ohledem na charakter ukládané hodnoty. PIN je hodnotou privátní, není vhodné, aby byl jednoduše k přečtení.. Typ TEST vypíše formát, v jakém se očekává, že budou zadány data pro zápis. Typ PARA zapíše zadané parametry do FLASH paměti. Současně provádí kontrolu, zda parametrem není nesmysl. Parametry jsou ukládány do paměti jako textový řetězec. Funkce kontroluje, zda délka řetězce není kratší než čtyři a delší než osm znaků, což je minimální a maximální délka PIN. Vstupní parametr je u tohoto příkazu jen jeden, tzn. musí být zadán. Pokud je zadán PIN nulové délky nebo kratší než čtyři znaky, je starý PIN z FLASH vymazán.

#### **4.2.4.3. +RTEL**

Tento příkaz umožňuje správu až pěti telefonních čísel, které lze uložit do FLASH paměti modulu. Tato čísla se ukládají na nejnižší volnou pozici v paměti. Při čtení se čtou všechna, která obsahují platnou hodnotu. Protože se čísla v paměti nijak neřadí ani neposouvají při vymazání některého, je jejich pořadí vesměs náhodné. Toto však nemá žádný vliv na funkčnost.

Z bezpečnostních důvodů se po přijetí SMS číslo odesílatele srovná se všemi platnými čísly uloženými na FLASH. Pokud jedno z nich souhlasí, SMS se předá k dalšímu zpracování, v opačném případě se zahodí.

Příkaz má definované stejné typy jako v předchozím případě, s tím rozdílem, že u něj není minimální a maximální délka nijak omezena. Pro typ WRITE jsou definovány dvě varianty.

První nastává, obsahuje-li příkaz pouze jeden parametr. Pak se očekává, že tímto parametrem je telefonní číslo. Pokud není paměť plná, uloží se toto číslo na nejnižší volnou pozici. Pokud paměť plná je, dojde k chybě. Jelikož modul nemá jak zjistit platnost zadaného čísla, je toto čistě na uživateli. Proto ani nekontroluje, zda se jedná o číslo či nikoliv. Jakákoliv neplatná hodnota bude mít za následek pouze zbytečné obsazení jednoho místa v paměti.

Druhá varianta nastává, pokud příkaz obsahuje parametry dva a druhý parametr je číselný a je roven nule. V tomto případě se projde celá paměť a pokud v ní existuje číslo stejné jako první parametr příkazu, vymaže se. Pokud neexistuje nebo je paměť již prázdná, dojde k chybě.

#### **4.2.4.4. +RFOR**

Příkaz nastavuje první slovo, které musí SMS obsahovat, aby byla akceptována. Je to další zabezpečovací prvek proti nechtěnému přístupu k zařízení. Pokud není parametr zadán, použije se předdefinovaný, není tedy nutno jej zadávat. Parametr není nijak délkově omezen, ale aby měl smysl, nesmí být delší než 151 znaků. Pak by se do SMS nevešla řídicí informace a všechny SMS by byly zahozeny. Používá stejné typy jako v předchozích případech.

#### **4.2.4.5. +RINT**

Příkaz +RINT zadává délku intervalu při resetování výstupu mezi vypnutím a jeho opětovným zapnutím. Délka se zadává v sekundách, minimální doporučena délka resetu je 5s, pod tuto hodnotu nemusí dojít ke korektnímu resetování zařízení připojeného na daný výstup.

Příkaz používá stejné typy jako předchozí. Je nepovinný; v případě, že není zadán, použije se předdefinovaná doba resetu.

#### **4.2.4.6. +ROUT**

Tento příkaz se od ostatních poněkud liší. Není určen pro zápis do FLASH paměti, ale pro přímé nastavení zadaného výstupu. Nepoužívá tedy typ READ, ale pouze TEST, který výpíše požadovaný formát, a typ PARA.

Pro typ PARA se zadávají dva parametry, oba jsou povinné. První určuje číslo výstupu, se kterým se bude pracovat, druhý pak akci, která se na výstupu provede. První parametr může nabírat hodnot 1-10, druhý pak pouze tyto tři: „ON“, „OFF“, „RESET“. Pokud jsou oba parametry zadány správně, požadovaná akce se vykoná a současně se nový stav výstupu nastaví do FLASH. Příkaz lze tedy použít k prvnímu nastavení výstupu.

#### **4.2.4.7. +RAPP**

Tento příkaz je pouze informativního charakteru a používá pouze typ TEST. Po zadání příkazu ve formátu tohoto typu dojde k vypsání verze aplikace. Slouží k ověření, zda je aplikace v “resetátoru“ funkční a zda je v něm požadovaná verze.

#### **4.2.4.8. +RWEB**

Tímto příkazem se aktivuje a deaktivuje možnost ovládat „resetátor“ přes webové rozhraní. Používá typy TEST, READ a PARA. Pracuje pouze s jedním číselným parametrem, který může nabývat hodnot 0 – ovládání přes web je neaktivní a 1 – ovládání přes web je aktivní a zařízení se snaží navázat a udržovat spojení s webovým serverem. Po libovolné změně tohoto parametru je třeba zařízení restartovat, aby se změna projevila.

#### **4.2.4.9. +RSVR**

Tímto příkazem se nastavuje adresa webového serveru. Na tomto serveru pak běží služba, která s modulem dokáže komunikovat a ovládat jej pomocí webového uživatelského rozhraní. Lze nastavit i sekundární server pro případ, že by primární byl dočasně nedostupný. Adresy serveru lze ukládat buď jako IP adresu, nebo jako URL. IP vrstva GSM modulu obsahuje i protokol DNS a překlad je automatický. Není tedy třeba nic víc specifikovat.

Příkaz podporuje všechny tři typy; pro typ PARA lze zadat buď jeden, nebo dva textové parametry. K přepsání uložených hodnot dojde při každém zadání nových parametrů, což je nutno mít na paměti hlavně při změně jen jednoho z parametrů. Druhý je v tomto případě třeba opsat. Pro smazání obou parametrů je třeba je zadat jako prázdný řetězec.

#### **4.2.4.10. +RNAZ**

Příkaz nastavuje název spínače, kterým se GSM modul přihlašuje k webovému rozhraní a následně i další identifikaci ve webové aplikaci. Délka by měla být nižší než 45 znaků.

Toto je maximální délka povolená na serveru. Používá stejné tři typy příkazu jako v předchozích případech.

#### **4.2.4.11. +RHES**

Příkaz nastavuje heslo spínače, kterým se GSM modul přihlašuje k webovému rozhraní. Heslo se ukládá otevřeně jako text, nicméně příkaz nepodporuje typ READ, tudíž jej nelze nijak vyčíst, kromě příslušné webové aplikace. Ostatní typy příkaz podporuje obdobně jako v předchozích případech.

#### **4.2.5. Příjem SMS**

Funkce zajišťující správné rozpoznání SMS a výběr správné akce je jakýmsi jádrem celé aplikace. Je volána pokaždé, když přijde libovolná SMS. Jako první se tedy rozhoduje, zda SMS zpracovat nebo zahodit.

Funkce nejprve načte z FLASH paměti telefonní číslo a předdefinovaný začátek zprávy. Pokud telefonní číslo není zadáno nebo se liší od čísla, ze kterého SMS přišla, je zpráva zahozena a funkce se ukončí. To samé platí pro případ, kdy se začátek SMS liší od zadaného, s tím rozdílem, že pokud začátek není zadán, je použit předdefinovaný. Předdefinovaný začátek zprávy je „RESETATOR“.

Pokud je SMS úspěšně ověřena, načte se číslo výstupu, se kterým se bude pracovat, a druh akce, který se provede. Číslo výstupu musí v SMS mít ponechanou mezeru před i za sebou, aby šlo spolehlivě rozlišit výstup 1 od výstupu 10. Druh akce se určuje klíčovým slovem podobně jako u AT příkazu +ROUT, a to „ON“, „OFF“ nebo „RESET“. Podle druhu akce se zavolá příslušná funkce pro její provedení a jako jeden z parametrů se předá telefonní číslo, ze kterého byla SMS přijata, aby po úspěšném provedení povelu bylo prozvoněno. SMS se následně zahodí, tj. nikam se neukládá, nehrozí tedy zaplnění paměti zpráv na SIM kartě.

#### 4.2.6. Uskutečnění vybrané akce

Funkce pro zapnutí a vypnutí výstupu jsou v podstatě velmi jednoduché. Funkce pouze dle svého typu zavolá funkci pro nastavení výstupu s parametrem TRUE nebo FALSE. Pokud se nastavení nepovedlo, zkusí akci opakovat, a pokud se nepovedlo podruhé, ukončí se bez prozvonění. Pokud nastavení proběhlo v pořádku, krátce se prozvoní číslo, ze kterého byl příkaz přijat.

Funkce pro reset nejdřív výstup vypne a následně spustí časovač s nastaveným intervalem. Pokud není interval zadán, použije se předdefinovaný o délce 5s. Po vypršení intervalu se výstup opět zapne a teprve po úspěšném provedení zapnutí se prozvoní telefonní číslo, ze kterého byl příkaz přijat.

#### 4.2.7. Nastavení výstupu

Tato funkce zajišťuje samotné nastavení výstupních signálů do požadovaných hodnot. Nejprve dle převodní tabulky určí z čísla výstupu hodnoty, které mají být nastaveny na signály A, B, C a D. Tyto hodnoty pak zapíše na výstup. Následně nastaví signál Data podle toho, zda má být výstup sepnut nebo rozepnut. Vstup Data se nastavuje negovaně.

Jakmile jsou všechny hodnoty nastaveny správně, aktivuje signál globálního povolení zápisu na výstup EN a zároveň spustí časovač. Časovač určuje, za jak dlouho se má signál EN deaktivovat. Po uběhnutí tohoto intervalu se časovač automaticky deaktivuje a na signál EN se zapíše 0, čímž se výstupy opět zablokují.

Po nastavení signálu Data se stav výstupu uloží do tabulky stavů do paměti FLASH, tak aby v případě výpadku napájení mohly být po startu všechny výstupy obnoveny.

Po zapnutí „resetátoru“ se spustí zvláštní varianta této funkce, která začne výstupem 1 a vždy po dokončení nastavování výstupu (deaktivace EN) vyvolá nastavení dalšího výstupu v pořadí podle tabulky stavů načtené z paměti FLASH.

### 4.3. *Webové rozhraní modulu*

#### 4.3.1. **Popis WIP plug-inu**

WIP (Wavecom<sup>®</sup> IP) plug-in [17] je soubor funkcí, metod a předdefinovaných struktur pro umožnění snadného přístupu k technologiím založeným na TCP/IP protokolu [17]. Je zaměřen na maximální efektivitu, umožňuje nezávislost vrstev TCP/IP protokolu na sobě a používání více otevřených kanálů zároveň. Z hlediska síťové architektury se dají specifikovat tři vrstvy.

Nejnižší je vrstva síťová, zajišťuje implementaci IP protokolu na zvolený typ nosiče. V současné době jsou podporovány tyto tři: PPP, CSD a GPRS. Vybraný nosič pak může být provozován jako klient anebo jako server, tj. neustále poslouchá a čeká na příchozí spojení. V současné době není možné provozovat server na GPRS. Inicializace nosiče, jeho připojení do sítě nebo uvedení do datového stavu je pak součástí inicializace samotné síťové vrstvy. Zařízení takto asociované jako nosič pro IP vrstvu WIP plug-inu pak již není přístupné pro ostatní aplikace.

Vyšší vrstvou je vrstva transportní. Ta definuje jednotlivé komunikační kanály. Počet komunikačních kanálů, jež mohou být nadefinovány, je závislý pouze na velikosti paměti RAM. Lze tedy nadefinovat i více kanálů zároveň a provozovat je paralelně. Na této úrovni jsou podporovány tři protokoly, TCP, UDP a ICMP [17]. TCP a UDP mohou být pak provozovány buď jako server, nebo jako klient. Ke každému kanálu lze přistupovat dvěma způsoby. Buď lze do něj zapisovat a vyčítat přímo kompletní TCP/UDP/ICMP paket a s ním pak dále pracovat, anebo lze nastavit parametry spojení a data vkládat pouze do datové oblasti protokolu s tím, že hlavička bude vygenerována automaticky. Je ještě jeden způsob komunikace, ale ten již nevyužívá přímého přístupu k transportnímu protokolu, a to předáním identifikátoru kanálu některému z protokolů nejvyšší vrstvy.

Nejvyšší vrstvu tvoří skupina aplikačních protokolů. V současné době jsou podporovány tyto čtyři: FTP, HTTP, POP3 a SMTP [17]. Každý z nich pracuje nad příslušným transportním protokolem a dále zvyšuje uživatelský komfort automatickým vyplňováním řídicích polí protokolu. Pro zachování přijatého systému každý z nich vytváří opět abstraktní komunikační kanál. Třebaže je definice kanálu stejná jako pro transportní vrstvu, každý typ protokolu má jiné podporované funkce a jiné návratové hodnoty. Nicméně i zde je možno zvolit mezi předáváním kompletního rámce anebo vyplněním pouze datové části a ponecháním automatického vyplnění řídicích značek.

Pro tento projekt používanou kombinací je nosič GPRS, transportní protokol TCP a aplikační protokol HTTP. Protokol HTTP podporuje verze 1.0 a 1.1 a umožňuje odesílání dat metodami GET, PUT, POST a HEAD [17]. Umožňuje i připojení pomocí proxy serveru a přímou definici HTTP hlavičky.

Přestože je tento plug-in pořád ještě ve vývoji a některé z protokolů jsou podporovány jen okrajově (obzvláště u HTTP je práce s protokolem zatím docela složitá a bez přesné znalosti specifikace HTTP nemožná), určitě je to mocný nástroj, který výrazně rozšiřuje možnosti GSM modulů firmy Wavecom®.

### **4.3.2. Inicializace**

Inicializace webového rozhraní a připojení k serveru probíhá v několika fázích. V každé z nich se uvedený postup v případě neúspěchu opakuje dvakrát. Pak se inicializace pozastaví a je nutný zásah obsluhy, aby chybu odstranila.

Jako první se vybere a inicializuje datový nosič. Tím je v tomto případě GPRS. Jakmile se GPRS rozhraní inicializuje, vytvoří se HTTP kanál, pracující nad protokolem TCP. Jako poslední v této fázi se definuje samotné HTTP spojení. Zatím se ale k serveru nepřipojuje, pouze se alokuje potřebná paměť a naplňují potřebné řídicí struktury nastavenými hodnotami.

K připojení k serveru dochází až v druhé fázi, kdy se specifikuje inicializační soubor a ten se ze serveru stáhne. Soubor by měl obsahovat povel „WEB\_HI“. Pokud z nějakého důvodu modul tento povel neobdrží, zkusí se připojit k záložnímu serveru (pokud je specifikován). Pokud ani tady neuspěje, webová služba se ukončí.

Následně se modul již řídí povely ze strany serveru. Povel „WEB\_HI“ ho vyzývá k odeslání přihlašovacích údajů. Povel „WEB\_INIT“ znamená, že server požaduje zaslání kompletního nastavení a aktuálního stavu výstupů. Tato data se před odesláním zformátují do podoby argumentu pro SQL příkaz a tímto zjednoduší serveru jejich zpracování a uložení. Posledním povelom inicializace je „WEB\_NONE“, který udává, že vše proběhlo v pořádku a modul má přejít do vyčkávacího režimu.

### **4.3.3. Zpracování povelů**

Pokud je modul již inicializován, mohou nastat z hlediska webového rozhraní následující situace. Modul je vyzván, aby změnil své nastavení, modul je vyzván, aby

změnil stav výstupu a nastavení modulu, nebo stav výstupu změní jiná událost a je třeba provést synchronizaci ze serveru.

První dva případy jsou co se týče zpracování totožné. Po obdržení povelu „WEB\_ORDER“ se z těla staženého souboru přečte AT příkaz. Ten se následně vykoná a současně se definuje funkce pro zachytávání odpovědi. Odpovědi se vyhodnotí a odešlou zpět serveru na povel „WEB\_RESP“.

Třetí případ je jakoby opakováním prostřední fáze inicializace. Modul si vynutí zaslání povelu „WEB\_INIT“ a poté již popsáním mechanismem odešle nové údaje na server.

Zpracování se ukončuje opět povelu „WEB\_NONE“, kterým se modul uvede do vyčkávacího režimu.

#### **4.4. Nahrání aplikace do modulu**

Výpis celého programu, včetně definice funkcí knihovny ADL a definice adres portů nebo konstant pro jednotlivé API, je velmi dlouhý. Z tohoto důvodu není uveden v této zprávě, ale je dodán ve formě dodatečného souboru. Zbýlé hlavičkové soubory jsou všechny již předdefinované dodavatelem, takže jejich obsah lze najít v literatuře [7].

Aplikace se nahrává přes sériový port UART1. Před nahráním je třeba přepnout modul do downloaderu příkazem „at+wdw1“. Následně se binární soubor (přípona .dwl) z výstupního adresáře použitého kompilátoru odešle protokolem 1KXModem. Po ukončení nahrávání aplikace je nutno provést restart modulu, nejlépe softwarově, příkazem „at+cfun=1“. Pro nahrávání aplikace je nutno mít vždy zapnuté řízení toku RTS/CTS, jinak může dojít k chybám přenosu a zablokování modulu.

K řízení aplikace se používá příkaz „at+wopen“. Pokud příkaz zadáme s parametrem 0, aplikace se vypne, parametr 1 ji zapne. S parametrem 2 příkaz vypíše informace ohledně verze OpenAT<sup>®</sup>. Další parametry lze použít pouze při zastavené aplikaci. Parametr 3 vymaže obsah FLASH paměti, parametr 4 celou aplikaci. Parametr 5 zablokuje spuštění aplikace do doby, kdy bude přijat speciální příkaz na její spuštění nebo hardwarové přerušení na příslušném vstupu modulu. Parametr 6 se používá ke konfiguraci datového úložiště ve FLASH.

## 5. Webové rozhraní

### 5.1. Použité webové technologie

#### 5.1.1. Apache

Webovým serverem, na kterém celá aplikace běží, je Apache. Jde o freewarový server s otevřeným zdrojovým kódem a je použitelný na většině používaných operačních systémů. Webové rozhraní „resetátoru“ lze samozřejmě provozovat na jakémkoliv jiném serveru, avšak na Apache jsou dnes umístěny více než dvě třetiny webových prezentací, takže se dá říct, že Apache je dnes jakýmsi etalonem webového serveru a ukážeme na něm některé principy a funkcionality těchto serverů.

Webový server je v podstatě systémová služba, která poslouchá na portu 80. Klientským aplikacím zasílá pomocí protokolu HTTP buď webové stránky, nebo soubory. Zvládá jak statické, tak i dynamické stránky, ale interpret příslušného jazyka (v našem případě PHP) není součástí samotného serveru a musí být doinstalován zvlášť. Součástí odpovědi je i tzv. stavový kód odpovědi. Ten udává, zda byl požadavek vyřízen v pořádku, nebo došlo k nějakým potížím. Stavový kód je trojčíferný, kód 200 udává, že požadavek byl vyřízen správně. Další kódy indikují různé druhy chyb. Obecně se dají rozdělit do tří skupin, chyby směrování (3xx), chyby související z vyřízením požadavku (4xx) a chyby serveru (5xx).

Vývoj Apache začal v roce 1993 na Illinoiské univerzitě [15]. Projekt se původně jmenoval NCSA HTTPd (odvozeno od názvu tamějšího vývojového centra, National Centre for Supercomputing Applications ). Kvůli zdržení způsobenému personálními změnami ve vývojovém teamu se první veřejná verze objevila až v roce 1995. Tehdy se také projekt přejmenoval na Apache. Od té doby jeho popularita stále roste, odhaduje se, že jej používá 69% webových serverů [15].

Ačkoliv je Apache dnes již multiplatformní, své počátky bere v Linuxu. To určuje jeho vlastnosti, zejména adresářovou strukturu a způsob konfigurace. Ta probíhá, jak je pro Linux typické, pomocí textového konfiguračního souboru. Avšak díky rozsáhlým komentářům je základní nakonfigurování serveru poměrně snadné i pro začátečníky. Samotná konfigurace je již nad rámec této kapitoly.

Dnešní distribuce serveru jsou již nastavené tak, že není třeba konfiguraci nijak měnit. Přesto je webový server důležitou částí tohoto projektu a jelikož je i potenciálním zdrojem problémů, je dobré si jeho existenci přinejmenším uvědomovat.

### 5.1.2. PHP

PHP [13] je hypertextový preprocesor, který na straně serveru interpretuje HTML stránky před jejich odesláním klientovi. Umožňuje tak vkládat vlastní skripty přímo do kódu HTML. Interpretace na straně serveru znamená ve své podstatě to, že takovýto HTML soubor se před odesláním klientovi projde, vykonají se PHP skripty a výsledný čistý HTML soubor se teprve odešle. Klient tedy samotný PHP skript nikdy neobdrží. Tímto mechanismem je zajištěna nezcizitelnost takto napsaného kódu. Mezi další výhody lze pak zařadit menší objem přenášených dat, snadnější přístup k datovým strukturám nebo databázím na straně serveru a jednoduchost zobrazení i na slabších počítačích. Mezi nevýhodami lze uvést větší zatížení serveru, které při větším počtu dotazů se může stát až neúnosné, a nemožnost interakce s objekty na webové stránce. Proto je nutné kombinovat PHP s nějakou technologií interpretovanou na straně klienta.

Jazyk PHP je jazykem interpretovaným. To znamená, že je ukládán ve své zdrojové podobě a teprve až dojde na jeho spuštění, přeloží se a vykoná. Toto řešení je sice o něco pomalejší, než v případě kompilovaného kódu, ale umožňuje jednoduchou přenositelnost mezi různými platformami a hlavně snadnější možnost úprav, kdy efekt zásahu do kódu se projeví okamžitě a není třeba ho ještě předtím překládat [13].

PHP vznikl jako soukromý projekt Rasmuse Ledorfa. Postupem času jej vylepšoval, až nakonec uvolnil k veřejnému používání pod názvem Personal Home Pages, odtud zkratka PHP. Od té doby prošlo PHP řadou změn, jež jej změnilo z nástroje pro domácí použití na základ velkých serverů zpřístupňujících rozsáhlá data [13]. V současné době je již také dostupná rozsáhlá online dokumentace, částečně i v češtině.

Zdrojový kód se vkládá přímo do HTML stránek. Z toho plyne, že k psaní PHP aplikace postačí jednoduchý textový editor. Ovšem existuje celá řada editačních nástrojů, které psaní zjednodušují zvýrazňováním klíčových slov, automatickým doplňováním jmen funkcí či strukturovaných proměnných apod. Ke spuštění je pak potřeba minimálně interpret PHP a pokud má vytvořená aplikace mít smysl, také webový server.

Co se týče zpracování dat, umí PHP ukládat data do samostatných souborů anebo spolupracovat s některou databází. Množství přímo podporovaných databázových systémů je rozličné a díky podpoře standardu ODBC je možné přistupovat z PHP k téměř všem existujícím databázím [13].

Ačkoliv je PHP kombinovatelný s různými technologiemi, nejčastěji, a to je i případ této aplikace, se používá ve spojení s serverem Apache a databází MySQL.

### 5.1.3. JavaScript

JavaScript je multiplatformní, objektově orientovaný skriptovací jazyk vytvořený v roce 1995 společností Netscape®.[14]. Na rozdíl od PHP se kód v JavaScriptu spouští až na straně klienta. To sice značně omezuje oblast působnosti takto psaného kódu pouze na objekty, jež se již na klientském počítači nachází, na druhou stranu díky tomu může provádět změny objektů bez nutnosti odesílat data znovu na server. Šetří tedy čas i množství přenášených dat. I přesto, že v naší aplikaci je JavaScript použit pouze jako pomocný programovací nástroj, je vhodné uvést alespoň krátký úvod do této problematiky, vysvětlující alespoň částečně princip jeho použití.

JavaScript vznikl jako pomocný nástroj, který měl zpřístupnit nové rozhraní prohlížeče Netscape® Navigátor, umožňující spouštět Java Applty také vývojářům nepříliš obeznámeným s touto technologií a potřebujícím jen zlomek z toho, co Java nabízela. I přes své zjevné nedostatky se tak JavaScript ujal, zvláště díky své jednoduchosti. Už od svého počátku se od samotné Javy lišil v mnoha věcech, ale i přes prohlubující se rozdíly, původně omylem přijaté, spíše marketingový název JavaScript zůstal. [14]

Autorem JavaScriptu je Brendan Eich [14], který definoval jeho podobu, jež se zachovala do teď. JavaScript je tvořen jednotlivými funkcemi, které se umísťují přímo do HTML kódu, naprosto libovolně a nezávisle na umístění ostatních funkcí. Lze tedy vytvořit blok JavaScriptových funkcí na začátku, nebo na konci HTML dokumentu (tak je to použito v případě naší webové aplikace), anebo každou funkci psát pro větší přehlednost separátně tam, kde je volána. Volání funkcí se uskutečňuje pomocí předdefinovaných událostí, vázaných na konkrétní prvky, skupiny prvků nebo celou stránku. Příkladem takovéto události může být událost „onClick“, která se vykoná po kliknutí myši na objekt, např. obrázek, se kterým je svázaná. Volání umožňuje předávání parametrů odkazem, převážně se předává odkaz na objekt, který volání zapříčinil, aby s ním funkce mohla pracovat. Umožňuje i použití složitějších strukturovaných proměnných.

V dnešní době je i přes rozličnost různých daleko sofistikovanějších programovacích nástrojů JavaScript stále používán. Svou jednoduchostí, nenáročností, rozšířeností a široké škále použitelnosti si stále udržuje pozici jednoho z nejoblíbenějších nástrojů pro tvorbu webu, jak u statických html stránek, jež potřebují jen minimum co nejjednodušeji implementovatelných funkcí, tak u dynamických projektů, jako je tento, kde je třeba jen několik jednoduchých operací na straně klienta.

#### 5.1.4. MySQL

Data odeslaná „resetátorem“ na server se ukládají do databáze MySQL [12]. Tato kapitola pojednává zkráceně o tom, jak je tato databáze postavená a jak funguje.

MySQL je jedna z nejpoužívanějších relačních databází dneška. Je poměrně jednoduchá, hodně rychlá a je multiplatformní, tzn. běží na valné většině dnešních operačních systémů. Pro nekomerční projekty je k dispozici zdarma, pro komerční je k dispozici placená verze. Je víceuživatelská a používá systémové úložiště dat, což sice ztěžuje zálohování, ale vzhledem k tomu, že dnešní databáze jsou téměř všechny systémové, tedy (ačkoliv pro tento projekt by byla vhodnější databáze souborová) prakticky není jiná možnost. Systémová databáze ale na druhou stranu umožňuje lepší a rychlejší administraci dat, jelikož její část je součástí samotného operačního systému. Toto uspořádání také podstatně zjednodušuje přístupnost dat programům, které v systému běží.

MySQL je relační databází, což znamená, že data jsou ukládána do vzájemně propojených tabulek. Tabulku tvoří sloupce a řádky. Sloupce definují prostor pro uložení jedné položky záznamu. Každý sloupec má neopakovatelný název, jež ho identifikuje, a určený typ dat, jaký lze do něj zadat. Tím se definuje i množství paměti, které bude tabulka zabírat. Řádek tabulky je pak vlastní záznam obsahující data. Všechny řádky tabulky musí být stejně dlouhé a jejich struktura je přesně definována sloupci. Pokud to definice sloupce umožňuje, lze položku v řádku ponechat prázdnou, ale nikdy nejde přidat jednu položku navíc. Řádky jsou číslovány pomocnou datovou strukturou, které se říká index. Index slouží pro spojování tabulek a vyhledávání záznamů, také může hlídat, aby v jednom sloupci nebyly duplicitní hodnoty. Index reprezentuje fyzickou strukturu dat, jak jsou v tabulce uložena. Oproti tomu druhá hodnota zastávající obdobnou funkci, klíč, reprezentuje logickou strukturu. Jako klíč tak lze zvolit libovolný sloupec tabulky.

MySQL umožňuje současné připojení několika i vzdálených uživatelů, funguje tedy na principu klient – server. Server je systémová služba, která neustále běží v hostitelském systému a naslouchá na předem specifikovaných portech. Server fyzicky spravuje databázi, zapisuje a vyčítá hodnoty z disku a zajišťuje jejich virtualizaci do podoby tabulek. Také vykonává veškeré povely, které s databází manipulují, a odesílá jejich výsledky klientům. Klient je program, který se místně nebo vzdáleně připojí k serveru, předává mu příkazy a zobrazuje výsledky. Klientů existuje celá řada, od jednoduchých textových, po zpracované s grafickou vizualizací jednotlivých tabulek a jejich relací. Klientem může být také PHP aplikace.

Jazyků, kterými komunikují klienti se samotnou databází, je několik. MySQL používá jazyk SQL, definovaný v roce 1986 jako univerzální jazyk pro databáze. Většina dnešních databází je založena právě na něm, pro jeho jednoduchost a podobnost syntaxe s reálnou řečí. Od té doby prošel SQL několika úpravami, nejvýznamnější z nich definovala v roce 1992 standard SQL2. [12]

Databáze MySQL byla pro tento projekt vybrána z důvodu své jednoduchosti a rozšířenosti. Většina webhostingů ji automaticky nabízí ke svým službám, není tedy problém celou aplikaci umístit na nějaký z nich. Co se týče výkonu a spolehlivosti, je naprosto postačující, neboť naše aplikace ukládá jen málo dat a není příliš citlivá na jejich ztrátu.

## **5.2. Uživatelské rozhraní**

### **5.2.1. Struktura stránek**

Webové stránky na straně uživatele jsou tvořené kombinací HTML a PHP kódu. Základní rozdělení okna je vytvořeno dnes již možná trochu zastaralým, nicméně výborně funkčním systémem dělení okna na rámy. Horní vodorovný rám má fixní výšku a obsahuje logo stránky. Je neměnný a je tvořen jedním statickým HTML souborem.

Dolní část okna je rozdělena na dva svislé rámy. Levý rám má fixní šířku a obsahuje menu. Menu je jednoduché a přehledné, obsahuje šest položek: Úvod, Ovládání, Nastavení, Administrace, Info a Logout. Jednotlivé položky jsou definovány jako odkazy na cílové HTML soubory. Každá položka je tedy samostatným souborem, což zvyšuje přehlednost a zjednodušuje práci serveru, který tak zpracovává jen tolik kódu, kolik je pro aktuální zobrazení nutné. Globální proměnné nutné k tomu, aby jednotlivé stránky na sebe navazovaly, se předávají přes databázi, odpadá tedy nutnost jakkoliv upravovat odkazy položek v menu případným přidáváním proměnných do URL. Jednotlivé stránky se pak směřují do hlavního okna pomocí atributu „target“.

Hlavním oknem je pravý svislý rám. Jeho šířka není omezená a využívá tak zbytku volného okna. Všechny položky v hlavním okně mají šířku definovanou procenty, což umožňuje korektní zobrazení na všech rozlišeních od 800 x 600 nahoru.

Všechny HTML i PHP soubory, které tato část webového rozhraní obsahuje, používají jednotný styl definovaný v externím souboru. Vzhled jednotlivých prvků, jejich pozadí,

velikosti a umístění lze tedy uživatelsky měnit změnou příslušného CSS souboru. Defaultní nastavení je spíše graficky decentní a zaměřené na přehlednost a funkcionalitu.

### **5.2.2. Přihlášení a udržování relace**

První stránka, která se zobrazí, je přihlašovací. Jako jediná není umístěna v rámech popsaných o kapitolu výš, ale zobrazuje se přes celé okno. Toto řešení narušuje sice uniformitu stránek, ale zjednodušuje jejich fungování, hlavně blokování menu pro nepřihlášené uživatele by bylo zbytečně složité.

Přihlašovacími údaji jsou uživatelské jméno a heslo. Uživateli lze definovat libovolné množství, jediným limitujícím faktorem zde je velikost databáze. Při jakýchkoliv problémech aplikace vypíše chybové hlášení, jinak dojde k přesměrování na vlastní stránky. Pozdější prověřování platnosti přihlášení se provádí pomocí „cookies“, je tedy nutné je mít povolené ve webovém prohlížeči. Po úspěšném přihlášení tam uloží přihlašovací skript aktuálního uživatele a zadané heslo. Po každém zobrazení nebo obnovení kterékoliv další stránky se tyto údaje znovu prověří a pokud jsou z jakéhokoliv důvodu neplatné, dojde k přesměrování opět na přihlašovací stránku. Platnost těchto údajů je nastavená na půl hodiny a obnovuje se při každé aktualizaci libovolné stránky. Po uplynutí půl hodiny od poslední akce je tedy třeba se přihlašovat znovu.

I přes toto automatické odhlašování při neaktivitě je lepší využít poslední položky v menu - Logout - a po ukončení práce se tak manuálně odhlásit. Při tomto způsobu odhlášení dojde k vymazání použitých „cookies“ a vynulování globálních proměnných. Poté dojde opět k přesměrování na přihlašovací stránku.

Z důvodu lepšího zabezpečení se hesla ukládají do databáze šifrovaně. Šifrování probíhá pomocí jednosměrné funkce MD5 [16], která neumožňuje zpětné odhadnutí hesla. Šifrovaně se heslo ukládá i do „cookies“.

### **5.2.3. Hlavní stránka**

Hlavní nebo také úvodní stránka zobrazuje základní informace o vzdáleném spínači. V horní části stránky lze vybrat spínač, se kterým se bude pracovat, ze všech zapsaných v databázi. Lze vybrat i spínače, které z jakýchkoliv důvodů nejsou právě připojené, ale pak samozřejmě načtené hodnoty odpovídají stavu při poslední aktualizaci a rovněž nelze provádět žádné změny. Neaktivní nebo neaktuální spínače v databázi zůstávají až do jejich manuálního odstranění tlačítkem Odebrat, které se objeví u položky Status v případě, že je

status Offline. Další možné statusy jsou Ok, v případě, že je spínač přihlášen, a Error v případě, že někde (buď při přenosu dat, nebo uvnitř spínače) došlo k chybě.

Po výběru aktuálně zpracovávaného spínače se jeho název uloží do databáze, aby zůstal uchován i po obnovení stránky či vybrání jiné položky z menu. Následně se zobrazí zbytek stránky. Prostřední část zobrazuje identifikační údaje o vybraném spínači, tj. Název, IP adresu, Telefonní číslo SIM karty a jeho status. Všechny údaje jsou umístěny přehledně ve středu stránky.

Níže, v rámečku, je vypsáno aktuální nastavení spínače. Jedná se o hodnoty PIN, interval pro reset, počáteční slovo pro SMS a povolená telefonní čísla, tj. hodnoty známé již z SMS rozhraní (kapitola 2.2). Jelikož hodnota PIN je víceméně soukromou záležitostí, defaultně se zobrazuje skrytě, lze ji však zobrazit pomocí tlačítka Zobraz PIN. Pokud je spínač aktivní, lze nastavení změnit buď výběrem příslušné položky v menu, nebo přímo tlačítkem Změň nastavení, které se nachází dole na stránce.

#### **5.2.4. Ovládání**

Jak již název této položky naznačuje, na této stránce se nachází přímé ovládání všech deseti výstupů spínače. V horní části je opět možnost zvolit si spínač, se kterým se bude pracovat. Dokud není vybrán, zbytek stránky se nezobrazí.

V poli Stav výstupů je výpis aktuálního stavu všech výstupů a zaškrťovací políčko pro každý výstup. Tímto se vybírá, které výstupy se budou měnit. Pomocí políčka Vybrat vše pak lze jedním klikem označit všechny. Akce, která se na vybraných výstupech provede, se volí trojicí tlačítek dole na stránce. Jednotlivé významy a průběhy těchto akcí jsou totožné s řízením přes SMS, popsáním v kapitole 2.2.

#### **5.2.5. Nastavení**

Menu Nastavení umožňuje měnit většinu nastavitelných parametrů modulu. V horní části je opět výběr spínače, který se bude nastavovat. Dokud není vybrán, zbytek stránky se nezobrazí. Pod ním je vypsána IP adresa, která jako uživatelsky neměnný parametr slouží k identifikaci aktuálně nastavovaného spínače. Název, který je jinak hlavním identifikátorem, lze totiž měnit.

Pod IP adresou následuje seznam parametrů, které lze změnit. Většina z nich je popsána v kapitole 2.2, kromě nich lze nastavit název spínače a heslo, kterým se bude přihlašovat do webové aplikace. Heslo a PIN, jakožto soukromé údaje, se zobrazují zástupnými znaky a tak se pro jistotu musí zadat dvakrát, aby nedošlo k překlepu a následným potížím. Povolené

telefonní číslo lze přidat pomocí textového pole nad příslušným tlačítkem, odebrat jej lze označením v seznamu a následným stiskem tlačítka Odebrat.

Provedené změny lze nakonec pomocí trojice tlačítek dole na stránce buď uložit, neukládat, nebo nastavit formulář na původní hodnoty.

### **5.2.6. Administrace**

Menu Administrace slouží k nastavení vlastních stránek. Jako první lze nastavit databázový server. Pokud není vyplněn, pokusí se aplikace kontaktovat přednastavený databázový server běžící na lokálním počítači. Jelikož spojení s databází je nutné pro samotné přihlášení do aplikace, je vhodné s touto volbou zacházet s nejvyšší opatrností. Toto nastavení bylo ponecháno jako možnost, neboť vzhledem k oblasti použití navrhovaného spínače se předpokládá obsluha alespoň částečně obeznámená s jeho vlastnostmi.

Další položky slouží k administraci uživatelů. Uživatele lze přidat, odebrat nebo mu změnit heslo. Heslo lze nastavit na prázdné, ovšem toto není z důvodu bezpečnosti doporučováno. Uživatel „admin“ je základním uživatelem a nelze jej odebrat. Pouze admin má právo měnit nastavení databázového serveru. Kromě této jediné výjimky jsou si všichni uživatelé rovni a mají stejná práva.

Za uživatele se považují i vzdálené spínače, jež se do aplikace připojují. Jejich název a přístupové heslo je třeba tedy přidat do seznamu ještě před prvním pokusem o připojení. Vzhledem k předpokládanému počtu uživatelů a spínačů v řádu jednotek je toto řešení jednodušší a méně náročné, než udržovat dva různé seznamy.

Pokud dojde ke změně údajů u aktuálně přihlášeného uživatele, dojde k automatickému odhlášení a přesměrování na úvodní obrazovku. Tato malá nepříjemnost je dána způsobem autentizace a kvůli jejímu minimálnímu výskytu je zbytečné celý tento způsob měnit.

### **5.2.7. Info**

Poslední stránka je čistě statická HTML stránka obsahující základní informace o tomto projektu, použitých technologiích a účelu, za jakým byl vytvořen a pro jaký platí licence použitých nástrojů.

## 5.3. Databáze

### 5.3.1. Tabulky

Veškerá data se ukládají do databáze, umístěné na MySQL databázovém serveru. Název příslušné databáze je „resetator“, a obsahuje celkem pět tabulek. Jak paměťové nároky, tak nároky na připojení jsou velmi nízké a minimálně zatěžují hostitelský systém, je tedy možné tuto databázi provozovat na již existujícím serveru bez větších potíží. V následujících kapitolách budou rozepsány struktura a účel jednotlivých tabulek.

#### 5.3.1.1. Tabulka „gsm“

Tabulka „gsm“ je hlavní tabulkou obsahující základní údaje o spravovaných spínačích. Údaje v ní se aktualizují vždy po připojení spínače k serveru a nebo při jakékoliv změně, ať už iniciované z strany uživatelského rozhraní nebo ze strany spínače. Má deset sloupců, v závorce jsou uvedeny datové typy sloupců:

- ID (Integer) – je primárním klíčem k identifikaci konkrétního spínače.
- nazev (Varchar(45)) – obsahuje název spínače.
- stav (Varchar(20)) – obsahuje aktuální stav spínače.
- ip (Varchar(20)) – aktuální IP adresa spínače.
- cislo (Varchar(20)) – telefonní číslo SIM karty spínače.
- smsslovo (Varchar(45)) – úvodní slovo pro řídicí SMS.
- pin (Varchar(8)) – PIN SIM karty spínače.
- interv (Integer) – interval resetování výstupu spínače.
- telciska (Integer) – ukazatel na příslušný řádek v tabulce „telciska“.
- heslo (Varchar(45)) – přihlašovací heslo spínače.

#### 5.3.1.2. Tabulka „aktualne“

Tato tabulka slouží k uchování dvou hodnot, které musí být přístupné z většiny stránek. Funguje spíše jako paměť na globální proměnné, obsahuje vždy jen jeden řádek. Hodnoty, které uchovává, jsou jen dvě: název spínače, který je zrovna vybrán, a příznak, zda je tento výběr platný. Obsahuje tedy tyto sloupce:

- list (Varchar (45)) – obsahuje název vybraného spínače.
- vybrat (Varchar (45)) – obsahuje příznak platnosti výběru názvu.

Druhý řádek této tabulky pak obsahuje aktuálně vyslaný povel do spínače a stav jeho zpracování na straně serveru. Tento stav se nezobrazuje, využívá se pro řízení povelů do spínače. Stav spínače zobrazovaný uživateli se získává ze spínače.

### 5.3.1.3. Tabulka „uzivatele“

Uchovává jména uživatelů a přístupová hesla. Je společná jak pro uživatele, tak pro spínače. Hesla se před uložením „zhašují“ algoritmem MD5. Obsahuje pouze tyto tři sloupce:

- ID (Integer) – je primárním klíčem k identifikaci konkrétního uživatele.
- jmeno (Varchar(45)) – obsahuje uživatelské jméno
- heslo (Varchar(45)) – obsahuje MD5 otisk hesla.

### 5.3.1.4. Tabulka „telciska“

Obsahuje povolená telefonní čísla. Je svázaná s konkrétním spínačem z tabulky „gsm“, identifikačním číslem, které se ukládá na pozici „telciska“. Telefonní čísla mohou být ve sloupcích uložena libovolně; na pořadí ani na tom, který sloupec bude prázdný, nezáleží.

Má 6 sloupců:

- ID (Integer) – je primárním klíčem k identifikaci konkrétní sady čísel.
- cislo1 (Varchar(20)) – obsahuje jedno povolené telefonní číslo.
- cislo2 (Varchar(20)) – obsahuje jedno povolené telefonní číslo.
- cislo3 (Varchar(20)) – obsahuje jedno povolené telefonní číslo.
- cislo4 (Varchar(20)) – obsahuje jedno povolené telefonní číslo.
- cislo5 (Varchar(20)) – obsahuje jedno povolené telefonní číslo.

### 5.3.1.5. Tabulka „vystupy“

Tato tabulka obsahuje stavy jednotlivých výstupů. Je svázaná s konkrétním spínačem pomocí jeho názvů.

## 5.4. Rozhraní modulu

### 5.4.1. Obecný popis

Rozhraní mezi GSM modulem a webovým serverem je postaveno na snaze o co nejjednodušší komunikaci, nenáročnost na množství přenášených dat a využití již implementovaných mechanismů v obou směrech. Z tohoto hlediska je nejpohodlnější předávat vzájemně data pomocí http hlaviček a skrytých polí. Přenos dat ve směru od serveru k modulu probíhá dvěma způsoby. Pokud server požaduje po modulu nějaká data, zasílá své požadavky pomocí krátkých povelů, jež modulu naznačují, co se od něj očekává, spolu se seznamem skrytých polí, jež se mají vyplnit. Pokud je třeba zaslat data do modulu,

vygeneruje se již kompletní AT příkaz a ten se odešle. V modulu se tento přímo vykoná, aniž by se musel jakkoliv složitěji zpracovávat nebo dekodovat. Na druhou stranu, při odesílání informací z modulu na server jsou tyto informace ve tvaru, který lze jednoduše vložit do SQL dotazu a odeslat databázovému serveru.

Aktuální stav komunikace se zapisuje do databáze do tabulky aktuálně, na řádek s identifikačním číslem 2. Pokud je řádek prázdný, je GSM modul ve stavu klidu. Stav READY nastává tehdy, jsou-li v tabulce platná data k přenesení do modulu. Stav SENDING nastává tehdy, když se modul již připojil, data si stáhl a teď je zpracovává. Stav SENDING trvá do doby, než od modulu přijde odpověď OK nebo ERROR. Ta se pak uloží do sloupce „stav“ daného spínače.

Po celou dobu zpracovávání povelu, tj. stavy READY a SENDING, není možné odeslat další povel. Teprve až přijde odpověď modulu nebo vyprší časová prodleva, odblokuje se odesílání povelu. Toto omezení by se však v praxi nemělo nijak projevit, neboť požadavky na změnu stavů spínačů nebo jejich nastavení nebudou až tak časté.

#### **5.4.2. Inicializace**

Inicializací v této kapitole je míněn proces, při kterém se spínač připojí k webovému serveru, předá mu informace o svém statusu, nastavení a stavech výstupů. Samotná inicializace webového rozhraní GSM modulu spínače již musí být v tomto okamžiku hotová. Tento proces probíhá ve třech fázích.

První fází je připojení se k serveru. Odpověď serveru je pak povel „WEB\_HI“, kterým je spínač vyzván k odeslání svého názvu a přihlašovacího jména. Heslo je v databázi uloženo šifrovaně, před porovnáním se pomocí algoritmu MD5 vypočítá otisk hesla a ten se použije pro srovnání. Pokud údaje souhlasí, dojde k vyslání dalšího povelu, pokud ne, proces se opakuje.

Dalším povelém je „WEB\_INIT“. Udává GSM modulu požadavek na zaslání kompletního nastavení, povolených telefonních čísel a stavů výstupů. Kromě toho se znovu separátně odesílá název spínače, aby se usnadnilo serveru korektní uložení přijatých dat do databáze. Přijatý název se vyhledá v tabulce „gsm“ a podle jeho existence či neexistence se pak vybere, zda se budou data měnit nebo přidávat. Zbytek dat je odeslán ve formě argumentů pro SQL dotaz, takže se pouze dosadí do správného dotazu a odešle do databáze.

Posední fází je povel „WEB\_NONE“. Tento oznamuje, že vše proběhlo v pořádku a není třeba přenos dat případně opakovat. GSM modul se po přijetí tohoto povelu uvede do

stavu vyčkávání, kdy čeká buď na nový povel od webového serveru, nebo, v případě, že došlo ke změně nějakého parametru jinou metodou než skrze web, k opětovnému vyvolání povelu „WEB\_INIT“ a korekci hodnot ve webové databázi.

### **5.4.3. Zápis hodnot**

Zápis hodnot do modulu je pro zápis nastavení i provádění výstupních akcí stejný. Má dvě fáze. V první se modulu zašle povel „WEB\_ORDER“. Po něm následuje složený AT příkaz, který specifikuje typ akce, kterou má GSM modul provést. AT příkazy jsou totožné jako v kapitole 4.2.4. Po obdržení AT příkazu čeká modul na jeho vykonání a následně odešle výsledek na server. Výsledky jsou možné dva: OK v případě úspěchu a ERROR při neúspěchu. Při neúspěchu může modul společně se zprávou ERROR odeslat i číslo chyby, která nastala. Pak se výsledek uloží do pole „stav“ v tabulce „gsm“. Modulu se po přijetí výsledku odešle povel „WEB\_NONE“, který jej uvede do stavu čekání na další události.

## 6. Závěr

V dnešní době prudkého rozvoje síťových komunikačních technologií a nárůstu objemu komunikovaných dat a nároku na kvalitu obsluhy budou tato zařízení využívána stále častěji. GSM „resetátory“ se budou svou jednoduchostí obsluhy a možnostmi rozšíření v tomto ohledu řadit mezi nejvyhledávanější. Pro menší aplikace může být nevýhodou jejich cena, jelikož GSM moduly nejsou levnou záležitostí, ale pro větší serverovny a spojovací místnosti tento nedostatek vynahradí rozsah jejich použitelnosti.

Navržený „resetátor“ je postaven na solidním základě výkonného GSM modulu od výrobce, jež se ve světě řadí mezi špičku ve svém oboru. Díky propracovanému vývojovému prostředí je možno zařízení přizpůsobit přímo na míru. Počet spínaných výstupů je dostatečný i pro větší sítě a jejich provedení umožňuje připojit různé spotřebiče napájené z navzájem nezávislých zdrojů. Ovládání je jednoduché, po prvotním nastavení již není třeba do „resetátoru“ nijak zasahovat, i při výpadku napětí si všechny hodnoty pamatuje. Díky tomu, že GSM modul obsahuje watchdog, který zabraňuje zaseknutí, nehrozí riziko, že by se „resetátor“ dostal do stavu, ve kterém by jej bylo třeba ručně resetovat.

Webové rozhraní dále zvyšuje komfort obsluhy celého zařízení a posouvá jej o třídu výš, než obvyklé zařízení této kategorie. Určitou nevýhodou je nutnost externího webového serveru, ale s rozvojem technologií, na kterých je postaven, lze do budoucna očekávat možnost umístění celého rozhraní do samotného zařízení.

Nicméně i toto řešení má své výhody. Tou hlavní je možnost přehledného a uživatelsky intuitivního ovládání bez omezení, která by umístěním webu do modulu jistě nastala. Z hlediska zpracování tohoto projektu jako diplomové práce lze taky jako výhodu poukázat na demonstrativní popis možností vzájemného propojení různých technologií.

Během testování na zkušebních přípravcích se navržený „resetátor“ choval dle očekávání, nedostatky v návrhu, jež se musely během testů opravovat, byly minimální. U testování uživatelské aplikace v modulu jsem také nenarazil na větší problémy, několik drobností bylo úspěšně doladěno a neobjevil se žádný zásadní nedostatek. Webový server byl odzkoušen na různých platformách a zobrazení stránek bylo korektní ve všech nejpoužívanějších prohlížečích. Testování software pro GSM modul proběhlo za podpory firmy AMiT, spol. s r.o.

Návrh vzdáleného spínače ovládaného přes SMS s GUI lze tedy považovat za úspěšný.

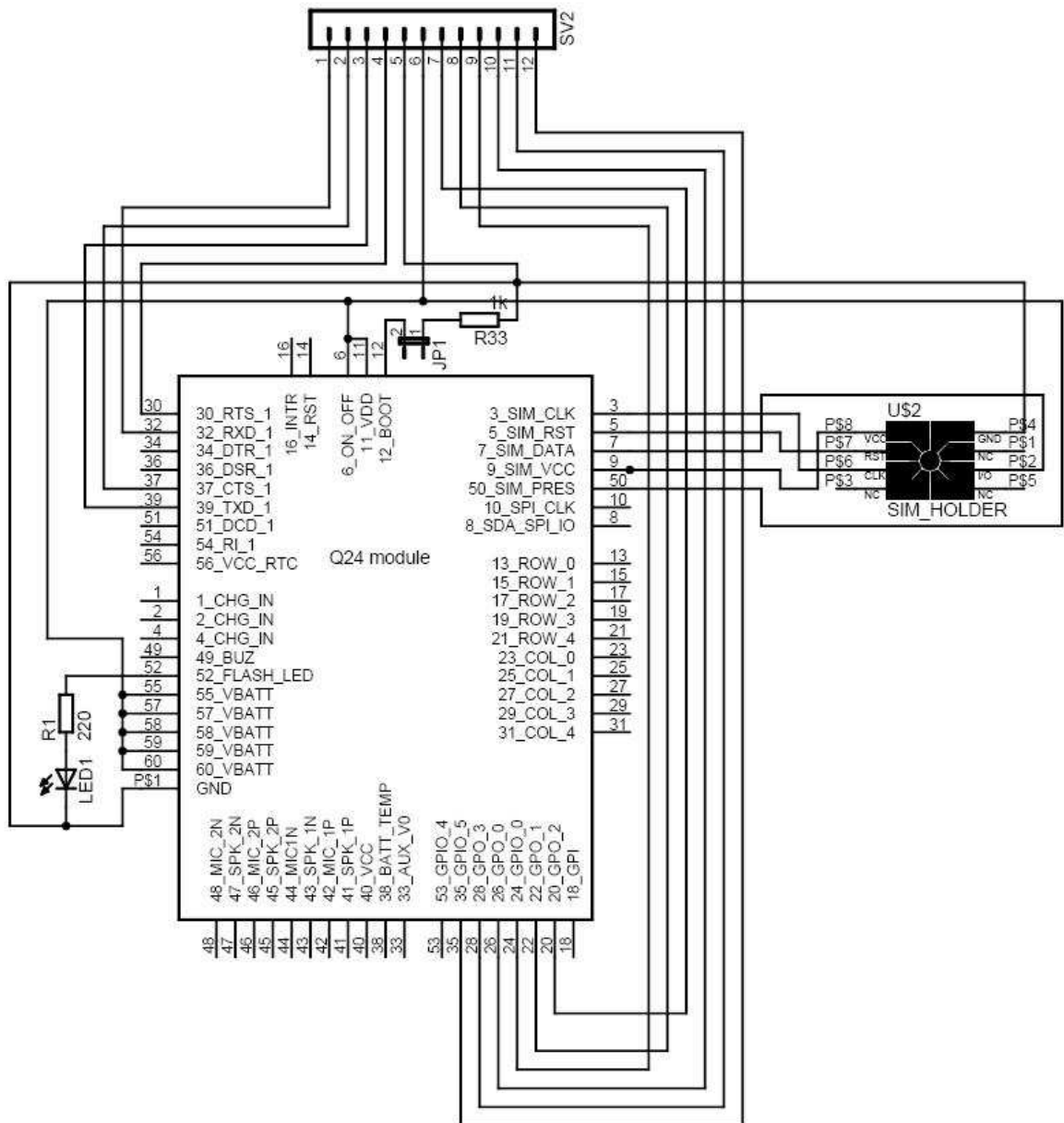
## 7. Použitá literatura

- [1] PRESSMAN, Abraham I. *Switching Power Supply Design*. 1st edition. [s.l.] : [s.n.], c2004
- [2] Wavecom<sup>®</sup> *Q2406 and Q2426 Product Specification*. Revision 003, 14.4.2003, [online]. Dostupné na WWW: <http://www.icenet.cz/supportwavecom.html>
- [3] Maxim<sup>®</sup> *+5V-Powered, Multichannel RS-232 Drivers/Receivers*. Revision 11, 2003, [online]. Dostupné na WWW: <http://pdfserv.maxim-ic.com/en/ds/MAX220-MAX249.pdf>
- [4] KREJČÍK Alexandr *Lineární napájecí zdroje - příklady výpočtů*. 1. vydání, BEN, 2002.
- [5] STMicroelectronics<sup>®</sup> *LF00 Series*. [online], 2003. Dostupné na WWW: [www.datasheetcatalog.com](http://www.datasheetcatalog.com)
- [6] GM Electronics *Katalog součástek*. 2007. Dostupné na WWW: <http://www.gme.cz/index.php?lk=&sk=&action=katalog>
- [7] Wavecom<sup>®</sup> *ADL User Guide for OpenAT v 3.10<sup>®</sup>*. OpenAT<sup>®</sup> Documentation, [online], Revision 007, 2005
- [8] Wavecom<sup>®</sup> *AT Commands Interface Guide for X55 release*. Revision 003, 2005, [online]. Dostupné na WWW: <http://www.icenet.cz/supportwavecom.html>
- [9] Modems.com *General Modem Operation*, 2003, [online]. Dostupné na WWW: <http://www.modem.com>
- [10] JEDLIČKA Petr *Přehled obvodů řady CMOS 4000*, 4. vydání, BEN, 2005
- [11] The PHP Group *PHPManual*, 2008, [online]. Dostupné na WWW: <http://www.php.net/manual/en/>
- [12] ZAJÍC Petr *MySQL- Pestrý svět databází*, 2005, [online]. Dostupné na WWW: [http://www.linuxsoft.cz/article.php?id\\_article=731](http://www.linuxsoft.cz/article.php?id_article=731)
- [13] BRÁZA Jiří *PHP4 učebnice základů jazyka*, 1. vydání, Grada Publishing, 2002
- [14] CHAMPEON Steve *JavaScript, How did we get there*, 2001, [online]. Dostupné na WWW: [http://www.oreillynet.com/pub/a/javascript/2001/04/06/js\\_history.html](http://www.oreillynet.com/pub/a/javascript/2001/04/06/js_history.html)
- [15] Apache Software Foundation *HTTP Server*, 2007, [online]. Dostupné na WWW: <http://httpd.apache.org/>
- [16] LACKO Luboslav *PHP5 a MySQL5 Hotová řešení*, 2007, 1. vydání, Computer Press a.s.
- [17] Wavecom<sup>®</sup> *OpenAT<sup>®</sup> IP Connectivity Development Guide*. Revision 004, 15.12.2006

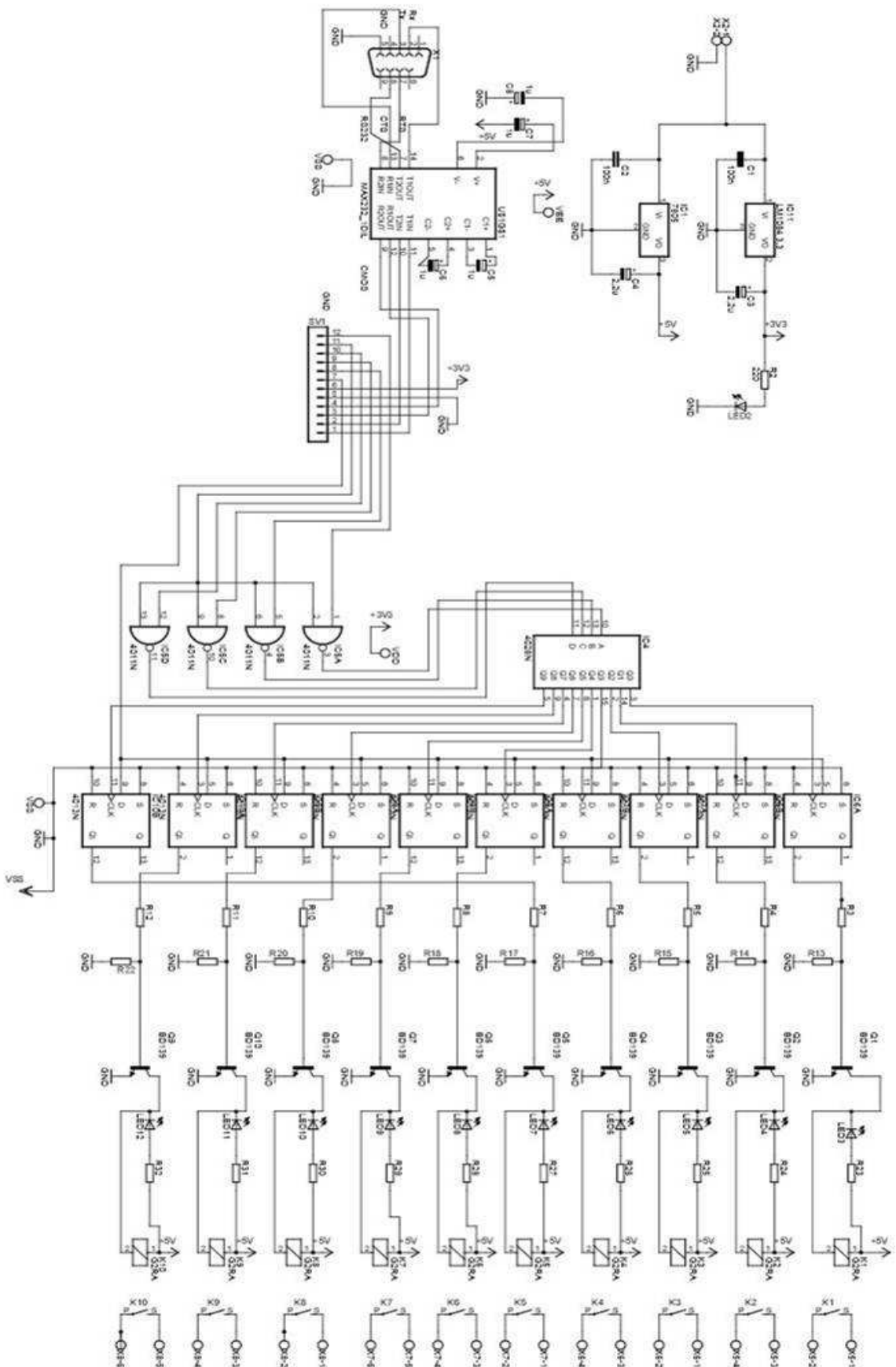
## 8. Seznam příloh

- A. Příloha 1 – Schéma zapojení GSM modulu
- B. Příloha 2 – Schéma zapojení pomocných obvodů
- C. Příloha 3 – Výsledky simulace chování výstupu
  - a. Vstupní úroveň 0V, tranzistor Q1 uzavřen, relé vypnuto, protéká jím pouze závěrný proud Q1.
  - b. Vstupní úroveň 3,3V, tranzistor Q1 otevřen, relé zapnuto, proud cca 100mA, proud LED cca 10mA.
- D. Příloha 4 – Návrh DPS pro pomocné obvody
  - a. Pohled shora
  - b. Pohled zespodu
- E. Příloha 5 – Návrh DPS pro GSM modul
  - a. Pohled shora
  - b. Pohled zespodu
- F. Příloha 6 – Rozložení součástek na DPS
  - a. DPS GSM modulu
  - b. Pohled zespodu
- G. Příloha 7 – Úvodní stránka webového rozhraní
- H. Příloha 6 – Nastavení spínače skrze webové rozhraní
- I. Příloha 6 – Ovládání spínače skrze webové rozhraní

## A. Příloha 1 – Schéma zapojení GSM modulu

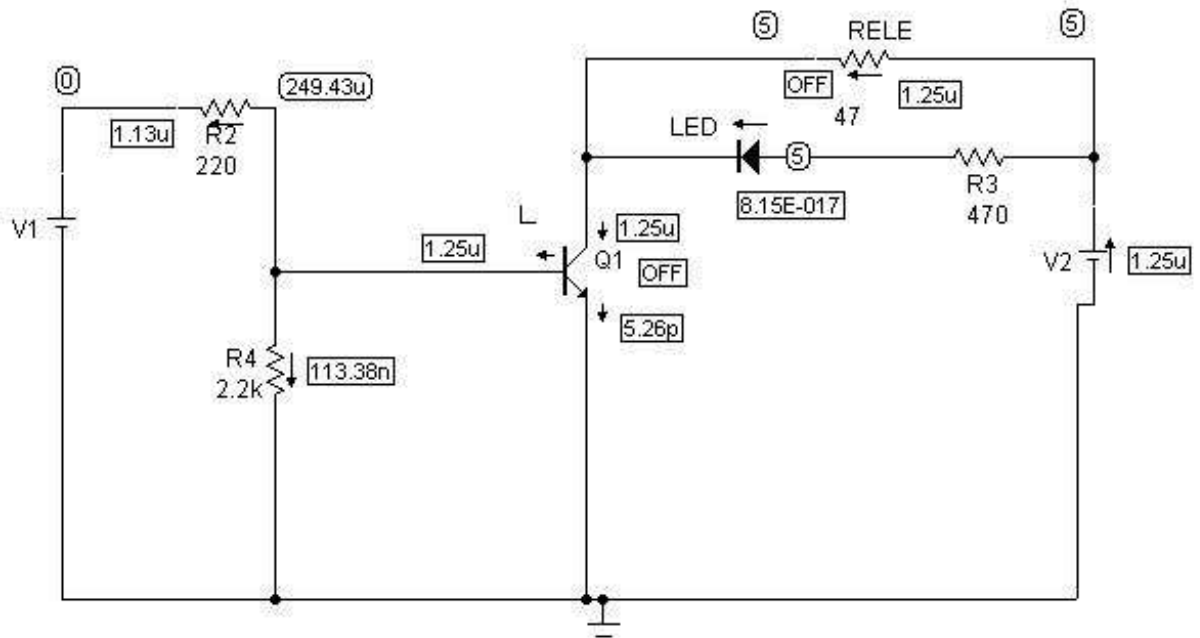


## B. Příloha 2 – Schéma zapojení pomocných obvodů

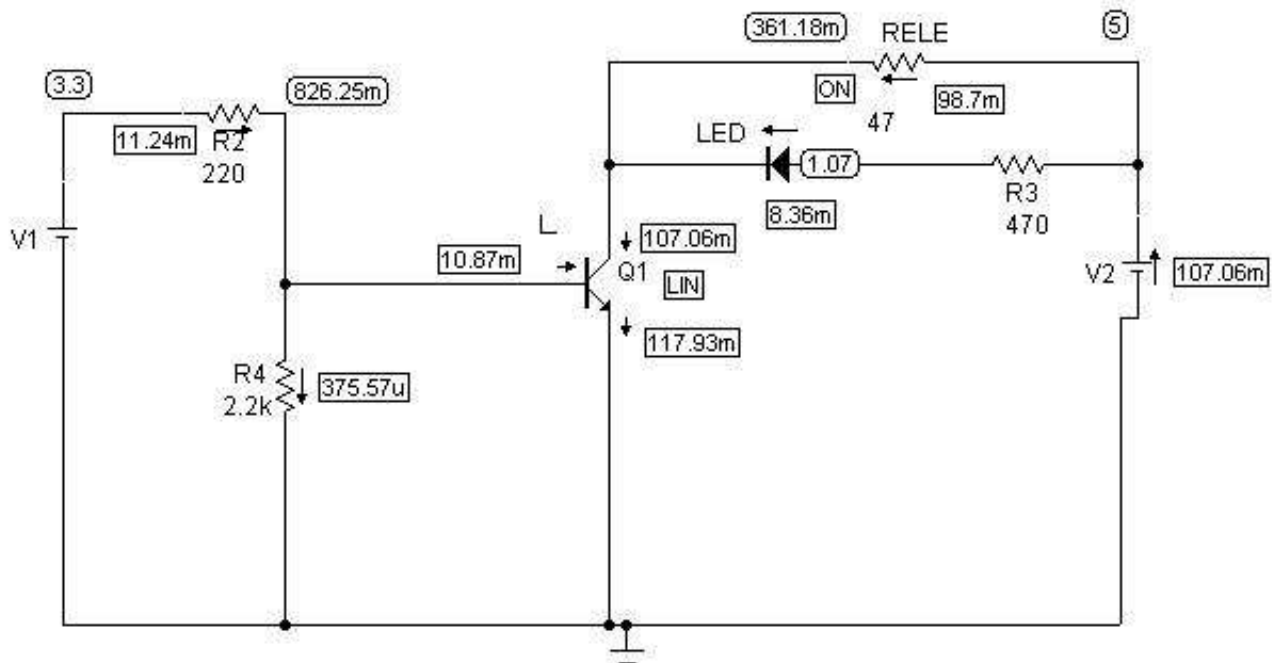


### C. Příloha 3 – Výsledky simulace chování výstupu

- a. Vstupní úroveň 0V, tranzistor Q1 uzavřen, relé vypnuto, protéká jim pouze závěrný proud Q1.

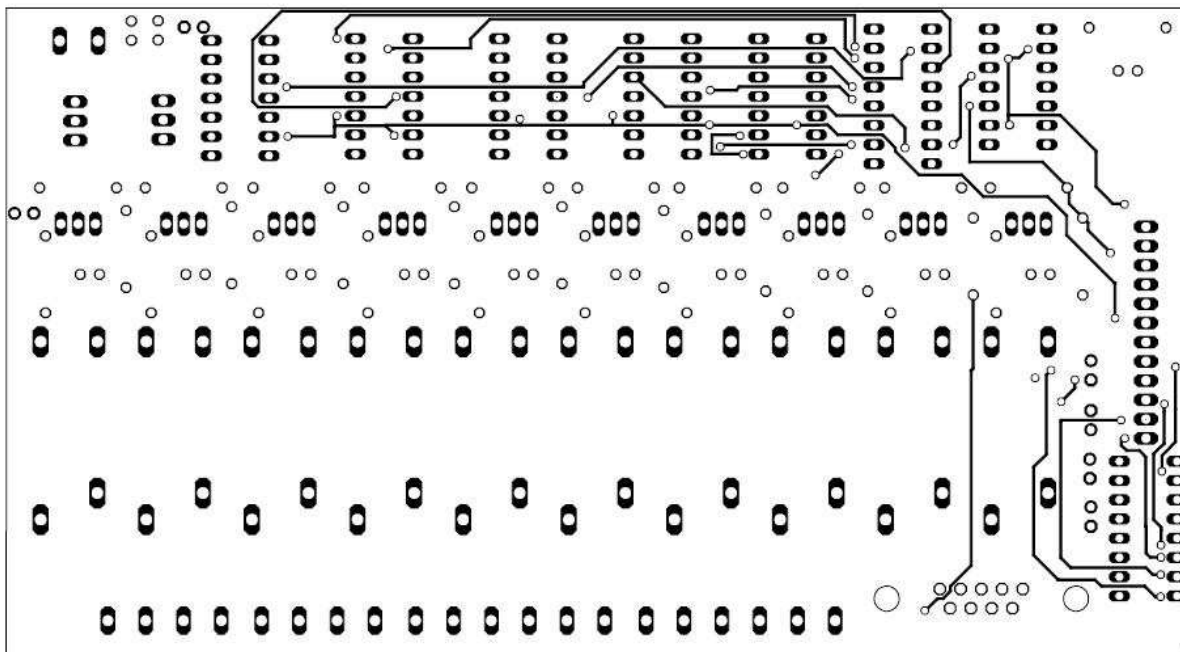


- b. Vstupní úroveň 3,3V, tranzistor Q1 otevřen, relé zapnuto, proud cca 100mA, proud LED cca 10mA.

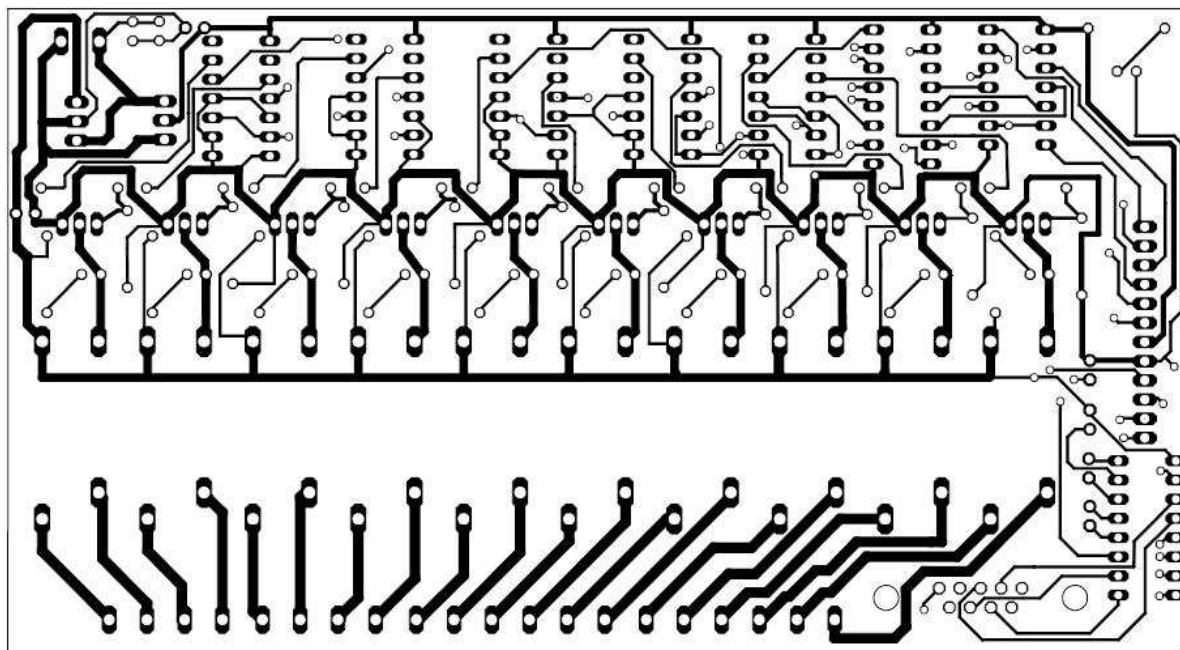


## D. Příloha 4 – Návrh DPS pro pomocné obvody

### a. Pohled shora

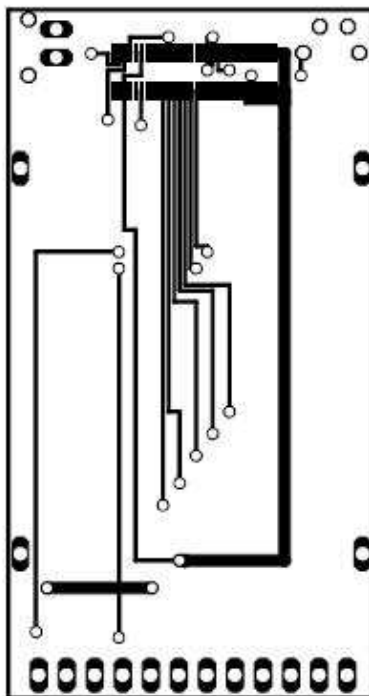


### b. Pohled zespodu

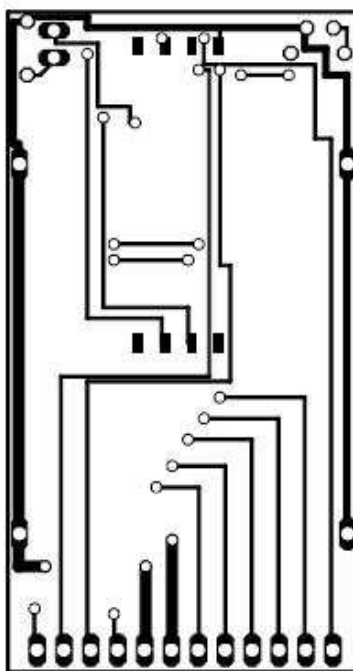


## E. Příloha 5 – Návrh DPS pro GSM modul

### a. Pohled shora

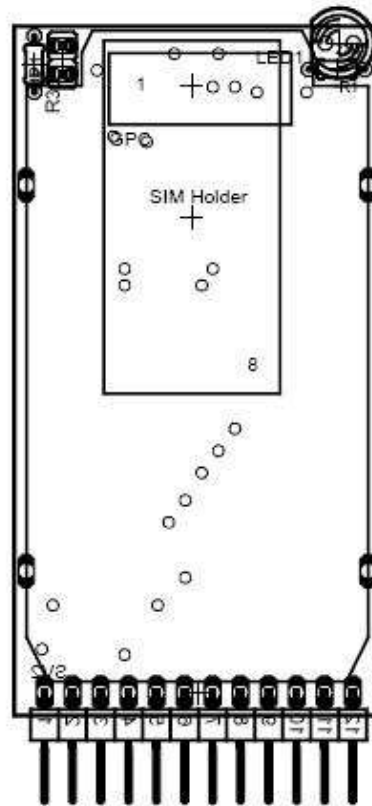


### b. Pohled zespodu

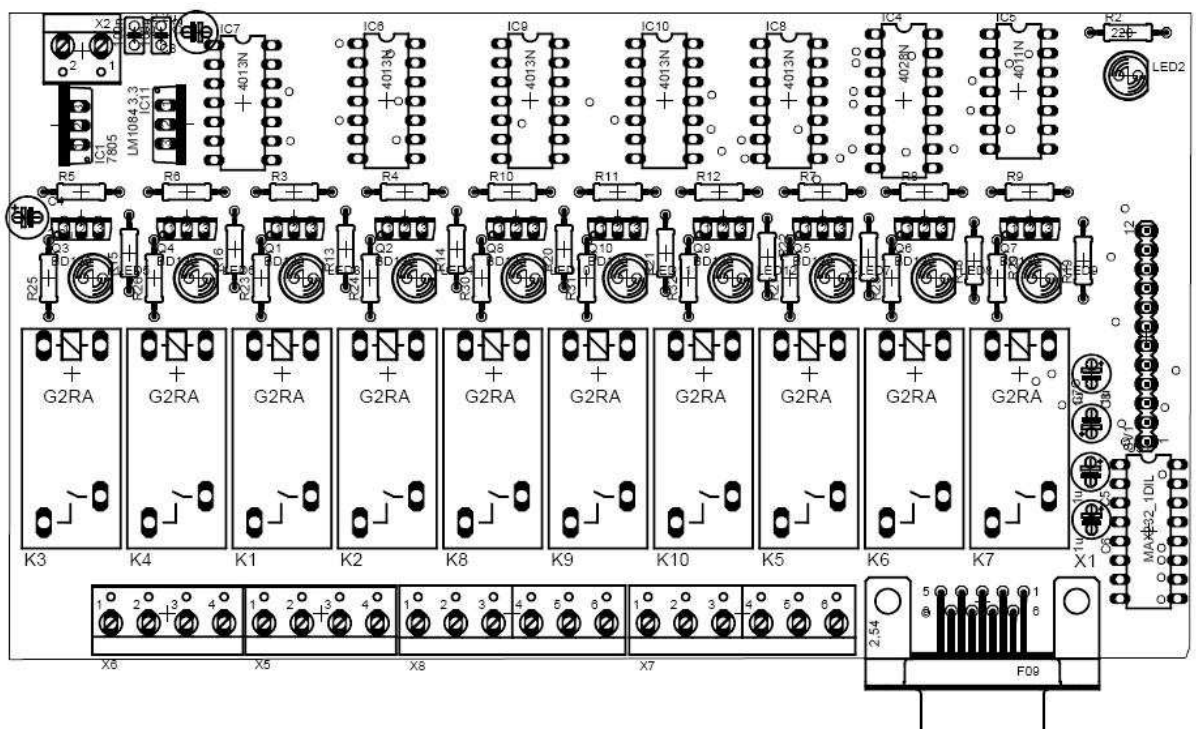


## F. Příloha 6 – Rozložení součástek na DPS

### a. DPS GSM modulu



### b. Pohled zespodu



## G.Příloha 7 – Úvodní stránka webového rozhraní

Ovládání vzdáleného GSM spínače

wavecom  
Smart wireless. Smart business.

Menu

Výběr vzdáleného GSM spínače

verze 0.05

Úvod

Ovládání Aktuální spínač: první ▾ Vybrat Obnovit

Nastavení

Administrace

Info

Logout

Název: první Stav:NC Odstranit z DB

IP adresa: 192.168.168.xx Tel. číslo: 777666555

Nastavení

Úvodní slovo v SMS: RES

PIN: \*\*\*\*\* Zobrazit PIN

Interval pro reset: 20

Povolené tel.čísla: 723334543, 234556754, 66666666,

Změň nastavení

## H. Příloha 6 – Nastavení spínače skrze webové rozhraní



### Ovládání vzdáleného GSM spínače



Menu

Aktuální spínač:

IP: 192.168.168.xx

Název:  Starý PIN:

Heslo:  Nový PIN:

Potvrď Heslo:  Potvrď PIN:

Úvodní slovo pro SMS:

Interval pro reset:

Povolené telefonní čísla:

723334543	<input type="text"/>
234556754	<input type="button" value="Přidej"/>
66666666	<input type="button" value="Smaž"/>

# I. Příloha 6 – Ovládání spínače skrze webové rozhraní

The screenshot shows a web interface for controlling a remote GSM switch. At the top, there is a blue header with a logo on the left, the title "Ovládání vzdáleného GSM spínače" in the center, and the Wavocom logo with the tagline "Smart wireless. Smart business." on the right. Below the header, there is a navigation menu on the left with items: Menu, Úvod, Ovládání, Nastavení, Administrace, Info, and Logout. The main content area features a dropdown menu for "Aktuální spínač:" with "první" selected, and two buttons: "Vybrat" and "Obnovit". Below this is a section titled "Stav výstupů" containing a list of ten outputs, each with a checkbox and a status label: 1: Vypnuto, 2: Vypnuto, 3: Zapnuto, 4: Zapnuto, 5: Zapnuto, 6: Vypnuto, 7: Zapnuto, 8: Vypnuto, 9: Vypnuto, and 10: Zapnuto. At the bottom of this section are three buttons: "Zapni", "Vypni", and "Reset".

Menu

Aktuální spínač: první ▼ Vybrat Obnovit

Úvod

Ovládání

Nastavení

Administrace

Info

Logout

Stav výstupů

- Vybrat vše
- 1: Vypnuto
- 2: Vypnuto
- 3: Zapnuto
- 4: Zapnuto
- 5: Zapnuto
- 6: Vypnuto
- 7: Zapnuto
- 8: Vypnuto
- 9: Vypnuto
- 10: Zapnuto

Zapni Vypni Reset