

Sem vložte zadání Vaší práce.



ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE  
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
KATEDRA POČÍTAČOVÉHO INŽENÝRSTVÍ



Bakalářská práce

## **System pro ovládání turniketů přes ethernet**

*Albert Orsák*

Vedoucí práce: Ing. Jaroslav Borecký

11. května 2015



---

## Poděkování

Rád bych poděkoval vedoucímu práce Ing. Jaroslavu Boreckému za jeho ochotu, přínosné rady, připomínky a odbornou pomoc při tvorbě této bakalářské práce. Také děkuji své rodině za jejich podporu po celou dobu mého studia.



---

## Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval(a) samostatně a že jsem uvedl(a) veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 46 odst. 6 tohoto zákona tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou, a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla, a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené. Každá osoba, která využije výše uvedenou licenci, se však zavazuje udělit ke každému dílu, které vznikne (byť jen zčásti) na základě Díla, úpravou Díla, spojením Díla s jiným dílem, zařazením Díla do díla souborného či zpracováním Díla (včetně překladu), licenci alespoň ve výše uvedeném rozsahu a zároveň zpřístupnit zdrojový kód takového díla alespoň srovnatelným způsobem a ve srovnatelném rozsahu, jako je zpřístupněn zdrojový kód Díla.

V Praze dne 11. května 2015

.....

České vysoké učení technické v Praze  
Fakulta informačních technologií

© 2015 Albert Orsák. Všechna práva vyhrazena.

*Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí, je nezbytný souhlas autora.*

### **Odkaz na tuto práci**

Orsák, Albert. *Systém pro ovládání turniketů přes ethernet*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2015.



---

# Abstrakt

Tato bakalářská práce se zabývá návrhem a vývojem řídicího systému pro turnikety. Vyvíjený přístupový systém je postavený na poznacích z komerčních přístupových systémů a také na poznacích z již realizovaných systémů. Úspěšně vyvinutý a v praxi ověřený přístupový systém obsahuje většinu funkcí komerčních systémů a dále množství dodatečných funkcí, ve kterých systém předčí své konkurenty. Dále jsou součástí systému vlastní řídicí jednotky pro řízení turniketů a serverová aplikace, která jednotky řídí. Díky vlastním řídicím jednotkám je systém snadno modifikovatelný a v důsledku toho levnější než běžná konkurence na trhu.

**Klíčová slova** Přístupový systém, řídicí jednotka turniketu, turniket, kontrola průchodů, propojení s pokladnou

---

# Abstract

This thesis describes the design and development of control system for turnstiles. Developed access system is based on findings from commercial access systems and also on findings from previously implemented systems. Successfully developed and proven in practice access system contains most of the functionality of commercial systems, as well as the amount of additional functions, in which the system surpasses its competitors. Furthermore, part of the system's own control unit for controlling the turnstiles and server application that manages units. Thanks to own control units, the system is easily modifiable and consequently cheaper than the normal competition on the market.

**Keywords** Access system, turnstile control unit, turnstile, check passes, connecting with POS

---

# Obsah

Úvod	1
<b>1 Analýza přístupových systémů</b>	<b>3</b>
1.1 Původní systém . . . . .	3
1.2 Přehled přístupových systémů na trhu . . . . .	4
1.3 Závěry z analýzy . . . . .	7
<b>2 Analýza obvodů pro PLCP jednotku</b>	<b>9</b>
2.1 Výběr procesoru . . . . .	9
2.2 Síťový čip . . . . .	13
2.3 RS232 . . . . .	14
2.4 Spínání turniketů . . . . .	15
2.5 Napájení . . . . .	17
<b>3 Analýza entrance serveru</b>	<b>19</b>
3.1 Programovací jazyk . . . . .	19
3.2 Databáze . . . . .	19
3.3 Komunikace . . . . .	19
3.4 Správa systému . . . . .	20
3.5 Licence . . . . .	20
<b>4 Návrh řešení PLCP jednotky</b>	<b>21</b>
4.1 Návrh řešení hardwaru . . . . .	21
4.2 Analýza vývoje plošného spoje . . . . .	22
4.3 Návrh řešení firmwaru . . . . .	23
<b>5 Návrh řešení entrance serveru</b>	<b>25</b>
5.1 Jednovláknový systém . . . . .	25
5.2 Více vláknový systém . . . . .	25

<b>6</b>	<b>Řešení HW PLCP jednotky</b>	<b>27</b>
6.1	Ochrana proti přepětí . . . . .	27
6.2	Napájecí modul 5V větev . . . . .	28
6.3	Napájecí modul 3.3V větev . . . . .	29
6.4	Napájení procesoru . . . . .	30
6.5	JTAG konektor . . . . .	31
6.6	Senzory pro protočení turniketu . . . . .	32
6.7	Relé . . . . .	33
6.8	Ethernet . . . . .	34
6.9	Serialový čip . . . . .	34
<b>7</b>	<b>Výroba plošného spoje PLCP jednotky</b>	<b>35</b>
<b>8</b>	<b>Řešení firmwaru PLCP jednotky</b>	<b>37</b>
8.1	Třída PLCP . . . . .	39
8.2	Třída IOWORKER . . . . .	39
8.3	Třída ip_arp_udp_tcp . . . . .	39
8.4	Třída enc28j60 . . . . .	40
<b>9</b>	<b>Řešení softwaru entrance serveru</b>	<b>41</b>
9.1	Funkce systému . . . . .	41
9.2	Model tříd . . . . .	44
9.3	clsEntranceServer . . . . .	44
9.4	clsTurniketManager . . . . .	44
9.5	clsTurniket . . . . .	44
9.6	TurniketComm . . . . .	45
9.7	clsLog . . . . .	45
9.8	clsCfg . . . . .	45
9.9	clsLanguage . . . . .	45
9.10	clsLogFile . . . . .	45
9.11	clsSettings . . . . .	46
9.12	clsTicketManager . . . . .	46
9.13	clsDatabase . . . . .	46
9.14	clsDatabaseModifier . . . . .	46
9.15	Structs Enums . . . . .	46
	<b>Závěr</b>	<b>47</b>
	<b>Literatura</b>	<b>49</b>
<b>A</b>	<b>Seznam použitých zkratk</b>	<b>53</b>
<b>B</b>	<b>Obsah příloženého CD</b>	<b>55</b>
B.1	Bakalářská práce . . . . .	55
B.2	Aplikace bakalářské práce . . . . .	56

B.3	Pomocné aplikace bakalářské práce . . . . .	57
<b>C</b>	<b>Instalační a konfigurační manuál</b>	<b>59</b>
C.1	Instalace . . . . .	59
C.2	Odinstalace . . . . .	59
C.3	Závislosti . . . . .	60
C.4	Komunikační porty . . . . .	60
C.5	Omezení . . . . .	60
C.6	Konfigurace . . . . .	60
C.7	Logy . . . . .	63
C.8	Licence . . . . .	64



---

## Seznam obrázků

1.1	Řídící jednotka přístupového systému Alveno . . . . .	5
1.2	Systém Acslime . . . . .	6
2.1	ATmega8 . . . . .	10
2.2	ATmega32 . . . . .	11
2.3	ATmega128 . . . . .	12
2.4	ENC28J60-I/SO . . . . .	13
2.5	MAX232D . . . . .	14
2.6	ST232CDR . . . . .	14
2.7	IRF540NPB . . . . .	15
2.8	AZ850-5 . . . . .	16
2.9	LM2576T-ADJG . . . . .	17
2.10	LM317T . . . . .	18
4.1	PLCP modelové řešení . . . . .	21
4.2	Doménový model firmwaru . . . . .	23
5.1	Doménový model softwaru . . . . .	26
6.1	Schéma ochrany proti přepětí . . . . .	27
6.2	Průběh napětí na LM2576T před stabilizací . . . . .	28
6.3	Schéma zapojení LM2576T . . . . .	28
6.4	Schéma zapojení LM317T . . . . .	29
6.5	Schéma zapojení napájení procesoru . . . . .	30
6.6	Schéma zapojení napájení procesoru2 . . . . .	30
6.7	JTAG konektor . . . . .	31
6.8	Schéma zapojení senzoru . . . . .	32
6.9	Schéma zapojení relé . . . . .	33
6.10	Schéma zapojení ethernet čipu . . . . .	34
6.11	Schéma zapojení ST232CDR . . . . .	34

7.1	PLCP jednotka . . . . .	35
8.1	Firmware PLCP jednotky . . . . .	38
9.1	Model softwaru entrance serveru . . . . .	44



---

## Seznam tabulek

2.1	ATmega8 parametry . . . . .	10
2.2	ATmega32 parametry . . . . .	11
2.3	ATmega128 parametry . . . . .	12
2.4	Napětové nároky . . . . .	17



---

# Úvod

Úkolem bakalářské práce je navrhnout a postavit řídicí systém turniketových zařízení (systém slouží k ovládnání turniketů a jiných zařízení kontrolující pohyb osob po objektu).

Systém se bude skládat z ovládacích hardwarových jednotek (dále PLCP), které prostřednictvím navrženého řídicího softwaru (dále Entrance Server) budou ovládat jednotlivé turnikety.

Nově navržený přístupový systém vychází z poznatků původního systému, který používal pro ovládnání turniketů mikropočítače s MS-DOS a neobsahoval běžné funkce přístupových systémů.

Ovládací jednotka PLCP turniketu je spojená s řídicím systémem Entrance Server. Komunikace PLCP s Entrance Serverem probíhá přes Ethernet.

Entrance Server umožňuje přidávání, mazání a editaci záznamů přes ethernetové rozhraní, dále Entrance Server obsahuje funkce pro kontrolu platnosti záznamů, vč. doby propadnutí identifikátoru (platí pro tištěné vstupenky s čárový kódem, bezkontaktní karty a podobně).

Každá PLCP jednotka umožňuje připojení až dvou čteček, tzn. umožňuje ovládat dva turnikety. Jednotka také snímá impulzy ze senzoru, který je součástí turniketu, a snímá realizovaný průchod nikoliv pouze přiložení identifikátoru.



---

# Analýza přístupových systémů

## 1.1 Původní systém

Původní systém navržený pro první použití v praxi se skládal z pokladního systému třetí strany a z hardwarových jednotek využívající mikropočítač s MS-DOS, které se již nevyrábí. Pokladní systém ovládal přímo řídicí jednotky.

### Nedostatky původního systému a navrhované řešení:

- Vysoká cena jednotky. Je potřeba vyvinout řídicí jednotku, která poběží na jednoduchém micročipovém procesoru. Díky mikroprocesoru bude možné vyvinout levnou řídicí jednotku.
- Pokladní systém musel implementovat modul pro komunikaci s turnikety. Mezi pokladním systémem a samotnými řídicími jednotkami je umístěn software, který přijímá data od pokladního systému a dále je interpretuje řídicím jednotkám. Tento software také implementuje funkce usnadňující práci pokladnímu systému.
- Čidla otáčení turniketu. Při používání systému se občas stávalo, že uživatelé systému přiložili identifikátor ke čtečce, ale neprošli. Což vedlo k odečtení vstupu. Tomuto problému zabránilo čidlo otočení turniketu.
- Funkce anti-passback. Je potřeba implementovat funkci anti-passback, aby se zabránilo průchodu více osob na jeden identifikátor.
- Absence funkce, která by umožnila přesměrování rozhodnutí o platnosti lístku na jiný systém. Tato funkce by sloužila pro placené služby, protože řídicí jednotka může obsluhovat např. pračku a systém nemůže rozhodovat o placených platbách.  
Uživatel systému přiloží identifikátor ke čtečce, rozhodnutí se přeměruje a systém, který obstarává elektronickou peněženku rozhodne zda

má uživatel dostatečný kredit. Pokud ano, předá přístupovému systému informaci, aby zapnul pračku.

- Chybějící funkce pro nastavení platnosti po průchodu turniketem. Tohle je typická funkce pro bazény, uživatel systému si koupí lístek, který může celý den vstoupit do bazénu. Avšak po průchodu turniketem lístek platí pouze dvě hodiny.

### 1.2 Přehled přístupových systémů na trhu

Na našem trhu je velké množství přístupových systémů, ale ve většině se jedná o autonomní systémy, které neumožňují vzdálenou správu. Další nevýhoda komerčních přístupových systémů je, že ve většině případů pracují pouze se specifickými čtečkami, které dodává výrobce přístupového systému. Další velká nevýhoda komerčních přístupových systémů spočívá v omezeném počtu identifikátorů v přístupovém systému, vyšší počet záznamů se musí dokupovat. Obvyklý maximální počet záznamů se pohybuje od 25 do 100. Firmy, které prodávají přístupové systémy si také účtují velké sumy za přidání nových funkcí specifických pro potřeby zadavatelů.

Jednotky pro ovládání turniketů se na českém trhu liší převážně typem a počtem čteček. Nejpoužívanější protokoly pro komunikace se čtečkami jsou RS232, Ethernet a WIEGAND.

#### 1.2.1 Systém Alveno

Firma Alveno má velké množství různých řešení. Pro analýzu jsem zvolil řešení s kontrolérem ACi 1 [1], protože je mému řešení nejbližší.

*"ACi je moderní přístupový kontrolér pro rozsáhlejší a bezpečnější instalace s ověřováním pomocí otisků prstů a čipů. Pro komunikaci se zařízením je možné použít dnes již standardní rozhraní TCP/IP. Tento centrálně řízený typ přístupového systému ACi je cenově výhodnější pro střední a rozsáhlejší instalace. K přístupovému kontroléru je možné připojit až 8 zařízení pro otevírání dveří, dle toho, kterou variantu kontroléru si zvolíte. K dispozici jsou 3 varianty pro 1/2/4 dveře. Tyto varianty přístupového kontroléru ACi lze kombinovat i s našim docházkovým systémem Alveno, můžete tak využít kompletní řešení pro společnosti jakékoli velikosti."* [2]



Obrázek 1.1: Řídící jednotka přístupového systému Alveno [2]

### 1.2.1.1 Přístupové systémy Alveno[1] umožňují:

- Nastavení časové platnosti identifikátoru.
- Dále podporují funkci anti-passback, která má za úkol zabránit průchodu více lidmi na jeden identifikátor. Funkce pracuje tím způsobem, že přes jeden konkrétní turniket je daný identifikátor platný pouze jednou, aby mohl návštěvník projít znovu musí projít jiným turniketem.
- Také jsou chráněny proti přepětí, tzn. zařízení je odolné proti chybnému zapojení nebo zapojení na příliš vysoké napětí.
- Podporují funkci interlock. Tato funkce umožňuje nastavení například dvou dveří, tak aby najednou mohli být otevřeny pouze jedny dveře.
- Další součástí funkcí řídicí jednotky je nastavení minimálního intervalu mezi jednotlivými vstupy.

### 1.2.1.2 Shrnutí systému Alveno

Nevýhoda přístupového systému Alveno spočívá ve složitější správě identifikátorů, které se nastavují fyzicky u řídicí jednotky a také neumožňují nastavit počet vstupů. Některé z těchto funkcí umožňuje dodatečný software, který má mimo jiné i funkce docházkového systému. Ovšem cena nejlevnějšího řešení přesahuje 17.000,- Kč.[1]

### 1.2.2 Systém Acslime

Acslime je další velká firma, která nabízí velké množství různých řešení přístupových systémů. Z možných řešení přístupového systému jsem zvolil jednotku AS3003 a čtečky EDK2B, protože byly funkčně velmi podobné přístupovému systému, který vyvíjím. Také se jedná o jedno z nejlevnějších řešení firmy Acslime. [3, 4]



(a) Řídicí jednotka Acslime [4]



(b) Čtečka čipů Acslime [4]

Obrázek 1.2: Systém Acslime

#### 1.2.2.1 Přístupové systémy Acslime[3, 4] umožňují

- Každému identifikačnímu médiu v systému lze nastavit libovolná časová oprávnění pro vstup do vybraných prostor.
- V systému jsou evidovány všechny pohyby osob včetně pokusů o neoprávněné vstupy.
- Pomocí grafické nadstavby lze ON-LINE monitorovat a ovládat prostory v reálném čase.
- Napojení na systémy EZS, EPS, MaR, CCTV.
- Další součástí funkcí řídicí jednotky je nastavení minimálního intervalu mezi jednotlivými vstupy.

#### 1.2.2.2 Shrnutí systému Acslime

Přístupový systém Acslime umožňuje pohodlnější správu identifikátorů díky řídicímu softwaru. Dále podporuje většinu obvyklých funkcí přístupových systémů. Bohužel vyžaduje ve většině dedikované čtečky, které dodává výrobce systému. Další nevýhody systému jsou stále omezený počet vstupů, u většiny řešení přístupových systémů - včetně řešení, které jsem vybral, je omezeno na 100 identifikátorů. Další nevýhoda je, že v systému nelze nastavit počet průchodů identifikátoru.[3, 4]



## 1.3 Závěry z analýzy

Data pro závěr vychází z [1, 3, 4].

Žádný z komerčních systémů v cenové relaci do 20.000,- Kč nespĺňuje požadavky uživatelů původního systému. Ve většině se jedná o omezený počet identifikátorů, také nemožnost propojení přístupového systému s pokladním systémem a nutnost použití dedikovaných čteček.

Nový systém zahrnující řídicí jednotky a centrální systém běžící na PC umožní dodatečné funkce, které jiné systémy postrádají. Dále jednotka bude umožňovat připojení běžných komerčních čteček přes protokol RS232, který podporuje většina čteček.

Řídicí jednotka bude navrhnutá, tak aby umožňovala spínat/napájet i elektricky náročnější zařízení. Zároveň jednotka bude obsahovat potřebné funkce pro kontrolu průchodu turniketem.

Systém běžící na PC bude umožňovat nastavení platnosti lístku (od – do), platnosti vstupu v jednotlivých dnech v týdnu. Dále funkci, která umožňuje přestavit čas platnosti při průchodu turniketem. Což znamená, že můžeme prodat lístek, který je platný celý rok, ale při průchodu se doba platnosti přepíše například na hodinu. Tuhle funkci konkurenční systémy postrádají. Dále funkci anti-passback, možnost přesměrování rozhodnutí vstupu na externí pokladní systém a pod.



---

# Analýza obvodů pro PLCP jednotku

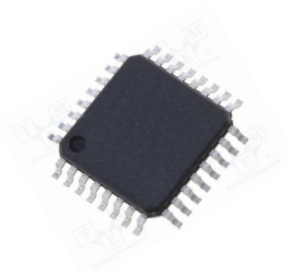
Řídící jednotka přístupového systému bude komunikovat se serverem po protokolu Ethernet. Dále jednotka umožňuje připojení dvou čteček po protokolu RS232, snímání senzorů otočení turniketu a také spínání turniketů a jiných zařízení.

Řídící jednotka se tedy skládá se 4 částí. Procesorové části a s částí realizující komunikaci po Ethernetu. Další dvě části zajišťují spínání turniketů a přijímání dat z senzorů otočení turniketu.

## 2.1 Výběr procesoru

Na trhu je velké množství vhodných procesorových jednotek. Mikrokontrolery od firmy Pic, Atmel dále také procesory technologie ARM. Jako vhodnou procesorovou jednotku jsem vybral mikrokontroler od firmy Atmel a to řadu ATmega. Mikrokontrolery ATmega zajišťují za nízkou cenou dostatečný výkon a také jednoduché ovládání. ARMové procesory sice disponují vysokým výkonem, ale jejich ovládání je znatelně složitější a na aplikaci tohoto typu není jejich výkon potřeba.

### 2.1.1 ATmega8



Obrázek 2.1: ATmega8 [5]

Procesor ATmega8 je 8-bitový RISCový procesor od firmy Atmel, který ovládá 130, ve většině Single-clock Execution instrukcí. O výkonu 16 MIPS při frekvenci 16MHz. Cenově je dostatečný pro aplikaci.[6]

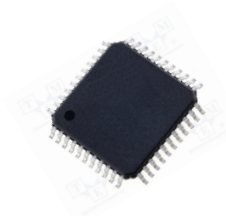
MIPS čili Million Instruction Per Second (česky milion instrukcí za sekundu) je jednotka výkonnosti počítače, která udává počet zpracovaných instrukcí za sekundu.

Tabulka 2.1: ATmega8 [6]

Výrobce	ATMEL
Typ integrovaného obvodu	mikrokontrolér AVR
Organizace paměti Flash	8k x 8bit
Kapacita paměti EEPROM	512B
Kapacita paměti SRAM	1024B
Pouzdro	TQFP32
Kmitočet taktování	16MHz
Periférie	RTC, komparátor
Počet vstupů/výstupů	23
Počet kanálů PWM	3
Počet čítačů 8-bit	2
Počet čítačů 16-bit	1
Montáž	SMD
Pracovní napětí	2.7...5.5V
Počet převodníků A/D 10-bit	8
Rozhraní	1*SPI,1*TWI,1*USART

Procesor je sice výkonově pro aplikaci vhodný, ale obsahuje malou Flash paměť, vyvíjený program by mohl být příliš velký. Také obsahuje pouze jedno USART rozhraní. Jelikož z analýzy vyplynulo, že PLCP jednotka musí umožňovat připojení 2 čteček, je tento procesor nevhodný i z tohoto důvodu.[6]

### 2.1.2 ATmega32



Obrázek 2.2: ATmega32 [7]

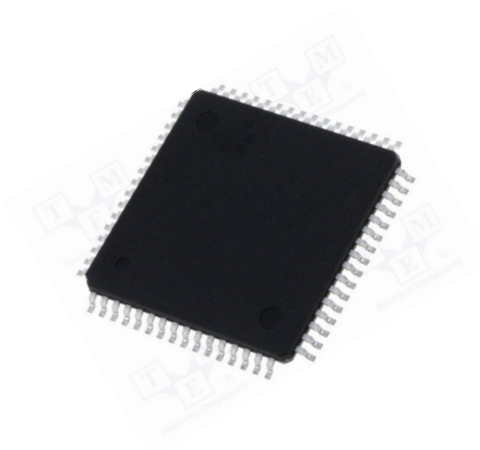
Procesor ATmega32 je 8-bitový RISCový procesor od firmy Atmel, který ovládá 131 instrukcí. O výkonu 16 MIPS při frekvenci 16MHz. Cenově i výkonově je dostatečný pro aplikaci.[8]

Tabulka 2.2: ATmega32 [8]

Výrobce	ATMEL
Typ integrovaného obvodu	mikrokontrolér AVR
Organizace paměti Flash	128k x 8bit
Kapacita paměti EEPROM	4096B
Kapacita paměti SRAM	4096B
Pouzdro	TQFP64
Kmitočet taktování	16MHz
Periférie	RTC, komparátor
Počet vstupů/výstupů	53
Počet kanálů PWM	8
Počet čítačů 8-bit	2
Počet čítačů 16-bit	2
Montáž	SMD
Pracovní napětí	2.7...5.5V
Počet převodníků A/D 10-bit	8
Rozhraní	1*SPI,1*TWI,1*USART

Procesor ve většině parametrů dosahuje potřebných hodnot pro aplikaci, ale má pouze jedno rozhraní USART. Jelikož z analýzy vyplynulo, že PLCP jednotka musí umožňovat připojení 2 čteček, je tento procesor také nevhodný. [8]

### 2.1.3 ATmega128



Obrázek 2.3: ATmega128 [9]

Procesor ATmega128 je 8-bitový RISCový procesor od firmy Atmel, který ovládá 133 instrukcí. O výkonu 16 MIPS při frekvenci 16MHz. Cenově i výkonově je dostatečný pro aplikaci. [10]

Tabulka 2.3: ATmega128 [10]

Výrobce	ATMEL
Typ integrovaného obvodu	mikrokontrolér AVR
Organizace paměti Flash	32k x 8bit
Kapacita paměti EEPROM	1024B
Kapacita paměti SRAM	2048B
Pouzdro	TQFP44
Kmitočet taktování	16MHz
Periférie	RTC, komparátor
Počet vstupů/výstupů	32
Počet kanálů PWM	4
Počet čítačů 8-bit	2
Počet čítačů 16-bit	1
Montáž	SMD
Pracovní napětí	2.7...5.5V
Počet převodníků A/D 10-bit	8
Rozhraní	1*SPI,1*TWI,2*USART

Procesor je vhodný pro PLCP jednotku. Především díky nízké ceně, jednoduchému ovládání a dostatečném výkonu a množství periférií. [10]

## 2.2 Síťový čip

Na trhu je velké množství různých síťových čipů. Definovaná kritéria pro síťový čip byla: potřeba podpory rozhraní SPI a typ ethernetu 10Base-T.

SPI (Serial Peripheral Interface) je sériové periferní rozhraní. Používá se pro komunikaci mezi řídicími mikroprocesory a ostatními integrovanými obvody.

Těmto kritériím vyhovuje pouze ENC28J60. SMD montáži vyhovují pouzdra SO28 a TSSOP28. Tyto pouzdra jsou funkčně totožná, ale TSSOP28 je zmenšená verze SO28 s roztečí 0.65mm. Z důvodu snadnější montáže jsem zvolil pouzdro SO28 s roztečí 2.56mm.

SMD (surface mount technology) je postup, kdy se elektronické součástky pájením osazují přímo na povrch plošného spoje.

### 2.2.1 ENC28J60-I/SO



Obrázek 2.4: ENC28J60-I/SO [11]

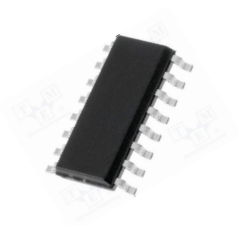
Síťový čip ENC28J60-I/SO podporuje, SPI rozhraní, přes které procesor přímá a vysílá pakety. Dále obsahuje buffery pro přenosy o velikosti 8192 B. Čip pracuje na MAC/PHY vrstvě s roztečí 2.56mm. [12]

### 2.3 RS232

Na trhu je opravdu velké množství RS232 linkových přijmačů/vysílačů se 2 transivery. Hlavním kritériem výběru je cena.

Standard RS232, resp. jeho poslední varianta RS-232C z roku 1969, (také sériový port nebo sériová linka) se používá jako komunikační rozhraní osobních počítačů a další elektroniky.

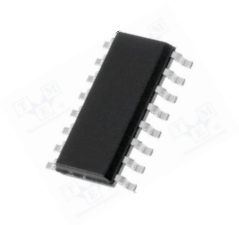
#### 2.3.1 MAX232D



Obrázek 2.5: MAX232D [13]

MAX232D je linkový vysílač/přijímač se dvěma vysílači/přijmači. S rychlostí 120kbps a rozhraním RS232. Tento čip jsem nevybral, protože existují levnější varianty. [13]

#### 2.3.2 ST232CDR



Obrázek 2.6: ST232CDR [14]

ST232CDR je linkový vysílač/přijímač se dvěma transivery. S rychlostí 220kbps a rozhraním RS232. Tento čip jsem vybral, protože je výkonnější a levnější než MAX232D. ST232CDR také vyžaduje menší kondenzátory pro výrobu 12V na sériovém rozhraní. [14]



## 2.4 Spínání turniketů

Turniket vyžaduje k otevření impuls na různých úrovních, např. 24V. Jednotka musí dále umožňovat spínání zámků dveří, které mohou vyžadovat až 0.5A při 24V.

### 2.4.1 IRF540NPBF

Jako zástupce skupiny hexfetů jsem vybral tranzistor N-Mosfet IRF540NPBF o vysokém výkonu 140W a velmi nízké ceně. [15]



Obrázek 2.7: IRF540NPBF [15]

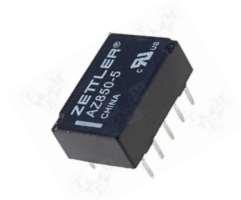
#### **Vlastnosti[15]:**

- Polarizace unipolární
- Napětí drain-source 100V.
- Proud drainu 33A

Výhoda hexfetů spočívá ve schopnosti spínat velké proudy, ačkoliv jsou pomalejší než jiné typy tranzistorů, ale to není v téhle aplikaci důležité.

Velká nevýhoda hexfetů spočívá v tom, že výstup je galvanicky propojený s procesorem. Při zkratu by se mohl zničit procesor. Z tohoto důvodu jsem zvolil relé.[15]

### 2.4.2 AZ850-5



Obrázek 2.8: AZ850-5 [16]

Vybral jsem relé AZ850-5 od firmy Zettler, které umožňuje spínat 30V při 1A, ale je schopné ve špičkách spínat až 220V. Díky tomu má relé vysokou izolační odolnost. Relé jsem také vybral z důvodu, že umožňuje bezpotenciálové spínání a galvanicky odděluje procesor od spínaného objektu. [16]

#### **Vlastnosti[16]:**

- Jmenovité napětí cívky 5V DC
- Zatížitelnost kontaktů AC (při odporové zátěži) 0.5 A / 125 VAC
- Zatížitelnost kontaktů DC (při odporové zátěži) 1 A / 30 VDC
- Spínané napětí max 250V AC, max 220V DC
- Odpor cívky 178  $\Omega$
- Doba sepnutí 2ms
- Doba rozepnutí 1ms
- Příkon přes cívku 142mW
- Mechanická trvanlivost 100000000 cyklů

## 2.5 Napájení

Vybrané čipy potřebují dvě různé napájecí úrovně. A to 3.3V a 5V. Síťový čip běží na 3.3V, procesor je od 2.7V do 5.5V a relé je spínané 5V. Také ST232 běží na 5V.

Tabulka 2.4: Napěťové nároky

ATMega128	max. 40mA
Enc28j60	max. 200mA
ST232CDR	max. 10mA
2*AZ-850-5	max. 60mA
Napájení čidla turniketu	max. 500mA
Celkem 3.3V větev	max. 240mA
Celkem 5V větev	max. 570mA

Vstupy PLCP jednotky, by měli být tolerantní do 36V. Lineární stabilizátor, který by měl stabilizovat z 36V na 5V potažmo na 3.3V by při odběru hodně přehříval.

Z toho důvodu jsem vybral impulzní stabilizátor, který bude stabilizovat na 5V a z těchto 5V bude druhý stabilizátor, který bude tentokrát lineární, stabilizovat na 3.3V. Touto kombinací dosáhneme ideálního napájení bez přehřívání obou stabilizátorů.

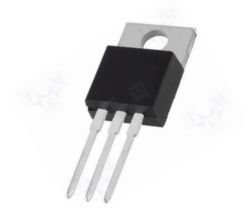
### 2.5.1 LM2576T-ADJG



Obrázek 2.9: LM2576T-ADJG [17]

Tento impulzní stabilizátor jsem vybral, protože splňuje napěťové nároky na 5V větev a je velmi levný. Stabilizátor je nastavitelný 1.3...37V s výstupním proudem až 3A. Stabilizátor má dostatečnou rezervu výkonu pro nepřetržitý provoz. [17]

### 2.5.2 LM317



Obrázek 2.10: LM317T [18]

Tento lineární stabilizátor jsem vybral, protože za nízkou cenu umožňuje stabilizovat napětí z 5V větve na 3.3V s výstupním proudem 1.5A. Stabilizátor má dostatečnou rezervu výkonu pro nepřetržitý přenos. [18]

---

# Analýza entrance serveru

## 3.1 Programovací jazyk

Pro vývoj serverové aplikace by byl vhodný programovací jazyk C++ [19] nebo C# [20].

Vývoj v jazyce C++ by byl složitější a časově náročnější. Jazyk C# umožňuje jednoduchý vývoj a zároveň C# je součástí .NET [21], který obsahuje širokou škálu podpory pro připojení k databázi a k síti.

Vývoj v C# je možný pouze ve vývojovém studiu Visual Studio [22]. Což je aplikace pro vývoj aplikací v různých programovacích jazycích pro platformu windows.

## 3.2 Databáze

Jako vhodná databáze se mimo jiné nabízí Microsoft SQL database [23] nebo Oracle SQL Developer [24]. Vybral jsem Microsoft SQL database, protože .NET pro ni obsahuje jednoduchý interface.

## 3.3 Komunikace

Komunikace po ethernetu bude probíhat na protokolu UDP, protože protokol TCP má vysoké odezvy. Od přiložení identifikátoru ke čtečce a otevřením turniketu musí latence být nižší než 100ms, při delší prodlevě musí uživatel systému čekat na otevření turniketu. Tento požadavek TCP protokol nesplňuje, kvůli kontrolním mechanismům. UDP protokol sice nemá žádné kontrolní mechanismy, ale systém bude nasazen tak, aby jednotlivé hardwarové prvky byly v oddělených lokálních kolizních doménách, takže nehrozí kolize či prohození paketů od 1 turniketu. V případě velmi nepravděpodobné ztráty paketu uživatel přiloží čip ke čtečce znovu. Aby nedošlo k odečtení vstupu při neúspěšné průchodu turniketem je v systému zabudována funkce kontroly protočení tur-

niketu. Tato funkce zajistí ochranu mimo jiné i proti případným výpadkům sítě. Dokud nebude proveden průchod turniketem nebude odečten vstup.

## 3.4 Správa systému

Pro konfiguraci systému jsou použity textové soubory, v praxi jsem na několika aplikacích ověřil, že administrátoři preferují snadnější konfiguraci textových souborů. Konfigurace přes registry bývá často nepřehledná a pro větší přehlednost jsou potřeba externí nástroje.

Logy o chodu systému jsou zaznamenávány jak v textové podobě, tak v databázi. V textové podobě jsou zaznamenávány logy o spuštění systému a popřípadě o jeho kritických chybách. Samotné logy o průchodech, neúspěšných průchodech a podobně jsou zaznamenány v databázi.

Logy podporují více jazyků, které jsou konfigurovatelné. Většina textů lze upravit v konfiguračních souborech podle přání uživatele systému.

Dále systém obsahuje mechanismus na správu starých záznamů v databázi. Konfigurovatelné, periodicky spouštěné SQL příkazy, které čistí databázi.

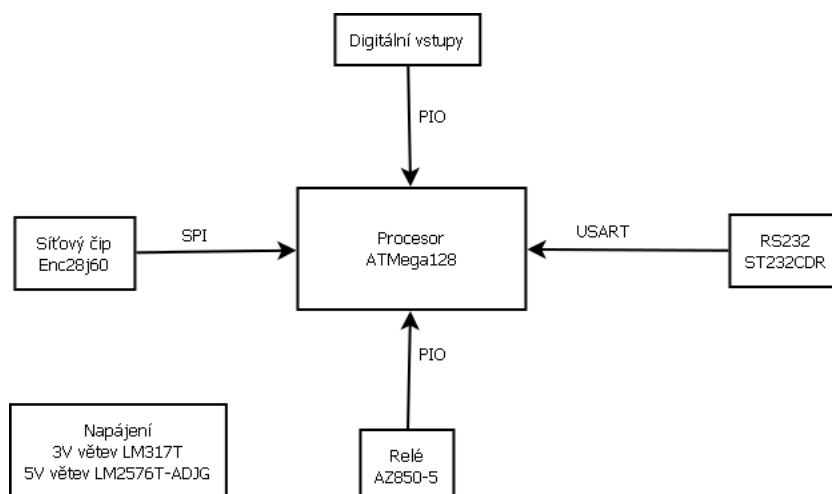
## 3.5 Licence

SW také obsahuje část pro kontrolu licencí. Při spuštění si systém ověří licenci vůči HW, na kterém běží. V případě neplatnosti licence se systém nespustí.

## Návrh řešení PLCP jednotky

### 4.1 Návrh řešení hardwaru

Na základě analýzy hardwarových prvků jednotky jsem zvolil řešení:



Obrázek 4.1: PLCP modelové řešení

- Procesor komunikuje se síťovým čipem přes SPI. Síťový čip dále vysílá do ethernet konektoru.
- Procesor přes digitální výstupy ovládá relé.
- ST232CDR posílá data procesoru ze čteček.
- Procesor přímá data s digitálních vstupů.
- PLCP jednotka je napájena LM2576T-ADJG pro 5V větev.
- PLCP jednotka je napájena LM317T pro 3.3V větev.

### 4.1.1 Napájení

Napájení zařízení je tolerantní do 36V. Zároveň zařízení musí být zabezpečené proti přehození napájecích vodičů a také musí být chráněno proti přepětí a napěťovým špičkám.

Ochranu proti prohození vodičů bude zajišťovat schottkyho dioda, která bude při opačné polaritě napětí v uzavřeném stavu.

Ochranu proti přepětí nad 36V bude zajišťovat kombinace polymerové pojistky a transilu, tato kombinace bude rozpojovat napájecí vodiče při přesažení 36V nebo 2A.

Dále ochranu před napěťovými špičkami přes 200V zajišťuje varistor, který je bude uzemňovat.

### 4.1.2 Digitální vstupy

Jednotka přes dva digitální vstupy umožňuje přijímat data ze senzoru turniketu. Na vstupech bude čidlo turniketu vyrábět impulzy při otočení turniketu.

Vstupy budou připojeny přímo do procesoru, a tak budou chráněny zenerovou diodou, která zvládá až 36V, toto napětí bude stabilizovat na 4.7V. Dále pro dodatečnou ochranu, a také pro uzemnění větve, bude použit pull-down o velikosti 1k $\Omega$ .

## 4.2 Analýza vývoje plošného spoje

Plošný spoj řídicí jednotky PLCP jsem navrhnul v programu DesignSpark PCB 7.0 [25]. Program jsem zvolil, protože je zdarma a poskytuje pohodlné prostředí pro návrh plošných spojů. Umožňuje také generovat soubory pro výrobu plošného spoje na výrobní lince.

Tento plošný spoj jsem vyrobil sám. Použil jsem metodu osvětlování UV světlem cuprexitových desek s pozitivní fotoemulzí přes poloprůhledný papír na kterém je zrcadlově vytištěný samotný plošný spoj a dále vyleptáním přebytečné mědi roztokem kyseliny solné a peroxidu vodíku.

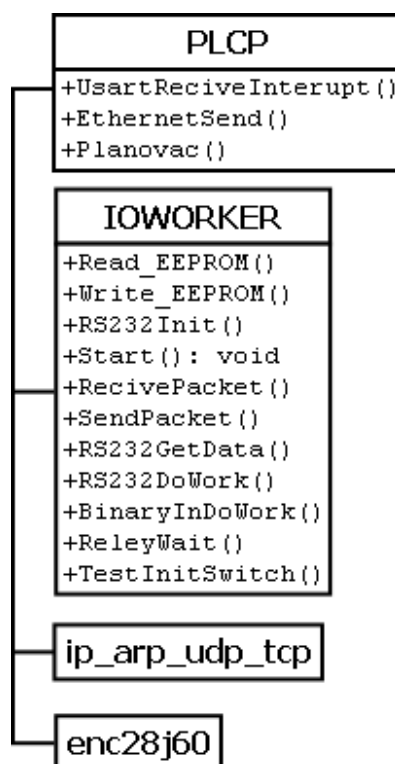
Firmware pro procesor ATmega128 budu vyvíjet v Atmel Studio 6.2 [26]. Toto vývojové studio jsem zvolil, protože je přímo od výrobce procesoru, z toho důvodu je vhodné pro vývoj.



### 4.3 Návrh řešení firmwaru

Plánovač úloh bude přerozdělovat procesor jednotlivým úlohám. A to při inicializaci zařízení, kde se budou načítat nastavení z eeprom paměti. Dále při příjmu paketů bude udělovat čas pro zpracování paketu a také pro práci ze senzory a práci z relé a čtečkami.

Pro práci se síťovým čipem a implementaci UDP protokolu využijí knihovny Procyon AVRlib [27].



Obrázek 4.2: Doménový model firmwaru

Zásadní při práci s periferiemi je plánovač úloh. Existují dva typy plánovačů úloh a to preemptivní a kooperativní.

#### 4.3.1 Preemptivní plánovač úloh

V preemptivním multitaskingu o přidělování a odebírání času jednotlivým úlohám plně rozhoduje samotný plánovač. Úlohy dostanou přidělený čas pro práci s procesorem a po vypršení dojde k uložení práce procesu a předání času jinému procesu.

Preemptivní plánovač úloh vyžaduje složitější implementaci a pro tuto aplikaci ho není zapotřebí.

##### **4.3.2 Kooperativní plánovač úloh**

V kooperativním multitaskingu plánovač pouze rozhoduje o tom, které úloha se spustí. Dokud se úloha neukončí plánovač nepřidělí čas dále.

Kooperativní plánovač je vhodný pro tuto aplikaci. Ale při psaní úloh je třeba věnovat důraz na krátkou dobu běhu.

---

## Návrh řešení entrance serveru

Entrance server podporuje všechny funkce, které vyplynuly z analýzy.

Aplikace běží jako služba, a to z důvodu, že bude instalována převážně na servery. Entrance server se tedy spouští ještě před přihlášením uživatele do systému, což umožňuje pohodlný provoz nezkušeným zákazníkům, kterým stačí pouze zapnout počítač a systém poběží.

Hlavní problémem je, že systém musí pracovat najednou s více turnikety najednou.

Tento problém lze vyřešit dvěma způsoby, jedno vláknovým řešením a více vláknovým řešením.

### 5.1 Jednovláknový systém

Aplikace bude obsahovat pro každou jednotku třídu, do které se bude zaznamenávat komunikace, stavy a hodnoty nutné pro práci jednotky.

Dále bude aplikace obsahovat kontrolní třídu, která bude číst třídy turniketů a bude provádět veškerou práci se všemi turnikety.

Tímto řešením se vyhneme více vláknovým problémům. Avšak toto řešení bude méně přehledné a povede k stavu, kdy bude jedna velká třída řešit všechny problémy.

### 5.2 Více vláknový systém

Aplikace bude obsahovat pro každou jednotku třídu, která bude řešit veškerou práci pouze s přidělenou jednotkou.

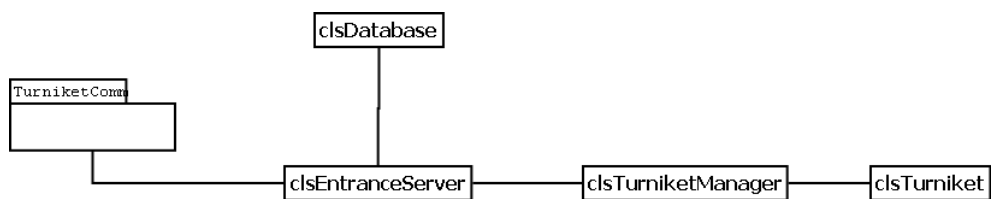
Dále bude obsahovat kontrolní třídu, která bude pouze přerozdělovat práci jednotlivým třídám.

## 5. NÁVRH ŘEŠENÍ ENTRANCE SERVERU

---

Tímto řešením vzniknou obvyklé problémy s řízením přístupu více vláken k datům a periferiím, které bude třeba řešit, ale díky navrženému řešení aplikace bude přehlednější a bude se také snadněji upravovat. Což je také jeden z požadavků na systém.

### Více vláknový model aplikace:



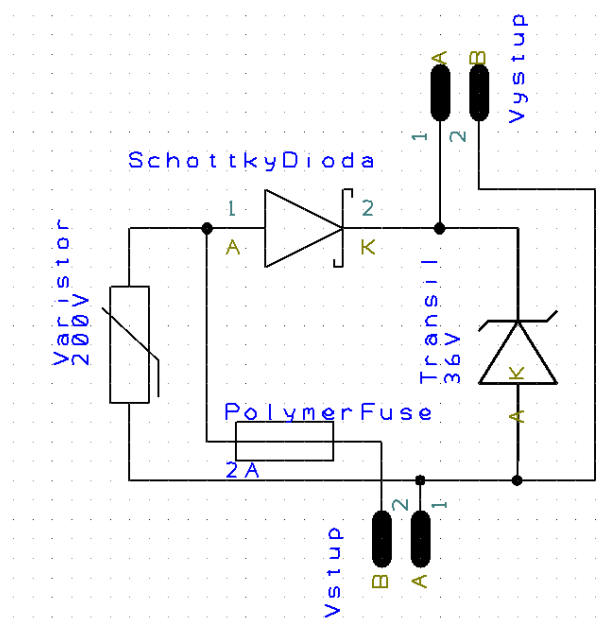
Obrázek 5.1: Doménový model softwaru

## Řešení HW PLCP jednotky

### 6.1 Ochrana proti přepětí

Pro ochranu proti přepětí jsem zvolil kombinaci polymerové pojistky a transilu. Polymerová pojistka obvod rozpojí při odběru větším než 2A a transil zkratuje napájecí větev při napětí větším než 36V, což vede k aktivaci pojistky. Dále je v obvodu zapojený varistor, který poskytuje ochranu před napěťovými špičkami.

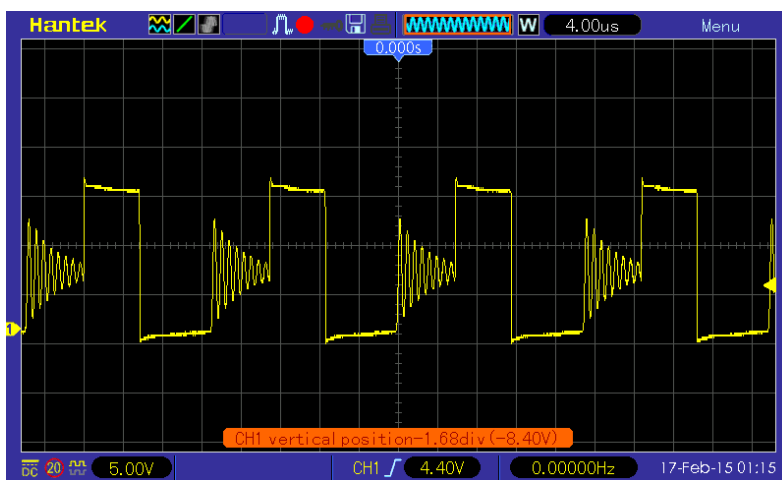
Dotatečnou ochranu proti obrácení napájecích vodičů poskytuje shottkyho dioda.



Obrázek 6.1: Schéma ochrany proti přepětí

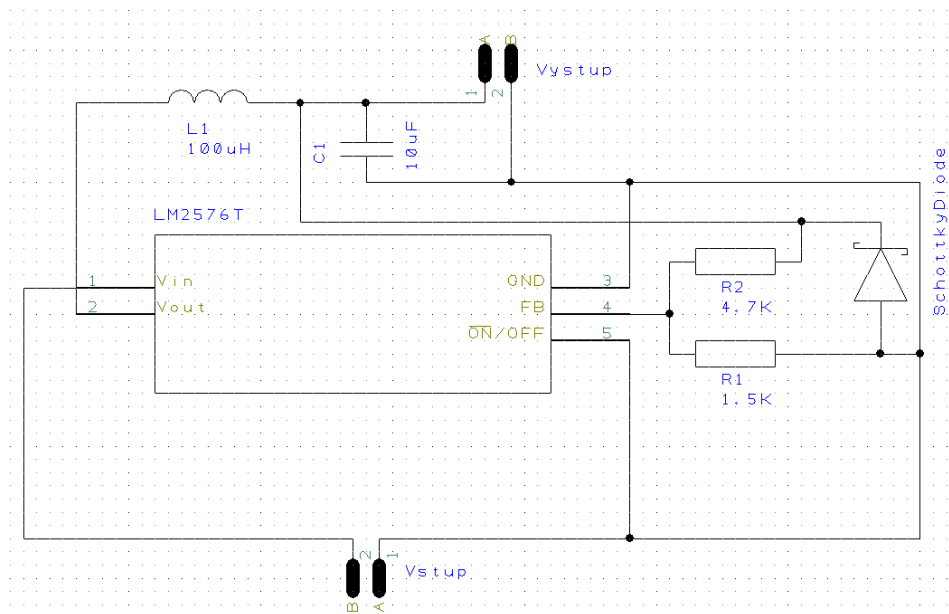
## 6.2 Napájecí modul 5V větve

Pro napájení 5V větve slouží nastavitelný impulzní stabilizátor LM2576T-ADJG. Z důvodu nestability napětí je u stabilizátoru cívka a kondenzátor pro stabilizaci napětí.



Obrázek 6.2: Průběh napětí na LM2576T před stabilizací

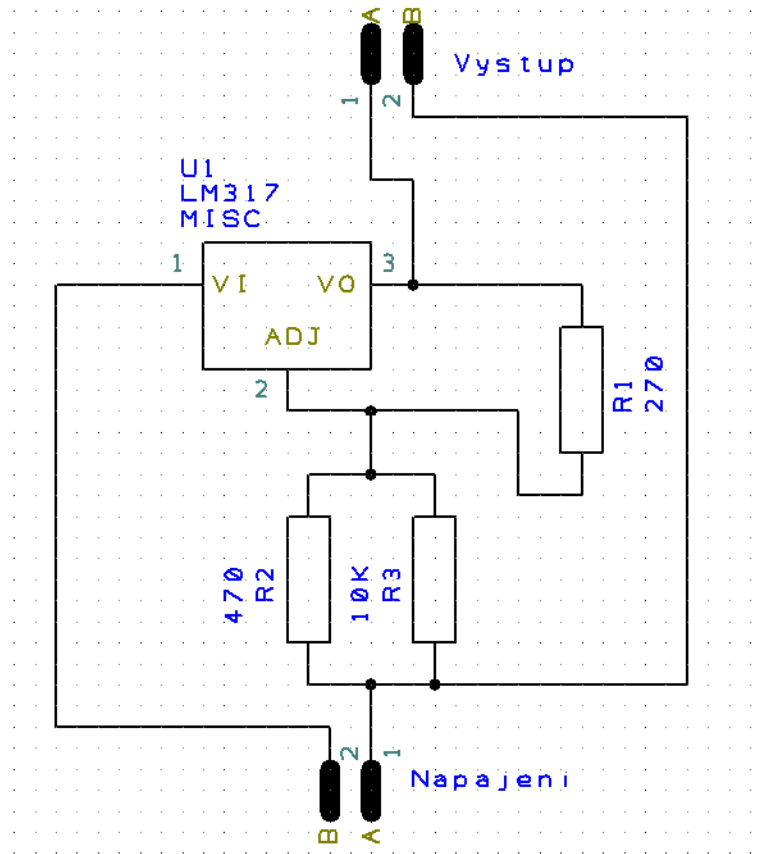
Po stabilizování větve cívkou a kondenzátorem je vše v pořádku. Stabilizátor je chráněn zenerovou diodou proti napěťovým špičkám vznikajícím na cívce při klesání napětí.



Obrázek 6.3: Schéma zapojení LM2576T

### 6.3 Napájecí modul 3.3V větev

Pro napájení 3.3V větve slouží nastavitelný lineární stabilizátor LM317T.



Obrázek 6.4: Schéma zapojení LM317T

## 6.4 Napájení procesoru

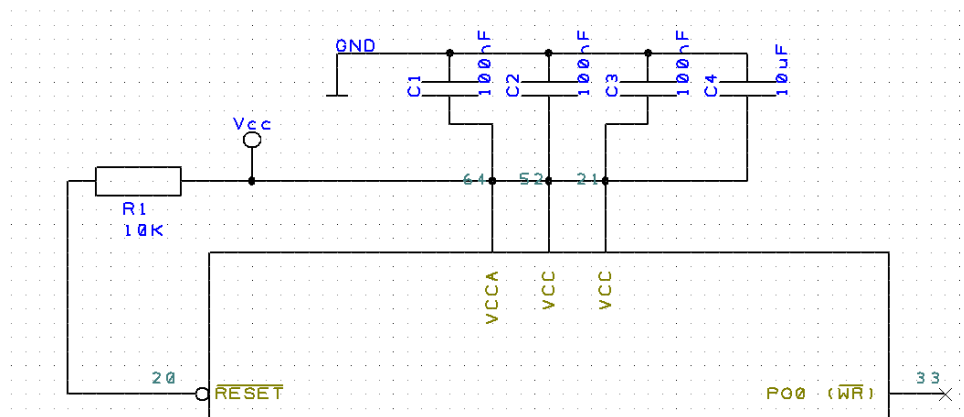
### 6.4.1 Napájení

Procesor je napájený z 3.3V větve. Na každém napájecím vývodu je 100nF keramický kondenzátor. Keramické kondenzátory se používají z důvodu nízkého ESR, čili dokáží rychle reagovat na změnu napětí.

ESR (equivalent series resistance), což je zdánlivý odpor kondenzátoru.

K napájení je připojený 10 $\mu$ F pro hrubou stabilizaci napětí, postačuje elektrolitický kondenzátor, bez ohledu na ESR.

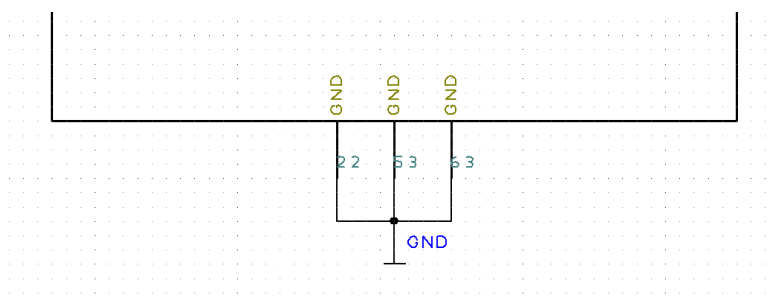
Na vývodu procesoru pro reset je umístěn 10k $\Omega$  pull-up. Pokud by se vývod dostal na logickou úroveň 0, procesor by se restartoval.



Obrázek 6.5: Schéma zapojení napájení procesoru

### 6.4.2 Zem

Vývody procesoru pro zem jsou připojeny na zem bez úprav.

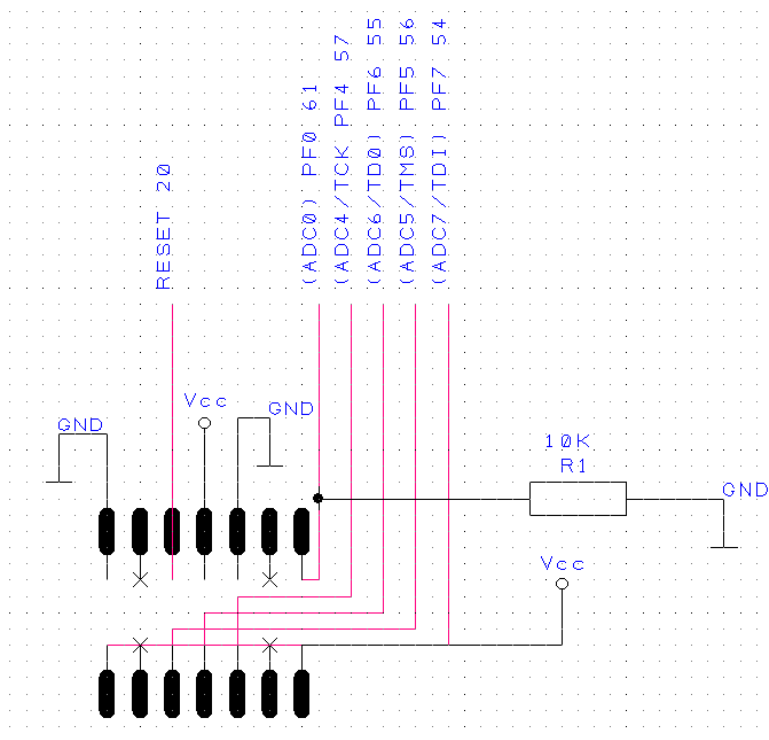


Obrázek 6.6: Schéma zapojení napájení procesoru



## 6.5 JTAG konektor

Konektor je pro programování přes JTAG. Součástí konektoru je i pin pro resetování IP adresy PLCP jednotky.



Obrázek 6.7: JTAG konektor

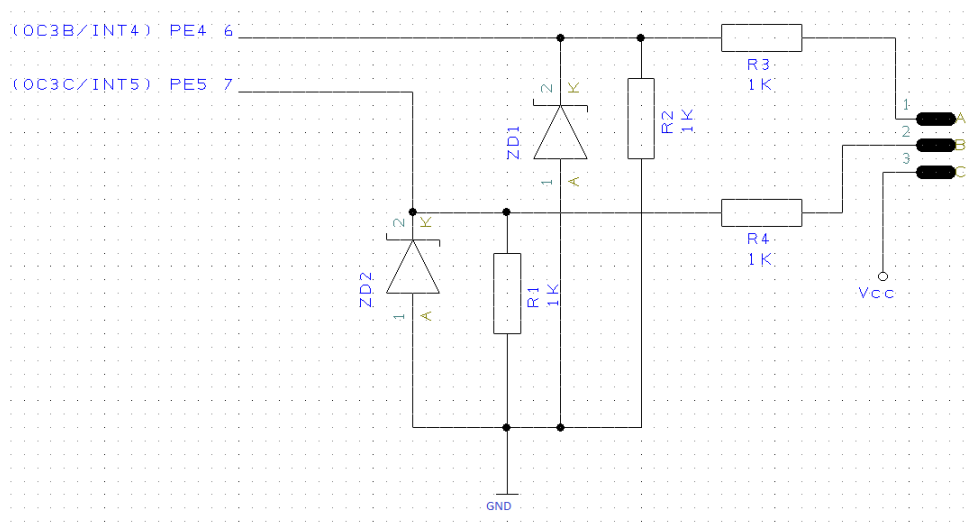
Pin konektoru, který je připojený na port F0 slouží k indikaci resetu nastavení PLCP jednotky. Na větvi připojené do procesoru je  $10K\Omega$  pull-down.

Pokud se propojí s protilehlým pinem konektoru bude na větvi logická 1. Při startu jednotky se pin testuje a pokud je na úrovni logické 1 resetuje se IP a MAC adresa.

## 6.6 Senzory pro protočení turniketu

Konektor pro senzory je přímo připojen k procesoru, k ochraně slouží zenerovi diody, které stabilizují napětí na 4,7V a zároveň je na každém kanálu pull-down odpor o velikosti 1k $\Omega$ , který poskytuje dodatečnou ochranu.

Současně je na konektoru pro senzory napájení pro senzor, které bude senzor posílat nazpět při své aktivaci.

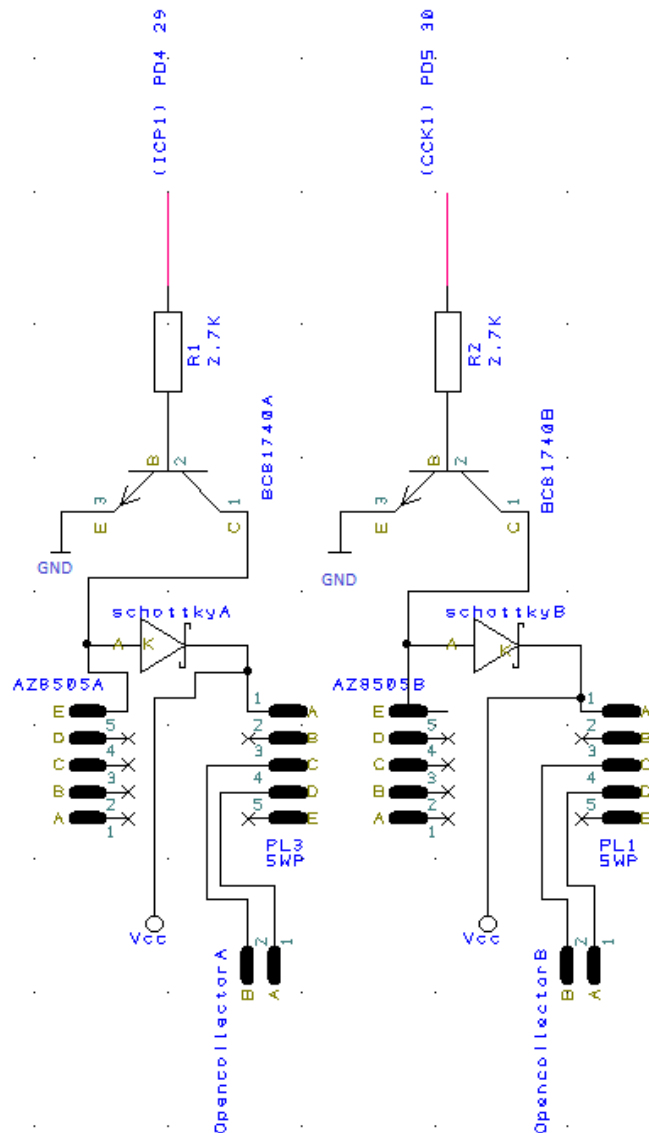


Obrázek 6.8: Schéma zapojení senzoru

## 6.7 Relé

Relé je napájené 5V. Bipolární NPN tranzistor, 5V větev sepne proti zemi pro rozpojení relé. Relé je zapojené pro bezpotenciálové spínání, při sepnutí spojí oba vodiče.

Proud tekoucí do báze bipolárního NPN tranzistoru je omezen odporem. Samotný tranzistor je chráněn schottkyho diodou proti napěťové špičce, která vzniká na cívce relé při náhlém odpojení napájení relé.

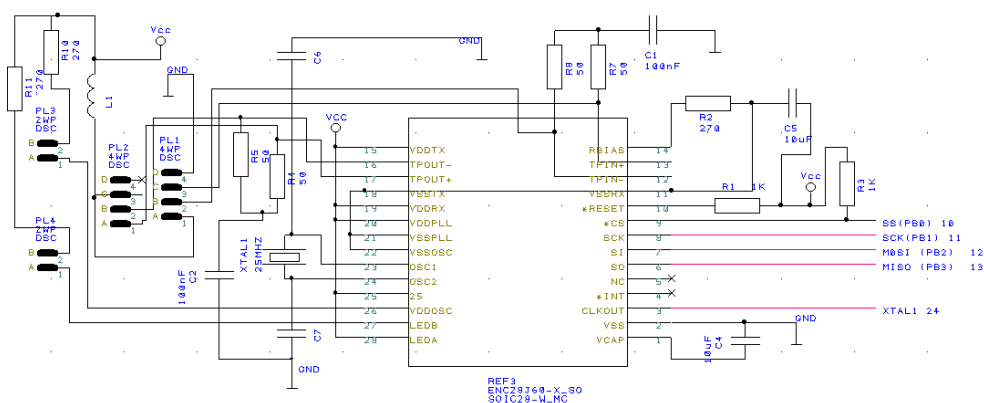


Obrázek 6.9: Schéma zapojení relé

## 6.8 Ethernet

Ethernet čip je řízen 25MHz krystalem a dále vyrábí 6MHz hodiny pro procesor. Procesor na základě těchto hodin přijímá data z SPI rozhraní ethernetového čipu. Ethernet čip je napájený z 3.3V větve.

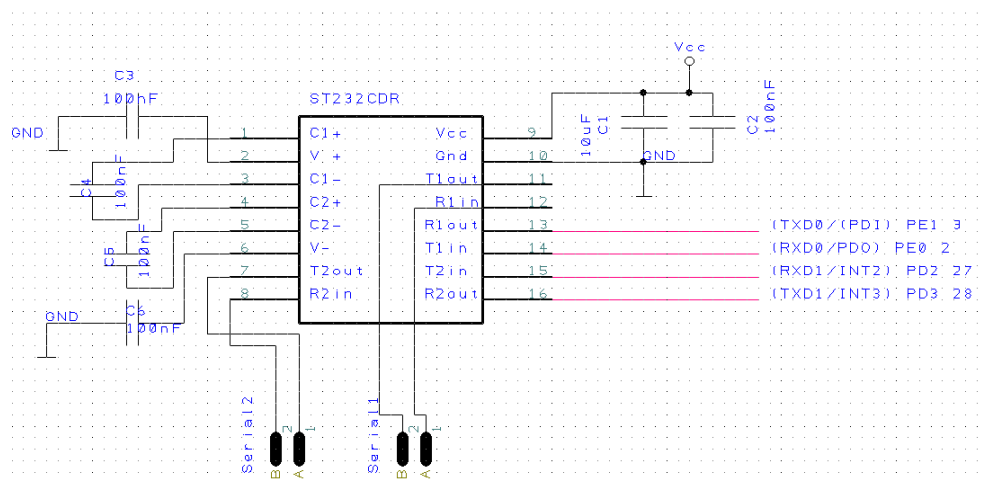
Součástí konektoru pro ethernet jsou trafo, nemusí být součástí plošného spoje.



Obrázek 6.10: Schéma zapojení ethernet čipu

## 6.9 Seriový čip

ST232 je zpojený podle dokumentace a jsou využity obě sériové linky. A je napájený z 5V větve.

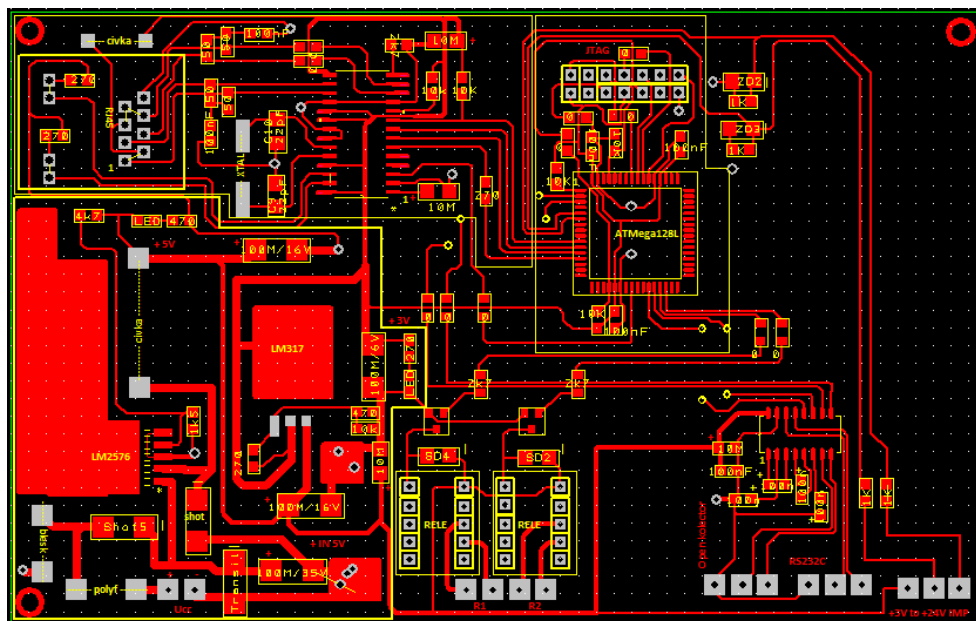


Obrázek 6.11: Schéma zapojení ST232CDR

## Výroba plošného spoje PLCP jednotky

Spodní část plošného spoje PLCP jednotky je použita jako zem, kromě 2 propojů pro sériovou linku.

Obsahuje konektory pro napájení, relé, sériové linky a čidla otočení turniketu.



Obrázek 7.1: PLCP jednotka

## 7. VÝROBA PLOŠNÉHO SPOJE PLCP JEDNOTKY

---

1. Vytisknutí plánu plošného spoje na laserové tiskárně na pauzovací papír.
2. Vyříznutí destičky o velikosti plánu z cuprexitu s vrstvou pozitivní fotoemulze o velikosti schématu.
3. Osvícení destičky UV zářením v osvitové jednotce přes pauzovací papír s plánem jednotky.
4. Odleptání fotoemulze narušené osvětlením, a to 1,5% roztokem hydroxidu sodného.
5. Odleptání nežádoucí mědi roztokem kyseliny solné a peroxidu vodíku.
6. Smytí zbylé fotoemulze acetonem.
7. Vyrtnutí otvorů pro prokvy a zapájení prokovů.

## Řešení firmwaru PLCP jednotky

Firmware v jednotce běží ve dvou základních třídách. První třída převážně implementuje kooperativní plánovač úloh. Druhá třída obsahuje jednotlivé úlohy a funkce potřebné pro jejich provoz.

Pro komunikaci po ethernetu jsem využil třídy `ip_arp_udp_tcp` a `enc28j60` z balíku knihoven Procyon [27]. Tyto třídy zajišťují komunikaci s `enc28j60` a `arp`, `udp` vrstvou.

Tuto knihovnu jsem po analýze vyhodnotil jako nejvhodnější, zároveň je tato knihovna velmi využívána jinými programátory, protože obsahuje velké množství různých tříd např. pro dispeje apod.

### Funkce PLCP jednotky:

- Čtení sériového portu 0
- Čtení sériového portu 1
- Čtení digitálního vstupu 0
- Čtení digitálního vstupu 1
- Komunikace s serverem přes ethernet
- Na základě komunikace otvírání relé

Další funkce, kterou jsem implementoval, je vzdálené nastavení ip a mac adresy a také reset na defaultní ip a mac adresu. Jednotka při startu kontroluje, zda byly propojeny 2 piny na programovacím konektoru. Pokud byly propojeny, resetuje svoje adresy. Tyto funkce byly zařazeny pro snazší implementaci PLCP jednotek do systému.

## 8. ŘEŠENÍ FIRMWARE PLCP JEDNOTKY



Obrázek 8.1: Firmware PLCP jednotky



## 8.1 Třída PLCP

Základní třída která zajišťuje prvotní inicializace, přerušení a spouštění kooperativního multitaskingu.

- Třída obsahuje buffery na packet a stav příjmu paketu.
- Dále obsahuje dvě přerušení od příjmu dat ze sériové linky. Tyto přerušení plní buffery s daty z třídy IOWORKER.
- Také obsahuje funkci pro posílání paketů.
- Funkce main je funkce zajišťující kooperační multitasking.

## 8.2 Třída IOWORKER

Třída zajišťující jednotlivé funkce jednotky a obsahující jednotlivé úlohy multitaskingu.

- Třída obsahuje pole pro nastavení, buffery na packet, buffery na RS232 a stavy relé a také vstupních senzorů.
- Dále funkce pro zápis a čtení EEPROM.
- Také obsahuje funkce pro inicializaci a práci sériové linky.
- Funkce pro vysílání a příjem paketů, které využívají ip\_arp\_udp\_tcp a enc28j60.
- Funkci pro zjištění sepnutí resetu IP/ARP adresy.

### Úlohy kooperativního multitaskingu:

- RS232DoWork –příjem dat ze sériové linky
- BinaryInDoWork –kontrola digitálních vstupů
- ReleyWait –práce s relé

## 8.3 Třída ip\_arp\_udp\_tcp

- Třída zajišťující arp udp a tcp vrstvu nad en28j60. Třídu jsem stáhnul z [27].
- Třída zajišťuje odpovědi na arp pakety, odpovědi na ping a odpovědi na ostatní udp pakety.

## 8.4 Třída `enc28j60`

- Třída zajišťující komunikaci s `en28j60`. Třídu jsem stáhnul z [27].
- Třída zajišťuje příjem paketů pro `PLCP.c`.

---

# Řešení softwaru entrance serveru

Entrance server běží jako služba, třída spouštěná službou je `clsEntranceServer`. Celý systém je plně konfigurovatelný přes textové konfigurační soubory a má několik logovacích souborů pro zjištění stavu nebo historie.

## 9.1 Funkce systému

- Připojení k pokladnímu systému pro editaci záznamů.
- Rozhodování o průchodech turnikety.
- Převedení rozhodnutí na pokladní systém.
- Vydávání lístků s čárovými kódy na stisk tlačítka.
- Logování stavů a jednotlivých průchodů ve více jazycích s konfigurovatelnými texty.
- Čištění a jiné úpravy databáze.
- Podpora licencování systému.

### 9.1.1 Připojení k pokladnímu systému

Pokladní systém může využít knihovnu `TicketManager_Clinet` pro připojení k přístupovému systému nebo použít vlastní tcp vrstvu pro komunikaci. Podporované funkce v přístupovém systému jsou pro přidání, editaci, smazání záznamu. Celková doba vypršení platnosti identifikátoru nebo doba vypršení identifikátoru při průchodu výstupním turniketem a také ping na přístupový systém.

Do systému lze přidat i lístek jehož platnost je upravená při průchodu turniketem. Toto se používá například na bazénech, kdy lístek je platný celý rok, ale při průchodu je platnost přeepsaná na hodinu.

### 9.1.2 Rozhodování o průchodech

První kontrola je zda přečtený identifikátor má přístupový systém v databázi, pokud není zapnutá funkce přesměrování neznámých identifikátorů pokladnímu systému, je zaznamenáno, že byl přečtený neznámý identifikátor a průchod končí. Pokud má dojít k přesměrování, pokračuje se dle kapitoly o přesměrování rozhodnutí.

Dále je provedena časová kontrola platnosti identifikátoru. Pokud se jedná o výstupní turniket, tak kontrolou projde i časově neplatný lístek, respektive lístek, kterému vypršela platnost. Tato je informace zaznamenaná v tabulce v databázi.

Poté je provedena kontrola, zda vstup není rozhodovaný pokladním systémem. Pokud má dojít k přesměrování pokračuje se dle kapitoly o přesměrování rozhodnutí.

Dále se provádí kontrola, zda je zapnutý anti-passback a pokud je, tak zda identifikátor může vstoupit. Funkci anti-passback lze zapnout i tak, že tato funkce bude ignorována, například po hodině. To znamená, že můžu vstoupit do jedné zóny dvakrát, ale interval mezi vstupy musí být minimálně hodina. Tato funkce může být samozřejmě vypnuta.

Po kontrole anti-passbacku je provedena kontrola na zjištění zbývajících vstupů a následně odečtení počtu vstupů. Identifikátor může mít i nekonečný počet vstupů. V konfiguračním souboru lze nastavit turniket, tak aby vstupy neodečítal a pouštěl i s nulovým počtem vstupů. Tato funkce řeší mimo jiné vnoření jednotlivých zón do sebe.

Problém vnořených zón je problém, kdy máme hlavní zónu 0 v objektu, ale uvnitř zóny 0 je další zóna 1, např. prohlídka pro VIP návštěvníky. Návštěvník má jeden vstup do zóny 0 a jeden vstup do zóny 1. Po vstupu do objektu již má nula vstupů do zóny 0 a při návratu z prohlídky by ho systém nepustil.

V poslední řadě je provedena kontrola na to, zda je zapnutý rotační senzor. Při zapnutém rotačním senzoru se počet vstupů odečítá až po průchodu. Pokud není proveden průchod do nastaveného intervalu z konfiguračního souboru, je celý vstup zrušen, relé otevření turniketu vypnuto a turniket se uzamkne.

Na závěr průchodu je vložena časová mezera zajišťující dostatečně dlouhý interval v průchodech návštěvníků.

### 9.1.3 Převedení rozhodnutí na pokladní systém

Pokladnímu systému je odeslán identifikátor a cílová zóna průchodu. Pokladní systém odpoví zda otevřít turniket. Poté při průchodu nebo ihned při vypnu-

tém senzoru otočení turniketu, je odeslána informace, tak aby si pokladní systém odečetl vstup nebo popřípadě peníze.

### 9.1.4 Logovací soubory

Systém využívá několik logovacích souborů.

- Log pro informace při spuštění systému, zda se vše úspěšně načetlo a spustilo.
- Log pro přidané záznamy pokladním systémem.
- Error log pro veškeré chyby vzniklé při běhu systému.
- Hlavní log pro záznamy průchodů je tabulka v databázi. Kde jsou zaznamenány všechny průchody, neplatné průchody a podobně. Tento log je čištěn při odebrání záznamu ze systému.

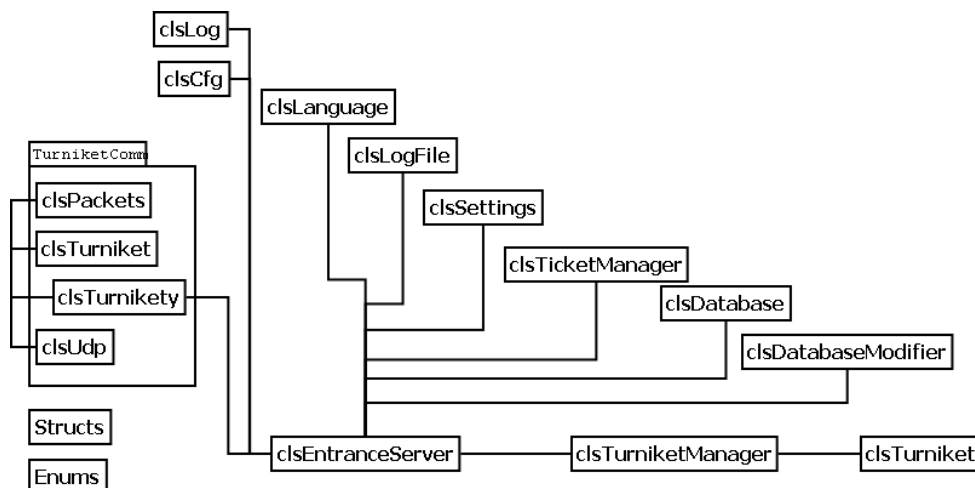
### 9.1.5 Čištění a úpravy databáze

Pro čištění a úpravy databáze slouží modul database modifier, který má vlastní konfigurační soubor. V tomto souboru jsou uvedené sql příkazy, které se ve zvolených intervalech spouští.

### 9.1.6 Licence

Systém obsahuje modul pro kontrolu licence, která je uvedena ve zvláštním konfiguračním souboru. Kontroluje od id procesoru a mateřské desky až po id hardisku, záleží na typu licence.

## 9.2 Model tříd



Obrázek 9.1: Model softwaru entrance serveru

### 9.3 clsEntranceServer

Třída nevykonává žádnou práci. Slouží pouze ke spuštění všech tříd kromě tříd pro turnikety. Po spuštění a inicializaci tříd předá odkazy ostatním třídám. Třída je také zodpovědná za řádné uzavření všech tříd mimo tříd s turnikety, které uzavírá clsTurniketManager.

### 9.4 clsTurniketManager

Třída načítá konfigurační soubory. Na základě těchto souborů spouští třídy s turnikety. Dále je za tyto třídy zodpovědná a při vypínání je i řádně ukončuje. Třídy s turnikety jsou implementovány jako více threadové. Třída TurniketManager tedy vyrábí hluboké kopie dat z konfiguračních souborů pro turnikety.

Třída přijímá data z modulu pro komunikaci s turnikety a dále je přerozděluje třídám jež zajišťují provoz turniketů, kterým paket patří a přijímá jejich odpověď. Odpovědi serializuje a přeposílá je zpět turniketům.

Dále načítá konfigurační soubor pro tisky a předává třídě clsPrinter.

### 9.5 clsTurniket

Třída zařizuje veškerou funkcionalitu PLCP jednotek. Při přečtení identifikátoru třída načte identifikátor z databáze. Dále probíhají kontroly na všechny

možné funkce, anti-passback, kontroly otočení turniketu, časové platnosti nebo počty průchodů.

Také umí funkce přesunout na pokladní systém. To jsou případy, kdy jde o placené vstupy a pokladní systém musí rozhodnout, zda má uživatel dostatek peněz na své elektronické peněženice v pokladním systému.

V neposlední řadě vytváří logy v databázi o proběhlých průchodech.

## 9.6 TurniketComm

Balíček tříd TurniketComm obsluhuje komunikaci s PLCP jednotkami. Obsahuje třídu s nastaveními pro pakety, třídu pro data turniketů a třídu která zajišťuje UDP komunikaci s PLCP jednotkami.

Tyto třídy řídí clsTurnikety, která zajišťuje rozhraní pro komunikaci s turnikety třídě clsTurniket entrance serveru.

## 9.7 clsLog

Třída clsLog je vytvořena ve 3 instancích. Log systému o spuštění, log o přidávaných záznamech pokladním systémem a jako globální error log.

Obsahuje funkce ke spuštění a připojení k souboru a dále k zápisu záznamu a uzavření souboru.

## 9.8 clsCfg

Třída clsCfg je vytvořena v několika instancích. Jednak pro třídu clsLanguageFile také pro clsSettings a clsDatabaseModifier.

Obsahuje funkce ke spuštění a připojení k souboru. Dále obsahuje funkce pro načítání čísel nebo řetězců, využívá k tomu funkce importované z kernelu windowsu a také obsahuje funkce na uzavření souboru.

## 9.9 clsLanguage

Třída slouží k načítání textů pro konfigurační soubory a pro logy do databáze. Využívá k tomu instanci třídy clsCfg.

## 9.10 clsLogFile

Třída využívá třídu clsLog pro zápis do souboru. Vytváří záznamy do souboru.

### 9.11 clsSettings

Třída načítá hlavní konfigurační soubor systému a informace předává dalším potřebným třídám. Ke své práci využívá třídu clsCfg.

### 9.12 clsTicketManager

Tato třída slouží pro příjem dat z ethernetového rozhraní. Je připojená do databáze a ukládá, popřípadě načítá záznamy. Také ukládá provedené úlohy do logovacího souboru.

Třída využívá pro komunikaci knihovnu tcpserver [28], která nabízí jednoduché rozhraní pro příjem a vysílání dat dat.

### 9.13 clsDatabase

Třída zajišťuje přístup k databázi. Umožňuje přidávání, editaci a mazání záznamů. Mimo jiné vytváří logy a umožňuje načítat dodatečné informace od záznamů v databázi, např. zda je záznam časově platný apod.

### 9.14 clsDatabaseModifier

Třída slouží k mazání a jiným úpravám databáze. Používá clsCfg pro svoji práci. Třída má ve svém konfiguračním souboru jednotlivé sql příkazy, které podle definovaných časových intervalů z konfiguračního souboru spouští.

### 9.15 Structs Enums

V souboru structs jsou uvedené struktury, které slouží pro přenos dat mezi třídami. Informace se předávají přes struktury z důvodu jednoduších úprav při změnách systému.

V souboru enums jsou uvedené výčtové typy pro návratové hodnoty funkcí, které předávají data jiným třídám.



---

## Závěr

V rámci bakalářské práce jsem se zabýval návrhem a vývojem nového řídicího systému pro turnikety. Na základě poznatků z praxe při vývoji prvního prototypu a při testování komerčních přístupových systémů, jsem realizoval systém splňující definované požadavky. Úspěšně vyvinutý a v praxi ověřený přístupový systém obsahuje většinu funkcí komerčních systémů a dále množství dodatečných funkcí, ve kterých systém předčí své konkurenty. Nejen, že jsem navrhl řídicí jednotku a software, který ji řídí, ale samotnou řídicí jednotku jsem vyrobil, včetně plošného spoje a osazení. Dále jsem napsal software pro jednotku a řídicí systém, který s jednotkou komunikuje. Celý systém jsem úspěšně zprovoznil.

Systém jsem dále rozvinul nad rámec bakalářské práce o displej, zvukové signály a další typy čteček. Systém je aktuálně nainstalovaný na desítky míst v ČR a na Slovensku, jedná se ve většině případů o muzea, stadiony nebo bazény. Nejmasivější nasazení systému je v objektu s 16 turnikety a několika set tisíci vstupy ročně.

Systém by se v budoucnu dal rozvinout o podporu funkcí docházkového systému.



---

## Literatura

- [1] Alveno: Přístupové systémy Alveno. <http://www.alveno.cz/cz/135/pristupove-systemy/>, 2008, [online]. [cit. 2015-03-17].
- [2] Alveno: Přístupový kontroler ACi. <http://www.alveno.cz/cz/369/pristupovy-kontroler-aci/>, 2008, [online]. [cit. 2015-03-17].
- [3] Acslime: Přístupové systémy Acslime. <http://www.acslime.cz/en/>, 2008, [online]. [cit. 2015-03-17].
- [4] Acslime: Koncový ceník systému ACS-line. [http://www.acslime.cz/media/document/cenik\\_acs-line\\_02\\_2015\\_konc.pdf](http://www.acslime.cz/media/document/cenik_acs-line_02_2015_konc.pdf), 2008, [online]. [cit. 2015-03-17].
- [5] Atmel: ATmega8-16AU. <http://www.tme.eu/cz/details/atmega8-au/mikroprocesory-atmel-avr-smd/atmel/#>, 2008, [online]. [cit. 2015-03-17].
- [6] Atmel: ATmega8-16AU. <http://www.tme.eu/cz/Document/26c82fa905cc57b0777eabdb6473af72/atmega8a.pdf>, 2008, [online]. [cit. 2015-03-17].
- [7] Atmel: ATmega32-16AU. <http://www.tme.eu/cz/details/atmega32-16au/mikroprocesory-atmel-avr-smd/atmel/#>, 2008, [online]. [cit. 2015-03-17].
- [8] Atmel: ATmega32-16AU. [http://www.tme.eu/cz/Document/5cc61c50912a387f5417dea1d0d6c1fc/atmega32\\_32l.pdf](http://www.tme.eu/cz/Document/5cc61c50912a387f5417dea1d0d6c1fc/atmega32_32l.pdf), 2008, [online]. [cit. 2015-03-17].
- [9] Atmel: ATmega128-16AU. <http://www.tme.eu/cz/details/atmega128a-au/mikroprocesory-atmel-avr-smd/atmel/#>, 2008, [online]. [cit. 2015-03-17].

- [10] Atmel: ATMega128-16AU. [http://www.tme.eu/cz/Document/872493b70c0a512004b66b2c33d60b52/atmega128\\_1281.pdf](http://www.tme.eu/cz/Document/872493b70c0a512004b66b2c33d60b52/atmega128_1281.pdf), 2008, [online]. [cit. 2015-03-17].
- [11] Microchip Technology: ENC28J60-I/SO. [http://www.tme.eu/cz/details/enc28j60-i\\_so/integrované-obvody-interface-ethernet/microchip-technology/#](http://www.tme.eu/cz/details/enc28j60-i_so/integrované-obvody-interface-ethernet/microchip-technology/#), 2008, [online]. [cit. 2015-03-17].
- [12] Microchip Technology: ENC28J60-I/SO. <http://www.tme.eu/cz/Document/472e1d0493a5e6b5d1fdddb421497a95/enc28j60.pdf>, 2008, [online]. [cit. 2015-03-17].
- [13] Texas Instruments: MAX232D. <http://www.tme.eu/cz/details/max232d/integr-obvody-interface-rs232422485/texas-instruments/#>, 2008, [online]. [cit. 2015-03-17].
- [14] St Microelectronics: ST232CDR. <http://www.tme.eu/cz/details/st232cdr/integr-obvody-interface-rs232422485/st-microelectronics/#>, 2008, [online]. [cit. 2015-03-17].
- [15] International Rectifier: IRF540NPBF. <http://www.tme.eu/cz/details/irf540npbf/tranzistory-s-kanalem-n-tht/international-rectifier/#>, 2008, [online]. [cit. 2015-03-17].
- [16] Zettler: AZ850. <http://www.tme.eu/cz/Document/b8fc14601611dc55f28666af3adb1be8/az850.pdf>, 2008, [online]. [cit. 2015-03-17].
- [17] On Semiconductor: LM2576T-ADJG. <http://pdf1.alldatasheet.com/datasheet-pdf/view/172056/ONSEMI/LM2576T-ADJG.html>, 2008, [online]. [cit. 2015-03-17].
- [18] St Microelectronics: LM317T. <https://www.fairchildsemi.com/datasheets/LM/LM317.pdf>, 2008, [online]. [cit. 2015-03-17].
- [19] Albatross: C++ description. <http://www.cplusplus.com/info/description/>, 2008, [online]. [cit. 2015-03-17].
- [20] Microsoft: C#. <https://msdn.microsoft.com/cs-cz/library/z1zx9t92.aspx>, 2008, [online]. [cit. 2015-03-17].
- [21] Microsoft: .NET. <https://msdn.microsoft.com/en-us/library/zw4w595w.aspx>, 2008, [online]. [cit. 2015-03-17].
- [22] Microsoft: Visual studio. <https://msdn.microsoft.com/cs-cz/vstudio>, 2008, [online]. [cit. 2015-03-17].

- [23] Microsoft: Microsoft SQL Database 2008 R2. <https://msdn.microsoft.com/en-us/library/ms165588%28v=sql.105%29.aspx>, 2008, [online]. [cit. 2015-03-17].
- [24] Oracle: Oracle SQL Developer. <http://www.oracle.com/technetwork/developer-tools/sql-developer/overview/index-097090.html>, 2008, [online]. [cit. 2015-03-17].
- [25] Design Spark: DesignSpark PCB Home. <http://www.rs-online.com/designspark/electronics/eng/page/designspark-pcb-home-pagel>, 2008, [online]. [cit. 2015-03-17].
- [26] Atmel: Atmel Studio 6. [http://www.atmel.com/microsite/atmel\\_studio6/1](http://www.atmel.com/microsite/atmel_studio6/1), 2008, [online]. [cit. 2015-03-17].
- [27] Stang, P.: Procyon AVRlib. <http://www.procyonengineering.com/embedded/avr/avr-lib/index.html>, 2008, [online]. [cit. 2015-03-17].
- [28] Baird, C.: C# TCP Server. <http://www.codeproject.com/Articles/488668/Csharp-TCP-Server>, 2008, [online]. [cit. 2015-03-17].



## Seznam použitých zkratek

**PLCP** PositronLabs communication processor

**SPI** Sériové periferní rozhraní

**RS232** Sériová linka

**MIPS** Počet zpracovaných instrukcí za sekundu

**SMD** Osazení součástek přímo na povrch plošného spoje

**ESR** Zdánlivý odpor kondenzátoru





---

## Obsah přiloženého CD

### B.1 Bakalářská práce

	readme.txt.....	stručný popis obsahu CD
	Firmware.....	adresář s projektem firmwaru pro procesor
	Software.....	adresář s Entrance Serverem
	Pomocné aplikace.....	adresář s pomocnými aplikacemi pro Entrance Server
	Bakalářská práce text.....	adresář s textem bakalářské práce
	Zdrojový kód práce.....	adresář se zdrojovým kódem práce
	Bakalářská práce.pdf.....	text bakalářské práce

## B.2 Aplikace bakalářské práce

readme.txt.....	stručný popis obsahu CD
└ Firmware .....	adresář s projektem firmwaru pro procesor
└ PLCP_V04_02C.atsln.....	projekt firmwaru PLCP jednotky
└ Software .....	adresář s Entrance Serverem
└ EntranceServerBAP	
└ EntranceServerBAP_projekt.sln.....	projekt Entrance Server
└ EntranceServerBAP	adresář s projektem knihovny Entrance Server
└ Bin	
└ Debug	
└ EntranceServerBAP.dll.....	knihovna Entrance Serveru spouštěná službou
└ TESTER....	adresář s projektem testovací aplikace Entrance Serveru
└ Bin	
└ Debug	
└ Language.....	adresář s nastavením textů
└ Language.cfg.....	konfigurační soubor textů
└ License.....	adresář s konfiguračními soubory pro licenci
└ License.cfg.....	konfigurační soubor licence
└ Error.log.....	záznam chyb při kontrole licence
└ Logs.....	adresář s logovacími soubory
└ ErrorLogFile.log....	záznam chyb Entrance Serveru
└ LogFile.log	záznam spuštění Entrance Serveru a jeho součástí
└ TicketManagerLogFile.log.....	záznam přidávaných identifikátorů pokladnou
└ Settings.....	adresář s konfiguračními soubory aplikace
└ DatabaseMod.cfg.....	konfigurační soubor pro modul DatabaseModifier
└ EntranceSettings.cfg....	hlavní konfigurační soubor Entrance Serveru
└ TESTER.exe .....	aplikace Entrance Serveru
└ ServiceTemplate ...	adresář s aplikací pro spuštění aplikace jako služba
└ EntranceServer Instalace...	adresář s aplikací Entrance Serveru pro instalaci
└ Pomocné aplikace.	adresář s pomocnými aplikacemi pro Entrance Server
└ Bakalářská práce text.....	adresář s textem bakalářské práce

## B.3 Pomocné aplikace bakalářské práce

readme.txt.....	stručný popis obsahu CD
Firmware.....	adresář s projektem firmwaru pro procesor
Software.....	adresář s Entrance Serverem
Pomocné aplikace.....	adresář s pomocnými aplikacemi pro Entrance Server
Licence.....	adresář s aplikacemi pro generování licencí
LicenseManagerClient.....	adresář s aplikací pro generování žádostí o licenci
LicenseManager_Client.exe.....	aplikace pro generování žádostí o licenci
LicenseManagerServer... ..	adresář s aplikací pro generování licencí
LicenseManager_Server.exe .	aplikace pro generování a zálohu licencí
PLCP tester.....	adresář s aplikací pro testování PLCP jednotky
PLIConfig_v2_1.exe.....	aplikace pro testování PLCP jednotky
POS_lite.....	adresář s aplikací jednoduchého pokladního systému
POS_lite.exe.....	jednoduchý pokladní systém
Bakalářská práce text.....	adresář s textem bakalářské práce



---

# Instalační a konfigurační manuál

## C.1 Instalace

1. Nakopírování souborů přístupového systému do požadované složky.
2. Spuštění cmd konzole pod právy administrátora.
3. Spuštění skriptu !RegistryAll.bat a kontrola že vše proběhlo správně.
4. Kontrola ve službách systému, přístupový systém by měl být zaregistrován jako služba pod názvem „Entrance server“ a měl by být závislý na službě „SQLAgent\$SQLEXPRESS”.
5. Spuštění skriptu „create database.sql“ v sql serveru na vytvoření databáze.
6. Vytvoření uživatele pro přístup k databázi. A nastavení připojení k databázi v Složka\_Přístupového\_systému\Settings\EntranceSettings.cfg pod sekci Database do hodnoty **ConnectionString** upravit a uložit: user id=<jméno uživatele>; password=<heslo>; server=<ip adresa serveru z databázi>; database=dbEntranceServer; connection timeout=30
7. Vytvoření žádosti o licenci pomocí aplikace „LicenseManagerClient.exe“ a její vytvoření v programu pro generování licencí. Licenci je třeba umístit do Složka\_Přístupového\_systému\License\License.cfg pod sekci License do hodnoty Key.

## C.2 Odinstalace

1. Spuštění skriptu !UnregistryAll.bat.
2. Smazání všech souborů.

### C.3 Závislosti

1. .NET Framework 4
2. Microsoft® SQL Server® 2008 R2
3. Windows 7 +

### C.4 Komunikační porty

1. Komunikace z turnikety na 1200,1201 [UDP] + porty z konfiguračních souborů [TCP].

### C.5 Omezení

Id identifikátoru nemůže obsahovat: # | < >. Pokud lístek bude obsahovat tyto znaky nebude systém funkční.

### C.6 Konfigurace

Soubor: Složka\_Přístupového\_systému\Settings\EntranceSettings.cfg

- 

#### *General*

**Language**=<text> ze které sekce se budou načítat jazyky ze souboru Složka\_Přístupového\_systému\Language\Language.cfg

- 

#### *POS\_Tickets*

**Port**=<číslo větší 0> port na kterém se poslouchá příchozí packety od knihovny „TicketManager\_Client.dll“ která slouží k přidávání lístku do přístupového systému.

- 

#### *POS\_RouteEntry*

**RouteUnknown**=<0,1> předávání rozhodování neznámých lístků pomocí knihovny „ElectronicWalletLib.dll“.

- 

#### *Turnikets*

**T<X>**=<text> IP adresa desky PLCP  
X musí být číslované od 0...n.

- 

### *TX\_Y*

X-číslo turniketu, Y-číslo čtečky

**TargetZone**=<číslo větší rovno 0> cílová zóna průchodu.

**OutputReader**=<0,1> zapnout funkci výstupní čtečka? (neodečítá vstupy).

**OutputTurniket**=<0,1> zapnout funkci výstupní turniket? (nekontroluje se čas <do>).

**OpenReley**=<číslo větší rovno 0> které relé otevřít při platném přístupu?

**OpenPulseLen**=<číslo větší rovno 0> doba otevření turniketu a pokud je zaplý APB tak zároveň nejdelší doba po kterou přístupový systém čeká na senzor protočení turniketu.

**RotationSense**=<0,1> zapnout rotační sensor?

**RotationSensor**=<číslo větší rovno 0> který rotační senzor sledovat.

**AntiPassback**=<0,1> zapnout funkci anti-passback?

**AntiPassbackTime**=<číslo větší rovno 0> doba v sekundách po úspěšném vstupu, po které se APB vypne.

**EntryTimeOut**=<číslo větší rovno 0> interval v ms. mezi průchody ve které turniket nereaguje na další vstupy.

Soubor: Složka\_Přístupového\_systému\Settings\DatabaseMod.cfg

- 

### *Querys*

**TimeStamp**=<číslo větší rovno 0> doba v sekundách za jak dlouho se má tenhle konfigurační soubor spustit znovu. Doba by měla být co největší.

- 

### *Query < X >*

musí být číslované 0...n

**TimeStamp**= <číslo větší rovno 0> doba v sekundách za jak dlouho se má tenhle script soubor spustit znovu.

**NextTimeDone**= <datum> datum dalšího spuštění scriptu. Pokud bude smazáno tak se do něj uloží aktuální čas při příští kontrole souboru.

**Query**=<text> SQL příkaz.

**LastTimeState**= status minulého spuštění příkazu.

Soubor: Složka\_Přístupového\_systému\Settings\Language.cfg

### Proměné:

- V hodnotě Log\_StartEntrance\_Message : <version>
- V ostatních hodnotách Db\_Entry\_... : <ticket>,<turniket>,<reader>,<zone>
- Zvolíme si název jak chceme a na tuto sekci musíme odkazovat v Složka\_Přístupového\_systému\Settings\EntranceSettings.cfg

### Hodnoty:

- **Log\_StartEntrance\_Message**=<text + proměné> spuštění entrance serveru.
- **Log\_StartEntrance\_Ok**=<text> start ok
- **Log\_StartEntrance\_Fail**=<text> start fail
- **Log\_LicenseCheck\_Ok**=<text> licence ok
- **Log\_LicenseCheck\_Fail**=<text> licence fail
- **Log\_StopEntrance\_Ok**=<text> stop ok
- **Log\_StopEntrance\_Fail**=<text> stop fail
- **Log\_EntranceSettings\_Load\_Ok**=<text>load cfg souboru (Složka\_Přístupového\_systému\Settings\EntranceSettings.cfg)
- **Log\_EntranceDatabase\_Connect\_Ok**=<text> připojení k databázi ok
- **Log\_EntranceDatabase\_Connect\_Fail**=<text> připojení k databázi fail
- **Log\_Language\_Load\_Ok**=<text> language file ok
- **Log\_LoadTurnikets\_Ok**=<text> nastaveni turniketu načteno
- **Log\_LoadTurnikets\_Fail**=<text> nastaveni turniketu nenačteno
- **Log\_TicketManager\_Ok**=<text> připojeno k modulu „TicketManager\_Client.dll
- **Log\_TicketManager\_Fail**=<text> nepřipojeno k modulu „TicketManager\_Client.dll
- **Log\_DatabaseMod\_Ok**=<text> database modifier spuštěn Složka\_Přístupového\_systému\Settings\DatabaseMod.cfg



- **Log\_DatabaseMod\_Fail**=<text> database modifier není spuštěn  
Složka\_Přístupového\_systému\Settings\DatabaseMod.cfg fail
- **Db\_Entry\_UnknownTurniket**=<text + promene> neznámý turniket
- **Db\_Entry\_TicketDBData\_Fail**=<text + promene> špatně nastavené zóny v turniketu ( dotaz na neexistující zónu)
- **Db\_Entry\_UnkownTicket**=<text + promene> lístek neexistuje
- **Db\_Entry\_NoMoreEntry**=<text + promene> lístek nemá vstupy
- **Db\_Entry\_ApbBlock**=<text + promene> APB block
- **Db\_Entry\_InvalidTime**=<text + promene> neúspěšná časová validace
- **Db\_Entry\_RotSenTimeOut**=<text + promene> vypršel časový limit pro rotační sensor
- **Db\_Entry\_Db\_Fail**=<text + promene> database error
- **Db\_Entry\_UknownFail**=<text + promene> + <message> neznama chyba
- **Db\_Entry\_Ok**=<text + promene> vstup OK
- **Db\_Entry\_PosResolvedMessage**=<text + promene> odeslani listku „ElectronicWalletLib.dll“
- **Db\_Entry\_PosResolved\_No**=<text + promene> zamitnuti „ElectronicWalletLib.dll“
- **Db\_Entry\_PosResolved\_timeout**=<text + promene> vypršení časového limitu na odpověď od „ElectronicWalletLib.dll“

## C.7 Logy

- Soubor: Složka\_Přístupového\_systému\Logs\LogFile.log  
Log o spuštění/vypnutí serveru
- Soubor: Složka\_Přístupového\_systému\Logs\ErrorLogFile.cfg  
Chyby v během běhu systému. Neměl by obsahovat žádné errorry.
- Soubor: Složka\_Přístupového\_systému\Logs\TicketManagerLogFile.cfg  
Záznam komunikace z knihovnou „TicketManager\_Client.dll“.

## C.8 Licence

Soubor: Složka\_Přístupového\_systému\License\License.cfg

- *License*

**Key**=<Licenční klíč>

- *Security*

**Level**=<0-2> úroveň zabezpečení