

ZÁPADOČESKÁ UNIVERZITA V PLZNI  
FAKULTA ELEKTROTECHNICKÁ

Katedra elektromechaniky a výkonové elektroniky

**BAKALÁŘSKÁ PRÁCE**

Vylepšení detektoru elektromagnetického pole

Autor: Kryštof Przechowski

2015

Vedoucí práce: Ing. Richard Linhart Ph.D.

ZÁPADOČESKÁ UNIVERZITA V PLZNI  
Fakulta elektrotechnická  
Akademický rok: 2014/2015

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Kryštof PRZECHOWSKI**  
Osobní číslo: **E12B0235P**  
Studijní program: **B2644 Aplikovaná elektrotechnika**  
Studijní obor: **Aplikovaná elektrotechnika**  
Název tématu: **Vylepšení detektoru elektromagnetického pole**  
Zadávající katedra: **Katedra elektromechaniky a výkonové elektroniky**

### Zásady pro vypracování:

Doplnění detektoru elektromagnetického pole mechanickými prvky, obslužným SW modulem a vylepšení datové komunikace.

1. Seznamte se s vlastnostmi detektoru elektromagnetického pole, navrhnete umístění a přívod vodičů tak, aby byl výsledek co nejméně ovlivněn.
2. Proveďte návrh a realizaci mechanických úprav zajišťujících vyšší mechanickou stabilitu a menší vliv okolí.
3. Sestavte a otestujte skript pro získávání naměřených dat z detektoru.
4. Upravte formát datové komunikace pro předávání výsledků do podoby vhodnější pro vyšší vzdálenosti bez vlivu rušení.



Rozsah grafických prací: podle doporučení vedoucího  
Rozsah pracovní zprávy: 20 - 30 stran  
Forma zpracování bakalářské práce: tištěná/elektronická  
Seznam odborné literatury:

1. Dostupné manuály k jazyku Python a detektoru elektromagnetického pole.
2. Další vhodnou literaturu si student vyhledá v dostupných pramenech podle doporučení vedoucího práce.

Vedoucí bakalářské práce: Ing. Richard Linhart, Ph.D.  
Katedra aplikované elektroniky a telekomunikací  
Datum zadání bakalářské práce: 15. října 2014  
Termín odevzdání bakalářské práce: 8. června 2015

  
Doc. Ing. Jiří Hammerbauer, Ph.D.  
děkan



  
Prof. Ing. Václav Kůs, CSc.  
vedoucí katedry

V Plzni dne 15. října 2014

# Abstrakt

Bakalářská práce se zabývá úpravou detektoru elektromagnetického pole, který snímá jeho aktivitu v oblasti dlouhých vln. Je zde řešena komunikace přístroje a počítače. Jsou zde vytvořeny programy pro přesun dat ze zařízení, následnou úpravu dat pro uložení do souboru a databáze a pro synchronizaci času zařízení. Pro data z databáze jsou vytvořeny internetové stránky a tabulky. Uvedeme systémy fungující na podobných principech jako naše zařízení. Hlavním cílem této práce je zpřístupnit data ze zařízení pro širokou veřejnost.

## Klíčová slova

detektor elektromagnetické aktivity, jazyk Python, jazyk PHP, databáze phpMyAdmin, vkládání databáze na internetovou stránku

# Abstract

The bachelor thesis deals with the regulation of electromagnetic field detector, which measures and records its activity in the area of long waves. There is communication solution between the device and a computer. It is developed programs for moving data from the device, their subsequent treatment for storage in the file and the database and for time synchronization of the device. It is created websites and tables for the data from the database. Afterwards there are mentioned systems operating on similar principles as our device. The main task of this work is to make the data from the device accessible to the general public.

## Key words

elmg. pulse detector, program Python, program PHP, database phpMyAdmin, insert database into website

# Prohlášení

Prohlašuji, že mnou vypracovaná bakalářská práce je vypracována samostatně, s použitím odborné literatury, která je uvedena v seznamu.

Dále prohlašuji, že mnou použitý software je legální.



Podpis

# Obsah

Úvod .....	1
<b>1 Konstrukce zařízení a princip jeho činnosti .....</b>	<b>2</b>
1.1 Anténa .....	2
1.2 Analogová část zařízení .....	2
1.3 Digitální část zařízení .....	4
1.4 Typy použité komunikace .....	5
<b>2 Zpracování a uložení dat z přístroje .....</b>	<b>8</b>
2.1 Databáze pro ukládání dat .....	8
2.2 Soubor pro ukládání dat .....	9
2.3 Sdílení dat .....	9
2.4 Příjem dat z přístroje.....	10
2.5 Zpracování dat získaných přístrojem .....	11
2.6 Ukládání dat .....	12
2.7 Synchronizace času .....	14
<b>3 Umístění dat z měření na internet .....</b>	<b>15</b>
3.1 Přístup k datům v databázi .....	16
3.2 Výběr a příprava dat pro graf .....	16
3.3 Tvorba grafu .....	17
3.4 Tvorba tabulky .....	18
3.5 Další alternativy sdílení dat .....	18
<b>4 Měření elektromagnetické aktivity .....</b>	<b>19</b>
<b>5 Podobná zařízení .....</b>	<b>22</b>
5.1 Systémy pro detekci blesků v Čechách .....	22
<b>Závěr .....</b>	<b>25</b>
Použitá literatura .....	26
Seznam obrázků .....	27
Seznam grafů .....	27
Příloha .....	28

# Úvod

Bakalářská práce se zabývá vylepšením detektoru elektromagnetické aktivity. Detektor je umístěn na Katedře aplikované elektrotechniky a telekomunikací FEL ZČU. Byl již vytvořen funkční prototyp, u kterého ovšem nebyly otestovány všechny funkce a při komunikaci s počítačem docházelo k rušení.

V úvodní části je krátké seznámení s konstrukcí zařízení, dále je vysvětleno, na jakých principech zařízení funguje a k čemu slouží. Je zde také zahrnuta problematika sériové komunikace a varianty, které byly testovány na zařízení.

Další část této práce se zabývá tvorbou skriptů. Bylo nutné vytvořit programy umístěné na počítači, které by automaticky pracovaly po sériové komunikaci s přístrojem. Má za úkol zpracovávat data, a následně je ukládat. Dále bylo nutné synchronizovat čas na přístroji. Pro data uložená v databázi by pak měla být vytvořena internetová stránka, ze které by data byla přístupná veřejnosti.

V závěru práce budou uvedeny podobné projekty, které již fungují a fungují na podobných principech, na jakých pracuje i naše zařízení. Nakonec bude celé zhodnocení tohoto projektu.



# 1 Konstrukce zařízení a princip jeho činnosti

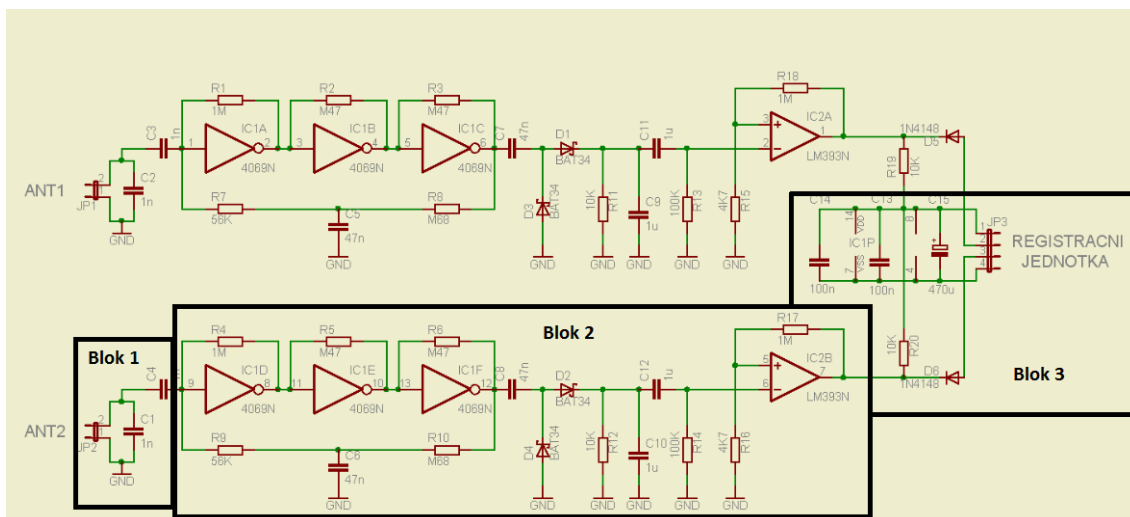
Detektor pracuje obdobně jako rádiový přijímač. Zachycuje vlnění v pásmu desítek kHz až stovek kHz. Zdrojem vlnění mohou být přírodní vlivy atmosférické jako je například bouřková činnost, nebo vlnění může produkovat činnost člověka. Výstup z detektoru bude zpracováván na počítači, proto bude nutné, aby výstup byl interfacem počítače.

## 1.1 Anténa

Na detektoru byla jako anténa použita feritová tyč, která byla ovinuta měděným izolovaným drátem. Drát má průměr 0,3mm a tvoří asi 100 závitů. Ferit je materiál, který má  $\mu_r$  vyšší jak jedna (udává se okolo 2000), tudíž zesiluje magnetické pole. Díky tomu můžeme volit anténu menších rozměrů při zachování stejné schopnosti zachycení vlnění. Pro vyšší všesměrovost je zde ještě druhá, shodně konstruovaná anténa. Obě jsou na povrchu zařízení a svírají spolu úhel přibližně 90°.

## 1.2 Analogová část zařízení

Detektor má dvě části, analogovou a digitální. Analogová část je vstupní snímač. Tento snímač je tvořen dvěma přímo zesilujícími přijímači, které jsou naladěny do pásma velmi dlouhých vln (desítky kHz až stovky kHz). Impulzy indukovaného napětí průchodem přes analogovou část jsou zesíleny a tvarovány tak, aby je mohla zpracovat digitální část.



Obr. č. 1. Schéma analogové části.

### Blok 1

V prvním bloku je k anténě paralelně připojen kondenzátor a tím vznikne rezonanční obvod. Rezonanční obvod je naladěn na frekvenci pásma, kde vyzařují sledované jevy například blesky hodně energie, ladění nemusí být příliš přesné.

### Blok 2

Za LC oscilátorem je zesilovač s inverty bez tvarování, Zesilovač je technologie CMOS. Po zavedení zpětné vazby se zesílení zmenší, tím se umožní lineární provoz, navíc tato zpětná vazba je frekvenčně závislá a upraví frekvenční charakteristiku. Toto zapojení je funkční, ovšem není optimální z hlediska šumu, avšak my potřebujeme pouze čítat impulzy a tak je to dostatečné řešení.

Blok 3.

Za zesilovačem je usměrňovač a za ním je komparátor s hysterezí. Jakmile prudce stoupne úroveň signálu zachyceného anténou, překlopí se komparátor a vyšle impuls do čtvrtého bloku.

## 1.3 Digitální část zařízení

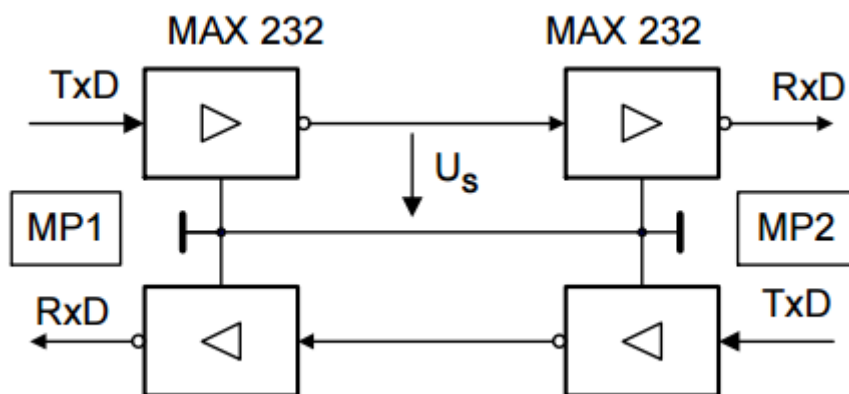
Digitální část funguje jako registrační jednotka. Tato jednotka má za úkol sčítat jednotlivé impulsy každou minutu. Data z měření se pak ukládají do paměti EEPROM. Registrační jednotka obsahuje mikroprocesor Atmel AT89S51. Jednotka obsahuje obvod RTC, který funguje jako hodiny a kalendář. Pro účely tohoto zařízení je to nezbytné, aby bylo možné měřit bez připojení k počítači a data ze zařízení později jen vyčíst. Indikaci stavů, aby bylo hned patrné, že zařízení vypadlo, nebo že zařízení začalo měřit, zajišťují tři led diody a zvuková signalizace. K ovládání slouží tři tlačítka, jedno z nich je pro reset zařízení. Přes sériovou komunikaci je možné zjistit aktuální počet měřených impulsů, aktuální čas a zaslat data, která se použijí pro nastavení času. Dále je možné po této komunikaci posílat data z paměti EEPROM. Paměť EEPROM má takovou kapacitu, že je možné do ní uložit data z měření, která jsou získávána po dobu více než dvou dnů.

## 1.4 Typy použité komunikace

Z mikropočítače je vyvedena sériová komunikace typu UART. Tato komunikace má typicky jeden start bit a jeden stop bit, plus dalších 8 bitů datových. Přenosová rychlost této komunikace je 9600 bd. Napětí  $U_L$  v této komunikaci je logická nula a  $U_H$  je logická jednička. Tato komunikace je nevhodná pro přenos signálu na větší vzdálenosti, a tak bylo nutné použít rozhraní k tomuto účelu vhodnější.

### RS-232

Tato komunikace slouží pro přenos dat. Přenosové stavy napětí u vysílače jsou  $U_L = -10V$  a  $U_H = +10$ . Pro přenos jsou potřeba tři vodiče, jeden je zemní vodič, další je pro vstupní data a třetí pro výstupní data.



Obr. č. 2. Přenosová komunikace RS-232.

TxD jsou výstupní data, RxD vstupní. Tato data se upravují pro přenos a pak zpětně pro příjem v zařízení. Mezi datovými vodiči je zemní vodič sloužící jako referenční napětí.

Přenosová rychlost je taková, kterou dodává UART. Tento druh přenosu pro naše zařízení byl nevhodný. Použitá přenosová rychlost produkovala

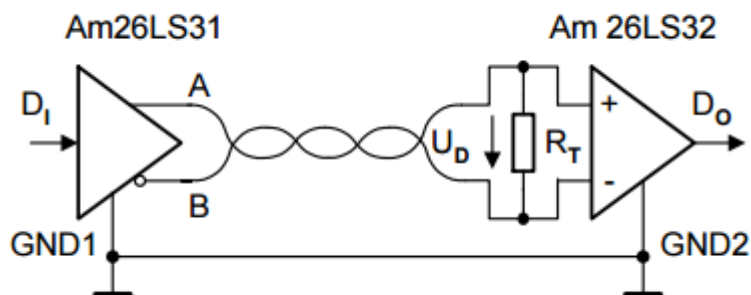
elektromagnetické vlnění ve frekvenčním pásmu velmi dlouhých vln, na které je laděn oscilátor a spolu s vysokými úrovněmi přenosového signálu to byl významný zdroj rušení. Toto rušení bylo zachyceno a zaznamenáno vždy, když probíhala komunikace [1].

## USB

Pro nevýhody přenosu pomocí RS-232 byla vyzkoušena komunikace pomocí USB. Pro přenos signálů USB jsou použity vodiče, které jsou vzájemně zatočeny, při změně signálu se mění jednotlivé polarity. Přenosová rychlost je mnohem vyšší než u přenosu pomocí RS-232, v řádech desítek MB. Pole vyvolané signály v těchto vodičích se od sebe téměř odečítá a tak je rušení zanedbatelné. Také přenosová rychlost po lince je mnohem vyšší (v řádech desítek MB) než u přenosu po lince pomocí RS-232. Díky vyšší přenosové rychlosti vzniklé vyzařování, které by mohlo rušit, by mělo mnohem vyšší frekvenci, a proto by nespadlo do oblasti velmi dlouhých vln, na které je laděn oscilátor. U tohoto provedení nedocházelo k žádnému rušení, které by vznikalo při komunikaci se zařízením. Nevýhodou USB je ovšem maximální délka připojení, která je maximálně pět metrů bez dalších úprav. Proto ani toto řešení není optimální, jelikož zařízení bude umístěno ve větší vzdálenosti od počítače.

## RS-485

Sériová komunikace RS-485 je fyzickou vrstvou. Výhodou této komunikace je, že je na ní možné připojit více zařízení. Jednotky jsou spolu propojené pomocí dvou do sebe kroucených vodičů (vodiče A a B) a země (GND). Podobně jako u USB důvodem zakroucení drátů je omezení vyzařování.



Obr. č. 3. Přenosová komunikace RS-485.

Úrovně vysílaného signálu oproti RS-232 jsou podstatně nižší, uvádí se +2V na vodiči A a -2V na vodiči B, při změně stavu z logické ,1' do logické ,0' se na obou kabelech změnila polarita. Na konce vedení se obvykle ještě přidává odpor pro impedanční přizpůsobení, tento odpor se volí na základě charakteristické impedance, která je 100 nebo 120  $\Omega$ . Přijímač rozlišuje logický stav A – B při rozdílu vyšším jak 200mV. Tato komunikace je odolná proti rušení, a proto je často používána v průmyslovém prostředí. Propojení zařízení s počítačem je možné až na vzdálenost 1200m pro přenosovou rychlost menší než 100kb/s. V dohledné době by měřící zařízení mělo přejít na tuto komunikaci. Přenosové rychlosti použité v této komunikaci by mohly generovat sice rušení pro přístroj v oblasti velmi dlouhých vln, ale díky zapleteným kabelům A a B je toto rušení zanedbatelné. Díky této komunikaci bude přístroj možné umístit na zamýšlenou polohu na střeše školy [1].

## 2 Zpracování a uložení dat z přístroje

Data, která byla zaznamenána a jsou uložena v přístroji je potřeba nějakým způsobem zpracovat. K práci s daty je nutné vytvořit skript, pro psaní skriptu byl vybrán programovací jazyk Python. V jiných programovacích jazycích, jako je například C nebo Pascal pro poměrně jednoduché vykonání úlohy je zapotřebí napsat dlouhý kód. Nevýhodou Pythonu je, že vykonávání operací, zejména početních zabírá více času, pro naši potřebu je ovšem rychlost vykonávání operací v Pythonu dostačující. Programování v Pythonu je objektově orientované a je možné použít řadu volně dostupných uživatelských balíčků, které značně usnadňují práci. Data ze zařízení, která jsou zpracována, se následně ukládají do databáze a souboru.

### 2.1 Databáze pro ukládání dat

Pro přechovávání dat byl použit databázový systém MySQL. Jeho obrovskou výhodou je, že je zdarma na rozdíl třeba od databázového systému Oracle a je postačující, pokud nepotřebujeme pracovat s obrovským množstvím dat. Pro práci s obsahem databázového systémem byl použit phpMyAdmin. Tento systém je psán v kódu PHP a slouží jako prostředek ke správě databáze MySQL. Systém phpMyAdmin běží na serveru (školním počítači), práce s tímto systémem je možná po internetové síti, ovšem pokud server vypadne, není možné se na něj dostat. Administrátor vytvořil pro použití k tomuto projektu uživatele przechowski, tomu přidělil potřebné oprávnění k funkcím v systému a přihlašovací jméno a heslo. Příkazem *CREATE DATABASE její název;* vytvoříme databázi, do které umístíme tabulku.

```
USE przechowski;
```

```
CREATE TABLE detektor (
```

```
  `tstmp` time NOT NULL DEFAULT CURRENT_TIMESTAMP COMMENT 'Cas  
nahrani',
```

```
  `raz` float COMMENT 'Pocet razu',
```

```
);
```

Nejprve systém musí vědět, s jakou databází chceme pracovat, a to i v případě, že máme vytvořenou jedinou databázi. *USE przechowski* se přesuneme do vytvořené databáze s názvem *przechowski*. Příkazem *CREATE TABLE detektor* v této databázi vytvoříme tabulku s názvem *detektor*. Mezi kulatými závorkami bude definice jednotlivých sloupců. *`tstmp`* vytvoříme sloupec, který bude datového typu *time*. *NOT NULL* znamená, že při vkládání tento sloupec musí být vyplněn. *COMMENT 'Cas nahrani'* již připojíme pouze komentář, aby bylo jasné, co sloupec znamená. Takto vytvořený sloupec, do kterého budeme vkládat hodnoty typu *time*, bude vyžadovat vkládání řetězce určitého formátu ‚rok-měsíc-den hodina:minuta:sekunda‘. Druhý sloupec se vytvoří obdobně, vkládat do něj budeme řetězec s číselnými znaky. Jednotlivé příkazy se ukončují středníkem [2].

## 2.2 Soubor pro ukládání dat

Samotné ukládání dat do databáze nestačí. Množství dat, které by bylo potřeba uložit za dobu nepřetržitého měření, by bylo příliš veliké. Možností, která by daný problém řešila, je data z databáze průběžně mazat a skladovat je v souboru.

## 2.3 Sdílení dat

Jestliže někdo bude potřebovat starší data z měření například pro výzkum, která již v databázi budou smazána a nahrazena daty novými, bude soubor s daty možné jednoduše zkopírovat a zájemci poslat po internetu nebo nahrát například na přenosnou flash. Další možností by bylo na počítači vytvořit složku, do které by se pravidelně data ukládala. Do této složky by byl umožněn přístup přes internet pomocí propojení dvou počítačů například přes SSH protokol.



Zájemci o data z přístroje by tak stačilo zaslat přihlašovací údaje, dále již bychom se nemuseli o víc starat, jelikož by zájemce měl přístup stále k aktuálním datům. Tento způsob sdílení dat nebyl vytvořen, avšak budou-li zájemci o data z přístroje, bylo by vhodné jej vytvořit.

## 2.3 Příjem dat z přístroje

Pro zpracování dat z přístroje uložených v paměti EEPROM je zapotřebí poslat tato data přes sériovou komunikaci do počítače. PySerial je balíček, který obsahuje funkce, které jsou určeny pro práci se sériovým portem. Je možné ho použít na Windows i Linux. Tento balíček není běžnou součástí Python, a proto je nutné ho stáhnout. Stáhnout ho lze ze stránek <http://pypi.python.org/pypi/pyserial>.

```
ser = serial.Serial(  
    port='/dev/ttyUSB0',  
    baudrate=9600,  
    timeout=1,  
    bytesize=serial.EIGHTBITS,  
    parity=serial.PARITY_NONE,  
    stopbits=serial.STOPBITS_ONE)
```

Funkce *serial.Serial* navazuje komunikaci se sériovým portem. Přiřadíme jí některé proměnné. Python umožňuje přiřazení objektů do proměnných a dále budeme pro práci se sériovým portem používat tuto proměnnou. Pro přístup na sériový port je nutné uvést některé další specifikace. Příkazem **port='/dev/ttyUSB0'** vybíráme, na který port v počítači se chceme připojit. Příkaz **baudrate=9600** nastavuje komunikační rychlost, pro náš přístroj je komunikační rychlost 9600db. Příkazem **timeout=1** nastavujeme časový limit, toto je důležité hlavně, pokud používáme funkci *readline*. Kdybychom to neuvědly, mohlo by dojít k zablokování komunikace. Příkazem **bytesize=serial.EIGHTBITS** nastavíme, po kolika bitech má být odeslána zpráva, standartní je 8 bitů použitých i v tomto případě.

**parity**=*serial.PARITY\_NONE* tímto příkazem říkáme, že nechceme posílat paritu a příkazem **stopbits**=*serial.STOPBITS\_ONE* nastavíme jeden stopbit, to je bit, kterým není posílána informace, ale který ukončuje zprávu, aby nedošlo při dalším čtení k chybě. Celá komunikace je ošetřena výjimkou, kdyby se k definovanému sériovému portu nepovedlo připojit.

K portu, pro který jsme nadefinovali ve funkci *serial.Serial*, nyní již přistupujeme přes *ser*, ke kterému je přiřazen *serial.Serial*. Pro poslání znaku přes sériový port slouží funkce *write("znak")*. Pro čtení řádků je zde funkce *readline()*, *radek=ser.readline()*, do proměnné *radek* přiřadí řetězec poslaný sériovým portem, jako odřádkování je zde použit znak *Enter* [3].

## 2.4 Zpracování dat získaných přístrojem

Jestliže vyšleme do přístroje po sériové komunikaci znak ‚x‘ přístroj z paměti EEPROM přečte naměřená data ve formátu CSV, která přes sériovou komunikaci pošle počítači. Tyto data řádek po řádku funkcí *ser.readline()* přečteme. Data odeslaná z přístroje mají následující podobu:

```
$b  
21:20;05.05;MSOK;EEOK;  
21:22;05.05;2;1;0;0;0;1;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;MSOK;EEOK;  
08:25;06.05;0;2;MSOK;EEOK;  
$e
```

Začátek souboru je značen symbolem ‚\$b‘. Na začátku řádku je datum a čas, první znaky, dvě dvojice cifer od sebe oddělené dvojtečkou je čas hodin přístroje a další dvě dvojice cifer oddělené od sebe tečkou je datum přístroje. Datum a čas je oddělen středníkem a následuje za ním počet Impulzů sečtených po minutě (dále jen impulzů), jednotlivé impulzy jsou od sebe odděleny středníkem. Řádek je ukončen značkou kontrolky CRC paměti RAM a EEPROM. Pro konec souboru je použit symbol ‚\$e‘.

Řádky se čtou v cyklu, kde se testuje symbol konec souboru. Takto získané řádky není možné jednoduše uložit do souboru a databáze, je nutná další úprava.

Vybere se řádek, který obsahuje informaci o čase a impulzech a s ním se dále pracuje. Funkcí `split(";")` se původní řetězec rozdělí na více částí, jednotlivé části jsou v původním řetězci od sebe odděleny středníkem. Z těchto částí vznikne seznam. První a druhý řetězec v seznamu připadá na čas a datum, poslední řetězce pak na kontrolní značky CRC paměti, počet těchto řetězců se nikdy nemění. Zbytek řetězců v seznamu připadá na impulzy. Čas vyčtený z řetězce platí pouze pro první impuls v seznamu.

Vytvoříme dva seznamy, jeden bude obsahovat impulzy a druhý čas, ve kterém měření impulzů probíhalo. Seznam intervalů impulzů můžeme převzít ze seznamu, který jsme již vytvořili a odstranit z něho řetězce data, času a kontrolní značky CRC paměti. Druhý seznam bude obsahovat informaci o čase impulzů. Čas prvního impulzu získáme z prvního vytvořeného seznamu, další časy seznamu inkrementujeme o jednu minutu. Abychom s těmito daty mohli provádět matematické operace, musí se nejprve přetypovat. Po vykonání matematických operací je nutné je přetypovat zase zpátky na řetězce, jelikož do souboru a databáze se zapisují řetězce. Počet prvků v seznamu pro čas a impulzy bude stejný. Řetězce pro ukládání do databáze musí mít podobu rod-měsíc-den hodina-minuta-sekunda. Vypadat to bude například:  
2015-05-30 10:12:00

## 2.5 Ukládání dat

Data z měření, která jsme upravili, jsou připraveny k uložení do databáze a souboru. Jak v souboru, tak v databázi budou stejného formátu. Pro práci s databází existuje balíček `MySQLdb`, který obsahuje funkce pro práci s databází `MySQL`. Abychom tento balíček mohli používat, je nutné ho nejprve nainstalovat. Tento balíček je možné stáhnout ze stránek <https://pypi.python.org/pypi/MySQL-python/>.

```
db = MySQLdb.connect(  
    host="localhost",  
    user="przechowski",  
    passwd="bcelmag21",  
    db="przechowski")
```

Funkce *MySQLdb.connect* se připojí k databázi. Toto spojení přiřadíme proměnné a dále budeme s tímto připojením pracovat přes tuto proměnnou. Pro navázání spojení s databází je nutné specifikovat další údaje. Do proměnné **host** přiřadíme řetězec, který bude obsahovat název hostitelského serveru. Proměnné **user** zadáme název uživatele, **passwd** bude obsahovat heslo uživatele a do proměnné **db** zadáme název databáze, ve které jsou data, se kterými chceme pracovat. Připojení k databázi je opatřeno výjimkou, kdyby se k databázi nepovedlo připojit, vyvolá se výjimka a odešle chybové hlášení.

*cur = db.cursor()* spouští kurzor, je to nutné pro další příkazy. Funkce *cur.execute(q)* je jedna z těch, co potřebují pro svoji práci kurzor. Tímto příkazem se odesílá proměnná *q* do databázového systému, který ji vyhodnotí stejně, jako bychom tento příkaz po patřičném otevření dané databáze napsali v systému phpMyAdmin. Funkcí *db.commit()* se aktivuje příkaz vložený do databázového systému funkcí *cur.execute(q)*, to znamená, že se až po příkazu *db.commit()* vložený kód vykoná. Funkce *db.rollback()* zruší to, co bylo odesláno *cur.execute(q)*. Je to užitečné použít v podmínce, když se zjistí chyba, například pokud v proměnné *q*, která byla odeslána na MySQL server ke zpracování, byl chybný příkaz.

Ukládání dat probíhá v cyklu, který končí posledním řetězcem v seznamu impulzů a seznamu času impulzů. Každým proběhlým cyklem se uloží jeden řetězec ze seznamu impulzů a jeden ze seznamu času impulzů, tím je zajištěno, že každému impulzu bude odpovídat správný čas. Za ukládáním do databáze bude kus kódu, který zajistí mazání dat. Mazat se budou data, která jsou starší více než 48 hodin, tím se zajistí neustálá aktualizace dat v databázi. Na konci skriptu bude funkce, která pošle po sériové komunikaci zařízení znak *,q'*, který toto zařízení vyhodnotí a vymaže paměť EEPROM.

Tím se zajistí, že po každém spuštění skriptu se stará data odstraní a nahradí se novými. Z toho plyne, že v databázi budou data stará 48 hodin od posledního spuštění skriptu a bude jich stále stejně, pokud bude měření probíhat více než 48 hodin. [4].

## 2.6 Synchronizace času

Hodiny uvnitř zařízení jsou dostatečně přesné, ovšem i tak je nutné je synchronizovat. Synchronizovány jsou podle hodin počítače. K synchronizaci vnitřních hodin byl vytvořen skript též v jazyce Python. Pro práci s časem byla použita knihovna *time*. Z této knihovny byla použita funkce *strftime*. Použití této funkce: `cas = time.strftime("%Y-%m-%d %H:%M:%S")`

Do proměnné *cas* se uloží aktuální čas jako řetězec ve formátu, který určuje posloupnost znaků v uvozovkách.

%Y Rok se stoletím jako desítkové číslo  
%m měsíc jako desítkové číslo [01, 12]  
%d den jako desítkové číslo [01, 31]  
%H hodina jako desítkové číslo [00, 23]  
%M minuta jako desítkové číslo [00, 59]  
%S Sekunda jako desítkové číslo [00, 61]

První číslo v hranatých závorkách znamená nejmenší cifru, druhé pak nejvyšší. U sekundy je nejvyšší číslo opravdu 61, takováto sekunda opravdu ve vzácných případech může nastat například jako přestupná sekunda. Z řetězce uloženému v *cas* vložíme jednotlivé cifry hodin, minut, dnů a měsíců do samostatných proměnných. Po sériovém portu budou posílány jednotlivě ve formátu: HHMMDDMM

Se sériovým portem budeme pracovat stejně, jako jsme s ním pracovali v předešlém skriptu. Nejprve je nutné zařízení poslat po sériovém portu znak ‚s‘, poté bude přístroj čekat na příjem osmi cifer, podle nichž nastaví svůj vnitřní čas.

Pokud zařízení pošleme špatný symbol mezi osmi znaky, na které čeká, celé zařízení bude signalizovat chybu a přestane fungovat. Nejprve se čas zařízení skriptem nedařilo nastavit, šlo to pouze, když po sériovém portu byly posílány znaky zadávané z klávesnice. Tento problém nakonec spočíval v tom, že znaky byly programem na sériový kanál posílány příliš rychle za sebou. Z klávesnice se nikdy nepodařilo dva znaky vyvolit tak rychle, aby došlo ke kolizi. Zařízení nemá hardwarové řešení pro příliš rychlé posílání dat po komunikaci, jako je například přímé uložení přijatých znaků do paměti FIFO, ze které by si bylo možné data postupně vyčíst. Proto je nutné kolizi zabránit softwarově. K tomuto ošetření byla použita funkce *sleep(argument)* z balíčku *time*, která pozastaví program. Za argument dosazujeme číslo, funkce ho chápe v sekundách, v našem případě stačilo pozastavit program po každém odeslaném znaku na 0.01 sekundy [5].

### 3 Umístění dat z měření na internet

Úpravy dat na počítači pomocí skriptů a umístění do databáze jsme vykonávali za účelem, aby bylo možné je umístit na internet pro širokou veřejnost, proto bude text internetové stránky anglicky. Umístění dat na internetovou stránku provede skript napsaný v PHP, text na internetové stránce generovaný tímto skriptem bude anglicky. PHP jsme si vybrali z důvodu širokého rozšíření PHP, pro tento jazyk existuje na internetu mnoho návodů, je zdarma a snadno dostupný. Kód PHP je vložen v kódu HTML. Překladač pozná, že se pracuje s kódem PHP, když narazí na příkaz `<?php`, práce s kódem PHP končí příkazem `?>`. Všechny webové prohlížeče, jako je například Google Chrome nebo Internet Explorer umí s produktem kódu PHP pracovat. Produkt kódu proto, jelikož server, na kterém je spuštěn skript PHP, zpracovává data od uživatele, který pomocí webového prohlížeče a internetového připojení na server pošle požadavek, skript na serveru vygeneruje kód a přepoše ho uživateli. Internetový prohlížeč pak tento kód již jen zobrazí.

## 3.1 Přístup k datům v databázi

Abychom mohli data z měření prezentovat na internetu, musíme definovat, odkud data vzít. I zde existuje podobná funkce, kterou jsme používali ve skriptu, který byl napsán v Pythonu. Ukázka funkcí:

```
mysql_connect (DB_HOSTITEL, DB_UZIVATEL, DB_HESLO)
mysql_select_db (DB_NAZEVDATABAZE)
```

První řádek definuje hostitele a uživatele, druhý řádek otvírá databázi, se kterou chceme pracovat. V takto otevřené databázi je nyní možné přistupovat k datům uloženým v libovolné tabulce.

## 3.2 Výběr a příprava dat pro graf

Po otevření databáze nyní můžeme přistupovat k datům z přístroje, která máme uložená v tabulce. Na internetové stránce si bude moci návštěvník vybrat, jestli bude chtít data včerejší, nebo dnešní.

```
$qr = mysql_query("SELECT * FROM detektor WHERE DATE(tstmp) !=
CURDATE() ORDER BY tstmp DESC")
```

Funkce *mysql\_query* odešle databázi požadavek, v tomto případě vybere z tabulky *detektor* data starší než dnešní. Nyní je nutné tato data zpracovat, aby bylo možné z nich vytvořit graf a tabulku. Do cyklu *while* vložíme podmínku *\$row = mysql\_fetch\_array(\$qr)*, funkce *mysql\_fetch\_array* čte z tabulky jednotlivé řádky, a ty vloží do proměnné *row*, když přečte všechny řádky, vrátí nulu, tím se ukončí cyklus. V tomto cyklu rozdělíme řádky s časy a impulzy na pole impulzů, pole času (hodin, minut a sekund) a pole hodin. Pole hodin vytvoříme funkcí *frachour*, která převede čas pouze na hodiny, hodiny budou v desítkové soustavě a budou mít desetinná místa. Pole impulzů bude společné jak grafu, tak tabulce, v těchto dvou formách si je návštěvník webové stránky bude moci přečíst. Pole času (hodina, minuta a sekunda) bude použito v tabulce a pole hodin v grafu.

Dále z prvního řetězce, který přečteme z databáze, použijeme rok, měsíc a den. Tyto údaje dále použijeme, budou sloužit pro určení období, ze kterého data pochází.

### 3.3 Tvorba grafu

Graf naměřených hodnot na první pohled řekne o měření víc, než pouhá data, která jsou umístěna v tabulce, a daleko více zaujme. Z těchto důvodů byl graf vytvořen. Tvorba grafu je v PHP přímo možná, ovšem má své nevýhody. Mezi tyto nevýhody patří dlouhý a poměrně složitý kód, který vytvoří jednoduchý graf, ve kterém budou vykresleny hodnoty, vyneseny osy a názvy os.

Další možností, která je na internetových stránkách používána a to nejen pro grafy, je vytvořit graf pomocí Java skriptu. Vytvoření Java skriptu, který by vytvořil graf, není také zrovna snadné, ale ani nutné. Na tvorbu grafů jsou na internetu již vytvořeny programy, na které z našeho serveru pošleme data, která jiný server zpracuje a odešle vykreslený graf. V našem projektu jsme zvolili možnost použití již vytvořeného grafu, který je dostupný na internetové síti. Zvolili jsme Google API, tuto aplikaci provozuje společnost Google a je zdarma. Abychom s touto aplikací mohli pracovat, musíme do záhlaví vložit:

```
<script type='text/javascript' src='https://www.google.com/jsapi'></script>
```

Tím, že zadáme příkaz `<script type="text/javascript">` překladač pozná, že odtud budou psány příkazy, které definují, jak má graf vypadat, a které grafu dodají hodnoty. Práce s tímto nastavením grafu se ukončuje příkazem `</script>`. Tento graf má obrovskou výhodu, jelikož není jako obrázek, ale pokud kurzorem myši najedeme na vynesenu hodnotu v grafu, zobrazí se, kde leží tato hodnota na obou osách. To pro návštěvníka bude velice užitečné vodítko pro představu měřených hodnot. Uživatel si na internetové stránce bude moci přepínat mezi dvěma grafy, jeden bude s dnešními a druhý se včerejšími hodnotami [6].



## 3.4 Tvorba tabulky

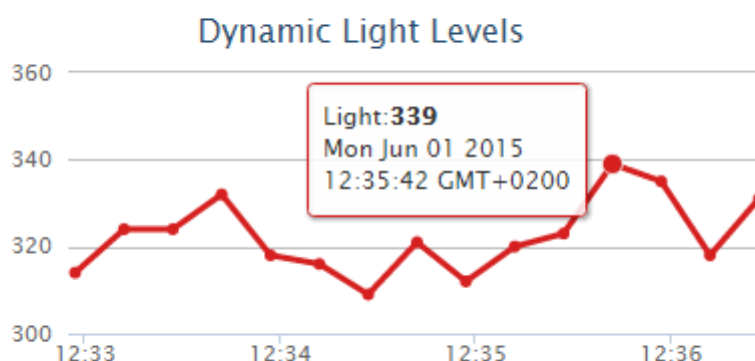
Přesto, že je možné z grafu zjistit přesně množství impulzů v určitý čas, i tak je vhodné tento graf doplnit tabulkou. Jednotlivá data v grafu jsou u sebe velice blízko a tak nalezení přesně jedné určité hodnoty je téměř nemožné. Jednodušší z programátorského hlediska bylo umístit tabulku před graf, ovšem pak by byl graf na konci stránky dole pod značným množstvím dat a návštěvník by ho mohl snadno přehlédnout, proto byla tabulka s daty nakonec umístěna až za graf.

## 3.5 Další alternativy sdílení dat

Další alternativou sdílení dat by bylo vložit data na internetovou stránku ThingSpeak. Tako internetová stránka je zdarma a je dostupná z: <https://thingspeak.com/> <https://thingspeak.com/>

Tato internetová stránka je přímo určena pro podobná měření. Pro práci v této aplikaci je nutné vytvořit uživatele. Po přihlášení je možné vytvořit kanál, do kanálu se dají uložit data, každý kanál má soukromou část a část pro veřejnost. Do soukromé části máte přístup pouze vy, ve veřejné části jsou prezentována data například v podobě grafů nebo popisků. Pro ukládání dat je zde vytvořena databáze, data z databáze se na kanálu aktualizují každých 15 sekund. Pro vizualizaci dat je zde více typů grafů, mezi tyto typy patří: line, bar, column, spline, a step.

Příklad Graf:



Obr. č. 4. Graf typu line generovaný systémem ThingSpeak.

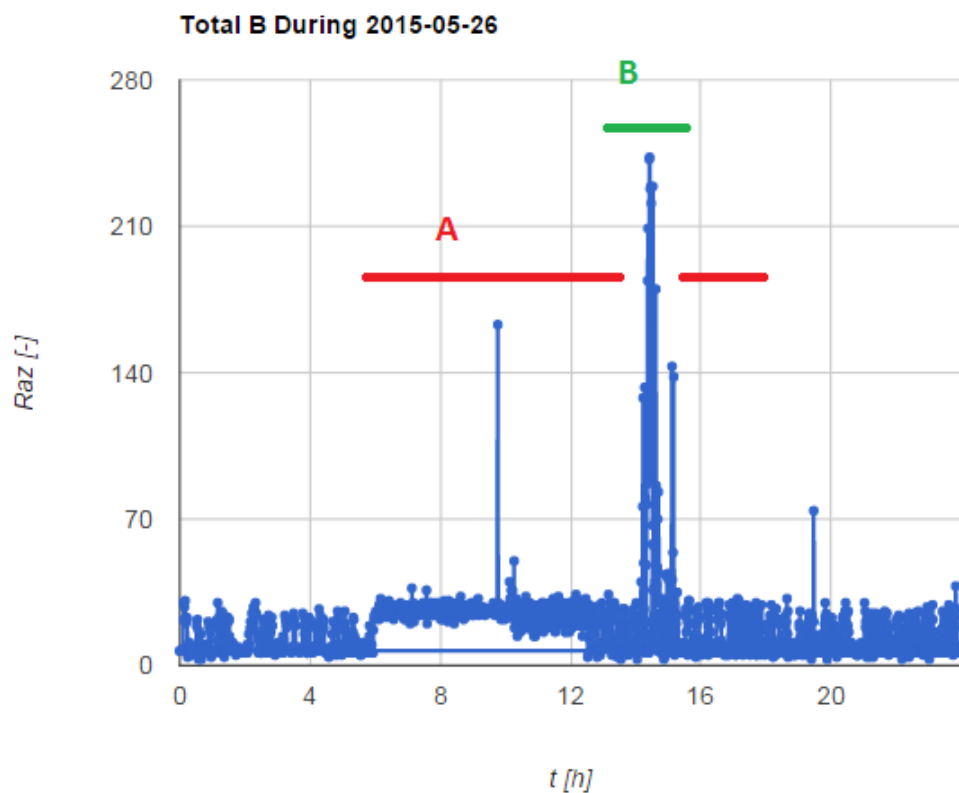
Graf typu line poskytuje základní vlastnosti, jako je popis os titul atd., najetím kurzoru myši na hodnotu vynesenu v grafu se zobrací informace o poloze této hodnoty na obou osách. Tento typ grafu je srovnatelný s grafem použitým v naší práci.

Pro umístění hodnot z měření jsme zvolili vlastní stránky hlavně pro nezávislost a možnost využití školního serveru (počítače). Toto měření můžeme umístit na internetové stránky z jakéhokoli počítače a nehrozí riziko, že stránka zanikne a bude nutné hledat jiné východisko. V budoucnu by ovšem stálo za zvážení umístit data z měření i zde [7].

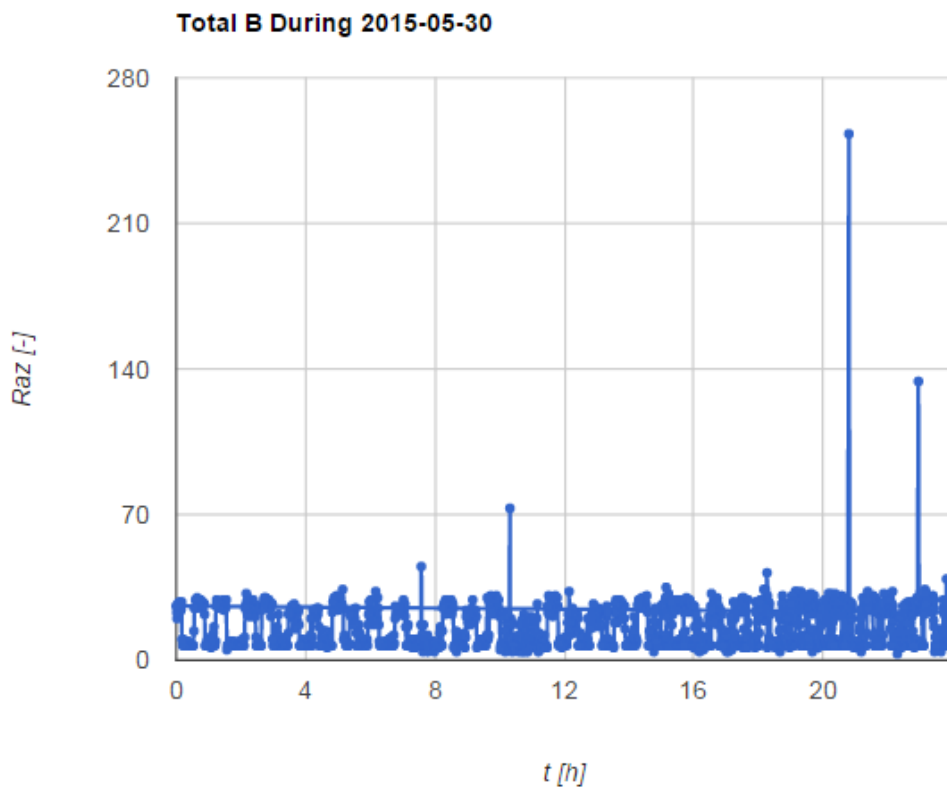
## 4 Měření elektromagnetické aktivity

Měření elektromagnetické aktivity probíhalo v budově školy na sedmém patře, zařízení bylo umístěno vedle počítače, blízko informačních kabelů, které naneštěstí byli zdrojem měřeného signálu.

Pro měření elektromagnetické aktivity lidské činnosti a přírodních dějů v širším okolí by bylo nutné zařízení umístit na střechu budovy z dosahu drobného rušení, které je produkováno v místnostech školy. Grafy pochází z vytvořených internetových stránek.



Obr. č. 5. Elektromagnetická aktivita ve všední den.



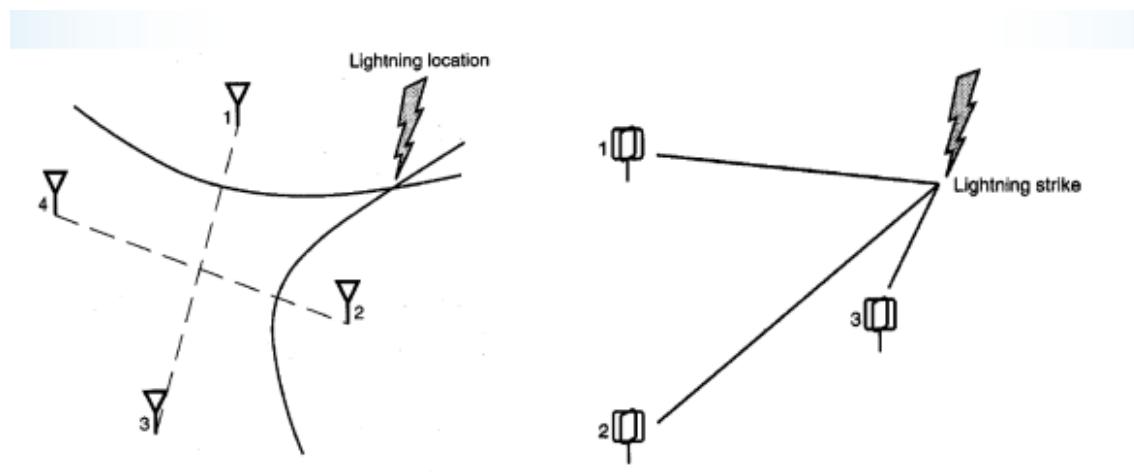
Obr. č. 6. Elektromagnetická aktivita o víkendu.

V grafu 2 vidíme konstantní elektromagnetickou aktivitu způsobenou aktivitou komunikace v kabelech vedených blízko zařízení kromě vyznačené oblasti A a oblasti B. Zvýšená aktivita v oblasti A, která trvá v rozmezí šesti až přibližně sedmnácti hodin je způsobena činností pracovníků školy. Oblast B je kratší, velký nárůst elektromagnetické aktivity způsobila zapnutá obrazovka blízkého počítače, která je zdrojem záření v naší frekvenční oblasti. V grafu 3 na rozdíl od grafu 2 k žádnému výraznému růstu aktivity průběhem dne nedochází. Tento graf byl pořízen z hodnot naměřených o víkendu, kdy ve škole nejsou pracovníci a komunikace je nižší, tomu odpovídá i nižší elektromagnetická aktivita. Lokální špičky s velkými impulzovými hodnotami budou nejspíš způsobeny zdroji vně budovy.

## 5 Podobná zařízení fungující v ČR

Už v druhé polovině minulého století se začaly vyvíjet systémy, které by sledovaly blesky na dálku. Tyto systémy přispěly k porozumění dějů, které při bouřkové činnosti probíhají. Je více metod, které se používají pro systémy pozorující tyto děje. Při bleskovém výboji tečou velké proudy, tyto proudy generují elektromagnetické pole. Toto pole se šíří do všech směrů rychlostí světla jako vlnění. Vlnění je nejintenzivnější ve frekvenčním spektru "

10 – 300kHz. Těchto principů využívá i detekční systém. Nejčastější metody určování jsou: určení směru zdroje magnetického pole a metoda stanovení času příchodu. První metoda z rozdílu časů, kdy zachytí tuto činnost, určí polohu výboje. Pro určení polohy jsou třeba čtyři stanice, k přesné synchronizaci časů je použit systém GPS. Druhá metoda určení místa zdroje blesku využívá Faradayova indukčního zákona. Pro stanovení polaritu a místa výboje je třeba alespoň třech detekčních stanic.



Obr. č. 7. Metody určení polohy výboje (vlevo metoda stanovení času příchodu blesků a vpravo metoda určení směru zdroje magnetického pole).

### 5.1 Systémy pro detekci blesků v Čechách

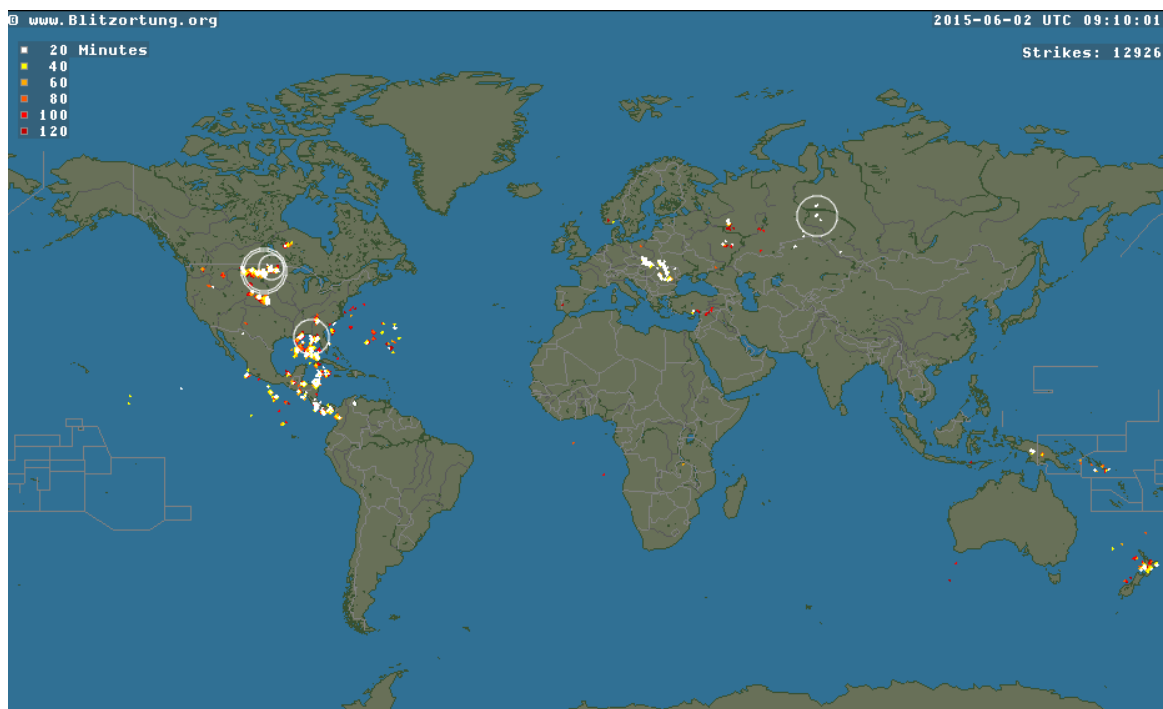
Mezi systémy pro detekci blesků patří CELDN, tento systém je součástí evropské sítě EUCLID. Data z tohoto systému přijímá ČHMU a data z tohoto systému jsou dostupná na stránkách In-počasí nebo ČHMU. Přesnost tohoto systému na 500 metrů je s účinností 90% pro blesky mezi oblaky a zemí, pro blesky mezi oblaky je tato účinnost pouze 30%.

Další systém je LINET a používá stejných metod jako CELDN. Byl vytvořen na Mnichovské univerzitě, čidla jsou od sebe vzdálená 200 až 250 kilometrů a některá jsou i v Čechách. Přesnost určení blesku mezi zemí a mrakem je 150 metrů a mezi mraky 100 metrů, účinnost se pohybuje mezi 60 a 90 procenty. Data z této sítě ovšem nejsou volně dostupná.



Obr. č. 8. Detektor sítě LINET na střeše budovy MFF UK v Praze.

Částečně k dispozici jsou data z internetových stránek [Blitzortung.org](http://Blitzortung.org). Tento systém není jen pro Evropu, ale detekční stanice jsou například i v Americe. Síť detektorů používá metodu stanovení času příchodu a metodu stanovení času příchodu skupin. Jedná se o komunitu provozovatelů detekčních stanic, která svá data posílají do centrálního serveru, kde se zpracují. Přidat se může kdokoli, potřebuje ovšem zakoupit svůj přijímač, tím získá přístup k datům a samozřejmě může posílat i svá data centrálnímu serveru [8].



Obr. č. 9. Vizualizace detekce blesků z internetových stránek Blitzortung.org.

## Závěr

Cílem této bakalářské práce bylo zpracování dat měření a ukládání těchto dat. Uložená data měla být následně umístěna na internet.

V první části (kapitola 1) je uveden princip fungování zařízení a popis jednotlivých částí. Hlavní bod této kapitoly se věnuje sériové komunikaci, jelikož ta činila zařízení od jeho vzniku problémy, tyto problémy jsou částečně vyřešeny, ale dosud ne v plném rozsahu. Instalací posledního typu komunikace RS-485 se ovšem nakonec vyřeší i problém přenosu na větší vzdálenost.

Byly vytvořeny dva skripty v Pythonu. `Zap_databaze.py` (příloha 1) a `synchronizace_casu.py` (příloha 2). Pro data ze zařízení byla vytvořena databáze a v ní tabulka. Data měřená zařízením a uložená ve vnitřní paměti je pomocí skriptu `Zap_databaze.py` možné přemístit přes sériovou komunikaci, data jsou tímto skriptem upravena a uložena do databáze a souboru. Dále tento skript data, která ze zařízení přijal, zpracuje, a poté v zařízení smaže. Čas zařízení synchronizuje skript `synchronizace_casu.py`.

Pro data v databázi byly vytvořeny internetové stránky. Skripty, které tyto internetové stránky vytvoří, se nazývají `daytabF.php` (příloha 3) a `yesterdayF.php`. `daytabF.php` vytvoří stránku pro dnešní data a `yesterdayF.php` pro včerejší. Tyto skripty pracují ještě s funkcemi, které jsou ve skriptu `libmag.php` (příloha 4) a `header.php` (příloha 5).

Poslední část práce se věnuje datům, která byla získána z přístroje. Tyto data jsou výsledkem měření přístroje. Na naměřených datech je vidět elektromagnetická aktivita v měřeném místě.



## Použitá literatura dle ISO 690-2

- 1) FISCHER, J. *Aplikace vestavěných systémů* [online] Praha: ČVUT, 2013  
URL:  
<[http://measure.feld.cvut.cz/system/files/files/cs/vyuka/predmety/A4M38AVS/Pred\\_AV\\_S\\_2013\\_UART\\_RS232.pdf](http://measure.feld.cvut.cz/system/files/files/cs/vyuka/predmety/A4M38AVS/Pred_AV_S_2013_UART_RS232.pdf)>
- 2) SiteGround. *Tutorial PhpMyAdmin* [online]. © 2004-2015 [cit. 2015-4-24]. URL:  
<<https://www.siteground.com/tutorials/phpmyadmin/>>
- 3) LIECHTI, Chris. *Documentation pySerial's* [online]. © 2001-2013 [cit. 2015-5-8].  
URL: <<http://pyserial.sourceforge.net/>>
- 4) DUSTMAN, Andy. *MySQLdb: a Python interface for MySQL* [online].  
[cit. 2015-5-12]. URL:  
<<http://mysql-python.sourceforge.net/MySQLdb.html#introduction>>
- 5) Python Software Foundation. *Dokumentace balíčku time* [online]. May 23, 2015 [cit. 2015-5-30].  
URL: <<https://docs.python.org/2/library/time.html>>
- 6) Google Developers. *Google Loader Developer's Guide* [online]. January 8, 2014  
[cit. 2015-5-31]  
URL: <<https://developers.google.com/loader/>>
- 7) ioBridge. *ThingSpeak* [online]. © 2015 [cit. 2015-6-1] URL: <<https://thingspeak.com>>
- 8) InMeteo, s.r.o. *In-počasí Teorie detekce blesků* [online]. © 2015 [cit. 2015-6-1] URL:  
<<http://www.in-pocasi.cz/clanky/teorie/detekce-blesku-18.7.2014/>>

## Seznam obrázků

Obr. č. 1 Schéma analogové části; převzato z originální dokumentace k přístroji

Obr. č. 2 Přenosová komunikace RS-232.; převzato [1]

Obr. č. 3 Přenosová komunikace RS-485.; převzato [1]

Obr. č. 4 Graf typu line generovaný systémem ThingSpeak.; převzato [6]

Obr. č. 5 Elektromagnetická aktivita ve všední den.

Obr. č. 6 Elektromagnetická aktivita o víkendu.

Obr. č. 7 Metody určení polohy výboje.; převzato z MacGorman, D. a Rust, W. *The Electrical Nature of Storms*. Oxford University Press, New York, 1998.

Obr. č. 8 Detektor sítě LINET na střeše budovy MFF UK v Praze.; převzato [8]

Obr. č. 9 Vizualizace detekce blesků z internetových stránek Blitzortung.org.; převzato z Blitzortung.org [online]. © 2003-2015 [cit. 2015-6-2]. URL: <<http://www.blitzortung.org/Webpages/index.php?lang=en>>

# Příloha

Příloha č. 1 Detektor elektromagnetické aktivity

Elektronická příloha č. 2 Skript v souboru synchronizace\_casu.py psaný v Python

Elektronická příloha č. 3 Skript v souboru synchronizace\_casu.py psaný v Python

Elektronická příloha č. 4 Skript v souboru daytabF.php psaný v PHP

Elektronická příloha č. 5 Skript v souboru header.php psaný v PHP

Elektronická příloha č. 6 Skript v souboru libmag.php psaný v PHP

Příloha 1 – Detektor elektromagnetické aktivity

