

Západočeská univerzita v Plzni

Fakulta aplikovaných věd

Katedra informatiky a výpočetní techniky

Bakalářská práce

Monitorování prostředí založené na Arduino UNO

Plzeň, 2016

David Studnička

Prohlášení

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne 20. června 2016

David Studnička

Abstract

Environment monitoring based on the Arduino Uno

The purpose of this thesis is to describe a creation of the monitoring system based on Arduino UNO and some of available modules. The text discusses use cases for Arduino in monitoring environment and describes concepts and realisations of three parts of created system. It concerns applications for monitoring station, central station and PC. Monitoring station captures scene properties, sends them to central station and reacts to events in its surroundings or commands from central station. Central station manages communication between monitoring station and PC's application. PC application, through central station, configures and manages whole system and also stores and visualizes data. Created system can be used for example by huntsmen for tracking wildlife or by plantsmen for monitoring temperatures and air humidity.

Abstrakt

Monitorování prostředí založené na Arduino UNO

Účelem této práce je popsat tvorbu monitorovacího systému založeného na Arduino UNO a některých dostupných modulech. V textu je diskutováno možné využití Arduina pro monitorování prostředí a popsány návrhy a realizace tří částí vytvořeného systému. Jedná se o aplikaci pro monitorovací stanici, aplikaci pro centrální stanici a aplikaci pro PC. Monitorovací stanice snímá vlastnosti okolí, odesílá je centrální stanici a reaguje na události ve svém okolí a na příkazy od centrální stanice. Centrální stanice zprostředkovává komunikaci mezi monitorovacími stanicemi a PC aplikací. PC aplikace prostřednictvím centrální stanice konfiguruje a spravuje celý systém a dále uchovává a vizualizuje data. Vytvořený systém mohou využít např. myslivci pro sledování výskytu zvěře nebo pěstitelé pro sledování optimálních hodnot teploty a vlhkosti vzduchu.

Obsah

1 Úvod.....	1
1.1 Princip monitorovacího systému	2
2 Základní specifikace součástí	4
2.1 Arduino.....	4
2.2 Modul GPRS/GSM SIM900	6
2.3 Senzor PIR	6
2.4 Senzor DHT	7
2.5 Kamerový modul	7
2.5.1 Kamera JPEG TTL.....	7
2.5.2 Kamera OV7670	8
2.6 Modul SD/MicroSD	9
3 Návrh monitorovacího systému	10
3.1 Monitorovací stanice.....	11
3.2 Centrální stanice	12
3.3 Aplikace v PC	12
3.4 Propojení součástí k Arduino	14
4 Realizace monitorovacího systému.....	16
4.1 Program pro centrální stanici.....	16
4.2 Program pro monitorovací stanici.....	18
4.3 Program pro PC aplikaci	20
4.4 Odesílání a přijímání SMS zpráv	23
4.5 Přerušení.....	26
4.6 Záznamy vytvářené monitorovací stanicí.....	27
4.7 Vizualizace dat v PC aplikaci	28
4.8 Komunikační protokol.....	29

4.8.1 Komunikace PC aplikace s monitorovací stanicí.....	29
4.8.2 Komunikace monitorovací stanice s PC aplikací.....	31
4.8.3 Názorná ukázka komunikace.....	32
5 Diskuze	33
5.1 Problémy návrhu.....	33
5.1.1 Nadměrná vlhkost.....	33
5.1.2 Porušené snímky a prodleva snímání	33
5.1.4 Nedostatek paměti a stabilita systému	34
5.1.5 GSM modul	34
5.2 Testování systému	35
5.3 Získané zkušenosti.....	35
5.4 Cena výsledného systému a jiné alternativy.....	35
6 Závěr.....	37
6.1 Možné budoucí práce.....	37
Reference.....	39
A Uživatelská příručka	41
A1 Přeložení a spuštění aplikace	41
A2 Ovládání aplikace	41
B Obsah přiloženého média	44

1 Úvod

Cílem této bakalářské práce je vytvořit prototyp přenosného monitorovacího systému na bázi Arduina. Náklady na běžné monitorovací systémy bývají velmi vysoké a my je zkusíme co nejvíce minimalizovat právě použitím Arduina a jeho modulů. Systém bude obsahovat monitorovací stanice, které mohou být umístěny například v lese, na louce, ale i uprostřed domácností. Cílem monitorovací stanice bude snímat požadované veličiny, kterými jsou v našem případě teplota a vlhkost vzduchu a pohyb. Naměřené údaje budou posílány centrálnímu prvku, který bude sloužit jen jako komunikační prostředek mezi PC a monitorovací stanicí.

V monitorovacích stanicích i centrální stanici bude jako hlavní řídicí prvek použito Arduino UNO R3¹, které bude řídit rozšiřující modul pro přenos dat, PIR senzor pro detekci pohybu a snímač teploty a vlhkosti. Jelikož monitorovací stanice může být umístěna na místech špatně pokrytých mobilní sítí (např. v lese nebo u rybníka pro sledování pytláků), tak budou také zmíněny dostupné moduly umožňující přenos dat. Další část systému bude tvořit aplikace v PC, která všechna naměřená data zpracuje a zobrazí ve formě grafů. Aplikace také umožní prostřednictvím GUI² [1] nastavit monitorovací stanice podle požadavků uživatele (např. aktivace/deaktivace připojených senzorů, změna režimu zaznamenávání údajů apod.).

Následující kapitola 2 obsahuje informace o Arduinu a rozšiřujících modulech, které byly v rámci této práce prozkoumány. Kapitola 3 obsahuje informace o návrhu systému a funkcích monitorovací stanice, centrální stanice a PC aplikace. V kapitole 4 je vysvětlena realizace systému a v kapitole 5 je provedena diskuze problémů, možných vylepšení apod. Závěrečná kapitola 6 obsahuje zhodnocení vytvořeného systému a možné budoucí práce.

¹ Popis a specifikace produktu - <http://www.arduino.cc/en/Main/ArduinoBoardUno>

² GUI (grafické uživatelské prostředí) – Je množina grafických prvků sloužících jako pracovní rozhraní mezi uživatelem a aplikací.

1.1 Princip monitorovacího systému

Monitorovací systémy se dnes používají běžně. S jejich pomocí může zaměstnavatel neustále sledovat provoz na pracovišti a mít dokonalý přehled o tom, co se kde děje. Nemusí ovšem sloužit jen jako prostředek ke sledování zaměstnanců, ale také jako prostředek pro uchování pocitu bezpečí (tj. zabezpečovací systémy [2]) a to v podobě různých alarmů, které např. reagují na pohyb neoprávněné osoby (např. v noci, kdy je pracoviště nikým nehlídané a máme v něm důležité a cenné předměty). Reakce na pohyb neoprávněné osoby se může lišit podle potřeb zaměstnavatele. Může to být např. odeslaná SMS zpráva s informací o detekci pohybu, uložení snímku detekované osoby anebo rovnou upozornění policie, aby mohla co nejdříve zasáhnout.

Ovšem monitorovacím systémem nemusíme rozumět jen nainstalovanou kameru uvnitř či zvenku budovy, která nám snímá dané okolí. Může to být také stanice, která snímá a zaznamenává veličiny, s nimiž následně pracuje, tj. analyzuje je a vytváří z nich statistiky. Jsou to např. meteorologické stanice³ pro měření meteorologických údajů, které následně slouží pro další předpověď počasí. Existuje několik typů těchto stanic, které můžeme vidět v tabulce 1.

Náš monitorovací systém bude malá meteorologické stanice, která bude zaznamenávat naměřené veličiny teploty a vlhkosti vzduchu a bude detekovat pohyb ve svém okolí. Tyto údaje budou přenášeny PC aplikaci, kde budou zpracovány a následně vizualizovány.

³ Význam a popis stanic - <http://www.meteocentrum.cz/encyklopedie/meteorologicke-stanice.php>

Tabulka 1: Typy meteorologických stanic

Synoptické	S nejrozsáhlejší působností. Snímá data v hodinových intervalech a vytvářejí synoptické mapy o aktuálním vývoji a změně počasí.
Klimatologické	Ke sledování vývoje klimatu. Snímá data pouze třikrát denně.
Srážkoměrné	Zaznamenává údaje o spadlých srážkách (sníh, déšť) a to jednou denně.
Letecké	Využívané převážně v letecké dopravě ke sledování počasí.
Silniční	Slouží k informování řidičů o aktuálním stavu vozovky. Jsou to různé silniční senzory, které jsou schopny zaznamenávat stav a teplotu vozovku.
Amatérské	Slouží převážně k amatérskému sledování meteorologických údajů a k vlastnímu užití naměřených dat.

2 Základní specifikace součástek

V této kapitole je podrobná specifikace konkrétních součástek, se kterými jsme se při práci blíže seznámili. V následující části 2.1 popíšeme Arduino UNO, jeho využití a některé vlastnosti. Funkce a možnosti využití GPRS/GSM modulu jsou v části 2.2, práce s pohybovým senzorem PIR v části 2.3 a se senzorem teploty a vlhkosti v části 2.4. V částech 2.5 a 2.6 zmíníme kamerový modul a SD modul, s nimiž jsme také pracovali, ale které jsme ve výsledném systému nepoužili.

2.1 Arduino

Arduino je Open-Hardware platforma pro jednoduchý návrh a vývoj elektronických programovatelných zařízení. K jeho používání nejsou potřeba hlubší znalosti principů elektroniky. Bez nutnosti pájení a prostřednictvím jednoduchého vývojového prostředí je vhodné pro rychlou tvorbu prototypů zařízení, hraček, ovládacích a regulačních aplikací či robotů. Díky svým vlastnostem může být také použito jako součást monitorovacího systému.

Existuje několik typů desek Arduina. Mezi nejzákladnější z nich patří Arduino UNO (viz obrázek 1), DUE, MEGA, LEONARDO, DUEMILANOVE a YÚN. Všechny tyto desky jsou založené na mikrokontrolerech ATmega⁴ od firmy Atmel a liší se většinou jen svými rozměry, velikostí paměti a počtem vstupně/výstupních pinů. My zvolili pro prototypovací a školní účely vhodné Arduino UNO (dále jen Arduino), a proto si ho popíšeme více.

Základní řídicí jednotkou Arduina je mikrokontrolér ATmega328⁵ [3]. Arduino má 6 analogových vstupů a 14 digitálních vstupně/výstupních pinů (6 z nich lze použít jako výstupy PWM⁶ [4]). Může být napájeno přes USB⁷, napájecím adaptérem nebo baterií. Deska může pracovat s napájením 6-20V, ovšem pokud je napětí menší než 7V, tak výstupní napájecí 5V pin může být nestabilní a dodávat

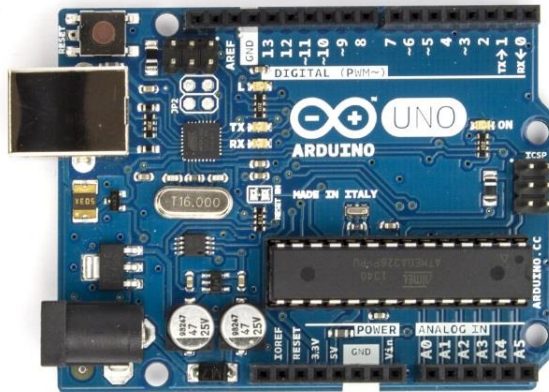
⁴ Popis a specifikace mikrokontrolérů ATmega - <http://www.atmel.com/products/microcontrollers/avr/megaavr.aspx>

⁵ Popis a specifikace mikrokontroléru ATmega328 - <http://www.atmel.com/devices/atmega328.aspx>

⁶ Pulzně šířková modulace (Pulse width modulation) – Diskrétní modulace pro přenos analogového signálu pomocí binárního signálu

⁷ USB - Univerzální sériová sběrnice pro propojení periférií s PC.

méně než 5V. Naopak pokud použijeme více než 12V, může se regulátor napětí přehřát a poškodit desku. Z těchto důvodů je doporučené napětí 7-12V. Vnitřní Flash⁸ paměť je 32KB a z toho 0,5KB používá zavaděč⁹. Dále má Arduino UNO operační SRAM¹⁰ paměť o velikosti 2KB a paměť EEPROM¹¹ o velikosti 1KB. Frekvence procesoru je 16MHz.



Obrázek 1: Arduino UNO

Pro Arduino lze vyvíjet ve vývojovém prostředí Arduino IDE¹², které je napsáno v jazyce Java [5, 6], je tedy multiplatformní (tj. využitelné ve všech operačních systémech podporujících JVM¹³) a obsahuje intuitivní ovládání a zvýraznění syntaxe kódu¹⁴. Existují také rozšíření pro vývojová prostředí Eclipse¹⁵ a NetBeans¹⁶. Samotné programování se provádí v jazyce C/C++ [7, 8]. Díky velkému množství funkcí (komunikace se vstupy/výstupy a sériovým portem, obsluha přerušení atd.) je také využívána dnes velmi rozšířená a doporučovaná

⁸ Flash paměť - Elektricky programovatelná paměť s libovolným přístupem.

⁹ Zavaděč (boot loader) – Program, který slouží k zavedení jádra jednoho z dostupných operačních systémů. Zpravidla je uložen v MBR (1. sektoru fyzického disku), nebo ve spouštěcím sektoru některého z diskových oddílů.

¹⁰ SRAM - Statická paměť, která k uchování dat nepotřebuje periodickou obnovu.

¹¹ EEPROM - Elektricky mazatelná paměť, kterou je nutno smazat před novým naprogramováním.

¹² Bližší informace o vývojovém prostředí - <http://www.arduino.cc/en/Main/Software>

¹³ JVM (Java Virtual Machine) – Je sada programů a datových struktur, která využívá virtuální stroj ke spuštění počítačových programů vytvořených v jazyce Java.

¹⁴ Zvýraznění syntaxe kódu (Syntax highlighter) – Zvýraznění programovacího kódu, které usnadňuje orientaci v kódu pomocí barev, velikostí, fontů písma.

¹⁵ Bližší informace o vývojovém prostředí - <https://eclipse.org/>

¹⁶ Bližší informace o vývojovém prostředí - <https://netbeans.org/>

knihovna Wiring¹⁷. Také existuje mnoho již vytvořených knihoven, např. pro práci s paměťovými kartami, kamerami apod., které se dají použít a jsou volně přístupné. Řešené příklady využití Arduina a dostupných knihoven a náměty pro práci s Arduinem lze nalézt např. v [9, 10, 11].

2.2 Modul GPRS/GSM SIM900

V této části popíšeme, co je GPRS/GSM a k čemu se používá. GPRS¹⁸ [12] je prostředek, kterým uživatelům mobilních telefonů umožňujeme GSM¹⁹ [13] přenos dat (hovory, SMS) a také poskytuje připojení k internetu. Idealizovaná rychlost přenosu dat se pohybuje v rozmezí 54-114 kbit/s. GSM síť, někdy také označována jako síť 2G²⁰, je mezinárodní komunikační standart pro mobilní telefony. Pro přístup k internetu musíme znát APN²¹ a uživatelské jméno a heslo, která většinou zjistíme od našeho síťového operátora.

GSM modul umožňuje mimo jiné zasílání textových SMS zpráv a uskutečnění telefonního hovoru. V našem případě pro ovládání Arduina využijeme SMS zprávy, přičemž Arduino podle nich příslušně zareaguje (např. nastaví monitorovací stanici, pošle požadovaná data apod.). V rámci práce byly prostudovány i jiné dostupné moduly pro přenos dat (např. Wi-Fi, ethernet nebo bluetooth), avšak všechny tyto moduly vyžadují v blízkosti mobilní telefon, či PC. Protože naší myšlenkou je umístění monitorovacího systému mimo dosah těchto přístrojů, tak použijeme právě modul GPRS/GSM SIM900 [14].

2.3 Senzor PIR

Senzor PIR je pasivní infračervený snímač používaný k rozpoznání pohybu v jeho okolí pomocí detekce infračerveného tepelného záření. Je velmi oblíbený a používaný díky své ceně a snadné obsluze. I přes svou jednoduchost spolehlivě slouží při odhalování pohybu osob či zvěře. Pracovní napětí tohoto senzoru se pohybuje v rozmezí 4,5-20V a odběr proudu má okolo 60uA. Samotný senzor nám

¹⁷ Bližší informace o knihovně - <http://wiring.org.co/>

¹⁸ GPRS (General Packet Radio Service)

¹⁹ GSM (Global System for Mobile Communications)

²⁰ 2G - 2. generace bezdrátové telefonní technologie

²¹ APN (Access Point Name) – Název brány mezi mobilní sítí a jinou počítačovou sítí.

umožňuje nastavit dobu, po kterou bude čidlo sepnuté (5-200 sekund) a také citlivost detekce. Dosahuje až do vzdálenosti 10 metrů a jeho úhel snímání je přibližně 110 stupňů.

2.4 Senzor DHT

DHT11²² je senzor určený k měření teploty a vlhkosti vzduchu. Jeho zapojení je velmi intuitivní. Data posílá pouze po jednom vodiči (používá vlastní protokol 1-wire²³) a zbylé dva vodiče slouží k napájení. Jeho pracovní napětí je 3,3-5V. Rozsah měření vlhkosti se pohybuje v rozmezí 20-90% a rozsah měření teploty v rozmezí 0-50 °C. Přesnost senzoru je u vlhkosti +/- 5% a u teploty +/- 2 °C, což je pro běžného uživatele dostačující. K jeho používání využijeme knihovnu DHT²⁴. Pokud bychom chtěli použít senzor s větší přesností, tak bych doporučil například DHT22, jenž je jeho nadstavbou a jeho rozsah měření vlhkosti se pohybuje v rozmezí 0-90% s přesností 2-5 % a rozsah měření teploty v rozmezí -40-50 °C s přesností 0,5 °C.

2.5 Kamerový modul

Kamerový modul měl sloužit ke snímání obrazu a při jeho změně měl být uživatel upozorněn SMS zprávou. Nicméně během testování funkčnosti kamerového modulu se vyskytly velice vážné problémy, kvůli kterým nemohla být kamera v systému použita. Tyto problémy jsou rozebrány v části 5.1.2.

Při práci jsme testovali dva typy kamer, které jsou doporučeny a určeny právě ke komunikaci s Arduinem.

2.5.1 Kamera JPEG TTL

Tento jednoduchý kamerový modul s obrazovým snímačem 6,35 mm má velké množství použití a může být začleněn do téměř každého monitorovacího či zabezpečovacího systému. Tento kamerový modul má rozměry 32 mm x 32 mm a

²² Specifikace senzoru DHT11 - <http://www.santy.cz/senzory-c24/arduino-dht11-modul-shield-mega-nano-teplota-vlhkost-mereni-i24/>

²³ Specifikace protokolu 1-wire - <http://vyvoj.hw.cz/navrh-obvodu/rozhrani/sbernice-1-wiretm.html>

²⁴ Knihovna dostupná pro senzor DHT - <https://github.com/adafruit/DHT-sensor-library>

je schopen zachytit jednotlivé snímky ve formátu JPEG²⁵. Výchozí přenosová rychlost jeho sériového portu je 38400 Bd, ale může dosáhnout rychlosti až 115200 Bd. Snímat okolí může do vzdálenosti maximálně 15 metrů s úhlem pohledu 60 stupňů. Snímky zachycuje ve velikosti VGA²⁶, QVGA²⁷ a QQVGA²⁸. Pro práci s touto kamerou lze použít dvě knihovny.

1. NewSoftSerial²⁹
2. SoftwareSerial³⁰

Mezi těmito knihovnami nejsou zásadní rozdíly. Knihovna SoftwareSerial byla vyvinuta dříve a je určena pro starší verze Arduina (do verze 1.0). U novějších verzí Arduina (od verze 1.0) tuto knihovnu nahrazuje NewSoftSerial. Arduino má vestavěnou hardwarovou podporu pro sériovou komunikaci na pinech 0 a 1. Zmíněné knihovny byly vyvinuty s cílem umožnit sériovou komunikaci na ostatních pinech Arduina.

2.5.2 Kamera OV7670

Kamerový modul s obrazovým SOC³¹ snímačem o velikosti 4,32 mm je vyroben firmou OMNIVISION³². Tento modul o rozměrech 37 mm x 43 mm je poměrně zastaralý oproti jiným standardům. Snímky zachycuje ve velikosti VGA a výstupními formáty jsou Monochrome³³, YUV/YCbCr³⁴, RGB³⁵. Úhel pohledu má 24 stupňů.

²⁵ JPEG – Formát pro ukládání fotografií využívající ztrátovou kompresi.

²⁶ VGA (Video Graphics Array) - Označuje rozlišení 640x480 pixelů.

²⁷ QVGA (Quarter Video Graphics Array) - Označuje rozlišení 320x240 pixelů.

²⁸ QQVGA (Quarter-Quarter Video Graphics Array) - Označuje rozlišení 160x120 pixelů.

²⁹ Knihovna dostupná na stránce - <https://github.com/sirleech/NewSoftSerial>

³⁰ Knihovna dostupná na stránce - <https://roboticsclub.org/redmine/projects/quadrotor/repository/revisions/master/show/arduino-1.0/libraries/SoftwareSerial>

³¹ SOC (Systém On Chip) - Má vestavěné schopnosti zpracování obrázků.

³² Stránka výrobce - <http://www.ovt.com/>

³³ Monochrome – Každý bod snímku je uložen jako hodnota z rozsahu 0 až 255 zastupující stupně šedé (0 je černá, 255 je bílá). Celý snímek je uložen jako posloupnost odpovídajících bytů.

³⁴ YUV/YCbCr – Barevný model, kde Y odpovídá množství bílého světla určité barvy, Cb a Cr jsou modrý a červený chrominanční komponent.

³⁵ RGB – Barevný model, kde je každý pixel uložen jako tři intenzity červeného, zeleného a modrého světla.

2.6 Modul SD/MicroSD

Tento modul měl původně sloužit jako prostředek pro ukládání snímků nebo naměřených veličin na SD kartu, kde by si je mohl uživatel později vyzvednout a dále s nimi pracovat. Bohužel při testování modulu se objevily potíže s nedostatkem vnitřní paměti Arduina, což se projevilo nestabilitou systému. Kvůli tomuto problému modul SD/MicroSD nebyl nakonec do systému zakomponován. Celá problematika je probrána v části 5.1.4.

K modulu SD/MicroSD existuje několik knihoven. Všechny jsou si velice podobné a liší se podporovaným souborovým systémem (FAT16/FAT36³⁶) používané SD karty. SD standard udává, že karty s kapacitou menší než 2 GB by měly být formátovány FAT16 a karty s kapacitou větší než 2 GB formátovány FAT32. Seznam knihoven seřazený podle podporovaného souborového systému a kapacity karty je k vidění v tabulce 2.

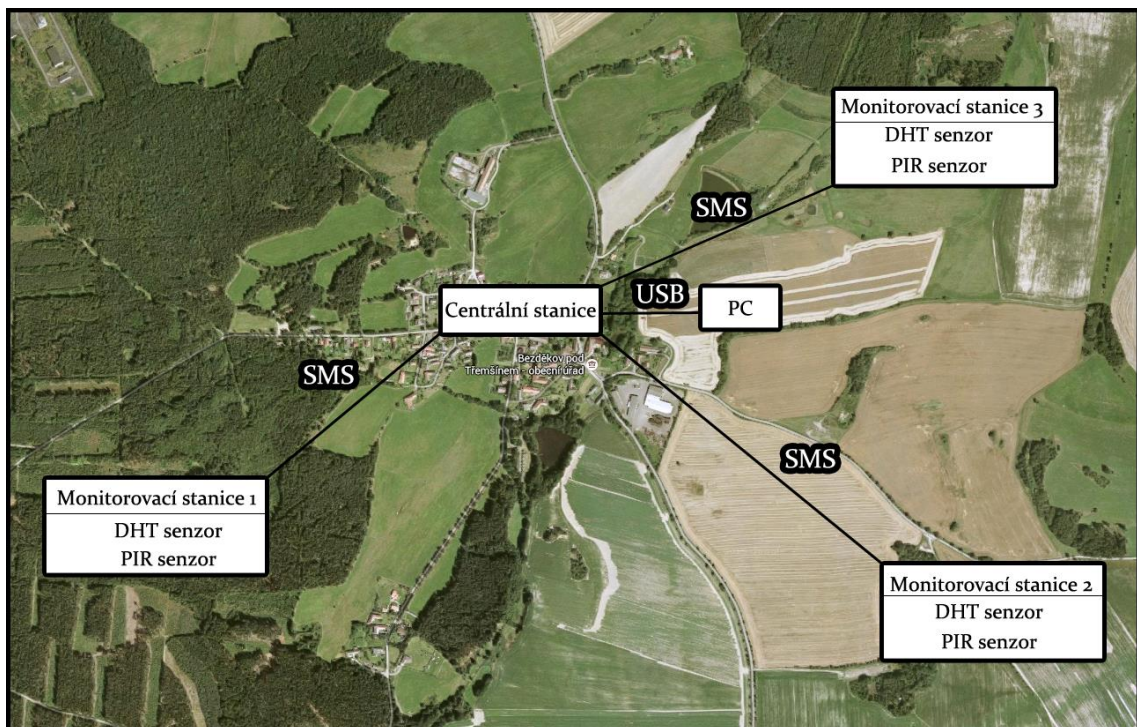
Tabulka 2: Knihovny pro práci s SD kartou

Formát FAT32 (pro karty s kapacitou větší než 2 GB)	Arduino SD
	Adafruit SD
	SDFatLib
Formát FAT16 (pro karty s kapacitou menší než 2 GB)	Fat16Lib
	Arduino-Filelogger
	tinyFat

³⁶ Popis souborového systému FAT16 a FAT36 - www.zive.cz/clanky/souborove-systemy-fat16-32x/sc-3-a-1312/default.aspx

3 Návrh monitorovacího systému

Model monitorovacího systému byl již nastíněn v kapitole 1, ale nyní bude popsán více podrobně. Monitorovací stanice umístěné např. v lese snímají své okolí a data odesílají centrální stanici využitím SMS. Centrální stanice řídí monitorovací stanice a přijatá data předává do PC. Data z centrální stanice jsou zpracována a vykreslena v PC aplikaci. Přenos dat mezi centrální stanicí a PC probíhá prostřednictvím USB kabelu. Hlavní řídicí jednotkou monitorovacích stanic i centrální stanice je Arduino. Na obrázku 2 je vidět možné nasazení celého systému v praxi.



Obrázek 2: Možné nasazení monitorovacího v praxi

V následujících částech je popsána funkčnost monitorovací stanice (viz část 3.1), centrální stanice (viz část 3.2) a PC aplikace (viz část 3.3). Propojení součástek a Arduina je v části 3.4.

3.1 Monitorovací stanice

K monitorovací stanici je pomocí drátěných propojek připojen senzor DHT a senzor PIR (popis propojení těchto senzorů s Arduinem obsahuje část 3.4). Každá monitorovací stanice se při zapnutí nahlásí (skrze centrální stanici) PC aplikaci posláním svého výchozího nastavení (viz část 4.8.2). Ve výchozím nastavení monitorovací stanice prostředí nesnímá, ale pouze čeká na příkazy uživatele, které stanici patřičně nastaví, tj. zapnou/vypnou senzor PIR a senzor DHT nebo uvedou stanici do jednoho z několika scénářů (viz dále v části 3.3). Na základě těchto scénářů odesílá monitorovací stanice záznamy (viz část 4.6) centrální stanici a ta je předává PC aplikaci.

Hlavním ovládacím prvkem monitorovací stanice je Arduino. Dále obsahuje GSM modul (příjem a odesílání SMS zpráv), senzor DHT (snímání teploty a vlhkosti vzduchu) a senzor PIR (detekce pohybu).

Funkce monitorovací stanice:

- Monitoruje své okolí.
- Komunikuje s centrální stanicí prostřednictvím SMS:
 - Odesílá data centrální stanici (dle scénáře, který zvolí uživatel, viz část 3.3).
 - Přijímá konfiguraci od centrální stanice.
- Na žádost centrální stanice mění svou softwarovou konfiguraci (tj. zapne/vypne DHT nebo PIR senzor, či změní časový interval zaznamenávání a odeslání naměřených veličin).
- Při prvotním spuštění aplikace odešle svojí konfiguraci centrální stanici.

3.2 Centrální stanice

Centrální stanice je pouze mezičlánkem mezi monitorovacími stanicemi a PC aplikací. Je přímo propojená s PC přes USB kabel, takže musí být v jeho blízkosti. Obsahuje v sobě identifikátory všech dostupných stanic, aby mohla odesílat příkazy z PC aplikace konkrétní monitorovací stanici, pro kterou je příkaz určen. Záznamy přijaté z monitorovacích stanic přeposílá přímo PC aplikaci.

Centrální stanice se skládá z ovládacího prvku Arduino a GSM modulu (příjem a odesílání SMS zpráv).

Funkce centrální stanice:

- Uchovává seznam stanic, které jsou v provozu.
- Zajišťuje komunikaci mezi monitorovacími stanicemi a PC aplikací, k čemuž využívá SMS zprávy a USB kabel:
 - Přijímá data z monitorovacích stanic (SMS zprávy).
 - Odesílá konfigurace do monitorovacích stanic (SMS zprávy).
 - Přijímá konfigurace z PC aplikace (USB kabel).
 - Odesílá data do PC aplikace (USB kabel).

3.3 Aplikace v PC

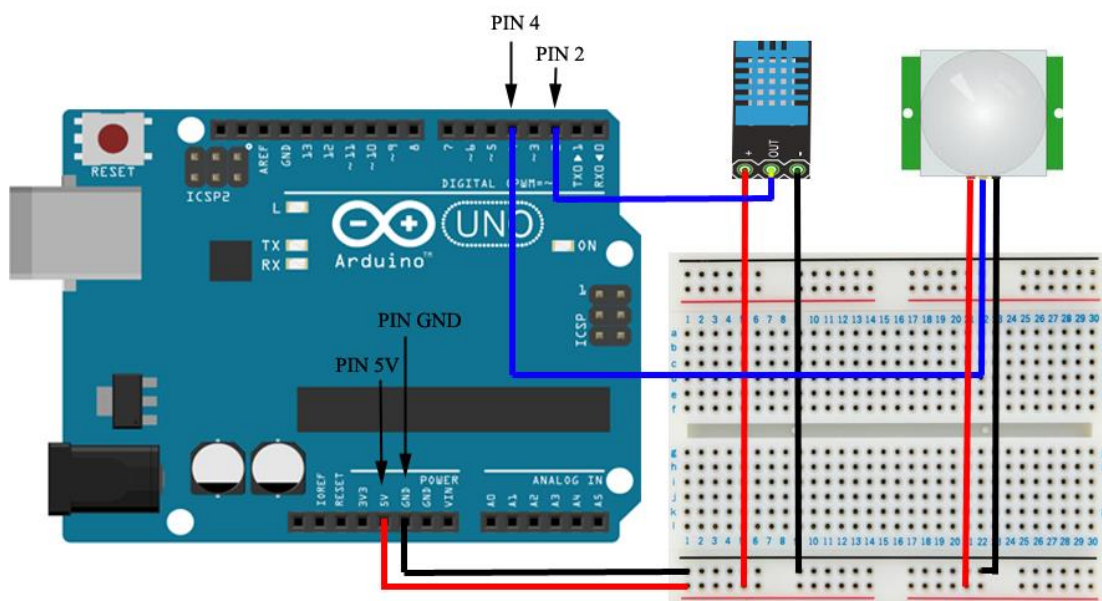
PC aplikace je vytvořena v programovacím jazyce Java a jejím hlavním úkolem je správa GUI pro snadnou konfiguraci monitorovacích stanic a vizualizaci záznamů pro koncového uživatele. Dostupné monitorovací stanice se při spuštění PC aplikace nahlásí skrze centrální stanici a tyto stanice je poté možné obsluhovat (tj. nastavení senzoru PIR a senzoru DHT, nastavení různých scénářů, které jsou popsány níže). Monitorovací stanice podle daného scénáře odesílají záznamy a PC aplikace je zpracuje.

Funkce PC aplikace:

- Zprostředkovává GUI pro konfiguraci stanic.
- Komunikuje s centrální stanicí přes sériový port (USB).
- Umožňuje změnu konfigurace monitorovacích stanic.
- Prostřednictvím centrální stanice přijímá data z monitorovacích stanic a ukládá je na úložiště zvolené v konfiguračním souboru (např. na pevný disk).
- Vizualizuje námi zvolená data v podobě grafu.
- Uživatel prostřednictvím PC aplikace řídí scénáře stanic:
 - a) Scénáře monitorovací stanice pro práci s PIR senzorem:
 - 1) Pokud je v blízkosti monitorovací stanice detekován pohyb, informace je okamžitě odeslána centrální stanici.
 - 2) Pokud je v blízkosti stanice detekován pohyb, tak je zvýšen čítač a počet zaznamenaných pohybů je odeslán centrální stanici vždy po uplynutí námi zvoleného intervalu.
 - 3) Na výzvu uživatele je zjištěna informace o aktuálním pohybu v blízkosti monitorovací stanice a následně odeslána centrální stanici.
 - b) Scénáře monitorovací stanice pro práci s DHT senzorem:
 - 1) Pokud je v blízkosti monitorovací stanice detekován pohyb, tak jsou naměřené hodnoty teploty a vlhkosti vzduchu odeslány centrální stanici. Z toho je ovšem zřejmé, že musí být aktivován PIR senzor.
 - 2) Naměřené hodnoty teploty a vlhkosti vzduchu jsou odesílány centrální stanici vždy po uplynutí námi zvoleného intervalu.
 - 3) Na výzvu uživatele jsou naměřeny hodnoty teploty a vlhkosti vzduchu a odeslány centrální stanici.

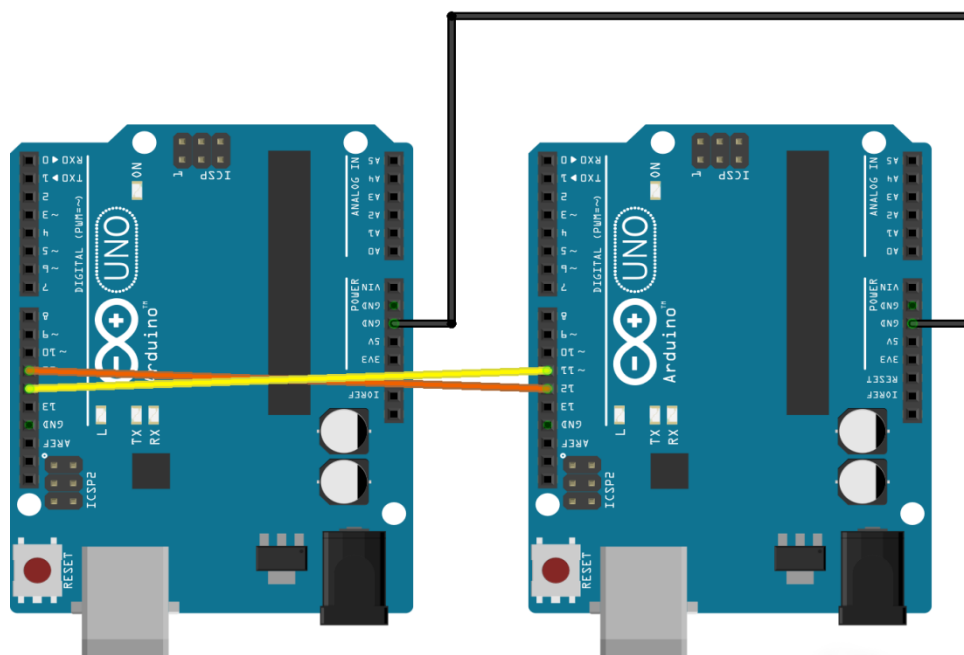
3.4 Propojení součástek k Arduino

Monitorovací stanice se skládá z Arduina, senzoru DHT, senzoru PIR, propojovacích drátů a nepájivého pole. Připojení těchto součástek je k vidění na obrázku 3. Modrou barvou jsou znázorněny datové vodiče. Datový vodič ze senzoru DHT vede na pin 2 a datový vodič ze senzoru PIR vede na pin 4. Červenou barvou jsou znázorněny napájecí vodiče. Oba senzory jsou prostřednictvím nepájivého pole připojeny na pin 5V. Černou barvou jsou znázorněny uzemňovací vodiče. Oba senzory jsou připojeny na pin GND přes nepájivé pole.



Obrázek 3: Připojení senzorů k Arduino

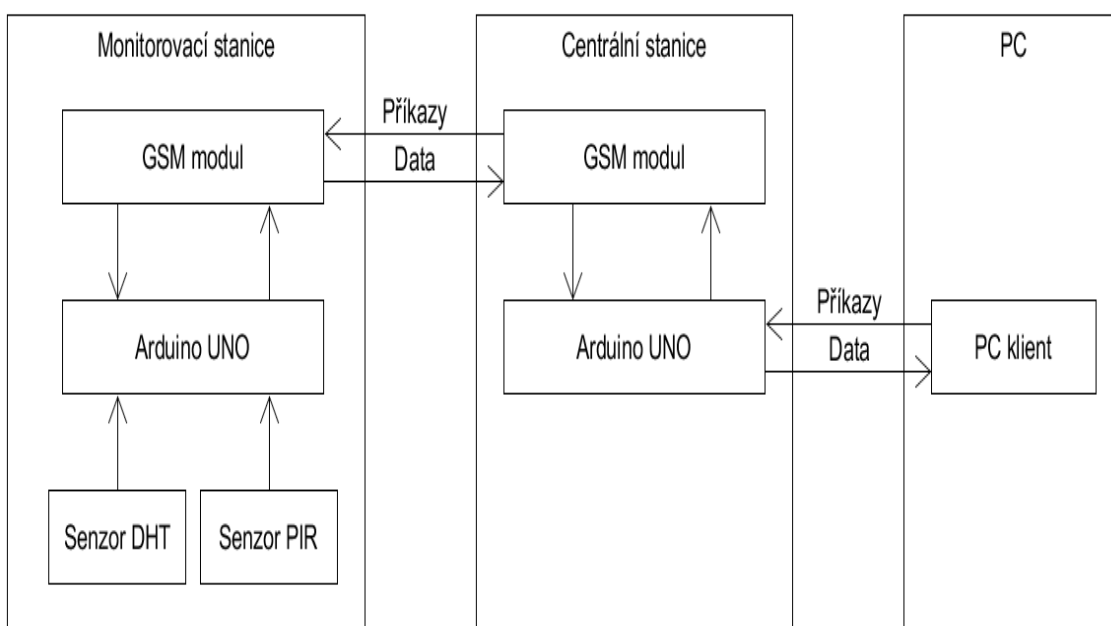
Pro usnadnění vývoje a testování systému byla Arduino monitorovací a centrální stanice propojena pomocí drátových propojek. Bylo to zejména z důvodu neustálého posílání SMS zpráv, protože pokud uživatel není držitelem neomezeného tarifu, tak se mu posílání SMS zpráv velice prodraží. Propojení můžeme vidět na obrázku 4. Barevnými vodiči jsou propojeny piny pro čtení (RX) a piny pro zápis (TX). Pin 11 (TX) na jednom zařízení je propojen s pinem 12 (RX) na druhém zařízení a opačně. Obě zařízení jsou navíc spojena ještě uzemňovacím vodičem, aby nedošlo ke zničení zařízení.



Obrázek 4: Propojení monitorovací stanice s centrální stanicí

4 Realizace monitorovacího systému

V této kapitole bude podrobně vysvětlen způsob realizace monitorovacího systému (na obrázku 5 vidíme diagram nasazení). V částech 4.1, 4.2 a 4.3 budou popsány jednotlivé programy pro centrální stanici, monitorovací stanici a PC aplikaci. Potřebná nastavení pro odesílání a přijímání SMS zpráv jsou v části 4.4. Způsobu použití přerušování v monitorovací stanici je věnována část 4.5. Dále budou v části 4.6 uvedeny typy záznamů, které jsou vytvářeny monitorovací stanicí, a jejich vizualizace v části 4.7.



Obrázek 5: Diagram nasazení monitorovacího systému

4.1 Program pro centrální stanici

Hlavní úkol centrální stanice je přeposílat data mezi monitorovacími stanicemi a PC aplikací. Pokud je komunikace s monitorovací stanicí prováděna prostřednictvím GSM modulu, tak všechna potřebná nastavení jsou vysvětlena dále, v části 4.4. Pokud je komunikace prováděna v rámci testování přes sériovou komunikaci (viz část 3.4), tak je nutné v konfiguraci programu nastavit piny pro čtení a zápis příkazem `SoftwareSerial mySerial(11, 12)`. Nyní už k funkčnosti programu. V metodě `setup()` nastavíme přenosové rychlosti dat pro sériovou komunikaci s PC (`Serial.begin(9600)`) a pro sériovou

komunikaci s druhým Arduinem (`mySerial.begin(9600)`). Funkce programu je obsažena v nekonečné smyčce `loop()`, která je rozdělena na dvě části podle toho, zda přijímáme data z PC aplikace nebo z monitorovacích stanic.

Nejdříve se zaměříme na příjem dat z PC aplikace. V tomto případě je potřeba zjistit, zda jsou na sériovém portu dostupná nějaká data. To se provádí funkcí `Serial.available()>0`, která ověřuje, zda počet bytů (znaků) dostupných pro čtení ze sériového portu je větší než 0. Pokud ano, uživatel obsluhující PC aplikaci poslal nějaký příkaz, který je potřeba uložit do proměnné typu *String*³⁷ příkazem `Serial.readString()`. Přijatý textový řetězec je poté rozdělen na několik částí metodou `parseString(String, char38, int)`. Prvním parametrem je textový řetězec, který má být rozdělen. Druhým parametrem je znak, podle kterého má být rozdělen. Posledním parametrem je index ukládané části řetězce (tzn. pokud bude metoda na rozdělení textového řetězce ve tvaru `parseString("station1;S;DHT", ';', 0)`, tak bude vracet `station1`). Když už je textový řetězec rozdělen na jednotlivé části, není problém ho přepsat cílové monitorovací stanici určené podle jména stanice. Prostřednictvím GSM modulu je příkaz odeslán `sms.SendSMS("+420775017937", buffer)`, přes sériovou komunikaci je odeslán `mySerial.write(buffer)`.

Nyní bude popsán příjem dat z monitorovacích stanic. Ten se liší podle toho, zda přijímáme data GSM modulem (reálný systém) nebo sériovou komunikací (testování). Přijatou SMS zprávu GSM modulem pouze odešleme na sériový port příkazem `Serial.print(sms_text)` a PC aplikace si text zpracuje. Při přenosu dat přes sériovou komunikaci ukládáme přijaté znaky do zvolené proměnné (`char c`) příkazem `mySerial.read()` a následně zřetěžením (funkce `concat`) těchto znaků sestojíme textový řetězec, který je opět odeslán do PC.

³⁷ Bližší informace o datové struktuře - <https://www.arduino.cc/en/Reference/String>

³⁸ Bližší informace o datové struktuře - <https://www.arduino.cc/en/Reference/Char>

4.2 Program pro monitorovací stanici

Jak je již vysvětleno v předchozí části, nejdříve je nutné nastavit piny pro přenos dat mezi Arduinem a GSM modulem, nebo Arduinem a druhým Arduinem (`SoftwareSerial mySerial(11, 12)`). Jelikož jsou k monitorovací stanici připojeny senzory DHT a PIR, je potřeba vytvořit nejenom hardwarové připojení (část 3.4), ale i v programu určit, na kterých pinech jsou součástky připojeny. Senzor DHT je připojen k Arduinu na pin 4, takže jej definujeme také v programu příkazem `#define DHTPIN 4`. Knihovna DHT umí pracovat s více typy senzorů (DHT11, DHT21, DHT22), takže je nutné definovat, jaký typ senzoru DHT používáme. Učiníme tak příkazem `#define DHTTYPE DHT11`. Jakmile je nastaven pin, na který je senzor daného typu připojen, můžeme senzor DHT inicializovat příkazem `DHT dht(DHTPIN, DHTTYPE)`. Dále definujeme příkazem `#define PIRPIN 2` připojení senzoru PIR na pin 2.

V metodě `setup()` zvolíme přenosovou rychlost dat a to jak v případě GSM modulu `gsm.begin(9600)`, tak také v případě sériové komunikace `mySerial.begin(9600)`. Nesmíme zapomenout ani na zapnutí senzoru DHT příkazem `dht.begin()` a nastavení přerušení (viz část 4.5). Každá monitorovací stanice se při svém spuštění nahlásí centrální stanici. Jelikož metoda `setup()` proběhne právě jednou při spuštění programu, tak je vhodné v této metodě odeslat svojí konfiguraci příkazem `sms.SendSMS("+420737754853",buffer)` v případě GSM modulu nebo příkazem `mySerial.write(buffer)` v případě sériové komunikace.

Příjem dat probíhá v nekonečné smyčce `loop()` a to stejným způsobem, jako příjem dat z monitorovacích stanic v centrální stanici, viz předchozí část. Avšak v tomto programu nás v přijatém příkazu nezajímá pouze název stanice, nýbrž i ostatní části příkazu z důvodu nastavení senzorů a vytváření záznamů. Sensory je možné nastavit do následujících scénářů:

- `dht_model()` - Při detekci pohybu zaznamená teplotu a vlhkost vzduchu a okamžitě je odešle centrální stanici.

- `dht_mode2()` - Zaznamenává teplotu a vlhkost vzduchu a odesílá je centrální stanici po uplynutí intervalu.
- `pir_mode1()` - Při detekci pohybu pošle okamžitě informaci o pohybu centrální stanici.
- `pir_mode2()` - Pohyby jsou detekovány využitím přerušení, viz část 4.5 a jejich počet odesílán centrální stanici po uplynutí intervalu.

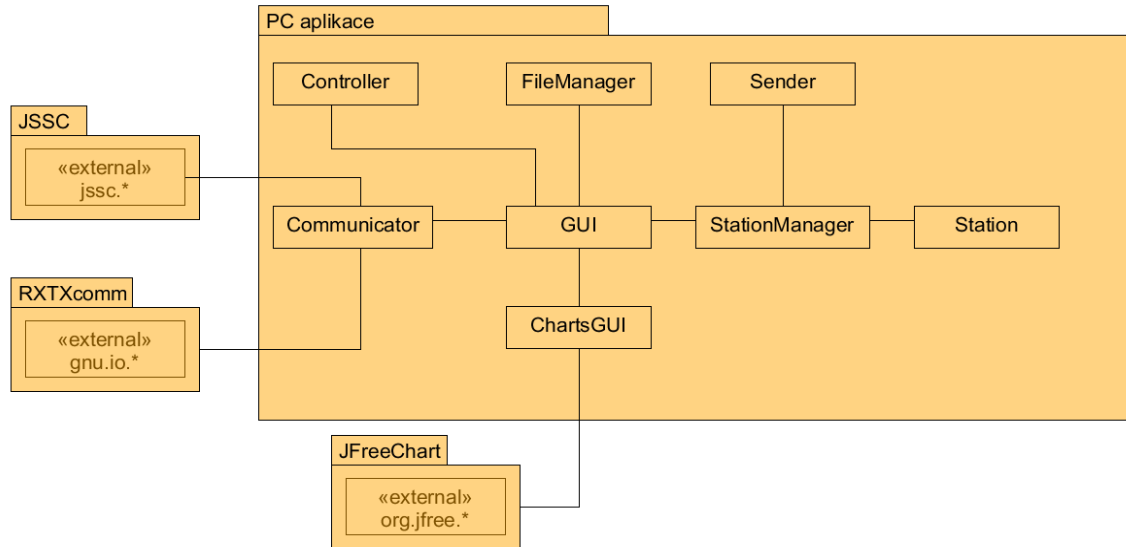
Nežli budou vykonány obslužné metody `dht_mode2()` a `pir_mode2()`, pro odeslání informací centrální stanici, je potřeba detekovat uplynutí zvoleného časového intervalu. To nemůže být provedeno příkazem `delay()`, protože ten by celý program zastavil na určenou dobu. Jelikož potřebujeme neustále přijímat nové příkazy, využijeme tedy následující podmínku:

```
unsigned long currentMillisDHT = millis();
if (currentMillisDHT - previousMillisDHT >= dhtinterval) {
    previousMillisDHT = currentMillisDHT;
    dht_mode2();
}
```

Metoda `millis()` vrací počet milisekund od zapnutí či restartování Arduina. Tento počet na začátku smyčky `loop()` uložíme do proměnných `currentMillisDHT` a `currentMillisPIR`, tj. aktuální čas. Poté už jen ověřujeme, zda požadovaný interval `dhtinterval` (u senzoru PIR je to `pirinterval`), ve kterém se mají metody spustit, již proběh či nikoliv. Pokud ano, je uložen aktuální čas, tj. `currentMillisDHT` nebo `currentMillisPIR` do proměnné `previousMillisDHT` či `previousMillisPIR`, které jsou na počátku rovny nule, a je spuštěna metoda `dht_mode2()` (u ověřování doby senzoru PIR je to metoda `pir_mode2()`).

4.3 Program pro PC aplikaci

V této části jsou stručně uvedeny třídy PC aplikace a jejich metody. Obecný digram tříd je na obrázku 6.



Obrázek 6: Obecný diagram tříd PC aplikace

Hlavní okno programu (viz obr. 7) je vytvářeno ve třídě `GUI.java` za podpory knihovny Swing [15], která nabízí velké množství nástrojů pro vytváření GUI. Třída obsahuje metodu `createPanel()` pro uspořádání prvků (tj. tlačítka, texty, přepínací tlačítka atd.), dále metodu `initComponents()` pro vytvoření obslužných událostí ke všem tlačítkům. Pro zpracování přijatých dat je zde několik metod:

- `parseInput()` – Rozdělení všech přijatých dat z centrální stanice pro další zpracování.
- `parseStarted()` – Slouží k zaregistrování monitorovací stanice, která se nahlásila.
- `parseToGraph()` – Rozdělení dat pro vizualizaci, tj. pro třídu `ChartsGUI.java`.

Třída `Communicator.java` zajišťuje veškerou komunikaci s centrální stanicí přes sériový port využitím knihoven `JSSC`³⁹ a `RXTXComm`⁴⁰ a obsahuje metody:

- `searchForPorts()` - K nalezení všech dostupných sériových portů.
- `connect()` - Připojení na zvolený sériový port.
- `disconnect()` - Odpojení ze sériového portu.
- `initListener()` - Spustí listener⁴¹ událostí, který kontroluje, zda nejsou k dispozici data ke čtení ze sériového portu.
- `serialEvent()` - Metoda, která zpracovává všechna přijatá data přes sériový port.
- `writeData()` - Metoda na odeslání dat z PC aplikace.

Třída `Controller.java` slouží k aktivaci/deaktivaci tlačítek obsažených v hlavním okně programu, tj. vytvořených ve třídě `GUI.java`. Její jedinou metodou je `toggleControls()`, ve které ověřuje, zda je navázáno spojení a pokud ano, tlačítka jsou aktivována, jinak jsou neaktivní.

Třída `FileManager.java` je určena pro práci se soubory. Obsahuje metody pro ukládání přijatých záznamů do souboru a metody k jejich opětovnému načtení k vizualizaci.

Třída `ChartsGUI.java` je vytvořena kombinací knihoven `Swing` a `JFreeChart`⁴². Prostřednictvím knihovny `Swing` je vytvořeno okno pro vizualizaci dat, včetně tlačítek a jejich obsluh. Samotnou vizualizaci dat zprostředkovává knihovna `JFreeChart`. Načtené záznamy ze souboru jsou zpracovány v metodách `createDatasetOfTemp()` (údaje teploty), `createDatasetOfHum()` (údaje vlhkosti vzduchu), `createDatasetOfMove()` (údaje počtu pohybu). Díky těmto datasetům jsou vytvářeny jednotlivé grafy pro teplotu `createChart(createDatasetOfTemp(), "Teplota", "°C"),`

³⁹ Bližší specifikace knihovny `jssc` - <https://github.com/scream3r/java-simple-serial-connector>

⁴⁰ Bližší specifikace knihovny `RXTXComm` - http://www.jcontrol.org/index_en.php

⁴¹ Listener (posluchač) - Je zodpovědný za generování reakcí na vzniklou událost.

⁴² Knihovna dostupná na stránce - <http://www.jfree.org/jfreechart/download.html>

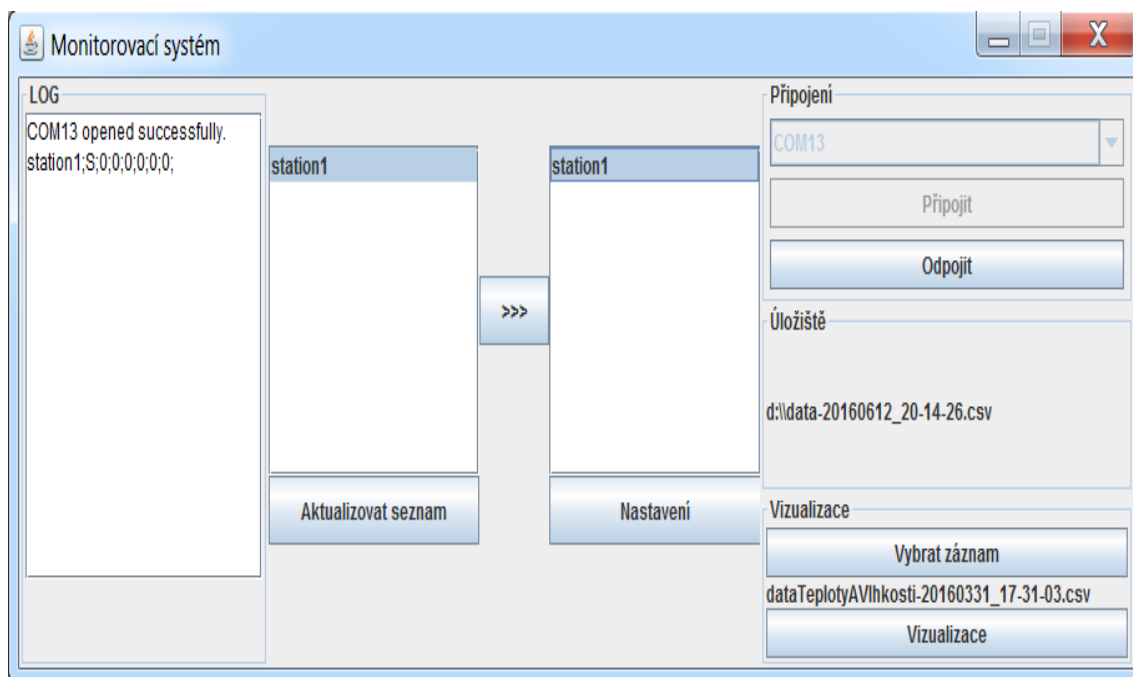
vlhkost vzduchu `createChart(createDatasetOfHum(), "Vlhkost", "%")`
a počet detekovaných pohybů `createChart(createDatasetOfMove(), "Pohyb", "Počet pohybů")`.

Třída `Sender.java` obsahuje pouze funkce pro odesílání příkazů po sériovém portu. Všechny metody si jsou velice podobné a liší se pouze vytvářeným textovým řetězcem, který se odesílá. Jsou to tyto funkce:

- `sendConfig()` - Odesílá konkrétní nastavení stanic (např. parametry `station1;D;1;1;0;1;1;0`, viz část 4.8).
- `sendPIR()` - Odesílá žádost o detekci pohybu (tj. parametry `station1;C;PIR`, viz část 4.8).
- `sendDHT()` - Odesílá žádost o záznam ze senzoru DHT (tj. parametry `station1;C;DHT`, viz část 4.8).

Třída `Station.java` obsahuje všechny atributy přidané stanice. Konkrétně jsou to `ID` (název stanice), `DHT` (stav DHT), `DHTmode` (DHT scénář), `timeDHT` (DHT interval), `PIR` (stav PIR), `PIRmode` (PIR scénář), `timePIR` (PIR interval). Abychom tyto atributy mohli pohodlně ovládat i z jiných tříd, jsou ke každému atributu vytvořeny gettery a settery.

Ve třídě `StationManager.java` je prostřednictvím knihovny Swing vytvořeno GUI pro nastavení monitorovacích stanic. Třída obsahuje dvě hlavní metody `createMainPanel()` a `initComponents()`. V metodě `createMainPanel()` jsou uspořádány všechny prvky GUI. V metodě `initComponents()` jsou vytvořeny obslužné události ke všem tlačítkům a také je zde možnost aktivace/deaktivace tlačítek podle daného nastavení monitorovacích stanic.



Obrázek 7: Ukázka GUI PC aplikace

4.4 Odesílání a přijímání SMS zpráv

GSM modul popsaný v části 2.2 umožňuje pracovat s SMS zprávami dvěma způsoby. Během práce jsem se seznámil s každým z nich, a je tedy vhodné si je vysvětlit oba pro lepší přiblížení dané problematiky.

V prvním způsobu pracujeme s knihovnou `SoftwareSerial` a některými konkrétními příkazy. Tento způsob se zdá být poněkud složitý, avšak znalost základních příkazů a jejich funkcionality pro ovládání GSM modulu rozhodně není na škodu. Ze všeho nejdříve nastavíme příkazem `SIM900(7,8)` pin 7 pro čtení a pin 8 pro zápis (RX a TX). Oba piny se mohou lišit v závislosti na tom, k jakému zařízení GSM modul připojujeme. Může se lišit například u zařízení Arduino Leonardo nebo Arduino Mega, které mají jiné rozložení pinů než Arduino UNO. Voláním metody `SIM900power()` zapneme/vypneme GSM modul (jinou možností je stisknutí tlačítka na modulu). Tato metoda neobsahuje nic jiného, než příkaz `digitalWrite(9,HIGH)` a příkaz `digitalWrite(9,LOW)`, kde přivádíme pouze logickou 1 nebo 0 na pin číslo 9, který nám modul zapne/vypne. Tyto příkazy nemohou být ovšem umístěny bezprostředně za sebou, protože

modul potřebuje nějaký čas, aby reagoval na příkaz zapnutí/vypnutí, takže je mezi nimi ještě umístěn příkaz `delay(5000)`, který pozastaví program na stanovený čas, v našem případě na 5 vteřin.

Odesílání zpráv funguje následovně. Vše se odehrává v metodě `sendSMS()`, která odesílání zpráv zajišťuje. Ta obsahuje dva AT⁴³ příkazy. První AT příkaz `SIM900.print("AT+CMGF=1\r")` slouží pro nastavení odeslání SMS zprávy v textovém režimu (tzn. že SMS zprávy jsou reprezentovány jako čitelný text, na rozdíl od PDU režimu, kde SMS zprávy jsou reprezentovány binárními řetězci zakódovanými jako hexadecimální znaky). Druhý AT příkaz `SIM900.println("AT+CMGS=\"+420123456789\"")` slouží pro zadání čísla mobilního telefonu příjemce v mezinárodním formátu. Pak už jen pošleme text zprávy pomocí `SIM900.println("Hello, world.")`, kde uvozovky značí text zprávy, který má být odeslán. Dále následuje příkaz `SIM900.println((char)26)` (v ASCII⁴⁴ znakové sadě je `ctrl+Z` označeno číslem 26). Na závěr je volána metoda `SIM900power()`, kterou modul vypneme.

Přijímání SMS funguje podobně jako odesílání. Arduino sketch⁴⁵, tj. hlavní část programu každého Arduina, obsahuje metody `setup()` a `loop()`. V metodě `setup()`, která proběhne jednou při spuštění programu, je použit za příkazy `SIM900power()` (zapnutí modulu), `SIM900.begin(9600)` (pro nastavení přenosové rychlosti komunikace) a `SIM900.print("AT+CMGF=1\r")` (nastavení textového SMS režimu) příkaz `SIM900.print("AT+CNMI=2,2,0,0,0\r")`⁴⁶. Ten slouží k nastavení způsobu zacházení s nově příchozími SMS zprávami. Můžeme např. požadovat, aby GSM modul nebo mobilní telefon předal nově příchozí SMS zprávy přímo počítači (např. přes USB), nebo je uložil do svého úložiště zpráv. Nyní už jen v nekonečné

⁴³ AT příkaz – Krátká sekvence příkazů k ovládní modemu. Výčet a význam základních příkazů - http://www.dhservis.cz/dalsi/at_prikazy.htm

⁴⁴ ASCII (American Standard Code for Information Interchange) - americký standardní kód pro výměnu informací

⁴⁵ Popis Arduino sketch na stránce - <https://www.arduino.cc/en/tutorial/sketch>

⁴⁶ Vysvětlení jednotlivých parametrů v příkazu - http://cellular.co.za/at_etsi.htm

smyčce `loop()` ověřujeme, zda GSM modul neobsahuje nově příchozí zprávu příkazem `if(SIM900.available()>0)`, a pokud ano, tak přijaté znaky ukládáme do zvolené proměnné příkazem `SIM900.read()`. To je vše nejdůležitější k prvnímu způsobu práce s SMS zprávami, který využívá knihovnu `SoftwareSerial`.

Nyní už k druhému způsobu práce s SMS zprávami, který je podstatně jednodušší a je založen na práci s knihovnou `GSM`⁴⁷. S touto knihovnou je práce s odesíláním a přijímáním zpráv mnohem jednodušší, protože všechny příkazy jsou schované uvnitř knihovny a my jen voláme příslušné metody. Z toho důvodu je v práci použit právě tento způsob.

Ze všeho nejdříve je nutné nastavit piny pro čtení a zápis (RX a TX), bez kterých nám nebude komunikace mezi Arduinem a GSM modulem fungovat. To uděláme uvnitř knihovny v souboru `GSM.cpp`, kde jsou zakomentovány jednotlivé řádky podle zařízení, se kterým pracujeme. V našem případě se jedná o Arduino UNO, takže odkomentujeme řádek `_GSM_TXPIN_7` (pro čtení) a `_GSM_RXPIN_8` (pro zápis). Po tomto kroku už nic nebrání samotné komunikaci. V metodě `setup()` voláme funkci `gsm.begin(9600)` pro zapnutí modulu s předávaným parametrem, který udává přenosovou rychlost komunikace. Tuto funkci je vhodné umístit jako podmíněný příkaz, abychom věděli, zda akce byla úspěšná či nikoliv.

Odesílání zpráv funguje velice jednoduše. Vše zařizuje pouze jedno volání funkce `sms.SendSMS("+420123456789", String)`, která na uvedené telefonní číslo posílá SMS s textem obsaženým v proměnné `String`. Pokud telefonní číslo nechceme zadávat přímo, lze zvolit pozici kontaktu v adresáři telefonu.

Příjem SMS se provádí v nekonečné smyčce `loop()`. V každém průchodu smyčky ověříme, zda není na vstupu nová nepřečtená SMS. Pokud ano, tak ukazatel na SMS uložíme do proměnné `sms_position`. K tomu slouží funkce `sms.IsSMSPresent(SMS_UNREAD)`, která vrací ukazatel na první nepřečtenou

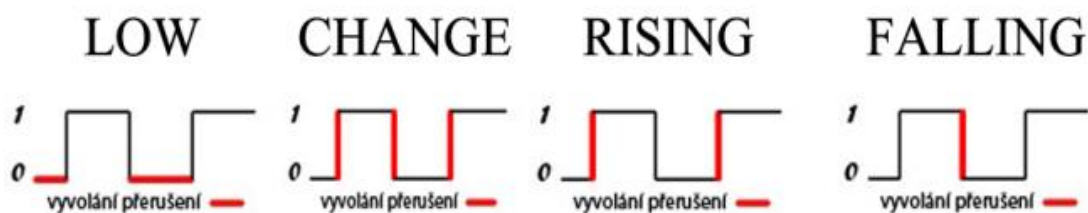
⁴⁷ Knihovna dostupná na stránce - <http://educ8s.com/Arduino/SMS/>

zprávu. Parametrem můžeme zvolit rovněž `SMS_READ` (přečtené SMS) a `SMS_ALL` (všechny SMS), ale tyto parametry pro nás nemají využití. Dále už je zde jen poslední funkce `sms.GetSMS(sms_position, phone_number, sms_text, 160)`, která obsah příchozí SMS na zjištěné pozici `sms_position` uloží do proměnné `sms_text`, číslo, ze kterého byla SMS odeslána, uloží do proměnné `phone_number` a maximální možnou délku SMS udává poslední parametr 160 (tj. délka SMS standardu GSM, podle kterého je pro text SMS povoleno celkem 1120 bitů na jednu SMS, což odpovídá u 7bitového kódování délce 160 znaků).

4.5 Přerušování

Přerušování⁴⁸ využívá monitorovací stanice pro reakce na pohyb, když počítá, kolikrát pohyb nastal v určeném intervalu. Arduino poskytuje celkem dvě externí přerušování. Jsou na digitálních pinech 2 a 3 a jsou označeny 0 a 1. Na těchto pinech mohou nastat 4 různé druhy událostí, viz obr. 8:

- `LOW` — Přerušování nastane vždy, když je pin v logické nule.
- `CHANGE` — Přerušování nastane při změně logické hodnoty na daném pinu.
- `RISING` — Přerušování s příchodem vzestupné hrany.
- `FALLING` — Přerušování s příchodem sestupné hrany.



Obrázek 8: Druhy událostí přerušování

Pro práci s Arduinem jsou navrženy také 4 funkce obsluhující hardwarová přerušování. Jsou to funkce:

⁴⁸ Přerušování (interrupt) - metoda pro asynchronní obsluhu událostí, kdy procesor přeruší vykonávání sledu instrukcí, vykoná obsluhu přerušování, a pak pokračuje v předchozí činnosti.

- `attachInterrupt()` — Slouží k registraci námi definované funkce pro obsluhu přerušení.
- `detachInterrupt()` — Odpojí funkci od daného přerušení.
- `interrupts()` — Zapíná přerušení.
- `noInterrupts()` — Vypíná přerušení.

Nejdříve je nutné v metodě `setup()` povolit přerušení výše zmíněnou funkcí `interrupts()`. Pokud by tato funkce nebyla zavolána, tak přerušení nebude fungovat z důvodu implicitního nastavení Arduina, tj. nereagovat na externí přerušení. V metodě `setup()` je např. nastaveno přerušení příkazem `attachInterrupt(0, pirInterrupt, CHANGE)`, tzn. že přerušení nastane při každé změně logické hodnoty na pinu 2 a v reakci na přerušení bude zavolána obslužná metoda `pirInterrupt()`. Tato metoda nedělá nic jiného, než že zvětší čítač pohybu o 1.

4.6 Záznamy vytvářené monitorovací stanicí

Monitorovací stanice poskytuje PC aplikaci několik druhů záznamů. Prvním záznamem je ID⁴⁹ stanice, podle kterého se identifikuje centrální stanici. ID je inicializováno jako textový řetězec a jeho tvar může být tedy téměř libovolný. Další údaje jsou teplota a vlhkost vzduchu získané ze senzoru DHT a počet pohybů, které jsou detekovány v okolí stanice PIR senzorem. Teplota je udávána jako dvouciferné číslo v rozsahu 0-50 °C a vlhkost vzduchu je udávána jako dvouciferné číslo v rozsahu 20-90%. Tyto minimální a maximální dosažitelné hodnoty, které je možné naměřit použitým senzorem, byly již zmíněny v části 2.4. Počet pohybů se liší podle nastavení monitorovací stanice. Pokud přijme požadavek na detekci aktuálního výskytu pohybu v okolí stanice, údaj je zaznamenán jako 0 (tzn. žádný pohyb) nebo 1 (tzn. zjištěn pohyb). Pokud má stanice reagovat pouze na pohyb, údaj bude zaznamenán vždy jako 1 (nemůže se zaznamenat 0, protože nebyl zjištěn žádný pohyb, tedy ani PIR senzor nijak nezareagoval). Počet pohybů, které byly detekovány v určeném intervalu, se zaznamená jako číslo v rozmezí

⁴⁹ ID – identifikace objektu ve výpočetní technice

0-65535 (tzn. 0 od žádného pohybu až po maximální hodnotu *unsignedInt*⁵⁰ na zařízení Arduino). Poslední údaj přidá PC aplikace a jím je čas ve formátu HH:mm:ss:SSS:dd:MM:yyyy. Jsou zde udávány i milisekundy (SSS) a to z toho důvodu, že může přijít několik SMS během jedné sekundy. V tomto případě by byly údaje zaznamenány pod stejným časem a nebylo by možné je následně vizualizovat při vytváření grafu. Všechny záznamy se ukládají do souboru s názvem `data-yyyyMMdd_HH-mm-ss.csv`, do adresáře, který je zadán v konfiguračním souboru `CONFIG.txt`. První řádek v uloženém CSV souboru obsahuje názvy sloupců, pod kterými jsou uvedeny záznamy: ID, čas, teplota, vlhkost a počet pohybů. V záznamech se může objevit místo naměřených údajů hodnota *null*⁵¹. To zapříčinilo nastavení dané stanice, např. pokud byl zapnut jen PIR senzor, tak údaje teploty a vlhkosti vzduchu zaznamenány nebyly a jsou zde zapsány právě jako *null*.

4.7 Vizualizace dat v PC aplikaci

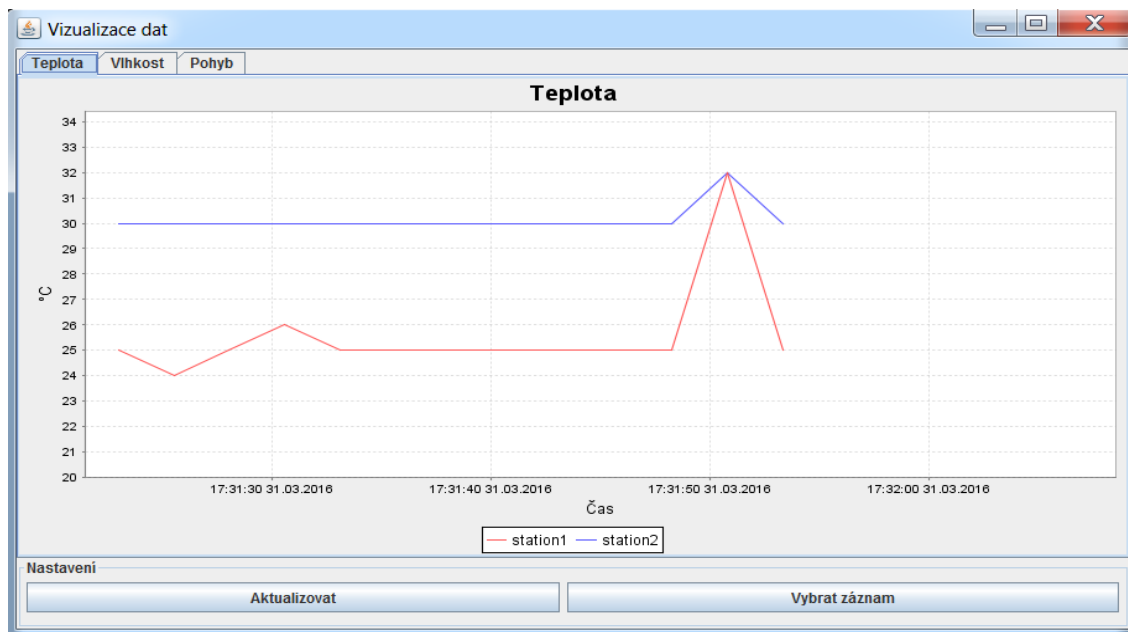
Všechny záznamy, které PC aplikace přijme a uloží do příslušného souboru, jsou vizualizovány třídou `chartsGUI.java`. Tyto záznamy nejsou nikterak modifikovány, ale pouze tříděny podle jména stanice, času, teploty, vlhkosti vzduchu a pohybu a následně zpracovávány, takže nehrozí jejich ztráta či změna. Třídění je prováděno nejdříve v metodě `parseDataByStation(String)`, které jsou předány všechny záznamy a roztrženy do `ArrayList52<String>` podle jména stanice. V metodě `parseOtherParts(String)` se již záznamy zpracovávají do podoby zobrazitelné v grafu. V něm jsou konkrétní hodnoty teploty a vlhkosti vzduchu a počtu pohybů zobrazeny v závislosti na čase, viz obrázek 9. V okně se kromě vizualizovaných záznamů, které je možné přepínat v horní liště, nacházejí ještě dvě tlačítka. První tlačítko `Aktualizovat` slouží k aktualizaci grafu a vykreslení nově příchozích záznamů. Druhé tlačítko `Vybrat záznam` poskytuje možnost vizualizovat jiné uživatelem zvolené záznamy. Knihovna `JFreeChart` obstarává možnosti nastavení grafu (např. změnu barvy

⁵⁰ Bližší informace o datové struktuře - <https://www.arduino.cc/en/Reference/UnsignedInt>

⁵¹ `null` - proměnná neukazuje na žádnou instanci

⁵² `ArrayList<String>` - generická kolekce, v závorce uváděn datový typ objektů, které ukládá.

pozadí, přiblížení/oddálení grafu, uložení grafu ve formátu PNG⁵³), takže tyto možnosti není nutné programově vytvářet.



Obrázek 9: Vizualizace dat – teplota snímaná dvěma stanicemi.

4.8 Komunikační protokol

Komunikační protokol byl navržen tak, aby přenášené zprávy byly co možná nejmenší. Nežli posílat příkazová slova, bylo raději zvoleno posílání příznaků (kromě jména stanice, které je napsáno slovem). Každá zpráva je složena ze znaků oddělených středníkem a zakončena znakem #. PC aplikace odesílá několik druhů zpráv, které můžeme vidět v tabulkách 3 a 4. Typy zpráv odesílané monitorovacími stanicemi jsou vidět v tabulce 5.

4.8.1 Komunikace PC aplikace s monitorovací stanicí

V tabulce 3 jsou obsaženy typy zpráv, které se odesílají při požadavku na aktuální stav monitorovací stanice (tzn. žádost o zaslání hodnoty teploty a vlhkosti vzduchu, které jsou aktuálně naměřeny senzorem DHT, a zda senzor PIR detekoval pohyb). Na první pozici ve zprávě (sloupec ID) je název cílové stanice, na kterou bude příkaz odeslán, na druhé pozici (sloupec Znak) je vždy znak C, podle kterého

⁵³ PNG (Portable Network Graphics) - formát pro bitmapovou grafiku využívající bezztrátovou kompresi.

bude příkaz zpracován v monitorovací stanici, a na posledním místě (sloupec *Senzor*) je buď DHT, nebo PIR podle toho, zda nás zajímá stav na senzoru DHT nebo PIR.

Tabulka 3: Typy zpráv odesílané PC aplikací – žádosti o aktuální stav

ID	Znak	Senzor
Název stanice	C	PIR
		DHT

V tabulce 4 jsou obsaženy typy zpráv, které nastavují monitorovací stanici do různých scénářů (viz část 3.3). První sloupec (*ID*) obsahuje název cílové monitorovací stanice, na kterou je příkaz odeslán, na druhé pozici je vždy znak *D*, podle kterého bude příkaz zpracován v monitorovací stanici. Sloupec *DHT* může obsahovat hodnoty 0 (senzor DHT vypnut) a 1 (senzor DHT zapnut), dále zde je sloupec *DHT scénář*, který obsahuje některou z hodnot 0-3, které zastupují vybraný scénář. Při nastaveném scénáři 2 je nutné zvolit *DHT interval* (tj. interval udávaný v milisekundách, který je označený v tabulce 4 písmenem *X*). V tomto intervalu mají být zaznamenávány veličiny v monitorovací stanici a následně odeslány. Sloupce *PIR*, *PIR scénář* a *PIR interval* (v tabulce 4 označený písmenem *Y*) mají stejnou funkcionalitu jen s tím rozdílem, že se hodnoty týkají senzoru PIR.

Tabulka 4: Typy zpráv odesílané PC aplikací –nastavení scénářů

ID	Znak	DHT	DHT scénář	DHT interval	PIR	PIR scénář	PIR interval
Název stanice	D	0			0		
		1	0		1	0	
			1			1	
			2	X		2	Y
setup							

PC aplikace odesílá ještě jeden typ zprávy. Jedná se o příkaz `setup` určený centrální stanici. Jeho odesláním se PC aplikace dotazuje na dostupnost všech monitorovacích stanic, tzn. že pokud se monitorovací stanice zapne a ohlásí se centrální stanici dříve, nežli je spuštěná PC aplikace (která logicky nemůže zaregistrovat spuštění monitorovací stanice), tak právě pomocí tohoto příkazu jsou všechny monitorovací stanice dotázány na svou dostupnost.

4.8.2 Komunikace monitorovací stanice s PC aplikací

Každá monitorovací stanice se při spuštění nahlásí PC aplikaci stejným typem zprávy, který je uveden výše v tabulce 4, avšak místo znaku D je použit znak S značící výchozí nastavení. Každá monitorovací stanice má výchozí nastavení, které je možné změnit pouze v programu Arduina. Toto nastavení vypadá následovně: `station1;S;0;0;0;0;0;0`, tj. ID; znak S; DHT; DHT scénář; DHT interval; PIR; PIR scénář; PIR interval.

Další typ zprávy, viz tabulka 5, je vytvářen reakcí na příkazy přijaté z PC aplikace (viz tabulka 3). Na první pozici (sloupec ID) je název stanice, ze které je zpráva odeslána, na druhé pozici (sloupec Znak) je vždy znak C, podle kterého bude příkaz zpracován v PC aplikaci, na dalším místě (sloupec Sensor) je buď DHT, nebo PIR podle toho, zda nás zajímá stav na senzoru DHT nebo PIR. Poslední je sloupec Hodnoty, kde písmeno P označuje, kolikrát byl detekován pohyb. Písmena T a V označují hodnoty teploty a vlhkosti vzduchu, naměřené senzorem DHT. Každá zpráva, která je odeslána z monitorovací stanice, je ukončena znakem # z důvodu oddělení jednotlivých zpráv.

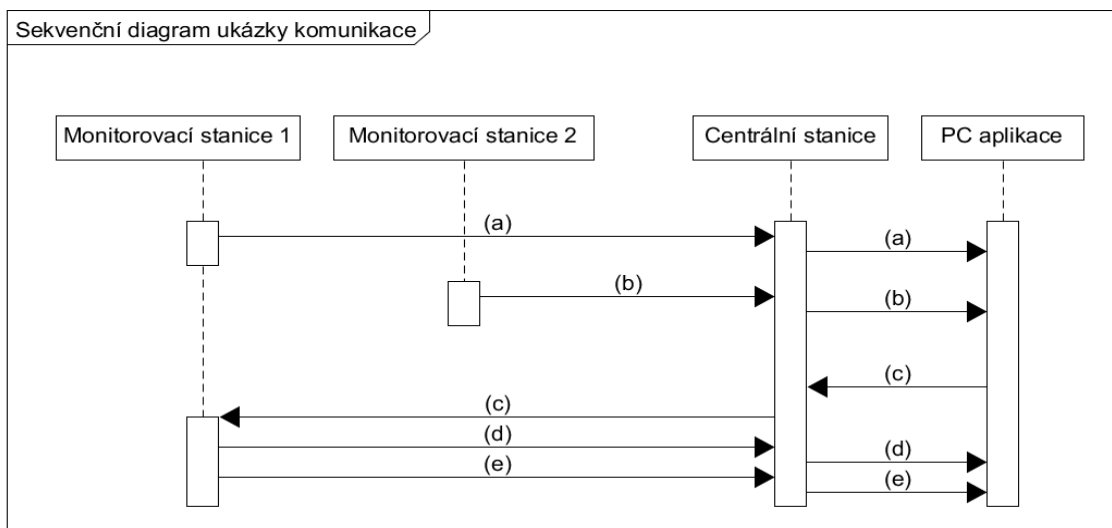
Tabulka 5: Typy zpráv odesílané monitorovacími stanicemi

ID	Znak	Senzor	Hodnoty	
Název stanice	C	PIR	P	
		DHT	T	V

4.8.3 Názorná ukázka komunikace

V následující ukázce se podíváme, jak probíhá komunikace mezi monitorovací stanicí, centrální stanicí a PC aplikací. Na situaci se díváme z pohledu PC aplikace. Diagram komunikace je na obrázku 10.

- a) PC aplikace přijala: `station1;S;0;0;0;0;0;0;0`, tj. nahlásila se stanice s ID `station1` s výchozím nastavením (tj. nemonitoruje prostředí a pouze čeká na příkazy).
- b) PC aplikace přijala: `station2;S;0;0;0;0;0;0;0`, tj. nahlásila se stanice s ID `station2` s výchozím nastavením (tj. nemonitoruje prostředí a pouze čeká na příkazy).
- c) PC aplikace odeslala: `station1;D;1;1;0;1;1;0`, tj. stanici s ID `station1` byl odeslán požadavek na aktivaci senzoru PIR a senzoru DHT, které budou pracovat dle scénáře 1.
- d) PC aplikace přijala: `station1;C;PIR;1`, tj. byla přijata informace od stanice s ID `station1`, která znamená, že byl detekován pohyb.
- e) PC aplikace přijala: `station1;C;DHT;27;32`, tj. byly přijaty naměřené hodnoty teploty (27 °C) a vlhkosti vzduchu (32%) od stanice s ID `station1`.



Obrázek 10: Sekvenční diagram komunikace.

5 Diskuze

V této kapitole bude diskutována praktická použitelnost systému a jeho možná vylepšení. V části 5.1 budou popsány problémy, se kterými jsem se během práce setkal, testování systému v části 5.2, nové zkušenosti, které jsem získal během vytváření monitorovacího systému, jsou v části 5.3. Dále budou zmíněny součástky a náklady na jejich pořízení (viz část 5.4), které jsou nutné k výrobě tohoto monitorovacího systému, a případné možné alternativy k nim.

5.1 Problémy návrhu

Během návrhu a realizace systému se vyskytly problémy, které jsou popsány v následujících částech. Některé bylo možné vyřešit použitím jiné součástky, ale některé se bohužel vyřešit nepodařilo.

5.1.1 Nadměrná vlhkost

Jedním z problémů může být nevhodné počasí. Arduino bude ideálně umístěno někde v přírodě a přírodní jevy, jako například déšť nebo nadměrná vlhkost, budou našemu Arduinu a dalším součástkám zcela jistě vadit. Mohlo by se to jistě řešit nějakým ochranným vodotěsným krytem, ale jelikož se v našem případě jedná pouze o tvorbu prototypu, tak tento problém zanedbáváme.

5.1.2 Porušené snímky a prodleva snímání

Bohužel Arduino jako takové není příliš vhodné pro komunikaci s kamerou. Při snímání obrázků kamera vytvoří první nepoškozený snímek, ale všechny ostatní jsou poškozené. Problém jsem se pokoušel řešit zmenšením obrázku na nejmenší velikost 160*120 (další dostupné jsou 320*240 a 640*480), ale s úspěchem jsem se nesešel. Dále mým předpokladem bylo, že čím rychleji se budou pixely zapisovat, tím je větší pravděpodobnost, že kamera nebude stíhat. To se ale bohužel nepotvrdilo, protože i po přidání velké časové prodlevy jsou snímky stále poškozené.

Knihovny používaných kamer neumožňují získat snímek (poškozený i nepoškozený) jako jeden celek, ale bylo potřeba získávat každý pixel zvlášť, což je

velice zdlouhavé. Při velikosti snímku 640*480 pixelů se doba sestrojení snímku pohybovala okolo 10 sekund.

5.1.4 Nedostatek paměti a stabilita systému

Při použití většího množství existujících knihoven se můžeme snadno setkat s nedostatkem operační paměti Arduina (pouze 32kB). To lze vyřešit několika způsoby. Například můžeme některé obsáhlé knihovny zmenšit, jelikož většinou obsahují i funkce, které vůbec nevyužijeme. Můžeme také více pracovat s ukazateli a poli, které přímo pracují s pamětí, takže nezvětšují velikost programu. Samozřejmě také můžeme vybírat mezi jednotlivými knihovnami. Například knihovnu SD.h, která slouží k interakci s SD kartou, můžeme nahradit knihovnou Fat16.h, která nám poslouží stejně dobře, jen funguje pro SD kartu o maximální kapacitě 2GB (rozdělení knihoven pro práci s SD kartami obsahuje část 2.6). To by nám ovšem vadit nemělo.

Při použití knihovny pro obsluhu GSM modulu společně s knihovnou pro obsluhu SD karty byl systém velice nestabilní. Paměť byla využita přibližně na 90% a systém nebyl schopen pracovat správně. Prvotní myšlenka byla, že to způsobuje nedostatečné napájení Arduina a jeho modulů, ale po přidání externího napájení tento problém setrval a příčina je tudíž jinde. Při odstranění jedné z knihoven žádný problém s nestabilitou nenastal. Při použití obou knihoven, avšak bez žádné vnitřní logiky (tj. bez použití několika jednoduchých funkcí či proměnných) byla paměť Arduina využita přibližně na 80%. Při této hodnotě by už program pravděpodobně pracoval správně, ale této hodnoty nebylo možné reálně dosáhnout. Z tohoto důvodu nemohl být SD modul použit.

5.1.5 GSM modul

Tento modul slouží dobře pro komunikaci prostřednictvím SMS zpráv, ale bohužel pokusy o připojení k internetu byly neúspěšné. Systém by tak mohl posílat snímky rovnou na nějakou webovou stránku. Bohužel ani při pokusech o připojení na úplně primitivní stránky modul nedokázal zajistit žádný přenos dat. Jednalo se o stránky, které neobsahovaly žádnou složitou strukturu, pouze měly přijímat

odeslané zprávy, což se při opakovaných pokusech nepovedlo. Využití GSM modulu tedy zůstane u komunikace SMS zprávami.

5.2 Testování systému

Arduino s připojenými součástky nabízí v této kombinaci velkou škálu možností. Před testováním celého systému bylo nutné prostudovat a otestovat komunikaci Arduina s PC, přenos dat využitím GSM modulu a funkčnost připojených senzorů. Jak už bylo zmíněno v části 3.4, v rámci testování celého systému neprobíhala komunikace z jednoho Arduina do druhého přes GSM modul a posílání SMS zpráv, ale přes sériovou komunikaci použitím drátěných propojek. Důvody, které k tomu vedly, byly zejména finanční. Mít dvě univerzální SIM karty a používat je při vývoji a testování systému není bez neomezeného tarifu levná záležitost. Testování PC aplikace proběhlo za podpory frameworku *JUnit*⁵⁴ [16] pro jednotkové testy a nástroje *EclEmma*⁵⁵ pro zjištění pokrytí kódu. Celý monitorovací systém byl testován nasazením v praxi.

5.3 Získané zkušenosti

Během práce s Arduinem jsem se blíže seznámil s vývojovým prostředím Arduino IDE a knihovnou Wiring, která je určena pro vývoj programů pro Arduino. S obojím jsem doposud neměl žádné zkušenosti. Dále jsem nahlédl do problematiky kamerových systémů, abych zjistil, na jakých principech pracují. Byla to velice zajímavá zkušenost, ačkoliv se ve výsledném systému nakonec nezuročila. Také jsem se zabýval celkovou problematikou monitorovacích systémů, tj. k čemu slouží, kde se používají, na jakých principech pracují a jaké technologie využívají.

5.4 Cena výsledného systému a jiné alternativy

Jedním z cílů práce bylo co nejvíce minimalizovat náklady na celkový systém, proto v této části popíšeme ceny jednotlivých komponent v době návrhu a vypočteme celkovou pořizovací cenu našeho konečného řešení. Pořizovací cena Arduina UNO v době realizace činila 679 Kč⁵⁶. Jiné alternativy lze pořídit

⁵⁴ Bližší informace o knihovně na stránce - <http://junit.org/junit4/>

⁵⁵ Bližší informace o knihovně na stránce - <http://www.eclEmma.org/>

⁵⁶ Arduina UNO je dostupné na stránce - <https://www.alza.cz/arduino-uno-rev3-d569244.htm>

za podstatně vyšší cenu. Cena například Arduino YUN se pohybuje okolo 2099 Kč⁵⁷ a cena Arduino Mega okolo 1129 Kč⁵⁸, což převyšuje základní cenu Arduina UNO skoro o dvojnásobek. GSM modul stál v době realizace 1199 Kč⁵⁹, ovšem pořídit ho na českém trhu není vůbec jednoduché. Dostupný je v podstatě jen v jednom obchodě a jiné alternativy jsou mnohem dražší. Příkladem je GSM 2 modul s integrovanou anténou a pořizovací cenou 2120 Kč⁶⁰. DHT11 senzor k měření teploty a vlhkosti vzduchu stojí 89 Kč⁶¹ a senzor pohybu PIR stojí také 89 Kč⁶². Možná alternativa k senzoru teploty a vlhkosti vzduchu je senzor DHT22, který stojí 189 Kč⁶³. Nedílnou součástí pro testování je nepájivé pole s pracovní částí 300 pinů a napájecí částí 100 pinů s pořizovací cenou 59 Kč⁶⁴. Lze pořídit nepájivé pole o větší pracovní a napájecí části, ovšem s růstem velikosti roste také cena. Sada drátových propojek stála v době návrhu 119 Kč⁶⁵. Ačkoli kamera není součástí systému, tak s její přítomností se původně počítalo, tudíž je také vhodné zmínit cenu. TTL kamera stála v době návrhu 39.95 \$⁶⁶, tj. 950 Kč, avšak na českém trhu je jí také obtížné pořídit. Kamera OV7670 stála v době návrhu okolo 300 Kč a lze ji pořídit na aukčních webových portálech. Cena monitorovací stanice (Arduino, GSM modul, senzor PIR, senzor DHT, nepájivé pole a drátěné propojky) tedy činí 2234 Kč a cena centrální stanice (Arduino, GSM modul) činí 1878. Celková cena při použití jedné monitorovací stanice je 4112 Kč.

⁵⁷ Arduino YUN je dostupné na stránce - <https://www.alza.cz/arduino-yun-d569253.htm>

⁵⁸ Arduino Mega je dostupné na stránce - <https://www.alza.cz/arduino-mega2560-rev3-d569248.htm>

⁵⁹ GSM modul je dostupný na stránce - <http://robotstore.cz/obchod/arduino/gprs-gsm-shield-arduino-stit-2/>

⁶⁰ GSM 2 modul je dostupný na stránce - <https://www.alza.cz/arduino-shield-gsm-2-modul-integrovana-antena-d2649561.htm>

⁶¹ DHT11 senzor je dostupný na stránce - <http://robotstore.cz/obchod/senzory/dht11-arduino-modul-pro-mereni-teploty-a-vlhkosti-5/>

⁶² PIR senzor je dostupný na stránce - <http://robotstore.cz/obchod/arduino/hc-sr501-pir-senzor-detekce-pohybu-arduino-modul-3/>

⁶³ DHT22 senzor je dostupný na stránce - <http://robotstore.cz/obchod/senzory/dht22-am2302-senzor-pro-mereni-teploty-vlhkosti-arduino/>

⁶⁴ Nepájivé pole je dostupné na stránce - <http://www.gme.cz/nepajive-kontaktni-pole-zy-60-p661-137>

⁶⁵ Drátové propojky jsou dostupné na stránce - <http://www.gme.cz/dratove-propojky-wjw-70-p759-885>

⁶⁶ TTL kamera je dostupná na stránce - <http://www.adafruit.com/product/397>

6 Závěr

Cílem této bakalářské práce bylo seznámení se s Arduino UNO a jeho moduly za účelem vytvoření funkčního monitorovacího systému. Za tímto účelem byly vytvořeny celkem tři aplikace. První aplikace je určena pro monitorovací stanici, která obsahuje senzor pro měření teploty a vlhkosti vzduchu a senzor pro detekci pohybu. Ze senzorů poskytovaných údajů jsou vytvářeny záznamy v závislosti na nastavení stanice. Nastavení je prováděno uživatelem prostřednictvím aplikace vytvořené pro PC. Tato aplikace mimo jiné zpracovává a vizualizuje všechny záznamy v podobě grafů, které je možné uložit ve formátu PNG. Třetí aplikace je určena centrální stanici, která přeposílá data z monitorovacích stanic do PC aplikace a naopak. Přenos dat mezi monitorovacími stanicemi a centrální stanicí je uskutečněn GSM moduly posíláním SMS zpráv. Pro testovací účely probíhá přenos dat mezi monitorovacími stanicemi a centrální stanicí přes sériovou komunikaci. Centrální stanice a PC aplikace jsou propojeny kabelem USB.

Ačkoliv monitorovací systém je funkční a splňuje všechny požadavky uvedené v zadání práce, tak dle mého názoru použití Arduina pro monitorovací účely není příliš vhodné, zejména kvůli nedostatku operační paměti a špatné kompatibilitě některých součástí či modulů. Jako vhodná alternativa Arduina se nabízí použití jiného zařízení, např. Raspberry Pi, které by umožnilo připojení modulů, které v našem systému nebyly použity, např. kamerového modulu a SD modulu.

6.1 Možné budoucí práce

Pokud by na tuto bakalářskou práci navazovala další odborná práce, tak by jistě přicházelo v úvahu, podívat se blíže na problémy, se kterými jsme se při návrhu setkali, a bylo by vhodné se jimi blíže zabývat a pokusit se je vhodně vyřešit. Velmi zajímavé by bylo snímání obrazu prostřednictvím kamery, případně rozpoznávání charakteristických rysů osob (např. zachycením tváře, výšky postavy či oblečení osoby). Dále by přicházelo v úvahu rozšíření monitorovacích stanic o více senzorů. Mezi ně by mohla patřit infračervená závora, ultrazvukový senzor,

digitální displej na zobrazení údajů přímo na monitorovací stanici atd. Jistě by bylo zajímavé vyzkoušet pro přenos dat kromě GSM také jiné dostupné moduly. Po připojení k internetu (např. pomocí ethernet modulu) by se záznamy mohly posílat na webový server a být tak dostupné pro více uživatelů online. Z těchto záznamů by se daly vytvořit různé statistiky, např. o změnách počasí v určitých oblastech, o výskytu zvěře v jednotlivých ročních obdobích a jiné.

PC aplikace by se následně měla rozšířit o nastavení nově přidaných senzorů, získávání záznamů ze serveru a podobně. Přidáním těchto několika málo rozšíření by monitorovací systém získal řadu dalších možných využití.

Reference

- [1] HEROUT, Pavel. *Java: grafické uživatelské prostředí a čeština*. 2. vyd. České Budějovice: Kopp, 2007. ISBN 8072323288.
- [2] KŘEČEK, Stanislav. *Příručka zabezpečovací techniky*. Vyd. 2. S.l.: Cricerus, 2003. ISBN 80-902938-2-4.
- [3] KÜHNEL, Claus. *AVR RISC microcontrollers handbook*. Boston: Newnes, 1998. ISBN 0750699639.
- [4] DH SERVIS. *Pulzně šířková modulace* [online]. 2002 [Citace: 6. 2.2016].
Dostupné z: <http://www.dhservis.cz/psm.htm>
- [5] HEROUT, Pavel. *Učebnice jazyka Java*. 5., rozš. vyd. České Budějovice: Kopp, 2010. ISBN 978-80-7232-398-2.
- [6] PECINOVSKÝ, Rudolf. *Java 7: učebnice objektové architektury pro začátečníky*. Praha: Grada, 2012. Knihovna programátora (Grada). ISBN 978-80-247-3665-5.
- [7] DOSTÁL, Radim. *C/C++: hotová řešení*. Vyd. 1. Brno: Computer Press, 2009. K okamžitému použití (Computer Press). ISBN 978-80-251-2190-0.
- [8] VIRIUS, Miroslav. *Jazyky C a C++: kompletní kapesní průvodce programátora*. Praha: Grada, 2006. ISBN 80-247-1494-9.
- [9] MARGOLIS, Michael. *Arduino cookbook*. 2nd ed. Sebastopol, Calif.: O'Reilly, 2012. ISBN 1449313876.
- [10] MCROBERTS, Michael. *Beginning Arduino*. New York: Distributed to the book trade worldwide by Springer Science + Business Media, LLC., 2010. Technology in action series. ISBN 1430232404.
- [11] MONK, Simon. *30 Arduino projects for the evil genius*. New York: McGraw-Hill, 2010. ISBN 007174133X.
- [12] GPRS. *K čemu je GPRS* [online]. 2002 [Citace: 6. 2.2016].
Dostupné z: <http://www.earchiv.cz/b01/b0100001.php3>

[13] GSM. *Jak přenášet data v sítích GSM* [online]. 2002 [Citace: 6. 2.2016].

Dostupné z: <http://www.earchiv.cz/a008s200/a008s205.php3>

[14] GPRS/GSM SIM900. *SIM900 GPRS/GSM Shield* [online].

2002 [Citace: 6. 2. 2015]. Dostupné z:

http://linksprite.com/wiki/index.php5?title=SIM900_GPRS/GSM_Shield

[15] LOY, Marc. a Robert. ECKSTEIN. *Java Swing*. 2nd ed. Sebastopol, CA: O'Reilly, 2003. ISBN 0596004087.

[16] WALNES, Joe. *Java Open Source programming: with XDoclet, JUnit, WebWork, Hibernate*. Indianapolis, Ind.: Wiley, 2004. Java Open Source library.

A Uživatelská příručka

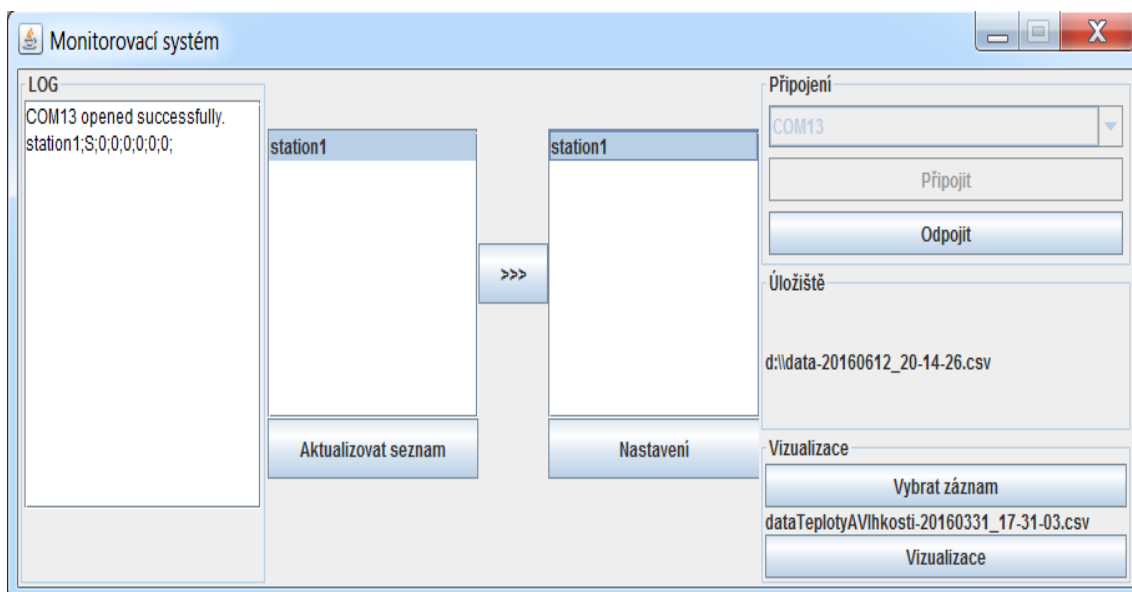
A1 Přeložení a spuštění aplikace

K přeložení a spuštění PC aplikace je nutné mít v PC nainstalovanou Javu JDK7 a nástroj Ant. Překlad lze provést na libovolném operačním systému a to v adresáři se zdrojovými soubory (adresář `src`), potřebnými knihovnamí (adresář `lib`) a souborem `build.xml`. Oba zmíněné adresáře a soubor jsou obsaženy na přiloženém paměťovém médiu (DVD).

Překlad je proveden příkazem `ant` zadaným v příkazové řádce. Tímto se vytvoří adresář `jar` se spustitelným `jar` souborem `system.jar`, který lze spustit např. příkazem `java -jar system.jar`.

A2 Ovládání aplikace

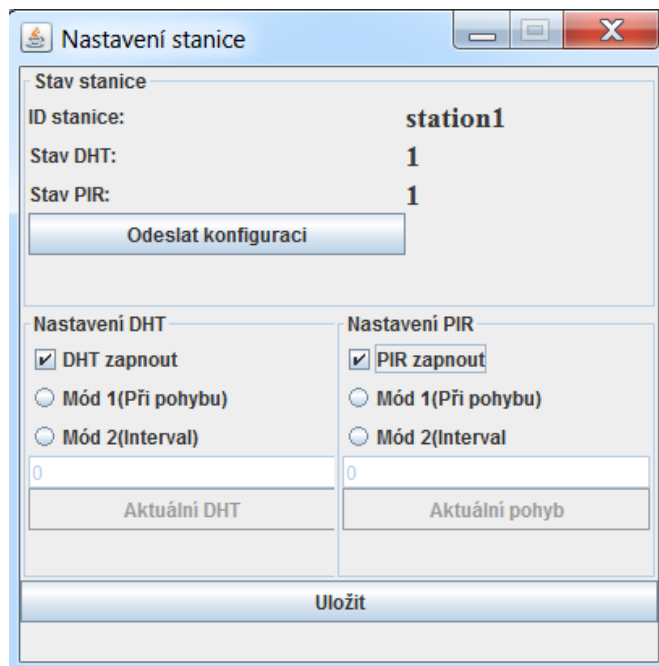
Spuštěním aplikace se otevře hlavní okno monitorovacího systému (viz obr. A1), v levém panelu LOG jsou přijaté zprávy z monitorovacích stanic. Dostupné stanice se automaticky zobrazí v poli vedle levého panelu. Tlačítko Aktualizovat seznam všechny dostupné stanice aktualizuje. Tlačítko >>> přidá dostupné vybrané stanice do dalšího pole, kde je možné vybrané stanice nastavit tlačítkem Nastavení. Pravý panel je rozdělen na tři části. První část je Připojení, kde se automaticky vyhledají dostupné porty a na jeden z nich se uživatel připojí tlačítkem Připojit a odpojí tlačítkem Odpojit. To je důležité pro zprovoznění komunikace s centrální stanicí. Další částí je úložiště, kde je zobrazen adresář a název souboru ukládaných záznamů. Úložiště se nastaví automaticky do složky `jar`, nebo je možné ho změnit v souboru `CONFIG.txt`. Poslední částí je Vizualizace, která obsahuje tlačítko Vybrat záznam pro vybrání souboru uložených dat. Tento vybraný soubor je zobrazen pod tlačítkem. Dále obsahuje tlačítko Vizualizace, které otevře okno s vizualizací vybraných záznamů.



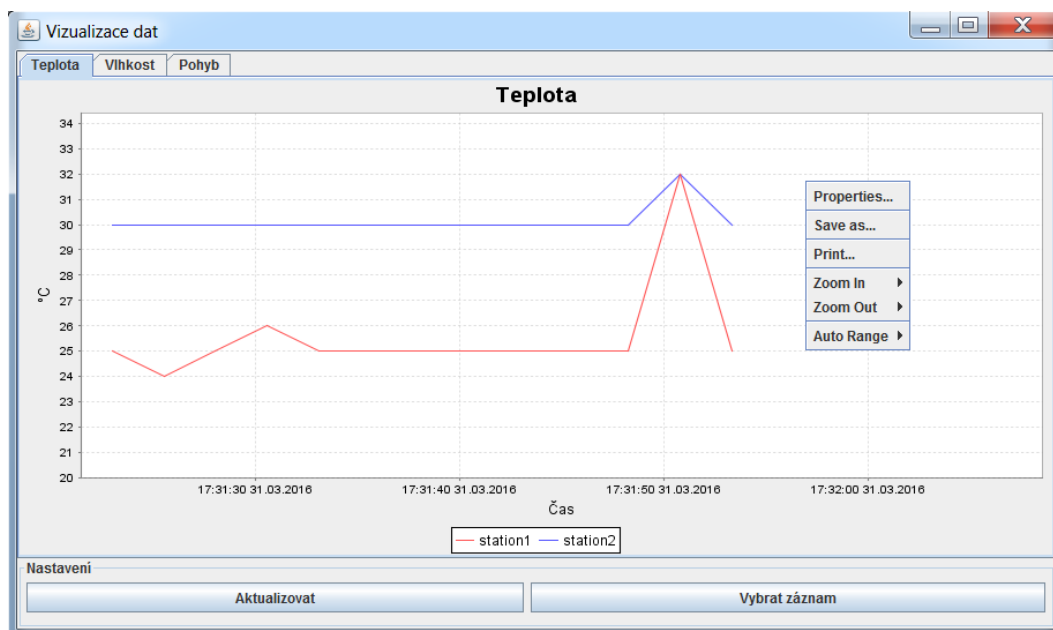
Obrázek A1: Hlavní okno aplikace

Stisknutím tlačítka *Nastavení* se zobrazí okno *Nastavení* stanice (viz obr. A2). V panelu *Stav* stanice jsou obsaženy základní údaje o stanici, tzn. ID stanice, stav DHT a stav PIR. Obsahuje také tlačítko *Odeslat konfiguraci*, které odešle zvolené nastavení monitorovací stanici. Dalšími panely jsou *Nastavení DHT* a *Nastavení PIR*, kde je zvolená stanice nastavena (interval se zadává v milisekundách). Obsahují také dvě tlačítka pro zjištění aktuálního stavu. Posledním tlačítkem *Uložit* se zvolené nastavení uloží a odešle dané stanici.

Stisknutím tlačítka *Vizualizace* v hlavním okně aplikace se zobrazí okno *Vizualizace dat* (viz obr. A3) k zobrazení vybraných záznamů v podobě grafu. V horní liště lze tyto grafy přepínat (graf teploty, vlhkosti vzduchu a počtu pohybů). Stiskem pravého tlačítka myši lze zobrazený graf nastavit, uložit ve formátu PNG, vytisknout, přiblížit/oddálit a automaticky vycentrovat. V dolním panelu *Nastavení* jsou dvě tlačítka. Tlačítko *Aktualizovat* k opětovnému nahrání a zobrazení záznamů a tlačítko *Vybrat záznam* ke změně vybraného souboru, který obsahuje záznamy.



Obrázek A2: Okno pro nastavení stanic



Obrázek A3: Okno vizualizace dat

B Obsah příloženého média

Paměťové médium (DVD) přiložené k této práci obsahuje tyto adresáře a soubory:

- system - Adresář obsahující všechny potřebné soubory ke spuštění PC aplikace.
- src_ard - Adresář obsahující zdrojové kódy monitorovací a centrální stanice.
- lib_ard - Adresář obsahující knihovny využité v monitorovací či centrální stanici.
- StudnickaDavidA12B0170P_BP.pdf – Text bakalářské práce ve formátu PDF.
- readme.txt – Popis struktury na přiloženém DVD.