



Středoškolská technika 2018

Setkání a prezentace prací středoškolských studentů na ČVUT

MeteoCam - Meteorologický kamerový systém

Jiří Pašek, Radek Švamberg

Střední škola informatiky a finančních služeb
Klatovská 200 G, Plzeň

MeteoCam - Meteorologický kamerový systém

MeteoCam - Meteorological Camera System

Autoři: Jiří Pašek, Radek Švamberg

Škola: Střední škola informatiky a finančních služeb, Klatovská
200G, 301 00 Plzeň

Kraj: Plzeňský kraj

Konzultant: Mgr. Bc. Petr Zima

Plzeň, 4. 1. 2018

Prohlášení

Prohlašuji, že jsem svou práci vypracoval/a samostatně a použil/a jsem pouze prameny a literaturu uvedené v seznamu bibliografických záznamů.

Prohlašuji, že tištěná verze a elektronická verze soutěžní práce jsou shodné.

Nemám závažný důvod proti zpřístupnění této práce v souladu se zákonem č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších předpisů.

V Plzni dne 4. 1. 2018

Jiří Pašek

V Plzni dne 4. 1. 2018

Radek Švamberg

Poděkování

Tímto bychom chtěli srdečně poděkovat panu Igoru Otto Singerovi, jehož nápadem jsme se dokázali dostatečně inspirovat až ke zpracování tohoto významného projektu v rámci našeho středoškolského studia.

Dále bych chtěl poděkovat svým rodičům, kteří mě v práci na projektu plně podporovali a zároveň mi finančně přispívali na nákup potřebné techniky a materiálů potřebných k realizaci našeho projektu. Poté také panu učiteli Zimovi, za vedení naší práce a poskytnutí užitečných rad při naší tvorbě.

- Jiří Pašek

Také bych rád srdečně poděkoval panu Singerovi. Dále chci poděkovat svým rodičům, kteří mě podporovali, našemu panu učiteli Zimovi, za vedení práce a také přátelům za konstruktivní kritiku a tipy na zlepšení naší práce.

- Radek Švamberg

Anotace

V našem projektu jsme se zabývali především meteorologickou stanicí a prezentací dat z ní. Využili jsme znalostí z oblastí programování hardwaru, elektrotechniky a designu a programování webových stránek. Hardwarově jsme využili Raspberry Pi s přídatnými moduly a webový server. Na serveru je uložena webová stránka a snímky pořízené Raspberry Pi. Dosáhli jsme toho, co jsme chtěli, čili máme funkční webovou stránku, kde prezentujeme nasbíraná data a funkční meteorologickou stanicí s využitím RPI zařízení. Také pokud by kdokoliv chtěl, tak mu rádi poskytneme veškeré zdrojové kódy, takže také naplňujeme naši open-source myšlenku. Dále jsme rozebrali systémy podobné tomu našemu a vypsali jejich klady, zápory a náš názor na ně. K našemu cíli jsme se dostali bez větších překážek, za což jsme rádi. Z celého projektu jsme se oba určitě hodně naučili a označujeme ho za velmi pozitivní zkušenost. Projekt plánujeme dále rozvíjet a přidávat nové funkce.

Klíčová slova

Kamera; Meteorologický systém; Internet věcí; Udělej si sám

Annotation

In our project, we've dealt with mostly meteorological station and presentation of the recorded data. We've also used our knowledge from the fields of hardware programming, electrical engineering and designing and programming websites. Hardware wise, we have used one device Rapsberry Pi with additional modules and a web server. There is a saved webpage and pictures taken by Raspberry Pi on the server. We've achieved what we've wanted, ergo we have a functioning website, where we are presenting collected data and we also have a working meteorological station using RPI device. Also if anyone would want, we will gladly provide them every source codes, so we are also fulfilling our open-source thought. Also we have analysed systems similiar to ours and wrote their pros, cons and our opinion of them. We got to our goal without any major hurdles which we are very happy about. We've both learned alot from the whole project and we would label it as very positive experience. We are planning to continue developing our project and add new functions.

Keywords

Camera; Meteorological system, IoT (Internet of Things), DIY (Do it yourself)

Obsah

Úvod	7
1. Dostupnost podobných projektů v online světě	8
1.1. Deutschland live	9
1.1.1. Zhodnocení - klady a zápory	12
1.2. Hydronet.cz	13
1.2.1. Zhodnocení - klady a zápory	15
1.3. Kamery.humlnet.cz	15
1.3.1. Zhodnocení - klady a zápory	17
1.4. Webcams.travel	17
1.4.1. Zhodnocení - klady a zápory	19
2. Ustanovení vhodných postupů při tvorbě takového projektu	20
3. Proces vývoje	22
4. Hardware	31
4.1. Programování hardwaru	31
4.2. Přídavné moduly	31
4.3. Vývoj designu a vzhledu zařízení	34
5. Generování animací	37
6. Návrh a provedení webové aplikace	39
6.1. Struktura	40
6.2. Databáze a dohledatelnost	40
6.3. Archivace	42
Závěr	45
Použitá literatura	46
Seznam obrázků a tabulek	47
Příloha DVD: Zdrojové kódy, skripty a programové struktury	48

Úvod

Náš projekt vznikl původně při praxích v Německu, které zařizuje každoročně škola. Zde se Jiří Pašek dozvěděl u pana Igora Otto Singera, že má fungující kamerový systém s webovou stránkou. Projekt ho hned zaujal a po návratu do ČR za mnou přišel, zda bych nechtěl s ním udělat naši vlastní meteorologickou stanici. Tímto náš projekt vznikl, načež jsme vybrali jméno - **Meteo(rologická)Cam(era)**. Cílem naší práce je vybudovat meteorologickou stanici, díky které by mohl kdokoliv vidět na místě kamery animace, snímky a informace o počasí. Samotné téma nás zaujalo a také otevírá možnosti zlepšit si dovednosti v oblasti IT, i zkušenosti s vytvářením nějakého projektu. Také jsme chtěli, aby zařízení bylo plně open-source a all-in-one device, což znamená, že kdokoliv by chtěl, tak by si mohl koupit své Raspberry Pi a pomocí našeho kódu si postavit stejnou meteorologickou stanici. Dále to umožňuje také ostatním uživatelům přijít a navrhnout v našem kódu vylepšení a např. přidávat nové funkce. Naším dalším cílem bylo přehledně prezentovat data, která sbíráme a použít různé metody jejich prezentování (grafy, statistiky).

1. Dostupnost podobných projektů v online světě

Jak už se zprvu může zdát, náš projekt není ani zdaleka jediný svého druhu, ať už se zaměříme na tuzemské nebo celkové světové četnosti podobných systémů, které jsou přímo zaměřeny na obdobnou problematiku jako např. sledování a záznam meteorologických jevů, pravděpodobně dojdeme při nejmenším až k počtu několika desítek tisíc. Počátkem tohoto globálního trendu může být v současné době hned několik důležitých faktorů. Jedním z nich je nejspíše cena kamerových snímacích čipů, které jsou již v současném světě 21. století naprosto běžnou součástí jakýchkoliv elektronických zařízení od chytrých telefonů až po chytré parkovací mechanismy u moderních vozidel. Dalším důležitým faktorem je zcela určitě globální pokrytí a dostupnost Internetu. Člověk tedy už může mít takovou IP kameru připojenou do sítě téměř všude, jelikož např. Česká republika je již pokryta z 99% při nejmenším mobilní bezdrátovou sítí. Dnes také zažívají velký rozmach všemožné projekty z řady Internetu věcí apod. Takové práce poté mohou např. sledovat a různým způsobem vyhodnocovat digitální data, která byla pořízena speciálními moduly navržených přesně pro sběr těchto informací. Tímto způsobem mohou být získávána a archivována data meteorologická a podnební, mezi něž může patřit právě i teplota v nejrůznějších stupnicích, relativní vlhkost vzduchu, atmosférický tlak, UV index slunečního záření ba i nadmořská výška v aktuální GPS geolokaci. Pomocí těchto technik v kombinaci s kamerovým modulem a popř. nějakým ze stále již výkonnějších a zároveň dostupných mikropočítačů, které je možno velice snadno programovat a modulovat, vznikne tím právě to užitečné zařízení, které dokáže pravidelně poskytovat místy i mnohem zajímavé informace. Takovéto meteostanice jsou na současném trhu již nepříliš neobvyklou záležitostí, ale téměř nikdy není spotřebiteli zcela dovoleno nahlížet do kódu a jednotlivých struktur zařízení samotného, natož jej poté nějak modulovat a přizpůsobovat přímo konkrétním potřebám daného spotřebitele.

Cílený zákazník může požadovat uzavřený systém nabízený na současném trhu, ve kterém bude mít spoustu struktur a funkcí naprosto uzavřené jeho uživatelskému oku, jelikož mu budou zobrazována pouze výstupní data a nic jiného. Kladem těchto zařízení může být jenom jejich spolehlivost zaručena samotnou značkou. Zákazník ~~by aleby ale~~ také mohl z opačného hlediska poptávat systém, který se přímo přizpůsobí jeho nejružnějším potřebám za cenu pouze krátké konfigurace uzpůsobené vlastností plně otevřeného hardwaru i softwaru. Tento proces by jednak přinesl spoustu praktických i dovednostních zkušeností, ale také by tím zákazník dostal mnohem větší povědomí o tom, jak celé zařízení funguje a v čem je jeho princip činnosti. Je ale pouze diskutabilní, zdali by se nadpoloviční většina spotřebitelů přiklonila spíše k variantě tržní, která je ale za cenu spolehlivosti značně omezená v již předem zmíněné modularitě, anebo spíše k variantě sofistikovanější, která je zákazníkovi naprosto otevřená ve všech směrech.

1.1. Deutschland live

Nejprve by bylo asi nejvhodnější si řádně zhodnotit a popsat jeden ze systémů, který nám přímo posloužil k inspirativním účelům při návrhu a poté i v samotném vývoji našeho projektu. Jedná se o službu s již výše zmíněným názvem Deutschland live. Z tohoto označení lze snadno odvodit fakt, že se nejedná o tuzemský systém, ale jde o zahraniční záležitost, konkrétně je tedy německého původu. O této zajímavé aplikaci jsme se dozvěděli v březnu roku 2017, kdy nám byla představena přímo jejím autorem, jehož jméno zní Igor Otto Singer, který ji až do dnešního data sám spravuje v plné míře. S jeho prací nás seznámil již v bavorském Waldmünchenu během naší odborné praxe v rámci našeho studijního oboru a to Informačních technologií. Během této praxe jsme se tedy podrobně seznámili se samotnou funkčností a strukturou takového kamerového systému, a jelikož nás projekt zaujal, rozhodli jsme se vytvořit si podobnou práci, ale pojmout ji mírně v odlišném smyslu a pokusit se ji více rozvést.

Tento kamerový systém je společně i s jeho výstupy všem plně dostupný v rámci internetové sítě. Lze na něj běžně přistupovat zadáním veřejné IP adresy serveru, na kterém ze kterého je celá služba řízena a konfigurována pro získání potřebných výstupů. Pro tento server bohužel není nakonfigurován žádný překlad doménových názvů, tudíž se spotřebitel musí přímo odkazovat na veřejnou IP adresu ve formátu 217.5.208.3, kde je již nakonfigurován http server na portu 80. Co se samotného serveru týče, jedná se o stanici, na níž je nainstalován operační systém Linux (konkrétně s obrazem Debian) a která je přímo dosažitelná z Internetu, díky její přidělené veřejné adrese a také za pomoci přístupové struktury http serveru s názvem Apache. „Apache HTTP Server je softwarový webový server s otevřeným kódem pro GNU/Linux, BSD, Solaris, Mac OS X, Microsoft Windows a další platformy. V současné době dodává prohlížečům na celém světě většinu internetových stránek.[1]“ Na tento server jsou poté v pravidelných intervalech odesílány snímky, které jsou zprostředkovány pomocí hned několika IP kamer. Původní systém Deutschland live je navržen až pro 1728 jednotlivých kamer. Každá z jednotlivých kamer je zejména v odlehlých krajinných oblastech připojena pomocí směrové antény. Napájení je -zejména -v -těchto -specifických -a -jen -těžce- přístupných- místech -vyřešeno -vskutku

příznivě a zároveň i ekologicky, využívá se zde totiž solární energie, která je poskytována přímo připojeným kompaktním panelem. Použité kamery disponují možným rozlišením pohybujičím se od 2 do 4 Mpx. Většina snímků je ale pořizována v základním rozlišení HD tj. 1280x720 px. Tyto snímky jsou poté v určité kaskádě převáděny do rozlišení 800x600, které je pro vystavení na webové stránky jednak příznivější, ale také se zde díky nižšímu datovému objemu ušetří i značné množství úložné kapacity. V tomto systému jsou také z výstupů několika kamer tvořeny i časosběrné animace, i když jejich princip tvorby zde funguje mírně jiným způsobem, než si může člověk zprvu představovat. Jsou zde totiž dostupné animace, které se spíše chovají jako tzv. loop. V českém jazyce bych toto označení definoval jako - zpětná animační fronta snímků.

Režim takového loopu bych popsal asi nejpřesněji tímto postupem. Nejprve je pořízen snímek z kamery, který je hned v zápětí odeslán na náš server. Určité programové funkce a skripty si tento snímek přeberou, vytvoří z daného snímku kopii s časovou značkou a menším rozlišením, které je lépe uzpůsobené pro webové prostředí a výstupní kopii již po tomto procesu přesouvá do adresáře, který je pak určen pouze pro vystavení na webové rozhraní. Syrový snímek v původním rozlišení se ale ovšem také nikam zcela neztrácí, je totiž znovu vzat určitou programovou strukturou a je zde provedena tvorba další kopie, která je opět převedena do jiného rozlišení, zde se jedná konkrétně o ~~velice~~[nízké](#)[velice nízké](#) rozlišení 160x120 px, které je zde používáno pro generování zpětných animací. Tato kopie je poté přesunuta do adresáře, který slouží jako fronta. Když uplyne interval a je přidán nový snímek do fronty, je tím vyvolán proces pro zpracování dostupných materiálů do animace ve formátu GIF. Tato výsledná animace je tedy vždy dostupná webovým stránkám, ale hlavně je vždy plně aktuální a nachází a obsahuje v každém čase i svůj nejaktuálnější snímek z dané kamery, když ovšem zanedbáme nepatrnou prodlevu způsobenou jednak přenosem snímku z kamery a také procesem tvorby animace. V tomto projektu jsou v současné době zavedeny tři struktury tvorby zpětných front, které zachycují časy 90 minut, 24 hodin a 7 dní. Navržený algoritmus dokáže také zachycovat i mnohem delší časové frekvence, je vysoce škálovatelný a snadno přizpůsobivý změnám v jeho konfiguraci, tudíž není nemožné s touto strukturou zobrazovat zpětné fronty snímků např. z posledního roku nebo rovnou několika po sobě jdoucích let. Pan Singer, který je jak jsme již zmiňovali tvůrcem systému Deutschland live, nazývá svoji vyvinutou technologii zpětných animačních front snímků jako ICS.

Další zajímavou vlastností pro práci se snímky je ještě jedna ze zajímavých technik, která je na webových stránkách projektu označována jako IRS. Tato funkce načítá stejně jako ICS snímky pořízené kamerou do jednoho konkrétního adresáře na serveru ve frontě od nejnovějšího po nejstarší v předem zvolené časové hranici. Tyto snímky se poté díky vyvolanému JavaScript skriptu převádějí a načítají se na webové stránce do sekvenční lišty, která je seřazena směrem z levého okraje od nejnovějšího až po nejstarší snímek v dané časové hranici ke kraji pravému. Když poté návštěvník najede kurzorem myši na jeden z obrázků v sekvenci, vyvolá se JavaScript příkaz, který zapříčiní promítnutí daného snímku v plném rozlišení a s časovou značkou v prostoru webové stránky přímo

pod umístěním časové lišty obrázků. Reálná AJAX technika by znamenala, že se hned po najetí kurzorem na danou miniaturu spustí na serveru příkaz, který klientovi stáhne a zobrazí již tedy předem zvolený požadavek. Tato technologie ve své podstatě využívá pouze tzv. pseudo-Ajax principy sloužící pro dynamické načítání obsahu přes Javascript, což znamená, že v tomhle případě již není vyvoláván žádný požadavek na server, ale všechna data jsou načtena klientovi už při samotném načtení stránky. Nejedná se tedy o reálnou AJAX techniku. Tato metoda ale není ve všech případech ~~plně~~ spolehlivě spolehlivá, některé prohlížeče ji příliš nerozumí a poté zde dochází ke zkrácení chtěného výstupu. Zde si můžeme uvést i malý příklad pseudo-Ajax skriptu. Tento konkrétní skript je napsán ve webovém skriptovacím jazyce Javascript, který je sestaven na základě javascriptové knihovny jQuery, která jednak podporuje celou řadu nejrozličnějších prohlížečů, ale také klade značný důraz na spolupráci mezi samotným JavaScriptem a značkovacím jazykem pro webové stránky HTML. Zde je příklad provedení na webové stránce systému a pod ním se dále nachází část javascriptového kódu pro lepší povědomí o samotné funkci.

Lišta pro dynamický výběr snímku:



Obrázek 1: Výstřižek z webu: 217.5.208.3

V horní části se nachází lišta pro již zmíněný dynamický výběr snímku, pod touto sekvencí se již promítá daný zvolený snímek. Levý a pravý okraj již poté tvoří odkazy pro zobrazení výstupů z ostatních kamer. Ukážeme si část kódu, jak to celé pracuje. Pro

objasnění můžete využít krátké vysvětlivky v komentářích jednotlivých příkazů.

```

this.imagePreview = function(){ // Vytvoření a nastavení názvu funkce

// Struktura a tělo funkce:

    $("a.preview").hover(function(e){ // Metoda hover slouží jako spouštěč, když je najeto na obj-
        this.t = this.title; // ekt s ním spjatým kurzorem myši.
        this.title = "";
        var c = (this.t != "") ? "<br/>" + this.t : "";
        $("body").append("<p id='preview'><img src='"+ this.href +"' alt='Image preview' />" + c + "</p>");
        $("#preview")
            .css("top",1)
            .css("left",1)
            .fadeIn("fast");
    },
    function(){ // Když kurzorem z objektu odejedeme, zobrazí se opět výchozí snímek.
        this.title = this.t;
        $("#preview").remove();
    });

```

Obrázek 2: Ukázka zdrojového kódu z: 217.5.208.3

1.1.1. **2.1.1. Zhodnocení - klady a zápory**

Systém Deutschland live se nám zdá velice zajímavý a jistě užitečný, ale nemyslíme si, že by byl plně využit jeho obrovský potenciál, což taky bylo jedním ze příčin rozhodnutí započít vlastní projekt. V celkové struktuře je sice zapojeno velké množství kamer, ale spotřebiteli resp. návštěvníkovi může přijít systém přepínání mezi výstupy z jednotlivých kamer mírně matoucí a nepříliš přehledný. Tento problém by bylo potřeba vyřešit, návštěvník se totiž musí za každou cenu umět bez starostí pohybovat ve struktuře webu, ale i celé domény. Zde přímo narážím na další nepříjemnost. Celý systém je sice dosažitelný pro lidi v celém Internetu, ale za vskutku nepříznivou cenu. Konkrétní webovou stránku lze dohledat pouze zadáním již dříve zmiňované veřejné IP adresy serveru <http://217.5.208.3/>, což samozřejmě vyžaduje přímou znalost dané adresy, jelikož žádný ze známých a dosti používaných vyhledávačů neindexuje a tudíž je nedokáže vyhledat, a tak vzniká samozřejmě obrovský dopad na návštěvnost serveru. Tento fakt lze tedy mírně ulehčit tím, že systém poskytuje spíše odkazy webům jistých měst a lokalit, které zkrátka obsahují jednoduché widgety stahující aktuální snímky a data ze systémových stránek. Kamery jsou také propojeny do českého systému cz.webcams.travel, kde jsou již indexovány webkamery téměř z celého světa. Co se tedy stránek samotných týče, připadají mi převážně neudržované. Hned na úvodní stránce je např. několik odkazů pro přístup k výstupům jednotlivých kamer, které ale bohužel nejdou otevřít, jelikož nejspíše sloužily jako odkazy v minulosti, ale poté byly vypnuty. Tudíž se návštěvník nedokáže na většinu kamer z úvodní stránky odkázat. Další zajímavou vlastností tohoto projektu, je množství kamer a jejich celkové území, kterým jsou kamery pokryty, jelikož se nenacházejí pouze v okolí bydliště jejich správce, ale ačkoli se systém nazývá „Deutschland live“, několik kamer se nachází i mimo Německo. Jedna z kamer zachycuje např. park s dětským hřištěm v Brně, jiná zase kdysi zobrazovala výhled na Středozevní moře z italského pobřeží. Snímky jsou ale pořizovány i z míst s extrémními podmínkami např. z vrcholu nejvyšší německé hory Zugspitze. Spolehlivost dodání snímku z kamery v daném intervalu se u jednotlivých aparátů liší, některé jsou totiž napájeny solární energií, tudíž když je zataženo, tma, nebo se vyskytuje mlha, dochází tím k poklesu energie v solárně nabíjeném akumulátoru, a tudíž dochází ke značnému časovému deficitu, kdy server neobdrží žádný nový snímek a

kamera žádné nepořizuje. Příkladem může být např. stanice na šumavském kopci Čerchov.

Na celém tomto systému je jistě zajímavé např. pokrytí, ale nejvíce nás zde zaujaly principy tvorby animací ICS a zároveň také pseudo-Ajax technologie IRS. Proto jsme se rozhodli do našeho projektu převzít technologii ICS, která by ale nebyla příznivá pro výkon malého jednoduchého jednodeskového počítače plošných spojů Raspberry Pi, který jsme se díky jeho snadné dostupnosti rozhodli použít jako základ celého kamerového meteorologického zařízení. Pseudo-Ajax technologii IRS jsme se zatím rozhodli nepoužívat, jelikož jednak není zcela příhodná pro výkonnost serveru, na kterém běží naše webové rozhraní a navíc není plně kompatibilní se všemi dostupnými webovými prohlížeči. Dále jsme se shodli rozšířit náš projekt oproti tomuto zakomponováním struktur pro získávání reálných meteorologických hodnot za pomoci speciálních modulů. Práce pana Singera se nám ovšem velice líbí a jednoznačně v ní vidíme jistý potenciál, byla by zde ale potřeba plně dokončit funkčnost a vzhled webového rozhraní a také dodat překlad doménových názvů. Každopádně děkujeme za nápad, který nám pan Singer vnukl, tento projekt nám posloužil jako velká inspirace.

1.2. ~~2.2.~~Hydronet.cz

Dalším vybraným pro rozbor je již tuzemský systém nesoucí název Hydronet. K této práci se nám bohužel nepodařilo dohledat takové množství materiálů a informací jako např. u výše rozebírané práce, jelikož jsme se zkrátka s tvůrci projektu nedokázali spojit a tudíž můžeme daný projekt hodnotit a rozebírat pouze jako běžní návštěvníci určené webové stránky prostřednictvím výstupů a dat, které jsou na zmíněném zdroji poskytnuté. Jak se poté chovají dané programové struktury, které jsou nám skryté, můžeme jenom spekulovat a domýšlet jejich funkčnosti na základě předem rozebíraných prací.

Hydronet je označení pro spolek několika lidí, kteří se zabývají volným šířením online meteorologických a obrazových informací, zejména se poté specializují na snímání české přírody. Zde si uvedeme několik oficiálních informací o spolku ze serveru Hydronet.cz.

„Spolek vznikl původně jen jako diskusní klub, reakce na povodně roku 2002. Tvůrcem původních stránek spolku, používaných (například diskusní stránka i další) až do nedávna, byl tehdejší student MFF UK Vojtěch Hála. O této nejstarší historii spolku se lze dočíst v <http://hydronet.cz/vzkazy.php>. Zaměření na výstavbu vlastních stanic a kamer přišlo až se vstupem Josefa Zemana, který již předtím provozoval četná automatizovaná měření na stanici Pernštejn. Koncem roku 2003 následovala v rychlém sledu výstavba stanic jím navržených a realizovaných: Železný Brod a Jizerka. V dalším roce pak Příchovice, Držkov, Kamenec, Lysá hora, Čertovo břemeno a roku 2005 Vrbatka, Kořenov a Střezimíř. Roku 2009 přibýly stanice Svatý Jan a Rejdice. (Některé z dřívějších stanic z různých důvodů později zanikly, resp. přerušily činnost na určitou dobu.) Postupem doby se hlavní osobou spolku, na jehož bedrech spočinula většina technické práce, stal Jiří Přibáň.[2]“

Nápad tohoto společenství několika nadšenců do podobného projektu je velice dobře promyšlený, protože má každý z jednotlivých členů nějaký svůj specifický obor práce, o kterou se dále starají a zajímají. Jejich webové stránky jsou skvěle provedeny, nevidíme zde žádný problém potenciální neschopnosti návštěvníka se na stránce orientovat a dohledat si potřebné informace. Tento projekt není postaven na žádné komerční bázi, jedná se pouze o neziskové občanské sdružení, a tudíž nemá žádný člen spolku ani nejmenší finanční zisk. Veškerá činnost v tomto spolku je zcela dobrovolná. Na stránkách je ale mimo to dostupná i sekce sponzoringu, kde mohou různí investoři a příznivci projektu přispívat libovolné finanční částky pro náklady spojené na samotný provoz a administraci systému. Na stránkách se také nachází i jedna velice zajímavá a užitečná záložka, která slouží jako odkaz na diskuzní fórum této sítě. Zde se již poté nacházejí pouze diskuze a příspěvky daných členů společenství. Nejčastěji tyto příspěvky tvoří vcelku podrobné rozebírání meteorologických stavů za poslední den nebo několik hodin. Čas od času se zde i jednotliví správci určitých stanic chlubí i krásou a jedinečností snímků, které jim dané stanice za poslední období pořídily. Místy se zde objevují i různé články z oboru meteorologie a celkově vědy, o kterých zde účastníci diskuze bouřlivě diskutují. Pokud se zdá nějaký z pořizovaných snímků administrátorům velmi atraktivní, rozhodnou se jej poté vystavit do sekce Top snímků.

Tento projekt o sobě prohlašuje, že vznikl již v roce 2003 jako první struktura s volným online snímáním počasí, přírody a krajiny pomocí kvalitních digitálních kamer. Když se nyní zaměříme na samotné kamery, je nutno konstatovat, že jednotlivé stanice opravdu nepostrádají žádný prvek kvality, jelikož jsou snímky ve vskutku vysokém a neobvyklém rozlišení s dobrou ostroť, což není u podobných projektů zcela obvyklým standardem. Celý systém nabízí k současnému datu na svých stránkách celkem 22 pohledů jednotlivých kamer a stanic. Celková pokrytá území měření a snímání v současnosti tvoří oblasti Krkonoš, Jizerských hor, České Sibiře a Středního Povltaví. Umístění stanic je navrženo tak, aby bylo pro návštěvníka co nejvíce atraktivní a zajímavé např. s výhledem na zapadající slunce v Krkonoších nebo se záběrem na přímý říční tok Jizery v Kořenově. V



Obrázek 3: Pohled na říční tok: hydronet.cz

rozhraní několika stanic na webových stránkách jsou také dostupné zajímavé informace o místě umístění, jeho historii a také konfigurace dané stanice. Jednotlivé webkamery poskytují snímky v odlišných časových intervalech na základě jejich konfigurace, většinou ale každá z do systému přímo zapojených

stanic pořizuje celkem patnáct snímků po čtyřech minutách za jednu celou hodinu. Když kamera pořídí za daný interval nový snímek, nejprve ho označí meteodaty získaných pomocí speciálního čidla a poté jej posílá pravděpodobně přes FTP na server webových stránek, kde to již zvolená programová struktura zaregistruje a vystaví snímek jako nejnovější a nejvíce aktuální. Tento princip je většinou u veškerých takovýchto projektů téměř principiálně stejný. Na tomto systému nelze nalézt žádné podobné animační struktury nebo dynamické vybírání ve zpětné frontě pořízených snímků jako u projektu předchozího. Jako alternativa za tyto metody se na stránkách vyskytuje pouze jedna metoda zpětného vyhledání a promítnutí. Funguje to tím způsobem, že se jednoduše po rozkliknutí odkazu kamery načte pod nejnovějším snímkem pouze galerie vytvořená během nynějšího dne. Zobrazí se tedy přesněji struktura, kde jsou snímky z daného dne seřazeny od půlnoci až do aktuální hodiny vždy po patnácti snímkách za jednu celou hodinu. Je zde ještě možnost zobrazit si stejným způsobem i materiály pořízené v rozsahu sedmi předešlých dní. Nevyskytuje se zde bohužel žádná vlastnost, která by nám dovolila vyžádat si data pořízená během delších a starších časových intervalů, jediná příslušná možnost by mohlo být pouze pokusit se zkontaktovat samotné správce a získat data od nich. Na úvodu webové stránky ale lze dohledat informaci, že se pracuje na nových funkcích pro web, možná se zde i do budoucna objeví nějaká z nových a zajímavějších struktur.

1.2.1. ~~2.2.1.~~Zhodnocení - klady a zápory

Celý systém a vůbec myšlenku spolku Hydronet.cz bychom zhodnotili spíše kladnými dojmy nežli těmi zápornými. Líbí se nám zejména to, že se jedná o společenství několika nadšenců do daného oboru, jelikož v tomto případě plní každý svojí část zájmu, který díky konkrétní specializaci dokáže provádět kvalitněji a spolehlivěji. To bylo asi hlavním důvodem, určit si před prací na praktické části přední dovednosti členů týmu a jejich obory, kterým se budou během práce věnovat a zpracovávat je. Také nám přijde příznivá crowdfundingová taktika a možnost sponzoringu projektu. Kamery se vyskytují na zajímavých místech se skvělým výhledem a mimo to získávají i meteorologická data o počasí, což lze požadovat za další klad. Nicméně asi za jediný větší zápor tohoto projektu lze považovat celkovou absenci jakýkoliv animačních procesů s danými výstupními snímky. Jedinou alternativou pro tyto metody zde může být pouze galerie výstupů pořízených v posledních sedmi dnech. Možná se ale již v blízké budoucnosti doplní nějaké další zajímavé funkce, které tento nedostatek odstraní, vzhledem k informacím dostupným na webových stránkách. Dále jsou kamery v systému stabilní a spolehlivé v poskytnutí dat. Většina z nich je poté dále propojena s různými regionálními stránkami měst a obcí. Tento systém nám také posloužil částečnou inspirací v práci na našem projektu.

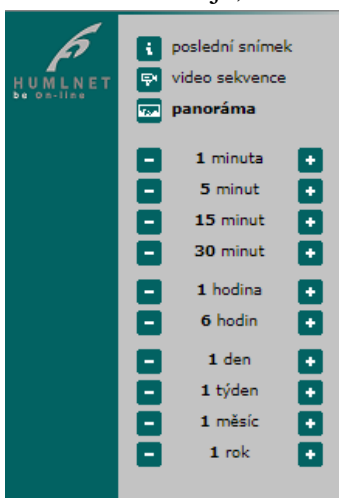
1.3. ~~2.3.~~Kamery.humlnet.cz

Dalším zajímavá práce, na kterou jsme narazili, nese název Humlnet.cz. Tento projekt nejprve začínal již v roce 1999 jako amatérská sdílená síť několika přátel mezi čtyřmi

panelovými domy ve městě Trutnov. Byla zde zprovozněna také poštovní schránka @humlak.cz, kterou používali členové této sítě pro komunikační účely. Postupem času se do této sítě začalo připojovat stále více uživatelů a tím se rozrůstala i samotná infrastruktura. Členové této sítě začínali již v té době objevovat potenciál Internetu a jeho roli v každodenním životě lidí ve 21. století, proto se rozhodli zapojit svoji síť do té internetové a to se jim dne 10. 5. 2000 skutečně podařilo uskutečnit. V tomto roce se také do sítě připojila první webová kamera, která snímala panoráma města Trutnov. V srpnu roku 2005 se přidala další webkamera, která již disponovala rozlišením 3 Mpix a byla použita pro snímání kultovního Open Air Music festivalu konaného v tomto čase. Postupem času poté začalo přibývat množství těchto kamer a meteorologických stanic zejména v okolí severovýchodní části České republiky a pohoří Krkonoše.

Webové rozhraní tohoto projektu je velmi dobře provedené a přehledné. Tvoří jej totiž pouze odkazy pro zobrazení snímků z kamer samotných a interaktivní pole, kde je možné si vyžádat konkrétní výstup ze zvoleného času a data času a data. Dále se zde ještě nachází pole s aktuálními meteorologickými údaji získanými pomocí meteostanic na místech s kamerami. Jedná se spíše ale pouze o teploty ve stupních Celsia. V systému je také možnost zúčastnit se a řešit různé aktuální problémy a témata v dostupném diskuzním fóru, které je velice podobné tomu z předchozího rozebíraného systému, kde se objevovali příspěvky orientované na zachycené dění stanicemi, ale i jiné předměty např. z přírodních věd. Velkým kladem tohoto diskuzního fóra je, že se diskuze může zúčastnit zcela každý

Location	Temperature
Trutnov	-0.1°C
455 m n.m.	
Bouda na Muldě	-4.6°C
1046 m n.m.	
Pec pod Sněžkou	-2.3°C
834 m n.m.	
Velká Úpa	-3.9°C
780 m n.m.	
Sněžka	0.0°C
1602 m n.m.	
Bouda pod Sněžkou	-1.8°C
m n.m.	
Luční bouda	-5.3°C
1410 m n.m.	
Prkeňák	-4.4°C
780 m n.m.	
Dvořákova bouda	-4.6°C
1235 m n.m.	
Litoměřice	0.0°C
163 m n.m.	



Obrázek 5: Výstřižek z: kamery.humlnet.cz

snímek obsahující v názvu tento řetězec. Pokud již není hledaný snímek v archivaci nalezen, je uživatel upozorněn příslušnou hláškou. Je zde také odrážka pro zobrazení

návštěvník stránky bez nějaké potřebné registrace, stačí totiž pouze vyplnit jméno a daný text zprávy pro přidání příspěvku.

Kamery, které jsou již součástí projektu, jsou nakonfigurovány pro pořizování nového snímku každých šedesát vteřin. Pro každý pořízený snímek je také možnost volby z až 4 dostupných rozlišení - 640x480, 1024x768, 1600x1200 a 2048x1536. Jak už již dříve bylo zmíněno, tento systém disponuje velice neobvyklým mechanismem zpětného zobrazování a vyhledávání snímků. Funguje to tím způsobem, že je nám dáno speciální pole s možností přičítání a zpětné odečítání předem zvolených časových délek za určením snímku označeným daným datem a časem. Pracuje to pomocí javascriptové programové struktury, jež podle zvoleného řetězce znaků dle data a času vyvolá z archivačního adresáře

panoramatického snímku, ale ten se nám bohužel nepodařilo zobrazit, jelikož bude nejspíše v současné době jednoduše nefunkční nebo v údržbě.

Pokud bychom hledali nějaké animační funkce na této práci, můžeme se pouze odkázat na odrážku videosekvencí. Zde jsou poté dostupná časosběrná videa, která jsou ale ve webovém prohlížeči spustitelná pouze za doprovodu nejnovější verze pluginu Flash. V případě nedostupnosti tohoto pluginu je zde ještě řešení stáhnout si danou videosekvenci na lokální disk svého počítače. Tato stažená videa jsou poté dostupná v základním rozlišení a formátu flv. Videosekvence zde bohužel nemají žádný řád a většinou zachycují pouze několik dní v měsíci, někde ale dokonce žádný, což je jistě velká škoda. Pravděpodobně tento proces tvorby a generování spočívá pouze na správcích samotných.

1.3.1. ~~2.3.1.~~ Zhodnocení - klady a zápory

Na tomto projektu se nám líbila zejména jeho preciznost a způsob provedení. Webové rozhraní sice nepoužívá zcela nejmodernější design, ale to nevytváří celkově žádný dopad na orientaci v prostoru webových stránek pro jakéhokoliv návštěvníka. Velice nás oslovila již výše popisovaná funkce, jež využívá pro vyhledání požadovaného snímku přičítání a odečítání jednotlivých časových délek. Zpočátku jsme přemýšleli nad možností tvorby podobné funkce, ale jelikož jsme již od započetí práce měli omezenou kapacitu možného úložiště pro pořízené snímky, rozhodli jsme se pracovat pouze s databází slouženou z URL odkazů na snímky, které jsou nahrávány do bezplatné online Image sharing služby, nesoucí název Imgur. Na tuto databázi by poté byla aplikována funkce zpětného vyhledávání. Tímto způsobem se nám podařilo zajistit velice značnou nezávislost množství a objemu dat nad našimi úložnými prostředky.

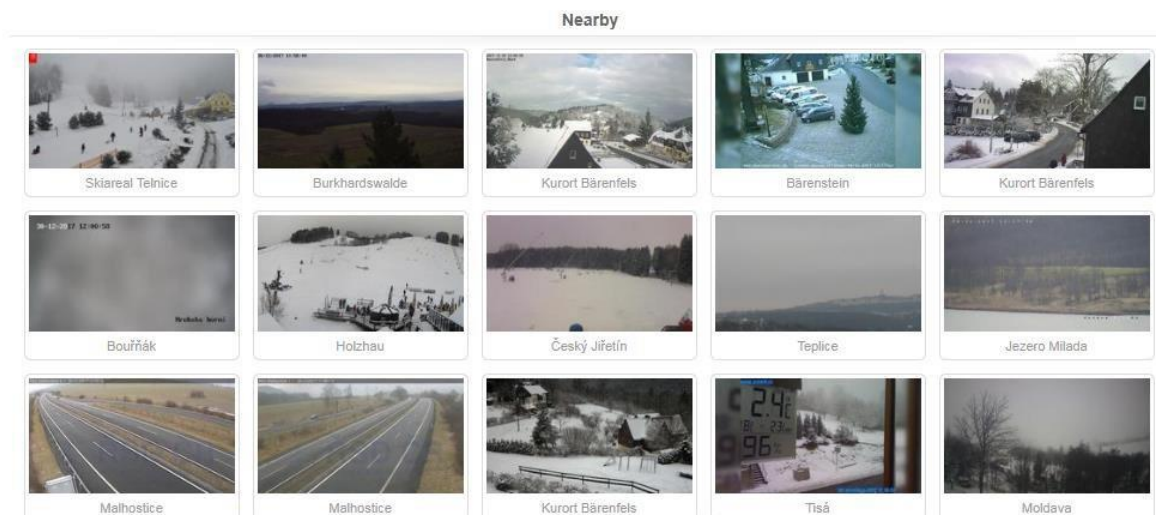
Co se animačních a časosběrných záznamů týče, vidíme v tomto projektu velký nedostatek, ale zdaleka ne tak razantní, jako např. u předchozí práce. Existuje zde pouze jediná struktura video sekvenčních záznamů, která bohužel není nijak pravidelná a nemá žádný řád, proto se může stávat, že tímto vznikne např. prázdné okno v rámci celého měsíce. Navíc jsou tyto sekvence spustitelné ve webovém prohlížeči pouze za nutného předpokladu pluginu Flash, popř. je zde možnost stažení, ale ~~pouze v neuspokojivé~~ pouze v neuspokojivé kvalitě ve formátu flv. To se nám zdá jako problém a nedostatek, tudíž jsme se rozhodli využívat formátu animací GIF, jelikož jsou vždy dostupné bez jakýkoliv nutností. Tento projekt nám rovněž posloužil částečnou inspirací.

1.4. ~~2.4.~~ Webcams.travel

Na posledním projektu, který ohodnocujeme, je web webcams.travel. Tento web má celkově 34437 kamer z celého světa a zaměřuje se především na turistický ruch. Velmi zajímavě je vyřešené hledání daných kamer - možností je několik. Buď můžeme dané kamery vyhledat na Google Mapách, kde jsou zobrazeny na odpovídajícím místě. Dále je zde možnost kamery filtrovat pomocí klíčových slov - toto je velmi hezky provedeno a filtry fungují tak, jak by se dalo očekávat. Poslední možností je hledání samotných kamer

podle názvu lokace, kamery atd. Vzhledem k množství kamer je nezbytné nějaké filtrování či vyhledávání kamer a to si myslíme, že stránka má zvládnuto dobře. I přes velké množství kamer se jednoduše dá najít ta, kterou zrovna chcete.

Dále je na stránce zobrazená náhodná kamera se živým přenosem - toto je podle nás velmi hezká funkce a do stránky dobře zapadá. Jinak se nám ale vzhled stránky příliš nelíbí, působí poměrně nepřehledně a je na ní moc informací - je zde mapa, reklama na různé mobilní aplikace, Google Earth widget a k tomu ostatní informace o kamerách. Nyní k samotným kamerám. Kamery mají velmi hezky vyřešené animace a jejich přehrávač vypadá hezky a elegantně. U některých kamer bohužel animace ale chybí a místo nich se na stránce nachází přehled posledních snímků za nějakou dobu, který se přehraje. U zobrazování animací jsou zde 4 možnosti - posledních 24 hodin, posledních 30 dnů, posledních 12 měsíců a záznam za celou dobu fungování kamery. Stejně jako my mají 24 hodin, 30 dnů a poslední rok, ovšem záznam za celou dobu fungování ne, ale je zajímavý. U daných webkamer je také hezky vyřešeno to, že pokud si nějakou rozklikneme, tak vidíme dole na stránce kamery v okolí.



Obrázek 6: Výstřižek z webu: cz.webcams.travel

Na stránce je také elegantně vyřešeno zobrazování informací o počasí. Provádí se přes aplikaci Wetter. Na další straně je dostupný ilustrační snímek.

Weather Altenberg



Obrázek 7: Výstřížek z webu: cz.webcams.travel

1.4.1. ~~2.4.1.~~ Zhodnocení - klady a zápory

Na tomto posledním projektu nás nejvíce zaujalo především několik různých možností, jak vyhledávat webkamery - přes vyhledávání na mapě, po vyhledávání podle názvů a filtrů, až po zobrazování kamer v okolí u každé z webkamery. Zajímavé je také řešení přehrávače animací, který je velmi moderní, jednoduchý a snadno ovladatelný. Mezi zápory bychom dali především nepřehlednost úvodní stránky - je zde příliš informací na jednom místě a pro nás osobně se zde dá špatně vyznat.

2. Ustanovení vhodných postupů při tvorbě takového projektu

Při prvotní myšlence, jak bychom měli postupovat při tvorbě tohoto projektu, jsme se nejdříve zamysleli, jak by bylo nejvhodnější postupovat a jaký postup bychom měli zvolit. Vycházeli jsme zejména ale z námi ustanovených hodnot, které jsme si již rozebrali v předchozí části rozborů podobných online projektů. Výhody tohoto přístupu byly patrné celou tvorbou - měli jsme práci jasně rozdělenou, ale zároveň jsme si vypomáhali, pokud jsme narazili na nějakou překážku. Nejprve jsme se rozhodli projít si, jak bychom chtěli, aby náš projekt vypadal. Přesně jsme si určili naše cíle - vytvořit funkční a přístupný meteorologický kamerový systém, který by běžel na jednom zařízení. Tímto jsme odstranili, podle nás, poměrně velikou nepřehlednost některých již hotových systémů, kde součástí systému je např. kamera a pak další zařízení, které data z kamery zpracuje. Vše potřebné pro vytvoření nové meteorologické stanice, jako máme my, je možné pouze na jednom zařízení. Díky tomu si může kdokoliv, kdo by se zajímal např. o meteorologii jednoduše vytvořit vlastní stanici. Naším dalším cílem bylo prezentovat data, která získáváme, v přehledné a jednoduše přístupné formě - webové stránky. Ustanovili jsme si přibližné schéma stránky a jaká data by se na ní měla nacházet. Toto se postupně rozšiřovalo - díky modulům do Raspberry Pi jsme časem mohli zjišťovat nejen teplotu, ale i detailnější statistiky, jako je např. vlhkost vzduchu, atmosférický tlak atd. Dále jsme řešili, jakým způsobem budeme data na stránce prezentovat - rozhodli jsme se pro animace, grafy a vypsané hodnoty. Díky tomu data nabízíme v přehledné formě, ve

které se dá jednoduše vyznat. Poté jsme chtěli, aby byla data přístupná i zpětně - ne jen aktuální data. K tomuto účelu vznikl náš archiv, kde se uchovávají staré animace, které jsou nyní přístupné veřejnosti. Dále jsme si jasně určili rozdělení práce a tím tvorba našeho projektu začala.

3. Proces vývoje

Proces vývoje jsme začali nakoupením potřebného hardwaru - samotného zařízení Raspberry Pi a modulů k němu. Pořídili jsme tento model Raspberry Pi:

Rapsberry Pi 3 Model B - Jedná se svým způsobem o malý jednodeskový počítač plošných spojů. Je velice kompaktní a také lehce dostupný, díky jeho nízké ceně. Pomocí integrace tzv. GPIO pinů lze tyto počítače propojovat s dalšími moduly a obvody téměř pro jakoukoliv činnost.

„Raspberry Pi je jednočipový počítač, který je srovnatelný se (slabším) stolním počítačem. Obsahuje vývod pro monitor (HDMI), přes USB je možné připojit klávesnici a myš. Vyvinuto bylo již několik generací tohoto počítače, které se liší výkonem a zamýšleným použitím. Použitý mikroprocesor je z rodiny ARM, takže je srovnatelný s běžným smartphonem. Na počítači Raspberry Pi je možné provozovat různé distribuce Linuxu, RISC OS, jakož i Microsoft Windows 10 IoT Core. Na rozdíl od počítače Arduino je možné Raspberry Pi použít nejen k ovládání různých zařízení (pomocí GPIO kontaktů), ale i k samotnému vývoji příslušných aplikací.[3]“

Rozhodli jsme se vybrat model již ze třetí generace tohoto zajímavého zařízení, jelikož právě ten disponuje největším výkonem a řadou dalších vlastností, které nejsou u modelů předchozích generací k dostání. Je zde např. přímo integrovaný WiFi 802.11 b/g/n modul nebo podpora Bluetooth 4.1 LE. Dále tento model disponuje o 33% vyšší frekvencí procesoru než u postaršího modelu 2B (1,2 GHz x 0,9 GHz) a také modernějšími čtyřmi jádry s



Obrázek 8: Raspberry Pi

efektivnější instrukční sadou, která je zvláště vhodná pro operace pracující s nad 64-bitovými hodnotami. Model je vhodný i kvůli dostatečné velikosti paměti RAM, která je již 1024 MB.

S rostoucím výkonem se samozřejmě zvyšují i požadavky na napájení a chlazení, proto jsme zakomponovali také pasivní chladič a pro uspokojení spotřeby jsme použili microUSB adaptér s napětím 5V a proudem 2,5A. Vybrali jsme ten dosud nejvýkonnější model hlavně z toho důvodu, jelikož pro další práci s obrazovými informacemi pořízenými pomocí kamery a zejména s generováním samotných animací, by vznikl

značný výkonnostní nedostatek a zařízení by tedy tímto způsobem nebylo schopno data zpracovávat. Cena tohoto modelu se v současnosti pohybuje kolem 50\$.

Moduly jsme nakoupili následující:

1. **Adafruit MPL3115A2** - Tento modul nám nyní slouží pro získávání dat o tlakových podmínkách v daném prostředí (atmosférický popř. barometrický tlak). Dále je použit jako tepelné čidlo, v základním provedení poskytuje informace o nynější teplotě ve stupních Celsia, což poté ještě pomocí vzorce přepočítáváme do Fahrenheitovy stupnice. Modul také dokáže poskytnout informace o jeho momentální nadmořské výšce, čehož jsme také využili. Tento modul dále



Obrázek 9: Adafruit MPL3115A2, výstřižek z: adafruit.com

komunikuje s Raspberry Pi pomocí speciální sběrnice I2C, kde je propojen a využívá zde výchozí adresu 0x60. Mezi jeho výhody patří velice nízký nárok na množství dodávané energie, ale také vcelku příznivé přesnosti v poskytnutých hodnotách, např. odchylka u teploty není nikdy více než 1°C. Jeho teplotní rozsah není nijak omezen. Cena za jeden kus se pohybuje kolem 10\$.

2. **Adafruit Si7021** - Dalším modulem, který je využíván pro činnost naší stanice, je právě tento, který dokáže oproti výše zmiňovanému modulu poskytovat i informace o procentuální relativní vlhkosti v daném prostředí. Odchylka této jednotky může být nanejvýš 3% v rozpětí od 0-80% relativní vlhkosti. Dále dokáže poskytovat rovněž informace o [teplotě v prostředí](#)teplotě v prostředí, spolehlivě ovšem pouze ale v rozsahu od -10°C do +85°C. Stejně jako modul předchozí nemá ani tento žádné vysoké nároky na napětí, postačí mu pouze 3,3 V, které je mu dodáváno přímo z Raspberry Pi, se kterým komunikuje také pomocí speciální I2C sběrnice, kde využívá adresu 0x40. Rozhodli jsme se využít teplotní data z již výše zmiňovaného modulu, jelikož nám tento poskytoval i přes jeho vlastnosti teploty zvláštní a nepřesné. Proto je použit pouze pro získávání procentuální relativní vlhkosti. Cena za jeden kus se pohybuje kolem 7\$.

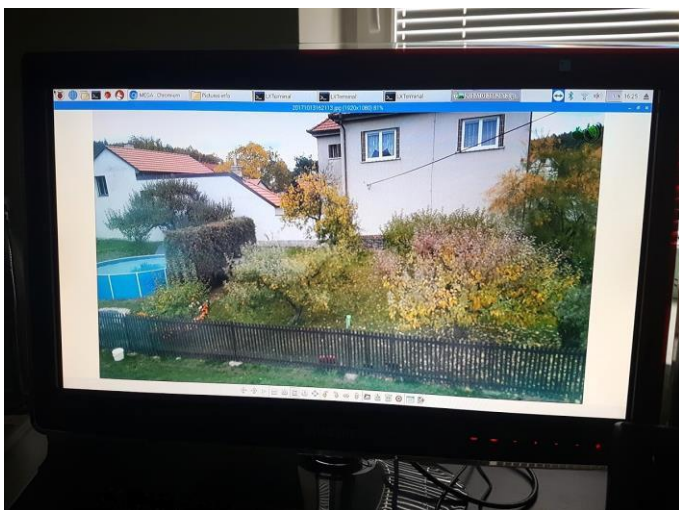


Obrázek 10: Adafruit Si7021, výstřižek z: adafruit.com

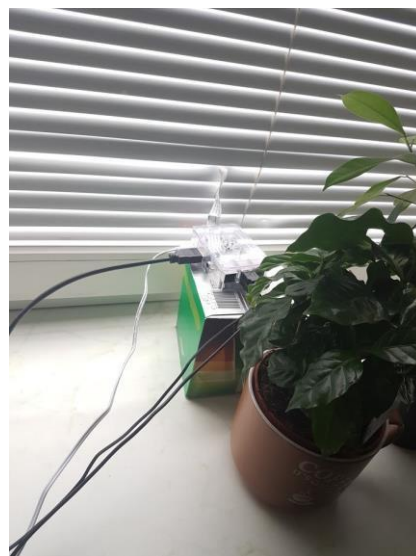
3. **Raspberry Pi Camera V2** - Naše práce je orientována velkou částí orientována přímo na snímání obrazu a poté další práci s poskytnutými obrazovými daty. Proto jsme museli nějakým způsobem zajistit také nějakou možnost, jakým nám bude umožněno snímání obrazových informací. Rozhodli jsme se právě pro kamerový modul, který je navržený speciálně -pro spolupráci s Raspberry Pi.

odložena hlavně z důvodu, že tento modul využívá stejné adresy (0x60) v I2C sběrnici jako modul MPL3115A2, který nám již poskytuje výše zmíněná data. Při integraci obou těchto modulů by vznikaly kolize, které by bránily získu jakékoli informace. Tyto nedostatky se doufejme dořeší do další verze systému.

Poté jsme museli samotné zařízení s moduly propojit a udělat základní nastavení. Zcela prvním praktickým úkolem bylo zprovoznit počítač Raspberry Pi. Po rozbalení se nejprve musí zvolit vhodná koncovka vstupu do elektrické sítě pro napájení, dle země kde se nacházíte. Poté bylo třeba také nainstalovat pasivní chladič na procesor počítače, jelikož operace, které budou prováděny, jsou dosti náročné a může docházet k možnému přehřívání jednotky. Součástí balení byl i speciální case pro ochranu počítače před okolními jevy, tudíž jsme neváhali a využili jsme jej. Poté bylo třeba také vložit SD kartu s obrazem systému Raspbian do již určeného slotu, je totiž potřeba, aby počítač byl schopný operovat na úrovni nějakého operačního systému. Dále jsme připojili kamerový modul do CSI portu na Raspberry Pi. Aby byla započata komunikace kamery a počítače, je třeba nejprve povolit danou funkci v konfiguračním okně, do kterého se dostaneme zadáním příkazu `raspi-config` v terminálu počítače. Po dokončení tohoto procesu vznikla první verze našeho systému, která již mohla obsluhovat kamerový modul. V této zcela první použitelné verzi byl z interiéru snímán pouze výhled z okna, protože zařízení nebylo připraveno na přesun do venkovního prostředí. První a ještě tehdy neúplně animační struktury, byly použity již v této verzi systému. Na fotografiích můžete vidět, jak to v té době vypadalo.



Obrázek 13: Prvotní snímky



Obrázek 14: První umístění

Dále jsme potřebovali zajistit webový hosting. Po vyzkoušení prvního webového hostingu jsme zjistili, že neposkytoval přístup ke konzoli samotného web serveru, který jsme potřebovali k ulehčení práce. Proto jsme daný hosting odhlásili a začali hledat nový, který by nám námi potřebný přístup ke konzoli dal. Také jsme měli další požadavky - chtěli jsme vlastní emailovou adresu, větší kapacitu na serveru, kvůli uloženým snímkům, animacím atd. a také aby server byl stabilní a nepadal (lepší přístupnost). Po chvílích

hledání jsme našli hosting, který nám plně vyhovoval a tak jsme ho zakoupili a začali tvořit samotnou webovou aplikaci. Sem jsme nahráli prvotní verzi stránky - odpočet do spuštění aplikace s alespoň základní funkcí. Tím jsme měli první reprezentaci našeho projektu a zároveň jsme měli dostatek času na to, vytvořit základní verzi.

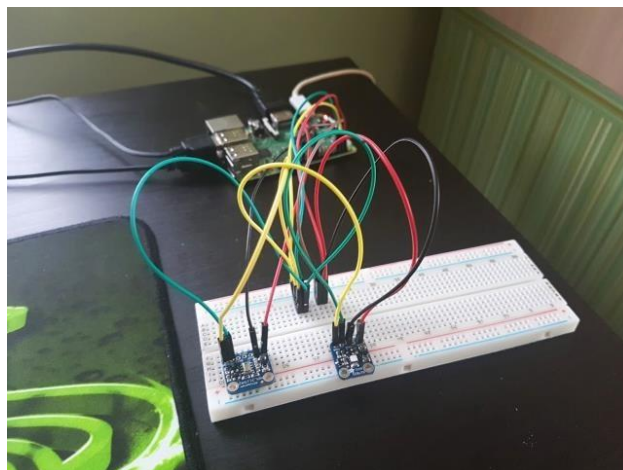
Když jsme měli již celkově dokončené animační struktury a práci se snímky, bylo na čase zprovoznit další fázi našeho systému, která se týkala propojení modulů pro získávání informací o prostředí. Tohoto jsme snadno docílili vytvořením pomocí nepojivého spoje paralelní obvod, který propojoval komunikační GPIO piny na pro I2C sběrnici na Raspberry Pi s příslušnými komunikačními porty na obou zvolených modulech. Napájení modulů bylo provedeno zvlášť, jelikož se na Raspberry Pi nachází dostatek GPIO pinů s výstupním napětím. Vyskytlo se zde nejprve několik problémů, jelikož se obvod zprvu nezdál příliš stabilní. Po krátkém procesu troubleshootingu a napsání programové struktury v jazyce Python sloužící pro obsluhu příslušných modulů, se ale zdálo vše plně funkční. Pro testovací účely zde bylo použito záložní Raspberry Pi. Na fotografiích si můžete prohlédnout progres vůči verzi předchozí.

```

pycompile      pypyclean      python3.5-...
pydoc          pypycompile  python3.5m-...
pydoc2.7      python        python3.5m-...
pydoc3        python2-config  python3.5m-...
pi@raspberrypi:~/Desktop $ python
python        python2-config  python3.5m-...
python2      python3        python3.5m-...
python2.7    python3.5      python3-conf...
python2.7-config  python3.5-config  python3m
pi@raspberrypi:~/Desktop $ python ttest.py
Pressure : 95.98 kPa
Altitude : 454.38 m
Temperature in Celsius : 25.88 C
Temperature in Fahrenheit : 78.58 F
pi@raspberrypi:~/Desktop $

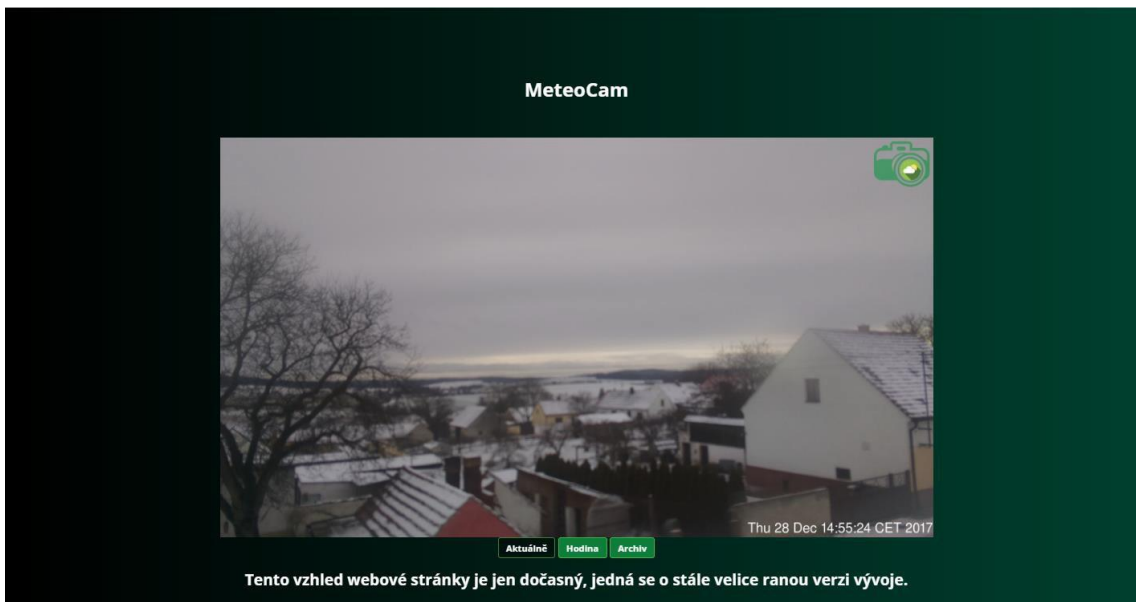
```

Obrázek 16: Úspěšný výpis hodnot



Obrázek 15: Prototypování modulů

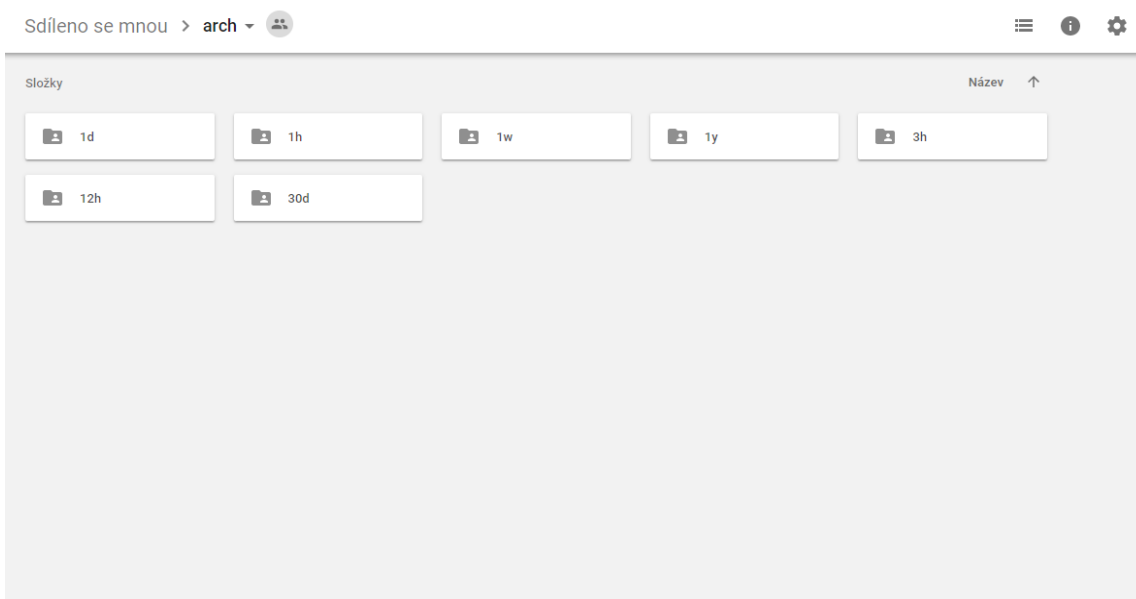
Stránka využívá HTML, CSS, PHP a Javascript. Po prvotních zkouškách základní aplikace, kde byl přístupný nejnovější pořízený snímek a hodinová animace, jsme potřebovali server propojit s Raspberry Pi. Toho jsme docílili poměrně snadno a rychle použitím linuxového balíčku ncfpput, pomocí něhož se v programové struktuře nahrávají nové snímky a animace do cílového adresáře webového serveru přes FTP (21) protokol. Po propojení jsme dosáhli našeho prvního cíle a na stránce jsme mohli zobrazovat nejnovější snímek, hodinovou animaci a také možnost zobrazit si archiv, který jsme si předem vytvořili na Google Disku. Archiv funguje vcelku jednoduše, všechny staré animace a snímky se ukládají a tím archivují do veřejně přístupného Google Disku, kam odkazujeme tlačítkem.



Obrázek 17: Výstup snímku na webu

Zde vidíme, jak vypadala prvotní verze - stránka má pouze základní funkcionalitu a 3 tlačítka

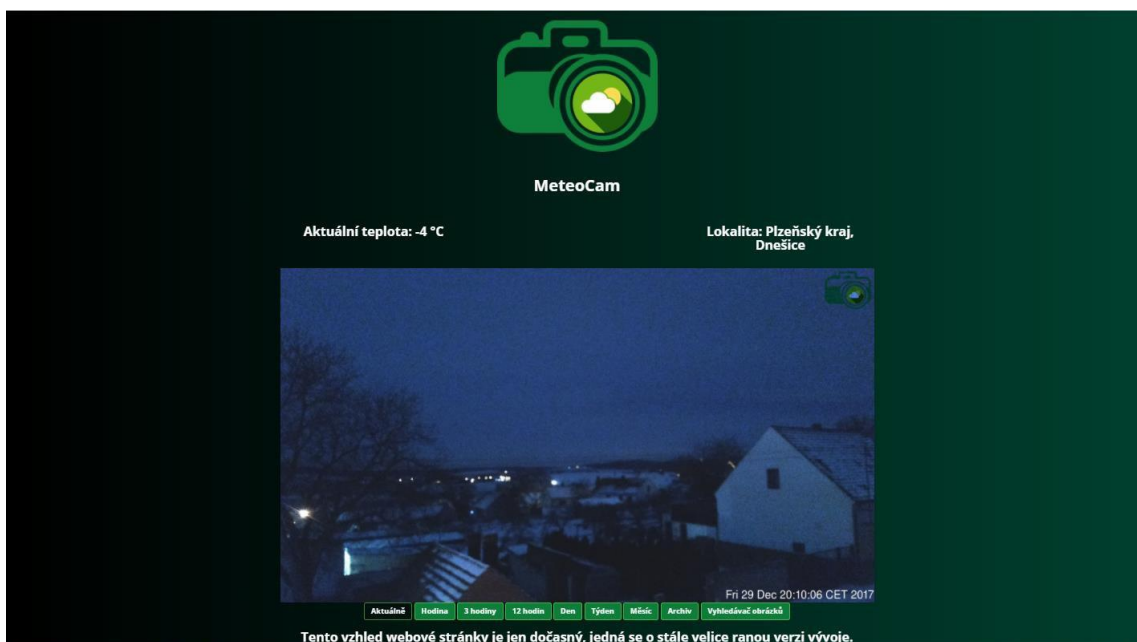
1. Aktuálně - zde byl vidět nejnovější snímek. Nový se pořizuje každou minutu a automaticky se na něj lepí 2 části - aktuální datum vpravo dolů a logo našeho projektu vpravo nahoru. Kameru máme naměřenou na panorama.
2. Hodina - zde se nacházela nejnovější hodinová animace. Stejně jako u aktuálního snímku je zde nalepené datum a čas a logo.
3. Archiv - toto tlačítko otevře uživateli v novém okně náš archiv na Google Disku, viz obrázek. Máme zde archivované animace všech časů - od 1h (jedna hodina) až po 1y (one year - jeden rok). Jako úložiště jsme vybrali Google Disk - především kvůli místu, které nabízí, ale také kvůli stabilitě a propojenosti s celým Googlem (jednoduché sdílení přes Gmail).



Obrázek 18: Archivace na Google disk

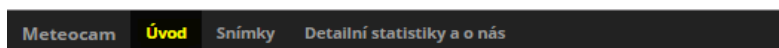
Nahoře na stránce se také ještě nenacházelo logo a byl zde pouze titulek MeteoCam.

Po této prvotní verzi jsme postoupili do fáze druhé - díky novým modulům jsme mohli zobrazovat aktuální teplotu na místě kamery. Ta se měří přes modul ve dvou jednotkách - stupně Celsia a Fahrenheit. Také je zde nově přidáné logo a lokalita dané kamery.



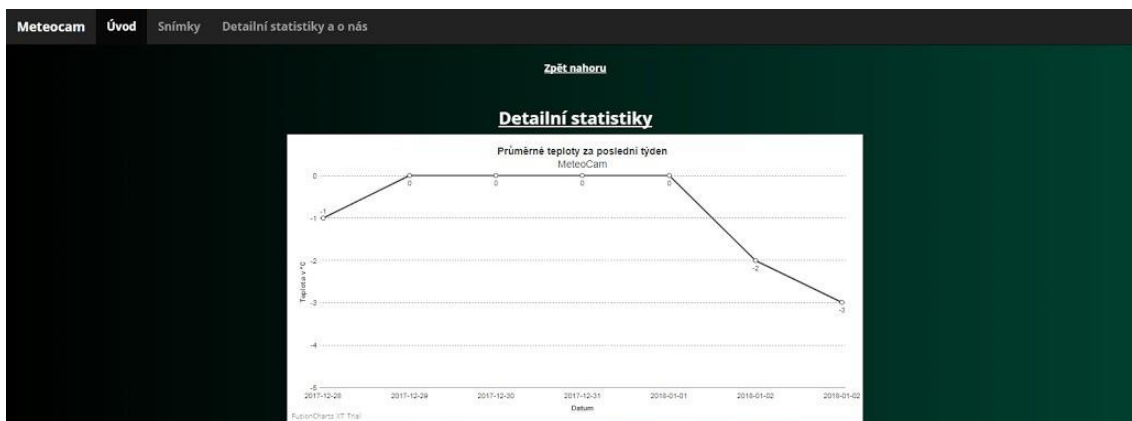
Obrázek 19: Výstup snímků a animací na webu

Poté už web přešel do předposlední fáze - stránka má přidáné detailní statistiky získané z našich modulů a informace o nás (autorech). Také je na stránce přidáný tzv. scrollspy - ten detekuje, kde se uživatel nachází na stránce a podle toho zvýrazňuje příslušnou sekci na vrchu stránky.



Obrázek 20: Banner na webu

Ve finální a poslední verzi stránky jsme přidali funkční graf, který ukazuje průměrné teploty za poslední týden. To se provádí díky FusionCharts, která generuje grafy na základě JSON.



Obrázek 21: FusionCharts graf

V této době jsme již měli vyřešen prvotní design zařízení, který bude plně přizpůsoben možným podmínkám ve venkovním prostředí. Nejprve jsme hledali nějaký ochranný box, do kterého by se dal celý systém ukrýt, ale zároveň musela být zachována úplná funkčnost a nesměly se zde objevovat ani žádné zkreslující vlivy. Na Internetu jsme tedy vyhledali jednu zajímavou možnost, kterou znázorňovala velmi odolná a prostorná elektroinstalační krabice s průhledným víkem, která je např. vhodná i do extrémních mrazivých podmínek.



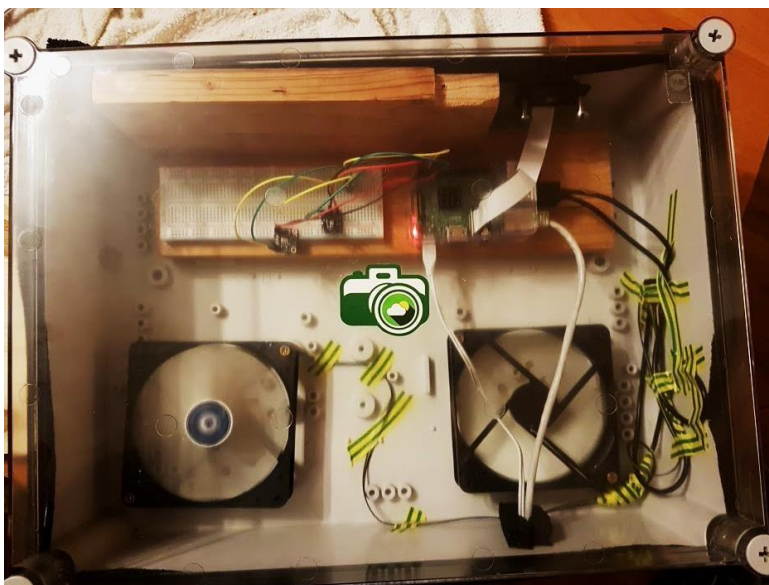
Obrázek 22: Elektroinstalační box, zdroj: www.famatel.cz

Bylo jasné, že ji bude potřeba pro účely projektu mírně poupravit, ale i přes to jsme ji nakonec pořídili. Nejprve jsme začali tím, že jsme vyvrtali otvor v jednom z boků krabice pro zorné pole kamery. Samotný kamerový modul jsme zde poté jednoduše upevnili pomocí dvou velkých šroubů. Dále jsme připravili malou konstrukci ze dvou dřevěných prkének, která bude sloužit jako základ pro upevnění nepájivého pole a počítače Raspberry Pi. Nepájivé pole poté bylo přichyceno pomocí vteřinového lepidla, Raspberry Pi bylo prostřednictvím svého case přivrtáno do konstrukce pomocí dvou malých šroubů. Ve dně boxu se také vyvrtalo několik otvorů pro přístup venkovního vzduchu. Dále se znovu zapojil celý obvod mezi všemi moduly a počítačem. Poté se Raspberry Pi připojilo k Internetu a udělalo se několik zkušebních testů. Když bylo vše vyhodnoceno jako funkční, přemístilo se celé zařízení na domovní terasu v prvním patře rodinného domu, kde se nechal systém několik desítek hodin testovat ve venkovních podmínkách. Zjistilo se, že kamera má nedostatečný výhled, tudíž byl vyroben i malý improvizovaný stožár pro lepší umístění.



Obrázek 23: Výhled pro zařízení

Časem se ale začala jevit větší teplotní odchylka při intenzivním slunečním svitu. V boxu tímto způsobem začal, vznikal skleníkový efekt, tudíž se v něm vzduch zahříval a to vedlo i ke zkreslení výstupní teploty. Tento se nám povedlo vyřešit umožněním lepšího vzduchového toku ve vnitru zařízení pomocí dvou počítačových větráků, které jsme sami uzpůsobily pro napájení z portu USB Raspberry Pi. Jeden z nich vhání venkovní vzduch do prostor zařízení, druhý zase na druhé straně vyhání vzduch z vnitru pryč. Od doby implementace tohoto řešení se nevyskytly žádné výrazné odchylky.



Obrázek 24: Testovací prototyp

4. Hardware

4.1. Programování hardwaru

V této části si ukážeme a popíšeme dvě programové struktury, které slouží pro pořízení snímku z kamery a pro obsluhu dalších modulů. Dále si poté ukážeme, jak funguje generování animací.

4.2. Přídavné moduly

Nejprve si ukážeme skripty pro práci se snímky, které jsme nazvali cam-mark a cam-get.

```
echo "Sleeping for $frequency seconds..."
sleep $frequency
echo "Taking image..."
$currentFolder/cam-get
```

Samotný skript cam-mark slouží jako nadřazený proces skriptu cam-get, to znamená, že si po vypršení určené časové frekvence v průběhu cam-mark volá cam-

get, který mu předá výstup v podobě snímku a s tím on poté provádí další nutné operace. Mezi tyto operace patří např. označení snímku našim logem v horním pravém rohu nebo implementace aktuální časové značky do dolního pravého rohu. Poté je také zodpovědný za nahrání aktuálního pořízeného snímku do adresáře na našem webovém serveru, ale také se stará o spuštění naší zálohovací programové struktury na obrázkovou službu Imgur.

```
# Přihlášení na FTP server a následné nahrání snímku.
ncftpput -u $userName -p $password $domain $destinationFolder "$outputFile.jpg"
echo "$outputFile has been successfully transfered to website hosting!"
echo "Transferring $outputFile to Imgur service..."
imgur "$outputFile.jpg" "$i" # Nahrání snímku na službu Imgur.
echo "$outputFile has been successfully transfered to Imgur!"
```

Obrázek 26: cam-mark skript ukázka

Skript cam-get obsahuje podmínky, které určují, zdali je čas pro denní nebo noční konfiguraci kamery. Dále je zde pomocí raspistill pořizován samotný snímek za užití

```
cd $pathToPictures/Pictures-unstamped
if [ $FLATTIME -gt 700 ] && [ $FLATTIME -lt 1600 ]; then
    raspistill -w 1920 -h 1080 -o $DATE.jpg
    echo "New daylight image has been taken!"
else
    raspistill -w 1920 -h 1080 -ISO 800 -ss 6000000 -br 80 -co 100 -o $DATE.jpg
    echo "New nightlight image has been taken!"
fi
```

Obrázek 27: cam-get skript ukázka

dané konfigurace, který se uloží do výchozího adresáře Pictures-unstamped, kde si jej bere právě výše rozebíraný skript a dále s ním pracuje.

Nyní si ukážeme funkci struktury námi označenou jako meteostats-get, která se stará o obsluhu modulů MPL3115A2 a Si7021, kteří nám poskytují celková data o - teplotě v °C i °F, nadmořské výšce, atmosférickém tlaku a procentuální relativní vlhkosti. Dodatkově byl do struktury implementován také výpočet průměrné teploty a vlhkosti během daného dne pro potřebu tvorby grafů.

```
while True:
    if divisor < 144:
```

Obrázek 28: Průměr

Pokud tato podmínka není splněna, je počet průběhů opět resetován a proces pokračuje dále pro následujících 24 hodin.

```
bus.write_byte_data(0x60, 0x26, 0xB9)
bus.write_byte_data(0x60, 0x13, 0x07)
bus.write_byte_data(0x60, 0x26, 0xB9)
```

Obrázek 29: Definice registrů

Nejprve se testuje podmínka, zdali ještě nebyl naplněn počet průběhů počet průběhů procesu za posledních 24 hodin, nový proces se spouští vždy po uplynutí časového intervalu 10 minut, tudíž je zde nastaven celkový počet 144 denních průběhů.

Ve struktuře kódu se nejprve definují potřebné registry, ze kterých budou výstupní data čtena a volána. —Když —jsou —informace —přečteny,

přepočítávají se dále do potřebného chtěného formátu. Nachází se zde také již výše zmiňovaný výpočet průměrné teploty a vlhkosti, jehož výstupy se poté předávají skriptu avg-temp, který se dále rozhoduje na základě podmínky, zdali daný vstup nahraje do databáze na webovém serveru, jelikož může data nahrát až ve předem vymezeném časovém úseku.

```
tHeight = ((data[1] * 65536) + (data[2] * 256) + (data[3] & 0xF0)) / 16
temp = ((data[4] * 256) + (data[5] & 0xF0)) / 16
altitude = tHeight / 16.0
cTemp = temp / 16.0
print cTemp

if cTemp > 128:
    cTemp = 256 - cTemp
    cTemp = -cTemp

print cTemp
fTemp = cTemp * 1.8 + 32
```

Obrázek 30: Ukázka výpočtu teploty

```
SET @avgTemp = (SELECT AVG(temperature_C) AS avgTemp FROM (SELECT temperature_C FROM stats ORDER BY timestamp DESC LIMIT 144) a);
SET @avgHumid = (SELECT AVG(humidity) AS avgHumid FROM (SELECT humidity FROM stats ORDER BY timestamp DESC LIMIT 144) a);
SET @oldID = (SELECT MAX(id) FROM avg_temp ORDER BY timestamp);
SET @newID = @oldID + 1;
INSERT INTO avg_temp (id, avg_temp, avg_humid) values(@newID, @avgTemp, @avgHumid);
```

Obrázek 31: Výpočet průměru

Každá nově pořízená hodnota se předává prostřednictvím argumentů skriptu uploader, který již netestuje, zdali data předat databázi na webovém serveru pomocí žádných

```
id=$1          # První argument = id v databázi
pressure=$2    # Druhý argument = atmosférický tlak
altitude=$3    # Třetí argument = nadmořská výška
cTemp=$4      # Čtvrtý argument = teplota v °C
fTemp=$5      # Pátý argument = teplota v °F
humidity=$6    # Šestý argument = vlhkost v ovzduší
DT=$(date '+%Y%m%d%H%M%S') # Tato proměnná vyjadřuje nynější datum a čas.
echo "Uploading time is: $DT "
```

Obrázek 32: uploader skript ukázka

podmínek, hodnoty jednoduše nahraje. Tento proces propojení se s webovým serverem je zprostředkován pomocí protokolu vzdálené správy SSH, kde se po úspěšném připojení vyvolá příkaz pro zapsání hodnot do databáze.

```
# Tato podmínka testuje, zdali je čas vhodný pro nahrání dat.
if [ $FLATTIME -ge 2349 ] && [ $FLATTIME -le 2359 ]; then
    sshpass -p          ssh -T $login -p $protocol <<cat
    mysql --host=localhost --user=          --password
    curl
```

Obrázek 33: Podmínka průměru

4.3.1.1. Generování animací

Nyní se podíváme na samotné struktury generování animací, které využíváme v našem projektu pro zpětné časosběrné zprostředkování obrazových informací. Nejprve bylo třeba definovat si časové délky, které by měli být těmito strukturami zachyceny. Zvolili jsme následující periody – 1 hodina, 3 hodiny, 12 hodin, 1 den, 1 týden, 1 měsíc a 1 rok.

Zde si popíšeme princip funkčnosti např. hodinové animace, která je v celkové kaskádě zcela první a poté předává vybrané snímky dalším strukturám v chronologickém pořadí.

Nejprve musí úspěšně proběhnout skript cam mark, který nám připravuje vždy ten nejaktuálnější pořízený snímek do adresáře Pictures-hist1. V dalším kroku si totiž skript pro tvorbu hodinové animace vždy po dané časové periodě 60 sekund vyžádá aktuální snímek, převede jej ve vhodném rozlišení pro generování animace do svého adresáře. Zde skript také ověřuje, zda je vůbec nějaký materiál pro přesun, proto se také nejdříve ptá na počet souborů ve zdrojovém adresáři. Před tím, než tak učiní, si ale vždy pomocí příslušné podmínky v cyklu zkontroluje počet uskutečněných průchodů cyklu (přesunů), aby určil, zda je již splněn počet snímků pro tvorbu animace.

```
while [ $takes -lt 51 ];
do
  sleep $frequency
  if [ $(ls -A "$directory" | wc -l) -gt 0 ]; then
    cd $directory
    convert $(ls -lt | head -1) -resize 640x480! $pathToPictures/Pictures-anim/Pictures-animlh/$takes'.jpg'
    echo "New image has been added to the animation!"
    mv $pathToPictures/Pictures-hist1/*.jpg' $pathToPictures/Pictures-hist2/
    let takes=takes+1
  else
    echo "There are no files!"
  fi
```

Obrázek 34: cam-anim-1hour ukázka

```
# Po dosažení dostatečného počtu snímků v adresáři pro animaci se pokračuje v samotné tvorbě animace.
DT=$(date '+%Y%m%d%H%M%S') # Tato proměnná vyjadřuje nynější datum a čas.
convert -morph 1 -delay 8 $pathToPictures/Pictures-anim/Pictures-animlh/{0..50}.jpg $pathToPictures/$DT.gif
echo "New animation: $DT.gif is ready!"
cd $pathToPictures/Pictures-anim/Pictures-anim-final/Pictures-animlh
$currentFolder/cam-anim-up-1hour "$DT" $pathToPictures $targetDirectory $userName $password $domain $destinationFolder &
```

Obrázek 35: Generování animace

Když se naplní adresář dostatečným počtem snímků, zahajuje se proces generování pomocí příkazu convert obsažený v balíku ImageMagick. Po vytvoření je animace předána jako argument dalšímu skriptu cam-anim-up-1hour, který ji poté pouze nahraje do cíleného adresáře na našem webovém serveru a také do Google drive archivu pomocí struktury gdrive-linux-rpi, kde je dále animace stále dostupná i v případě jejího nahrazení aktuálnější animací na našich webových stránkách.

```

DT="$1"
pathToPictures=$2
targetDirectory=$3
# FTP login attributes
userName=$4
password=$5
domain=$6
destinationFolder=$7
cd $pathToPictures/Pictures-anim/Pictures-anim-final/Pictures-animlh
ncftpput -u $userName -p $password $domain $destinationFolder "$DT.gif"
echo "File $DT.gif has been successfully transferred to website hosting!"
gdrive-linux-rpi upload --parent $targetDirectory $pathToPictures/Pictures-anim/
echo "File $DT.gif has been successfully transferred to Google drive!"

```

Obrázek 36: Upload animace

~~V posledním kroku se už pouze předávají určité vybrané snímky z hodinového adresáře do další struktury – 3 hodiny. Tento proces předávání probíhá jednotlivě ve třech vlnách. Je jasné, že aby mohla být vytvořena tříhodinová animace, je nejprve nutné vytvořit tři hodinové animace. Proto se vždy po dokončení hodinové animace předává 1/2 (resp. každý druhý) snímek v ní obsažených do tříhodinového adresáře, což k dosažení nutného počtu snímek pro tvorbu animace zobrazující tři hodiny přidává pouze 1/2. Ostatní nepřesunutá~~

```

while [ $copy -lt 51 ];
do
  if [ -e $pathToPictures/Pictures-anim/Pictures-anim3h/"$copy.jpg" ]; then
    let firstDuplicate=$(( $copy + $firstAddValue ))
    if [ -e $pathToPictures/Pictures-anim/Pictures-anim3h/"$firstDuplicate.jpg" ]; then
      let secondDuplicate=$(( $copy + $secondAddValue ))
      mv $pathToPictures/Pictures-anim/Pictures-animlh/"$copy.jpg" $pathToPictures/Pictures-
      echo "New picture: $copy.jpg has been moved to the next level animation directory as:
    fi
    mv $pathToPictures/Pictures-anim/Pictures-animlh/"$copy.jpg" $pathToPictures/Pictures-
    echo "New picture: $copy.jpg has been moved to the next level animation directory as:
  else
    mv $pathToPictures/Pictures-anim/Pictures-animlh/"$copy.jpg" $pathToPictures/Pictures-
    echo "New picture: $copy.jpg has been moved to the next level animation directory as:
  fi
  let copy=copy+2
done
rm -r $pathToPictures/Pictures-anim/Pictures-animlh/*.'jpg'
echo "Unnecessary pictures has been removed!"

```

Obrázek 37: Přesun snímeků do další struktury

~~snímky jsou odstraněny, protože už nejsou potřeba. Na podobném principu fungují i tvorby v dalších kaskádách až do animace jednoho týdne, měsíční a roční animaci vždy poskytuje denní kaskáda 1-2 snímky za daný den.~~

4.3. Vývoj designu a vzhledu zařízení

Již od začátku práce jsme se zajímali o design našeho cíleného zařízení. Jak design, tak i výsledný vzhled musí být pro zákazníka přívětivý. V první fázi bylo pouze samotné Raspberry Pi s připojenou kamerou, které snímalo pouze výhled z okna. Systém v této době nedisponoval žádnými ochrannými prvky vůči venkovnímu prostředí, bylo



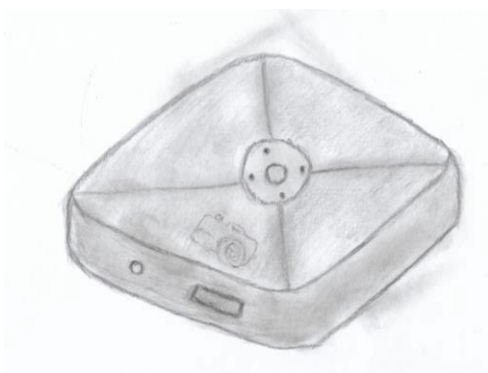
zde zcela vyloučené nějaké přesunutí
projektu do exteriéru.



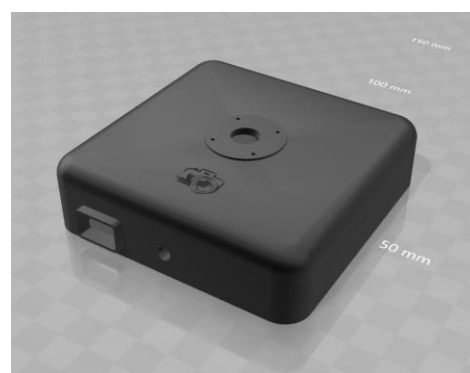
Obrázek 35: Zařízení v exteriéru

V další verzi jsme již řešili přesun systému kompletně do venkovního prostředí, jelikož zde byly zakomponovány i další přídavné moduly. Dříve jsme už v procesu vývoje rozebírali pořízenou elektroinstalační krabici, do které jsme nakonec celý systém úspěšně přesunuli a vystavili vlivům exteriéru.

Jelikož nám přišel tento design vcelku neohrabaný a velice nepraktický (i v případě plně očekávané funkčnosti zařízení), tak jsme se rozhodli, navrhnout si case na míru, který bude přímo navržen pro naše komponenty a potřeby. Zde si můžete prohlédnout prvotní návrhy.

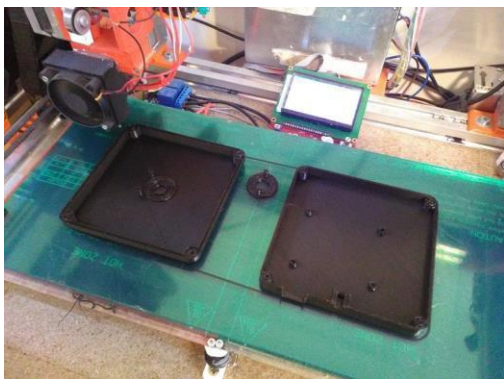


Obrázek 37: Náskres case

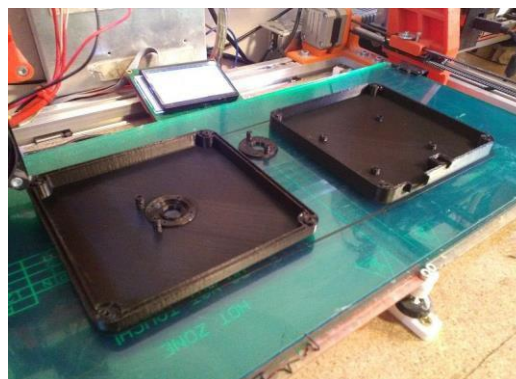


Obrázek 36: 3D design case

Po kompletním rozvržení a přeměření modelu jej bylo nutné vytisknout na 3D tiskárně. Pro tisk bylo spotřebováno 38,65 metrů tisknoucího materiálu ABS. Celkové náklady se rovnaly 93 Kč.



Obrázek 38: Tisk case



Zde se již nachází snímek výsledného produktu, do kterého budou příští systémy vkládány.



Obrázek 39: Výsledný produkt tisku

5. Generování animací

Nyní se podíváme na samotné struktury generování animací, které využíváme v našem projektu pro zpětné časosběrné zprostředkování obrazových informací. Nejprve bylo třeba definovat si časové délky, které by měli být těmito strukturami zachyceny. Zvolili jsme následující periody - 1 hodina, 3 hodiny, 12 hodin, 1 den, 1 týden, 1 měsíc a 1 rok.

Zde si popíšeme princip funkčnosti např. hodinové animace, která je v celkové kaskádě zcela první a poté předává vybrané snímky dalším strukturám v chronologickém pořadí.

Nejprve musí úspěšně proběhnout skript cam-mark, který nám připravuje vždy ten nejaktuálnější pořízený snímek do adresáře Pictures-hist1. V dalším kroku si totiž skript pro tvorbu hodinové animace vždy po dané časové periodě 60 sekund vyžádá aktuální snímek, převede jej ve vhodném rozlišení pro generování animace do svého adresáře. Zde skript také ověřuje, zdali je vůbec nějaký materiál pro přesun, proto se také nejdříve ptá na počet souborů ve zdrojovém adresáři. Před tím, než tak učiní, si ale vždy pomocí příslušné podmínky v cyklu zkontroluje počet uskutečněných průchodů cyklu (přesunů), aby určil, zda je již splněn počet snímků pro tvorbu animace.

```
while [ $takes -lt 51 ];
do
  sleep $frequency
  if [ $(ls -A "$directory" | wc -l) -gt 0 ]; then
    cd $directory
    convert $(ls -lt | head -1) -resize 640x480! $pathToPictures/Pictures-anim/Pictures-animlh/$takes'.jpg'
    echo "New image has been added to the animation!"
    mv $pathToPictures/Pictures-hist1/*'.jpg' $pathToPictures/Pictures-hist2/
    let takes=takes+1
  else
    echo "There are no files!"
  fi
```

Obrázek 40: cam-anim-1hour ukázka

```
# Po dosažení dostatečného počtu snímků v adresáři pro animaci se pokračuje v samotné tvorbě animace.
DT=$(date '+%Y%m%d%H%M%S') # Tato proměnná vyjadřuje nynější datum a čas.
convert -morph 1 -delay 8 $pathToPictures/Pictures-anim/Pictures-animlh/{10..50}.jpg $pathToPictures/$DT.gif
echo "New animation: $DT.gif is ready!"
cd $pathToPictures/Pictures-anim/Pictures-anim-final/Pictures-animlh
$currentFolder/cam-anim-up-1hour "$DT" $pathToPictures $targetDirectory $userName $password $domain $destinationFolder &
```

Obrázek 41: Generování animace

Když se naplní adresář dostatečným počtem snímků, zahajuje se proces generování pomocí příkazu convert obsažený v balíku Imagemagick. Po vytvoření je animace předána jako argument dalšímu skriptu cam-anim-up-1hour, který ji poté pouze nahraje do cíleného adresáře na našem webovém serveru a také do Google drive archivu pomocí struktury gdrive-linux-rpi, kde je dále animace stále dostupná i v případě jejího nahrazení aktuálnější animací na našich webových stránkách.

```

DT="$1"
pathToPictures=$2
targetDirectory=$3
# FTP login attributes
userName=$4
password=$5
domain=$6
destinationFolder=$7
cd $pathToPictures/Pictures-anim/Pictures-anim-final/Pictures-animlh
ncftpput -u $userName -p $password $domain $destinationFolder "$DT.gif"
echo "File $DT.gif has been successfully transferred to website hosting!"
gdrive-linux-rpi upload --parent $targetDirectory $pathToPictures/Pictures-anim/
echo "File $DT.gif has been successfully transferred to Google drive!"

```

Obrázek 42: Upload animace

V posledním kroku se už pouze předávají určité vybrané snímky z hodinového adresáře do další struktury - 3 hodiny. Tento proces předávání probíhá jednotlivě ve třech vlnách. Je jasné, že aby mohla být vytvořena tříhodinová animace, je nejprve nutné vytvořit tři hodinové animace. Proto se vždy po dokončení hodinové animace předává 1/2 (resp. každý druhý) snímků v ní obsažených do tříhodinového adresáře, což k dosažení nutného počtu snímků pro tvorbu animace zobrazující tři hodiny přidává pouze 1/3. Ostatní nepřesunuté

```

while [ $copy -lt 51 ];
do
  if [ -e $pathToPictures/Pictures-anim/Pictures-anim3h/"$copy.jpg" ]; then
    let firstDuplicate=$((copy + $firstAddValue))
    if [ -e $pathToPictures/Pictures-anim/Pictures-anim3h/"$firstDuplicate.jpg" ]; then
      let secondDuplicate=$((copy + $secondAddValue))
      mv $pathToPictures/Pictures-anim/Pictures-animlh/"$copy.jpg" $pathToPictures/Pictures-
      echo "New picture: $copy.jpg has been moved to the next level animation directory as:
    fi
    mv $pathToPictures/Pictures-anim/Pictures-animlh/"$copy.jpg" $pathToPictures/Pictures-
    echo "New picture: $copy.jpg has been moved to the next level animation directory as:
  else
    mv $pathToPictures/Pictures-anim/Pictures-animlh/"$copy.jpg" $pathToPictures/Pictures-
    echo "New picture: $copy.jpg has been moved to the next level animation directory as:
  fi
  let copy=copy+2
done
rm -r $pathToPictures/Pictures-anim/Pictures-animlh/*.'jpg'
echo "Unnecessary pictures has been removed!"

```

Obrázek 43: Přesun snímků do další struktury

snímky jsou odstraněny, protože už nejsou potřeba. Na podobném principu fungují i tvorby v dalších kaskádách až do animace jednoho týdne, měsíční a roční animaci vždy poskytuje denní kaskáda 1-2 snímky za daný den.

5.6. Návrh a provedení webové aplikace

Při návrhu webové aplikace jsme nejdříve přemýšleli, jaké využít nástroje a jazyky, popř. zdali bychom využili nějaký CMS systém. Po rozmýšlení jsme se rozhodli postavit stránku od základů novou a využít jako hlavní jazyk PHP - především kvůli tomu, že už jsme s ním měli předešlé zkušenosti. Dále samozřejmě HTML a CSS. V HTML je základní struktura stránky a CSS zas zajišťuje hezčí vizuální stránku. Nakonec také JavaScript, ve kterém jsou napsané skripty, např. na měnění snímků při zmáčknutí některého z tlačítek.

5.1.6.1. Struktura

Stránka se skládá z klasické HTML struktury - <head> a <body>.

```

▼ <head>
  <meta charset="UTF-8">
  <link rel="stylesheet" href="css/bootstrap.min.css">
  <link rel="stylesheet" href="css/style.css">
  <link rel="stylesheet" type="text/css" href="css/font-awesome.min.css">
  <link rel="stylesheet" type="text/css" href="css/animate.min.css">
  <link href="https://fonts.googleapis.com/css?family=Open+Sans:300i,700,700i"
  rel="stylesheet" type="text/css"> == $0

```

Obrázek 44: HTML struktura

Jak vidíme, v hlavičce se nejdříve určuje kódování, jaké se bude na stránce využívat. Zde máme nastaveno UTF-8. Dále zde odkazujeme na několik CSS souborů, které využíváme. Můžeme odkazovat i na soubory lokálně uložené na serveru, či uložené někde na přístupném internetovém úložišti. To můžeme vidět např. u posledního odkazu na font uložený na serverech Google.

```

▼ <body class="background" data-spy="scroll" data-target=".navbar" data-offset="50" cz-shortcut-listen="true" style>
  <!-- Preloader -->
  <div id="preloader" style="display: none;">...</div> == $0
  <!-- /Preloader -->
  <!-- Overlay -->
  <div id="section-home" class="home-section-wrap center">
    <div class="section-overlay"></div>
  </div>
  <!-- /Overlay -->
  <div class="containers">...</div>
  <!-- JS -->
  <script type="text/javascript" src="js/jquery-1.11.3.min.js"></script>
  <script type="text/javascript" src="js/bootstrap.min.js"></script>
  <script type="text/javascript" src="js/jquery.countdown.min.js"></script>
  <script type="text/javascript" src="js/main.js"></script>
  <script>...</script>
  <div class="celery celery-test">...</div>
</body>

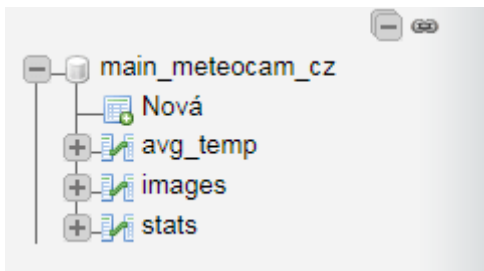
```

Obrázek 45: Ukázka zdrojového kódu webu

V těle se potom nachází to, co vidíme na stránce. Pod body se nachází několik menších sekcí <div>. Ty využíváme pro lepší rozdělení stránky a pro lepší manipulaci s jednotlivými částmi. U konce vidíme část, ve které odkazujeme na využití skripty.

5.2.6.2. Databáze a dohledatelnost

Pro uložení dat, která využíváme na stránce, používáme databázi MySQL s phpMyAdmin. Toto je uloženo na stejném serveru, kde nám běží samotná webová stránka. V databázi máme několik tabulek.



Obrázek 46: Databáze a tabulky

V tabulce `avg_temp` ukládáme průměrné statistiky za daný den - vlhkost a teploty. Tyto data využíváme dále při tvorbě grafů. Dále je zde tabulka `images`, ve které se ukládají informace o jednotlivých snímcích (viz 4.3.3. Archivace). Jako poslední je tabulka nejdůležitější - samotné statistiky, nasbírané z modulů.

#	Název	Typ	Porovnávání	Vlastnosti	Nulový	Výchozí	Komentáře	Další	Operace
1	<code>id</code>	<code>int(11)</code>			Ne	Žádná		Změnit	Odstranit
2	<code>humidity</code>	<code>decimal(10,0)</code>			Ne	Žádná		Změnit	Odstranit
3	<code>height</code>	<code>smallint(6)</code>			Ne	Žádná		Změnit	Odstranit
4	<code>air_pressure</code>	<code>decimal(10,0)</code>			Ne	Žádná		Změnit	Odstranit
5	<code>temperature_C</code>	<code>decimal(10,0)</code>			Ne	Žádná		Změnit	Odstranit
6	<code>temperature_F</code>	<code>decimal(10,0)</code>			Ne	Žádná		Změnit	Odstranit
7	<code>timestamp</code>	<code>timestamp</code>			Ne	CURRENT_TIMESTAMP		Změnit	Odstranit

Obrázek 47: Pole v tabulce

V tabulce se nachází 6 řádků.

- 1) `id` - toto je primární klíč, což je unikátní číslo, podle kterého se identifikuje každý nový prvek v tabulce
- 2) `humidity` - zde se ukládá v decimálním typu vlhkost vzduchu
- 3) `height` - zaznamenaná nadmořská výška
- 4) `air_pressure` - atmosférický tlak naměřený modulem
- 5) `temperature_C` / `temperature_F` - naměřené teploty ve dvou jednotkách Celsius a Fahrenheit
- 6) `timestamp` - zde se ukládá čas ve chvíli, kdy se přidá nový prvek do tabulky.

+ Nastavení									
		<code>id</code>	<code>humidity</code>	<code>height</code>	<code>air_pressure</code>	<code>temperature_C</code>	<code>temperature_F</code>	<code>timestamp</code>	
<input type="checkbox"/>		0	0	0	0	0	0	2017-12-27 15:34:43	
<input type="checkbox"/>		1	52	320	98	-2	29	2017-12-27 15:34:43	
<input type="checkbox"/>		2	52	321	98	-2	29	2017-12-27 15:34:43	
<input type="checkbox"/>		3	52	321	98	-2	29	2017-12-27 15:34:43	

Obrázek 48: Struktura tabulky

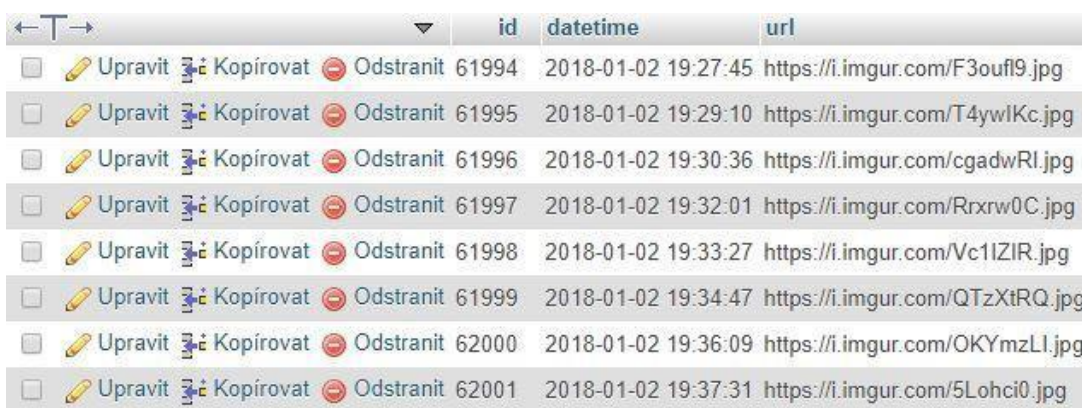
Zde vidíme příklad některých prvků, např. prvek s `id` 3 zaznamenal v 15:34:43 27. 12. 2017 teplotu -2 °C, čili 29 °F, tlak 98 hPa, vlhkost 52% a nakonec nadmořskou výšku 321 m.

Díky způsobu ukládání do databáze můžeme jednoduše získat informace i o starších statistikách, což se dá využít např. při tvorbě grafů.

5.3.6.3. Archivace

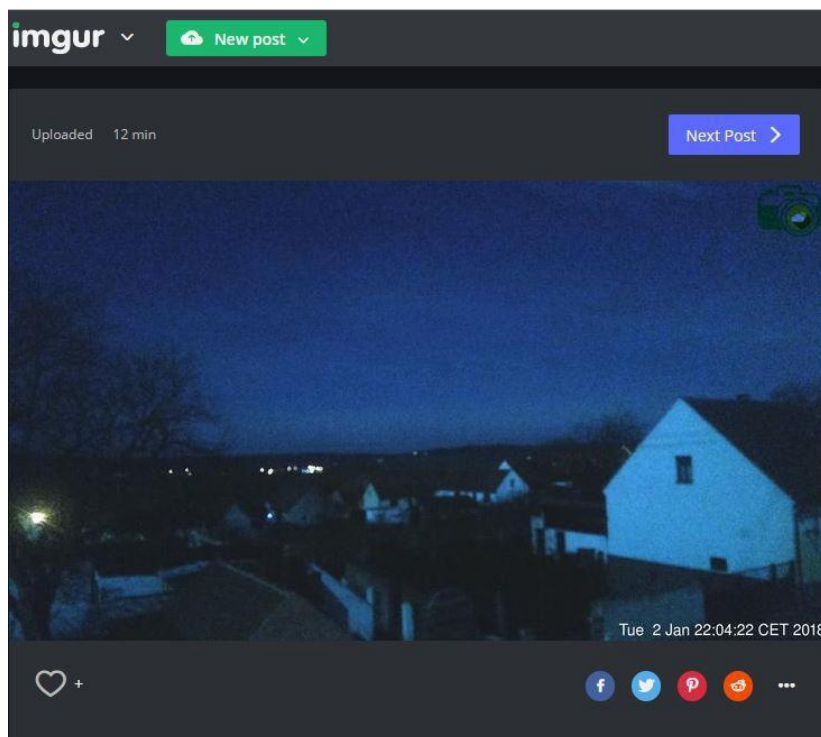
Byla by jistě velká škoda přicházet o jakákoliv data, která nám náš systém poskytuje. Zároveň se ale jedná o skutečně velký objem dat, který má vysoké nároky na naši možnou úložnou kapacitu. Proto jsme již od raného vývoje hledali řešení, jakou strukturou lze zde snadno a rychle data zálohovat a archivovat. Pracovali jsme nejprve s myšlenkou nějakého cloudového úložiště s neomezenou kapacitou, problém bylo, že přímo taková možnost není dostupná anebo pouze za splnění nějakých pro nás nepříznivých podmínek, jako např. omezená velikost nahrávaného souboru nebo možnost uploadu jen několikrát denně. I tak se nám ale podařilo přijít na vsutku zajímavá a efektivní řešení.

Pro neomezenou archivaci všech pořízených snímků jsme využili kombinaci Imgur API a naší MySQL databáze. Podařilo se nám sestavit funkční skript, který nahraje nově pořízený snímek na obrázkovou službu Imgur a poté výstupní URL odkaz zaeviduje do příslušné databáze. Imgur neposkytuje žádné struktury, jak bychom mohli uspořádat všechny námi nahrané snímky někde v prostředí Imgur webového rozhraní, což znamená, že v moment ztracení URL odkazu by již neexistoval žádný způsob, jak daný obrázek zpětně vyhledat. Proto máme dostupnou databázi s daty a odkazy na snímky.



			id	datetime	url	
<input type="checkbox"/>	Upravit	Kopírovat	Odstranit	61994	2018-01-02 19:27:45	https://i.imgur.com/F3ouf19.jpg
<input type="checkbox"/>	Upravit	Kopírovat	Odstranit	61995	2018-01-02 19:29:10	https://i.imgur.com/T4ywIKc.jpg
<input type="checkbox"/>	Upravit	Kopírovat	Odstranit	61996	2018-01-02 19:30:36	https://i.imgur.com/cgadwRI.jpg
<input type="checkbox"/>	Upravit	Kopírovat	Odstranit	61997	2018-01-02 19:32:01	https://i.imgur.com/Rrxrw0C.jpg
<input type="checkbox"/>	Upravit	Kopírovat	Odstranit	61998	2018-01-02 19:33:27	https://i.imgur.com/Vc1IZIR.jpg
<input type="checkbox"/>	Upravit	Kopírovat	Odstranit	61999	2018-01-02 19:34:47	https://i.imgur.com/QTzXtRQ.jpg
<input type="checkbox"/>	Upravit	Kopírovat	Odstranit	62000	2018-01-02 19:36:09	https://i.imgur.com/OKYmzLI.jpg
<input type="checkbox"/>	Upravit	Kopírovat	Odstranit	62001	2018-01-02 19:37:31	https://i.imgur.com/5Lohci0.jpg

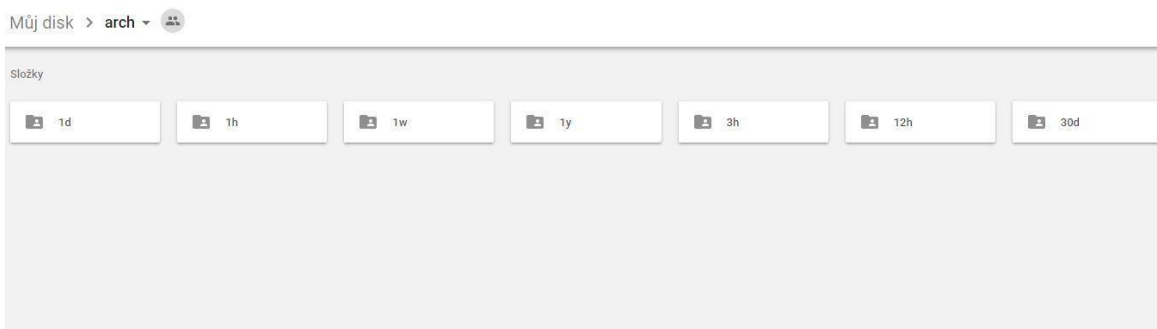
Obrázek 49: Archivační tabulka pro pořízené snímky



Obrázek 50: Pohled na snímek z webu Imgur

Jednou za čas se stane, že je nám nahrávání na Imgur znepřístupněno kvůli tomu, že nahráváme nový obrázek každou 1 minutu, což někdy cílová služba hodnotí jako příliš častý upload. Tento zákaz platí ale pouze po dobu pěti minut, poté nahrávání opět pokračuje. Tento případ dočasného zákazu není příliš častý, většinou ho proděláme maximálně jedinkrát za den, což požadujeme za dosti nepatrnou ztrátu.

Po dokončení otázky, jakým způsobem zálohovat a archivovat pořízené snímky, se rovněž dostalo i na možnosti archivace samotných výstupních animací. Ty pro nás představovaly již větší výzvu, protože mají mnohem větší datový objem (většinou až 20 MB a více) než snímky samotné (1-2 MB). To znamená, že na Imgur by nahrávání našich animací nebylo možné, protože přes API lze nahrávat soubory menší než 10 MB. Snažili jsme se využít spíše nějakého klasického cloud úložiště, a tak jsme prošli hned několik nepříliš příznivých možností a skončili u Google drive, na kterém máme k dispozici neomezenou úložnou kapacitu díky službě G suite, kde je registrována i naše škola. Nechali jsme si vygenerovat API přístupní klíč a vyhledali jsme jednu zajímavou strukturu, která nám dovoľovala nahrávat soubory na Google drive přímo ze skriptu.



Obrázek 51: Struktura archivace animací

Název tohoto procesu je gdrive-linux-rpi, který rovněž dokáže zpracovat několik parametrů příkazu a lze tak definovat např. ID cílové složky na cílovém úložišti. Jako zajímavost bychom ještě mohli uvést, že k datu 1. 1. 2018 se nám podařilo nasbírat 15 GB dat v oblasti animací a zhruba 60 GB dat v oblasti samostatných snímků. Tyto počty přesněji vyjadřují 2 měsíce aktivní činnosti našich struktur.

5.4.1.1. Vývoj designu a vzhledu zařízení

Již od začátku práce jsme se zajímali o design našeho cíleného zařízení. Jak design, tak i výsledný vzhled musí být pro zákazníka přívětivý. V první fázi bylo pouze samotné Raspberry Pi s připojenou kamerou, které snímalo pouze výhled z okna. Systém v této době nedisponoval žádnými ochrannými prvky vůči venkovnímu prostředí, bylo zde zcela vyloučené nějaké přesunutí projektu do exteriéru.



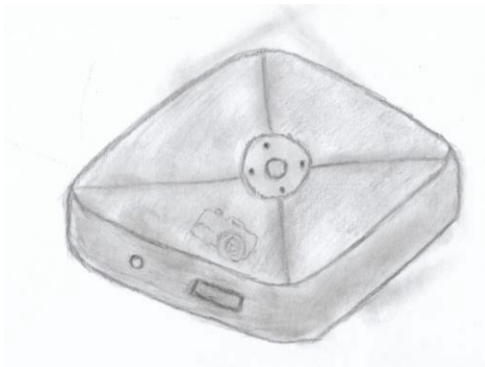
Obrázek 52: Zařízení v interiéru



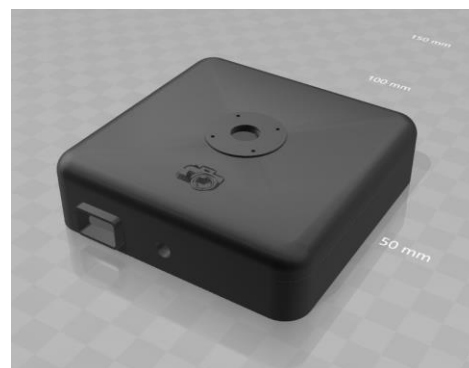
Obrázek 47: Zařízení v exteriéru

~~V další verzi jsme již řešili přesun systému kompletně do venkovního prostředí, jelikož zde byly zakomponovány i další přídavné moduly. Dříve jsme už v procesu vývoje rozebírali pořízenou elektroinstalační krabici, do které jsme nakonec celý systém úspěšně přesunuli a vystavili vlivům exteriéru.~~

~~Jelikož nám přišel tento design veelku neohrabaný a velice nepraktický (i v případě plně očekávané funkčnosti zařízení), tak jsme se rozhodli, navrhnout si case na míru, který bude přímo navržen pro naše komponenty a potřeby. Zde si můžete prohlédnout prvotní návrhy.~~

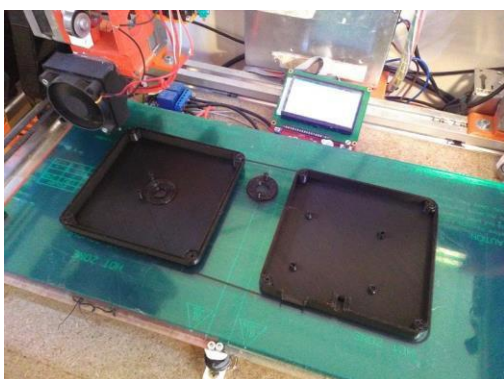


Obrázek 49: Náskres case

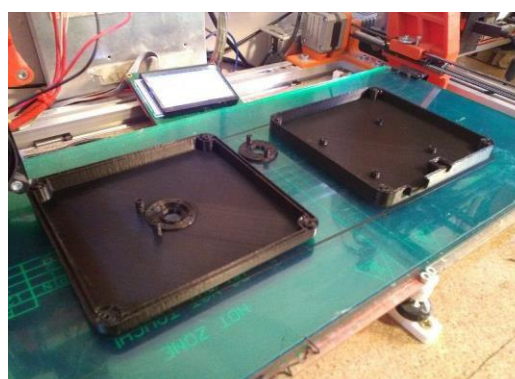


Obrázek 53: 3D design case

~~Po kompletním rozvržení a přeměření modelu jej bylo nutné vytisknout na 3D tiskárně. Pro tisk bylo spotřebováno 38,65 metrů tisknouceho materiálu ABS. Celkové náklady se rovnaly 93 Kč.~~



Obrázek 50: Tisk case



~~Zde se již nachází snímek výsledného produktu, do kterého budou příští systémy vkládány.~~



~~Obrázek 51: Výsledný produkt tisku~~

Závěr

Nyní si zhodnotíme celý průběh naší dlouhodobé práce, kterou hodnotíme velmi pozitivně vůči rozvoji našich dovedností a všeobecných schopností. Již na počátku projektu jsme si vztyčili několik cílů, které jsme poté logicky chtěli během tvorby zrealizovat. Mezi ně patřila zejména naše myšlenka udělat systém co nejvíce otevřený a přizpůsobivý, ať už jakýmkoli potřebám potencionálního spotřebitele, který by měl dostatečný zájem a vůli takového zařízení provozovat. Myslím, že se nám právě tento nejhlavnější úmysl povedl skvěle vypracovat až do finální podoby, uživatel by neměl shledat při konfiguraci a přizpůsobení vlastního zařízení žádný problém. Co se ostatních vytčených bodů a úkolů týče, tak se nám některé povedly lépe, než jsme měli původně vůbec v záměru. Největší radost máme ze stabilních programových struktur, dále z přehlednosti a vzhledu webové aplikace a také ze spolehlivosti našich hardwarových prvků. Také nám přijde velice impresivní náš archivovací systém, pomocí něhož se lze snadno orientovat v poskytnutých výstupech a zároveň nesmíme opomenout i dynamicky aktualizované grafy, které nakonec byly provedeny neočekávanou, ale mnohem efektivnější technikou. V projektu se nám zatím nepodařilo uskutečnit naše zařízení více mobilní za pomoci solárního napájení a dalších dodatkových modulů. V práci hodláme pokračovat i po konečné obhajobě, bude tedy mnoho času poté pro další vývoj.

Kohokoli by náš projekt zaujal a chtěl by v něm pokračovat, kontaktujte nás a my vám jistě poskytneme všechny materiály v plné míře. Jeho modifikace a další šíření není žádným způsobem omezeno, jen prosím zachovejte tento smysl otevřenosti a seznamujte s naší prací i ostatní zájemce.

Použitá literatura

1. Apache HTTP Server. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2017 [cit. 2017-12-27]. Dostupné z: https://cs.wikipedia.org/wiki/Apache_HTTP_Server
2. Informace o spolku Hydronet.cz. *Hydronet.cz: Projekt volného šíření online meteorologických a obrazových informací.* [online]. Železný Brod: Hydronet.cz, 2010 [cit. 2017-12-28]. Dostupné z: <http://www.hydronet.cz/content/cz/Ospolku.aspx>
3. Raspberry Pi. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2017 [cit. 2017-12-31]. Dostupné z: https://cs.wikipedia.org/wiki/Raspberry_Pi#Charakteristika

Seznam obrázků a tabulek

Obrázek 1: Výstřížek z webu: 217.5.208.3	11
Obrázek 2: Ukázka zdrojového kódu z: 217.5.208.3	12
Obrázek 3: Pohled na říční tok: hydronet.cz	14
Obrázek 4: Výstřížek z: kamery.humlnet.cz	16
Obrázek 5: Výstřížek z: kamery.humlnet.cz	16
Obrázek 6: Výstřížek z webu: cz.webcams.travel	18
Obrázek 7: Výstřížek z webu: cz.webcams.travel	19
Obrázek 8: Raspberry Pi	22
Obrázek 9: Adafruit MPL3115A2, výstřížek z: adafruit.com	23
Obrázek 10: Adafruit Si7021, výstřížek z: adafruit.com	23
Obrázek 11: RPi s kamerou, zdroj: https://tinyurl.com/y7nmkm2n	24
Obrázek 12: Pořízené moduly	24
Obrázek 13: Prvotní snímky	25
Obrázek 14: První umístění	25
Obrázek 15: Prototypování modulů	26
Obrázek 16: Úspěšný výpis hodnot	26
Obrázek 17: Výstup snímku na webu	27
Obrázek 18: Archivace na Google disk	27
Obrázek 19: Výstup snímků a animací na webu	28
Obrázek 20: Banner na webu	28
Obrázek 21: FusionCharts graf	28
Obrázek 22: Elektroinstalační box, zdroj: www.famatel.cz	29
Obrázek 23: Výhled pro zařízení	29
Obrázek 24: Testovací prototyp	30
Obrázek 25: cam-mark skript ukázka	32
Obrázek 26: cam-mark skript ukázka	32
Obrázek 27: cam-get skript ukázka	32
Obrázek 28: Průměr	32
Obrázek 29: Definice registrů	32
Obrázek 30: Ukázka výpočtu teploty	33
Obrázek 31: Výpočet průměru	33
Obrázek 32: uploader skript ukázka	33
Obrázek 33: Podmínka průměru	33
Obrázek 34: Zařízení v interiéru	31
Obrázek 35: Zařízení v exteriéru	33
Obrázek 36: 3D design case	33
Obrázek 37: Náskres case	33
Obrázek 38: Tisk case	33
Obrázek 39: Výsledný produkt tisku	36
Obrázek 40: cam-anim-1hour ukázka	37
Obrázek 41: Generování animace	37
Obrázek 42: Upload animace	38
Obrázek 43: Přesun snímků do další struktury	38
Obrázek 44: HTML struktura	40
Obrázek 45: Ukázka zdrojového kódu webu	40
Obrázek 46: Databáze a tabulky	41
Obrázek 47: Pole v tabulce	41
Obrázek 48: Struktura tabulky	41
Obrázek 49: Archivační tabulka pro pořízené snímky	42
Obrázek 50: Pohled na snímek z webu imgur	43
Obrázek 51: Struktura archivace animací	44

Příloha: zdrojové Příloha: zdrojové kódy, skripty a programové struktury

K této práci přikládáme veškeré programové materiály a struktury, které jsme během naší práce vytvořili. Všechny tyto struktury obsahují podrobné komentáře. Najdete zde i další prostředky, které jsou rovněž výstupy našeho projektu. Popř. můžete také sledovat doménu meteocam.cz, na kterou se nejsnadněji dostanete naskenováním následujícího QR kódu.

1. 3D modely
2. Adresářová struktura a skripty webového serveru
3. Adresářová struktura pro snímky a tvorbu animací
4. Skripty pro obsluhu lokální stanice RPi
5. Naše logo – MeteoCam
6. Odkaz na naši funkční webovou aplikaci
7. QR kód - www.meteocam.cz

