

GYMNAZIUM JANA KEPLERA

MATURITNÍ PRÁCE

Školní meteorologická stanice

Jaroslav Lžičař

Předmět: Informatika

Vedoucí práce: Pavel Zbytovský

Březen 2015

Prohlášení

Prohlašuji, že jsem jediným autorem této maturitní práce a všechny citace, použitá literatura a další zdroje jsou v práci uvedené.

Tímto dle zákona 121/2000 Sb. ve znění pozdějších předpisů uděluji bezúplatně škole Gymnázium Jana Keplera, Praha 6, Parléřova 2 oprávnění k výkonu práva na rozmnožování díla (§ 13) a práva sdělování díla veřejnosti (§ 18) na dobu časově neomezenou a bez omezení územního rozsahu.

V Praze dne 23. Března 2015

.....

Anotace

Cílem této práce je vytvořit amatérskou meteorologickou stanici určenou pro provoz ve škole. Stanice bude schopna měřit aktuální stav počasí a tyto informace zpřístupňovat přes webové stránky a ukládat do databáze. Součástí projektu je návrh a konstrukce zařízení s využitím vývojové platformy arduino, řešení jeho připojení k síti a komunikace se serverem a následně vizualizace a ukládání dat pomocí webové aplikace.

Obsah

1.	ÚVOD	5
2.	TEORIE	1
2.1.	TEPLOTA.....	1
2.2.	TLAK	1
2.3.	VLHKOST VZDUCHU.....	1
2.4.	ROSNÝ BOD	1
3.	HARDWAROVÁ ČÁST	2
3.1.	STRUKTURA.....	2
3.2.	SCHÉMA ZAPOJENÍ.....	2
3.3.	POUŽITÉ SENZORY.....	3
3.4.	ZAPOJENÍ SENZORŮ.....	3
3.5.	ARDUINO.....	4
3.6.	MODUL PRO PŘIPOJENÍ K SÍTI	4
3.7.	POUZDRA CHRÁNÍCÍ KOMPONENTY.....	5
4.	SOFTWAROVÁ ČÁST	6
4.1.	STRUKTURA.....	6
4.2.	ZDROJOVÝ KÓD PRO ARDUINO.....	7
4.3.	PRÁCE S DATABÁZÍ.....	8
4.4.	JAVASCRIPT A JEHO KOMUNIKACE S PHP	9
4.5.	GRAFY.....	11
4.6.	VZHLED STRÁNKY	12
5.	ZÁVĚR.....	13
6.	NÁVOD NA SPUŠTĚNÍ	14
7.	ZDROJE.....	14

1. Úvod

Když jsem vybíral téma svého maturitního projektu, usiloval jsem o to, aby nebyl jen čistě softwarový, ale zahrnoval i elektrotechnickou část, která by umožnila nějakou formu kontaktu s reálným světem. Pro tento účel jsem zvolil arduino, což je vývojová deska založená na jednoduchém mikroprocesoru, ke které je možno připojit ostatní součástky přes univerzální vstupy a výstupy. Arduino lze též ovládat přes počítač a rozšiřovat ho mnoha přídavnými moduly. Výhodou této platformy je, že její zdrojové kódy i schémata zapojení jsou open-source, je tedy možné je libovolně upravovat a vytvářet si vlastní kopie.

Základem stanice je tedy arduino, ke kterému jsou připojeny senzory pro měření meteorologických údajů. Na něj navazuje modul umožňující jeho připojení k internetu.

Systém funguje tak, že arduino přečte data ze senzorů odešle je na sever. Tam se nachází webová aplikace obsahující skript, který příchozí informace uloží do databáze. Když si uživatel zobrazí stránku, načtou se potřebná data z databáze, zobrazí se aktuální hodnoty a vykreslí se grafy. Po dalším měření se zobrazené hodnoty obnoví bez nutnosti aktualizace stránky.

Meteorologická stanice umožňuje sledovat jak aktuální stav počasí, tak i jeho krátkodobý vývoj, ze kterého lze odhadovat, jaké bude počasí v blízké budoucnosti. Zároveň dlouhodobý graf může zaznamenávat různé klimatické rekordy a anomálie.

2. Teorie

Meteorologická stanice měří teplotu, tlak a vlhkost vzduchu. Dále z nich potom dopočítává rosný bod a pocitovou teplotu. Tyto údaje stačí na to, aby nám podaly základní charakteristiku počasí. Zvažoval jsem i měření rychlosti větru, to by ovšem vyžadovalo zvláštní měřící jednotku na specifickém místě a celý projekt by se tím značně technicky zkomplikoval.

2.1. Teplota

Teplotou vzduchu se v meteorologii rozumí teplota naměřená ve stínu ve výšce 2 m nad zemí.^[1] Udává se v Celsiově stupnici. Nejvyšší denní teplota nastává kolem 14. hodiny, nejnižší několik minut po východu slunce. Teplotu vzduchu lze zjišťovat pouze ve stínu, jinak by bylo měření ovlivněno tím, že by se teplotní čidlo zahřívalo od slunce.

2.2. Tlak

Atmosférický tlak je síla, kterou působí atmosféra Země na jednotkovou plochu v daném místě. Měří se v Pascalech (Pa) a jeho násobcích. Tlak vzduchu klesá se stoupající nadmořskou výškou, závisí ale také na ročním období, teplotě vzduchu a obsahu vodní páry v atmosféře^[2]. Pro možnost porovnávání údajů se v meteorologii přepočítává na relativní hodnotu vzhledem k tlaku u hladiny moře.

2.3. Vlhkost vzduchu

Vlhkost vzduchu udává množství vodní páry ve vzduchu. Má významný vliv na to, jak člověk vnímá teplotu. Nejčastěji se používá relativní vlhkost, což je poměr okamžitého množství par ve vzduchu ku množství par, které by měl vzduch při plném nasycení. Udává se v procentech^[3].

2.4. Rosný bod

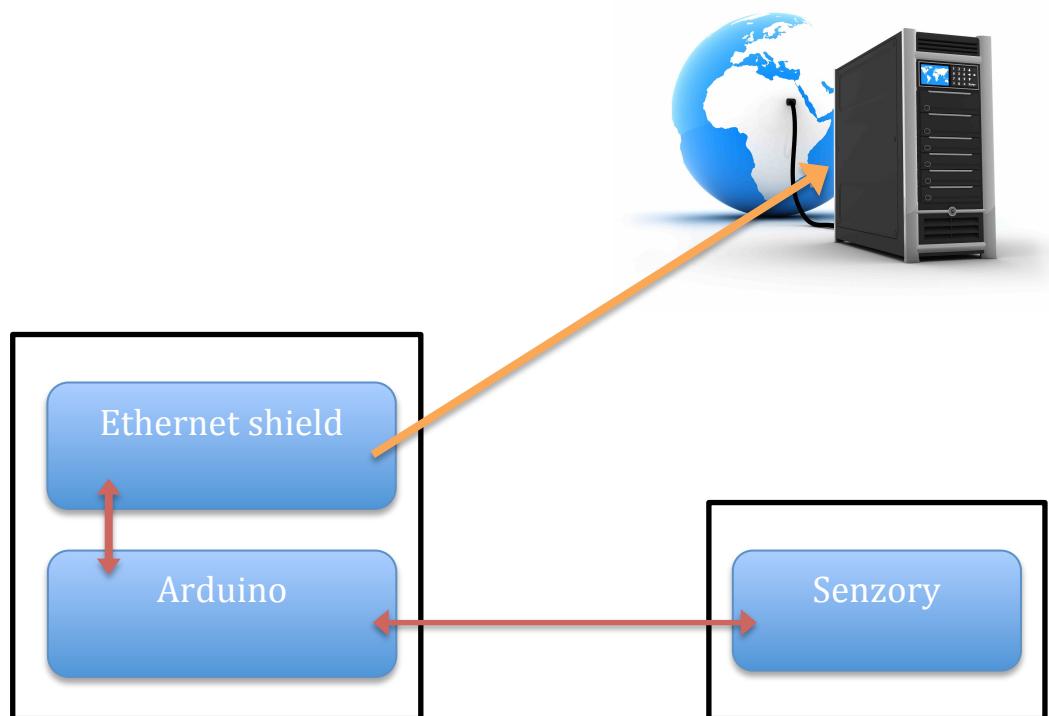
Rosný bod je teplota, při které je vzduch již maximálně nasycen vodními parami. Jestliže klesne teplota pod tento bod, dojde ke kondenzaci – pára obsažená ve vzduchu začne kapalnět.^[4] Čím vyšší je vlhkost vzduchu, tím vyšší je i rosný bod.

3. Hardwarová část

3.1. Struktura

Meteorologická stanice je tvořena dvěma krabičkami. První z nich obsahuje arduino a modul pro jeho připojení k internetu, zvaný Ethernet shield. Je do ní přivedeno napájení ze síťového adaptéru a ethernetový kabel. Senzory se nachází uvnitř druhé krabičky, která je s tou předchozí propojena datovým kabelem. Pouzdro se senzory je jediná venkovní část, zbytek systému je umístěn uvnitř budovy.

3.2. Schéma zapojení



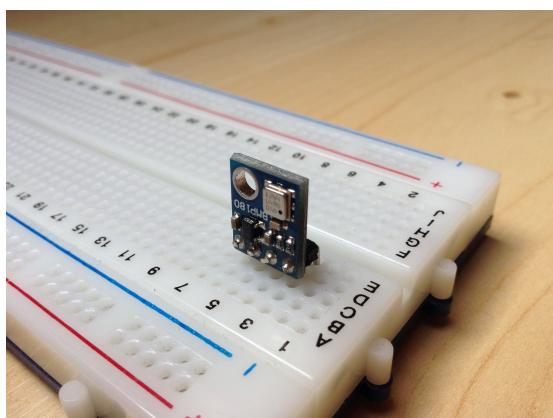
3.3. Použité senzory

Jak jsem již zmínil v teoretické části této práce, měřena bude teplota, tlak a vlhkost vzduchu. Jelikož měření tlaku vyžaduje kompenzaci v závislosti na teplotě, součástka měřící tlak obvykle obsahuje i teplotní čidlo.

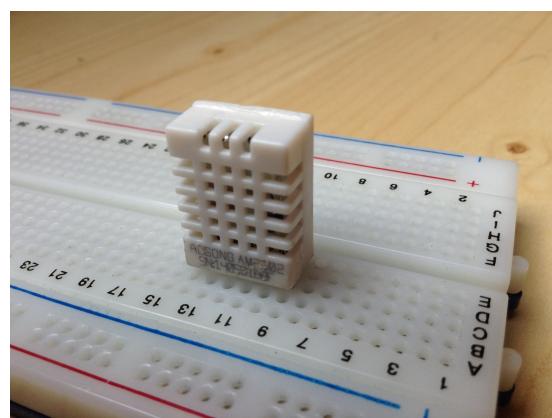
Právě takovým senzorem je BMP 180 od firmy BOSCH, pro jehož komunikaci s arduinem jsou navíc připravené knihovny. Je schopen měřit tlak s přesností 0.12 HPa a teplotu s odchylkou menší než 0.5 °C. Součástka je plně digitální a k arduinu se připojuje přes sběrnici I²C.

Jako senzor vlhkosti slouží DHT 22 od Aosong Electronics, který dokáže měřit relativní vlhkost s přesností 2%. Kromě toho nabízí též měření teploty, které ale ve svém projektu nevyužiji, neboť není tak přesné jako v případě předchozího senzoru. Komunikace probíhá také digitálně přes protokol 1-wire.

BMP 180

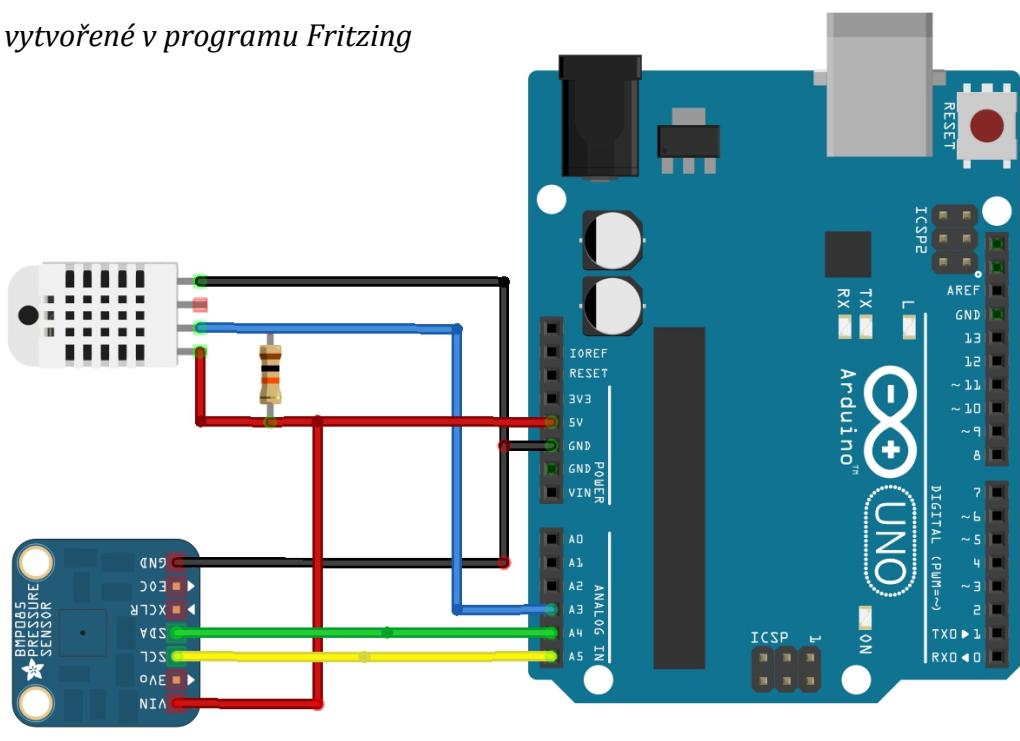


DHT 22



3.4. Zapojení senzorů

Schéma vytvořené v programu Fritzing



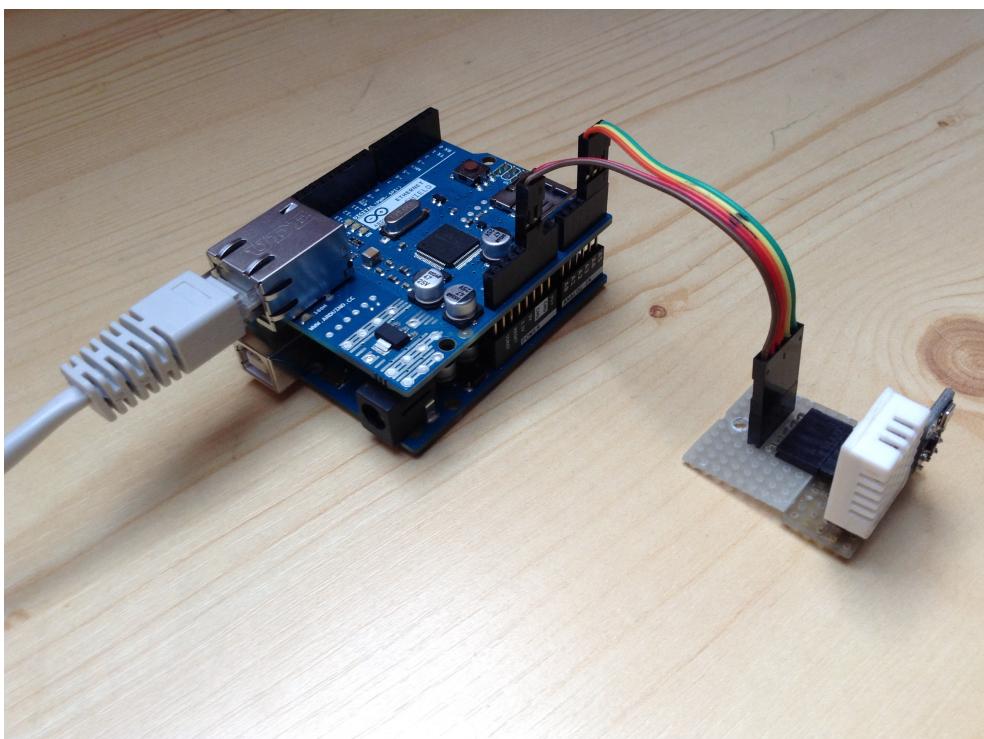
3.5. Arduino

Arduino je vývojová platforma založená na mikrokontroleru Atmel. Jedná se o nejdůležitější část stanice, neboť řídí veškerou její činnost. Arduino se vyrábí v mnoha verzích, které se liší použitým procesorem a počtem vstupů a výstupů. Já jsem zvolil základní verzi Arduino uno s procesorem ATmega328. Tento mikrokontroler má dostatečnou vnitřní paměť o velikosti 32 KB a disponuje 14 vstupně-výstupními piny. Navíc zaručuje kompatibilitu s rozšiřujícími moduly, které budu používat. K počítači se připojuje přes USB je ho možné programovat pomocí vývojového prostředí, které je volně ke stažení.

3.6. Modul pro připojení k síti

Jako nevhodnější se mi jevilo použít oficiální modul pro arduino zvaný Ethernet shield. Ten nabízí konektor RJ45 pro připojení k síti. Nasazuje se přímo na arduino prodlužuje jeho vstupní a výstupní piny. V základní výbavě vývojového prostředí je již připravená knihovna, které usnadňuje jeho programování.

Arduino s Ethernet shieldem a připojenými senzory

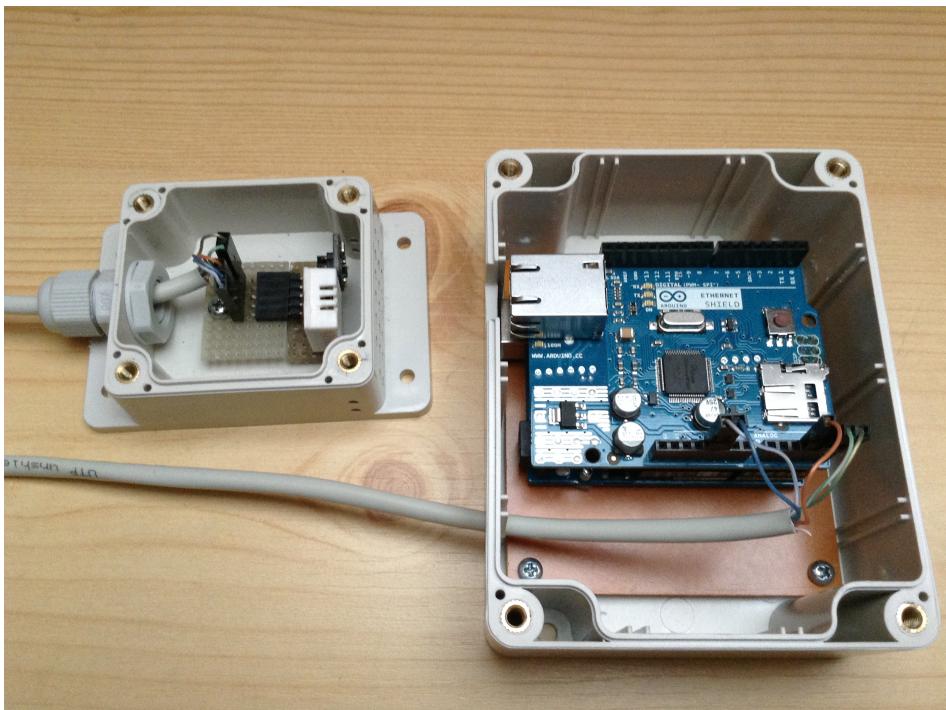


3.7. Pouzdra chránící komponenty

Venkovní část systému je ukryta v průmyslové krabičce splňující normu krytí IP65. Krabička je připevněna ke zdi ve vertikální poloze a všechny otvory pro průchod vzduchu se nachází ve spodní části, což vylučuje vniknutí vody a poškození součástek. Zároveň mají senzory zajištěn dostatečný přísun vzduchu zvenku. Datový kabel je vyveden horní částí skrz vodotěsnou průchodku. Celá krabička je nastříkána bílým lesklým lakem, což maximalizuje odraz světla, které by způsobovalo nežádoucí zahřívání senzorů.

U druhé krabičky není kladen důraz těsnění a odolnost, protože je instalována uvnitř budovy. Ukrývá arduino a Ethernet shield. V její pravé části nalezneme tři otvory – pro napájení, připojení síťového kabelu a pro datový kabel vedoucí k venkovní části. Oběma krabičkám lze odšroubovat víko a libovolné komponenty vyměnit.

Odkryté krabičky



Venkovní jednotka se senzory

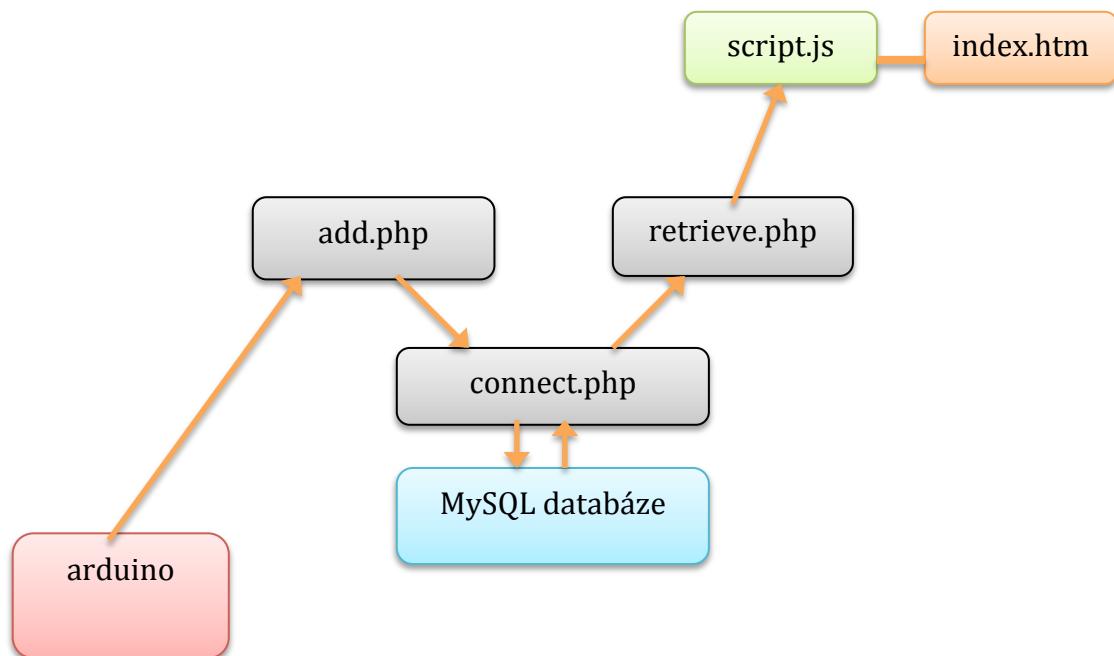


4. Softwarová část

4.1. Struktura

Softwarové řešení meteorologické stanice využívá mnoho technologií a různé programovací jazyky. Vše začíná u arduina, které přečte naměřené hodnoty ze senzorů a odešle je na server jako http požadavek souboru *add.php*. Ten se přes další soubor *connect.php* připojí k MySQL databázi a data do ní uloží.

Když uživatel zobrazí stránku, javascript v souboru *script.js* pošle AJAX žádost souboru *retrieve.php*, který se připojí k databázi, také přes *connect.php*, odtud načte potřebná data a předá je javascriptu ve formátu JSON. Soubor *script.js* potom data předá grafu, který je zobrazí na stránce. Po uplynutí 1 minuty, během které proběhne další měření, je vyslán nový AJAX požadavek a data jsou obnovena.



4.2. Zdrojový kód pro arduino

Kód pro arduino je psaný v jazyce Wiring, který vychází z C. Hlavními částmi jsou funkce setup, která se vykoná při spuštění a funkce loop, která je poté prováděna opakovaně. Její obsah je vidět v následující ukázce:

```
void loop(){

    hum=measureHumidity();
    temp=measureTemperature();
    pres=measurePressure(temp);

    printValues();

    data =(String)"temp=" + temp + "&pres=" + pres + "&hum=" + hum +
    "&pass*****";
    sendToServer(server, 80, data);

    //Wait between measurements
    delay(INTERVAL);
}
```

Nejprve jsou naměřeny hodnoty teploty, tlaku a vlhkosti, které se uloží do proměnných. Pro příklad zde uvedu funkci sloužící k měření vlhkosti.

```
int measureHumidity(){
    int value;
    value =(int) (dht.readHumidity()*10);
    if(isnan(value)){
        value = -999;
        Serial.println("error retrieving pressure measurement");
    }
    return value;
}
```

Poté jsou hodnoty začleněny do textového řetězce, který je odeslán na server v rámci http požadavku metodou *post*. Takto probíhá posílání žádosti:

```
void sendToServer(char server[], int port, String data){
    if (client.connect(server,port)) {
        Serial.println("-> Connected");
        client.println("POST /~xlzij01/meteo/add.php HTTP/1.1");
        client.print("Host: ");
        client.println(server);
        client.println("Content-Type: application/x-www-form-
urlencoded");
        client.print("Content-Length: ");
        client.println(data.length());
        client.println();
        client.print(data);
        Serial.println(data);
        Serial.println();
    }
    else Serial.println("Connection failed");

    if (client.connected()) {
        client.stop();
    }
}
```

4.3. Práce s databází

Údaje naměřené meteorologickou stanicí se ukládají do MySQL databáze. Ta obsahuje celkem dvě tabulky. Jedna uchovává hodnoty z každého měření, které se zobrazují v krátkodobém přehledu. Druhá tabulka ukládá pouze maximum a minimum teploty a průměrnou hodnotu tlaku za den a slouží jako zdroj dat pro dlouhodobý graf, kde by zobrazování hodnot za každou minutu způsobilo zbytečně velký objem dat. Uvedu zde některé části kódu ze souboru *add.php*, který obstarává zápis do databáze.

Takto se ukládají do databáze příchozí data:

```
if(isset($_POST["temp"]) && isset($_POST["pres"]) && isset($_POST["hum"])
&& isset($_POST["pass"])){
    if($_POST["pass"] == "*****"){
        $temp = $_POST["temp"];
        $pres = $_POST["pres"];
        $hum = $_POST["hum"];
        mysql_query("INSERT INTO `senzory2` (`pressure`, `temperature`, `humidity`)
VALUES ('".intval($pres)."','".intval($temp)."','".intval($hum)."')", $link);
    }
}
```

Ukládání maximálních a minimálních hodnot do druhé tabulky probíhá následujícím způsobem:

```
$row = mysql_fetch_array(
    mysql_query(
        "SELECT COUNT(*) FROM minmax
        WHERE DATE(date) = DATE(DATE_SUB(NOW(), INTERVAL 1 DAY))",
        $link
    )
);

// Insert yesterdays min, max and average values to the minmax table
if($row[0] == "0"){
    mysql_query(
        "INSERT INTO minmax (date, tmin, tmax, pressure) VALUES(
            (SELECT DATE(DATE_SUB(NOW(), INTERVAL 1 DAY))),
            (SELECT MIN(temperature) FROM senzory2
                WHERE DATE(time) = DATE(DATE_SUB(NOW(), INTERVAL 1 DAY))),
            (SELECT MAX(temperature) FROM senzory2
                WHERE DATE(time) = DATE(DATE_SUB(NOW(), INTERVAL 1 DAY))),
            (SELECT AVG(pressure) FROM senzory2
                WHERE DATE(time) = DATE(DATE_SUB(NOW(), INTERVAL 1 DAY)))
        )", $link
    );
}
```

Princip je takový, že jestliže v tabulce minmax ještě nejsou žádné hodnoty z předchozího dne (počet řádků je 0), ze včerejších záznamů z první tabulky jsou vybrána potřebná data, která jsou pak do tabulky pro maxima a minima vložena.

4.4. Javascript a jeho komunikace s PHP

V javascriptu je psaná velká část mého projektu, ačkoliv jsem původně myslel, že bude převažovat PHP. Důvodem je hlavně použití javascriptových knihoven pro grafy, ale také jeho jednoduchost a účinnost ve spojení s frameworkm jQuery. Komunikace s PHP, které získává data z databáze, probíhá přes AJAX žádost, a sice funkcí *getJSON*.

```
// Initialization...
$(document).ready(function(){

    $.getJSON(PHPFILE, {"type":"c"}, function(data){
        $("#t").text(temp(data)+" °C ");
        $("#p").text(pres(data)+" HPa ");
        $("#h").text(hum(data)+" % ");
    });

    $.getJSON(PHPFILE, {"type":"b"}, function(data){
        var chart2 = createRangeChart(data);
    });

    $.getJSON(PHPFILE, {"type":"a"}, function(data){
        var chart = createChart(data);
        setInterval(
            function(){ updateChart(chart); displayValues(); },
            INTERVAL
        );
    });
});
```

Souboru *retrieve.php*, na který odkazuje proměnná *PHPFILE*, je poslána žádost s parametrem *type*. Ten určuje, jaká data má soubor vrátit. Pokud *retrieve.php* odpoví vrácením požadovaných hodnot, tak je zavolána anonymní funkce, která předá data na potřebné místo. Poslední funkce pak spustí periodické obnovování grafu a ukazatelů aktuálních hodnot metodou *setInterval*.

retrieve.php na žádost odpovídá tímto způsobem:

```

if( isset($_REQUEST["type"]) ){

    $type = $_REQUEST['type'];
    switch ($type) {

        case "a":
            $result=mysql_query(
                "SELECT * FROM (SELECT * FROM senzory2 WHERE date(time) > DATE_SUB(NOW(),
                |INTERVAL 1 WEEK) ORDER BY time DESC) sub ORDER BY time",
                $link
            );
            echo json_encode(mysqlData($result));
            break;

        case "b":
            $result=mysql_query("SELECT * FROM minmax ORDER BY date", $link);
            echo json_encode(mysqlData($result));
            break;

        case "c":
            $result=mysql_query("SELECT * FROM `senzory2` ORDER BY time DESC LIMIT 1", $link);
            echo json_encode(mysql_fetch_row($result));
            break;

    }
    mysql_free_result($result);
}

```

Podle typu žádosti jsou z databáze vybrána požadovaná data. A takto probíhá automatické obnovování grafu:

```

// Live-update 1 week chart
function updateChart(chart){
    $.getJSON( phpFile, {"type":"c"}, function( data ) {
        chart.series[0].addPoint([parseTime(data[0]), temp(data)], true, false);
        chart.series[1].addPoint([parseTime(data[0]), pres(data)], true, false);
        chart.series[2].addPoint([parseTime(data[0]), hum(data)], true, false);
        console.log(data);
    });
}

```

Tentokrát je použita žádost typu c, po které PHP vrátí pouze nejnovější hodnoty. Ty jsou následně přidány na konec grafu. Pro zajímavost uvedu ještě jednu funkci, která při zmenšení nebo zvětšení okna prohlížeče dynamicky mění velikost písma.

```

// Dynamically update font size
function setFontSize(){
    $("#p,#h,#hi,#dp").css(
        "font-size", ($("#p").width() < 250) ? $("#p").width()/5.5 : "45px"
    );
    $("#t").css(
        "font-size", ($("#t").width() < 500) ? $("#t").width()/5 : "100px"
    );
    $("#time").css(
        "font-size", ($("#t").width() < 500) ? $("#t").width()/18 : "28px"
    );
}

```

4.5. Grafy

Pro vykreslování grafů jsem vybíral z několika frameworků, a nakonec jsem zvolil Highcharts. S touto knihovnou se dá pracovat velmi jednoduše, a přitom nabízí nespočet možností a některé velmi pokročilé funkce. Vlastnosti každého grafu jsou určeny jeho konfiguračním objektem který lze libovolně upravovat a přizpůsobit tak graf svým potřebám. Mezi klíčové vlastnosti patří podpora časové osy, jejíž rozlišení se mění v závislosti na aktuálním rozsahu, a možnost dynamicky do grafu přidávat další body po jeho vytvoření. Také se mne zaujala možnost přesné volby rozsahu pomocí navigátoru pod grafem, či nastavitelné shlukování blízkých bodů, které značně urychlují zobrazování grafu.

Na stránce *index.html* se nachází dva grafy. První zobrazuje vývoj teploty, tlaku a vlhkosti vzduchu za poslední 3 dny, přičemž jeho rozsah lze prodloužit až na 1 týden. Starší záznamy již neobsahuje, protože by se pak muselo načítat příliš velké množství dat. Z druhého grafu lze vypozorovat teplotní maximum a minimum za každý den, popřípadě i průměrné hodnoty tlaku. Jeho rozsah není nijak omezen.

Screenshot grafů

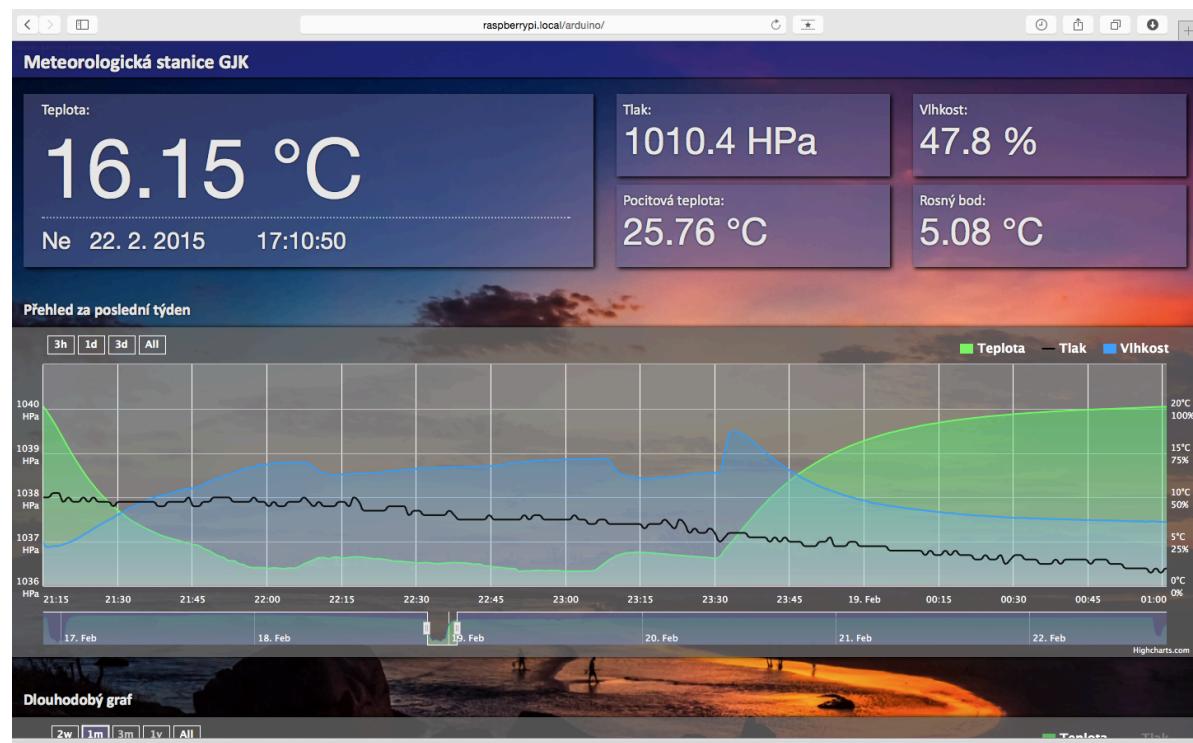


4.6. Vzhled stránky

Většina grafických prvků je tvořena pomocí kaskádových stylů, které jsou obsaženy v souboru *stylesheet.css*. Vzhled stránky se přizpůsobuje velikosti okna a nabízí i speciální zobrazení pro mobilní zařízení. To je implementováno pomocí *media queries*.

```
@media only screen and (max-device-width:800px), (max-width:800px){  
    .values{  
        width: 48%;  
        max-width: 100%;  
    }  
  
    #teplota{  
        width: 98%;  
        max-width: 100%;  
    }  
}
```

Screenshot webové stránky



5. Závěr

Projekt byl sice časově velmi náročný, ale během jeho tvorby se mi podařilo originálním způsobem vyřešit spoustu problémů a naučil jsem se mnoha novým dovednostem a teoretickým poznatkům. Jsem rád za to, že se mi podařilo dojít od pouhého nápadu až k funkčnímu celku, který zasahuje do mnoha různých oborů. Na závěr bych chtěl poděkovat všem, kdo se na mému projektu účastnili a poskythli mi svůj čas, prostor, zařízení, odborné rady, peníze nebo i kritiku a připomínky. Projekt sice takto odevzdávám, ale nevylučuji tím jeho budoucí úpravy nebo rozšíření.

6. Návod na spuštění

Jelikož se jedná o aplikaci v rámci webové stránky, stačí zadat do internetového prohlížeče následující odkaz:

<http://www.gjk.cz/~xlzij01/meteo/>

7. Zdroje

1. <http://www.garni-meteo.cz/cz/meteorologie/>
2. <http://www.meteocentrum.cz/encyklopedie/tlak-vzduchu.php>
3. http://cs.wikipedia.org/wiki/Vlhkost_vzduchu
4. http://cs.wikipedia.org/wiki/Rosný_bod