|  |
| --- |
| Gymnázium Jiřího Gutha- Jarkovského |
| Meteorologická stanice |
| Eva Horalíková |
|  |
| **Školní rok 2016/2017** |
| **Vedoucí Mgr. Jana Racková** |

|  |
| --- |
|  |

OBSAH

[1. Úvod 3](#_Toc480223600)

[2. Atmosferické jevy 3](#_Toc480223601)

[2.1 Teplota 3](#_Toc480223602)

[2.2 Relativní vlhkost 3](#_Toc480223603)

[2.3 Rosný bod 3](#_Toc480223604)

[3. Historie měření atmosferických jevů 3](#_Toc480223605)

[3.1 Rozhovor s dědou 4](#_Toc480223606)

[4. Součástky 5](#_Toc480223607)

[4.1 Arduino Uno 5](#_Toc480223608)

[4.2 Ethernet sheild 6](#_Toc480223609)

[4.3 Nepájivé pole a vodiče 6](#_Toc480223610)

[5. První verze 6](#_Toc480223611)

[5.1 Senzor TMP 36 6](#_Toc480223612)

[5.2 Zapojení 8](#_Toc480223613)

[5. Druhá verze 8](#_Toc480223614)

[5.1 Senzor DHT11 8](#_Toc480223615)

[5.2 Zapojení a konstrukce 10](#_Toc480223616)

[5.3 Sběr a vizualizace dat 11](#_Toc480223617)

[5.4 Filtrace dat 15](#_Toc480223618)

[6. Závěr 15](#_Toc480223619)

[6.1 Resumé v AJ 15](#_Toc480223620)

[7. Prameny 15](#_Toc480223621)

[8. Zdrojové kódy 16](#_Toc480223622)

[8.1 První verze 16](#_Toc480223623)

[8.2 Druhá verze 17](#_Toc480223624)

[8.3 Filtrace dat 22](#_Toc480223625)

# 1. Úvod

Jako ročníkovou práci, jsem se rozhodla letos sestavit domácí meteorologickou stanici. Hlavně jsem si chtěla prohloubit znalosti z elektrotechniky a naučit se programovat v jazyce v C. Od začátku jsem chtěla použít mikropočítač Arduino se kterým jsem se seznámila na Letní škole IT na Fakultě informatiky MUNI v Brně.

# 2. Atmosferické jevy

## 2.1 Teplota

Teplota je skalární fyzikální veličina, související s potencionální energií těles.

Ve většině meteorologických stanic se zaznamenávají následující měření:

* denní minimální teplota
* denní maximální teplota
* průměrná denní teplota (jedná se o [aritmetický průměr](https://cs.wikipedia.org/wiki/Aritmetick%C3%BD_pr%C5%AFm%C4%9Br) z teploty vzduchu naměřené v 7 hodin, teploty ve 14 hodin a dvojnásobně započtené teploty v 21 hodin)

Vedle toho se také sleduje

## 2.2 Relativní vlhkost

Relativní vlhkost udává množství vodní páry ve vzduchu, toto množství je velice proměnlivé a moje měření nebyly bohužel vůbec přesné. Dle mého názoru jsem to způsobila papírovou krabičkou kterou jsem přikryla senzor a uvnitř se vytvořilo mikroklima. Vlhkost vzduchu je ovšem klíčová pro předpověd počasí. Je zaznamenávana v procentech.

## 2.3 Rosný bod

Rosný bod je teplota, při které dosáhl vzduch maximální možné vlhkosti, tedy při které by se právě přítomná pára ve vzduchu stala nasycenou (při nezměněném tlaku vzduchu) a může tedy docházet ke kondenzaci (relativní vlhkost vzduchu je 100 %). Bývá nižší nebo rovna teplotě vzduchu.

# 3. Historie měření atmosferických jevů

Meteorolgie je věda zkoumající a popisující atmosféru a atmosférické jevy, toto slovo poprvé použil antický filozof Aristoletes ve svém spise Meteorologica v 4. století př.n.l. ve kterém se zabýval možností využití matematické analýzy, o vyjadřování reálné skutečnosti, příčinnosti jevů a procesů vývoje počasí matematickými prostředky. Určité znalosti z meteorologie byly zachycené v lidové slovesnosti v pranostikách. První vědecké pokusy měřit počasí provedl Leonardo da Vinci v 16. století sestrojením hygrometru, tedy měřícího přístroje ukazující relativní vlhkost vzduchu. Dalšími průkopníky měření počasí byli v 17. století Galileo Galilei (sestrojil kapalinový teploměr) a Jan Evangelista Torricelli (rtuťový tlakoměr). První záznamy z našeho území pochází z Kosmovy kroniky české z 10. století a mnoho staletí později vzniky žerotínské denní záznamy o počasí z let 1533-1534. Roku 1771 vznikla dodnes měřící stanice při jezuitské koleji v Praze Klementinu a 1804 klementinská řada srážek.

## 3.1 Rozhovor s dědou

V rámci této ročníkové práce jsem také udělala rozhovor s mým dědou Václavem Poláškem, který používal a vysílal meteorologické balóny v rámci svého výzkumu v telekomunikacích.

Co jsi měřil za veličiny?

Měřil jsem teplotu, vlhkost vzduchu a výšku sondy.

Proč jsi měřil počasí?

Sonda na balonu posílala data z prostoru mezi dvěma radioreleovými stanicemi, kde procházel paprsek digitálního signálu. Data jsem vyhodnocoval pomocí programu v počítači,kde na jedné ose je čas a na další výška sondy a výsledný graf. Souběžně s vyhodnocováním dat meteorologické sondy, jsem vyhodnocoval data kvality signálů měřených rádií.

Jaké jsi používal přístroje?

Používal jsem sondu jež byla vypouštěna na meteorologickém balonu o průměru 1.5 - 2 metrů, plněného héliem na silonové šňůře do výšky 400 - 700 metrů. Sonda měla senzory tlaku (na měření výšky, měřily ve stopách), teploměr (bimetalový) a pro vlhkost vzduchu hygrometer.

Jak byly přesné?

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX



Kde a kdy jsi prováděl měření?

Měření jsem prováděl většinou v noci kolem druhé třetí hodiny někde za Edmontonem v provincii Alberta v Kanadě, mezi léty 1989-1990.XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

Co jsi používal na vyhodnocení dat?

Sonda má rádio, kterým vysílá uvedené údaje na frequenci kolem 430 KHz. Přijímač údajů sondy data acquisition instrumemet byl napojen na počítač s tiskárnou. Interval vysílání dat bylo možno zvolit v rozsahu od  5 do 10 vteřin. Data byla zobrazována v řádcích. Jeden řádek obsahoval Datum, čas, teplotu, vlhkost a výšku.

# 4. Součástky

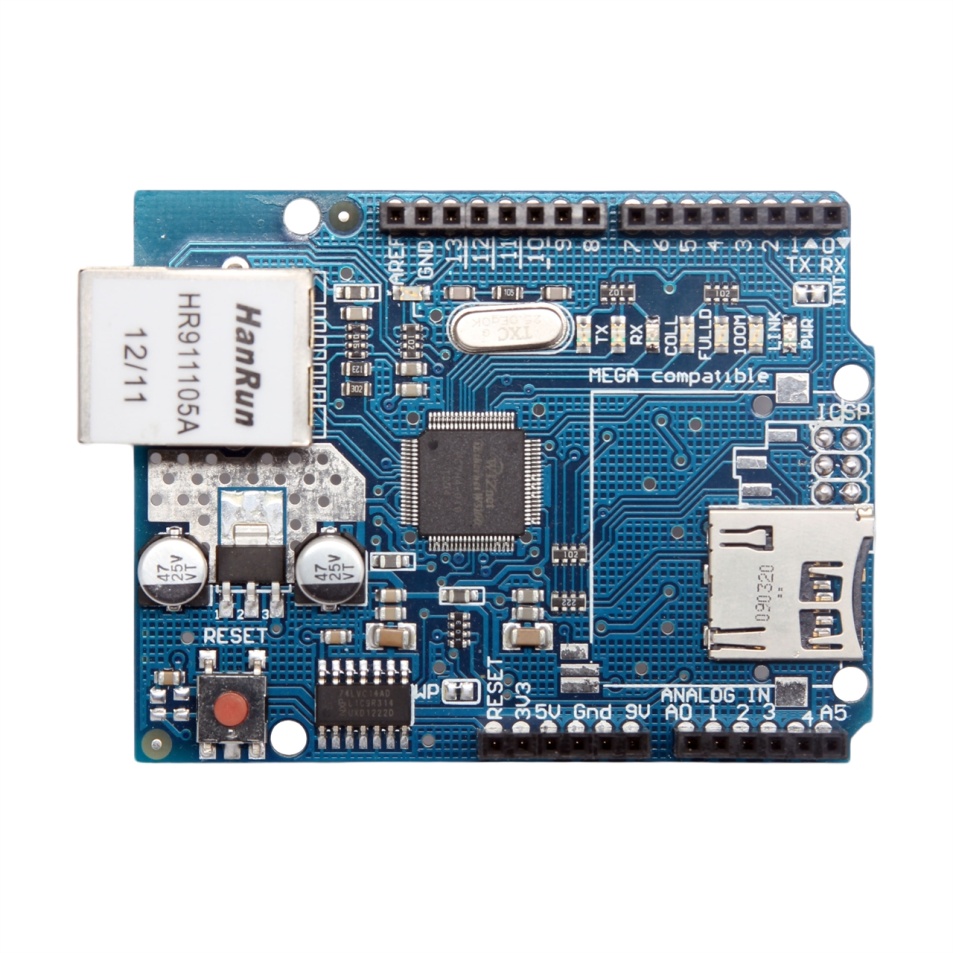
## 4.1 Arduino Uno

Arduino je jednoduchý počítač založený primárně určený na výuku programování a elektroniky. Je založený na mikrokontrolerech ATmega od firmy Atmel. Programuje se pomocí Arduino IDE v jazyku velice podobném jazyku C. Já jsem používala model Arduino Uno a to jednak originální Arduino a později klon (napodobenina, celá idea Arduina je dělat všechno tzv. opensource, tedy otevřeně a lze tedy legálně prodávat kopie).



## 4.2 Ethernet sheild

Sheildy jsou desky



XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

## 4.3 Nepájivé pole a vodiče

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

# 5. První verze

První verzi mé meteorologické stanice jsem sestavila už v listopadu a skládala se pouze z teplotního senzoru TMP 36 a Arduina uno. Později jsem za účelem prezentace připojila i obrazovku z tekutých krystalů a tilt switch. Zdrojový kód je v příloze.

Sestavila jsem obvod a nahrála kód. K mému velkému překvapení se hodnoty pohybovaly mezi -30 a +20 °C. Začala jsem propadat mírné panice, že jsem teprve u první verze a že mi to vůbec nefunguje. Zapojení a kód byl správně a nemohla jsem najít chybu. Nakonec po konzultaci s tátou jsme přišli na to, že jsem místo teplotního senzoru zapojila transistor. Jejich identická velikost také fakt, že jsem měla součástky neutříděné, vedly k tomu, že jsem nevěnovala pozornost číslům na senzoru a zapojila to první, co jsem našla bez toho, abych to ověřila. Nicméně jsem nic nezničila a poučila jsem se a později jsem pečlivě kontrolovala všechny obvody a zapojení.

## 5.1 Senzor TMP 36

TMP36 je nízkonákladový senzor teploty, který je součástí Arduino startovacího balíčku a proto jsem s mím začala pracovat jako s prvním.

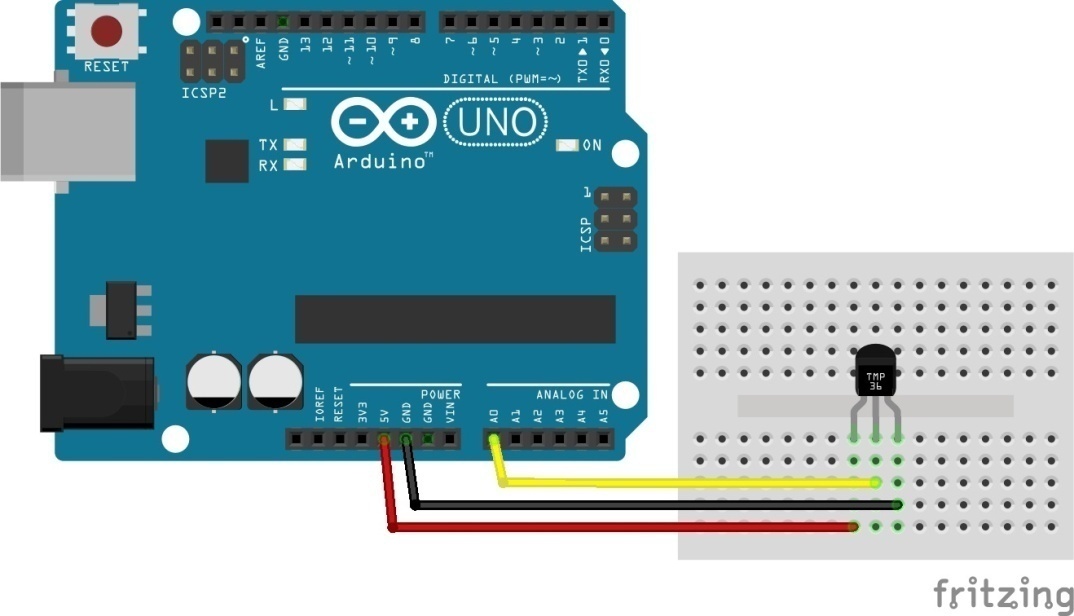
Senzor je termistor, tj. rezistor s proměnným odporem závislým na teplotě. Senzor tedy mění napětí v obvodu a Arduino z toho získá teplotu. Senzor posílá střídavý proud, tedy Arduino musí nejdříve změnit střídavý na stejnosměrný a poté změřit velikost napětí procházející obvodem. Čidlo je nakalibrované tak, každých 0,01 mV odpovídá jednomu stupni Celsia, Arduino tedy teplotu vypočítá a pošle po sériové lince, kde ji můžeme sledovat, nebo uložit na SD kartu.

Kladem tohoto senzoru je jeho velikost a také rychlá odezva.

Vlastnosti

|  |  |
| --- | --- |
| Napájecí napětí | 2.7 V až 5.5 V |
| Rozsah měření teploty | −40°C až +125°C |
| Přesnost měření teploty | ±2°C |
| Rozměry (mm) | 12 x 4 x3 |

## 5.2 Zapojení



# 5. Druhá verze

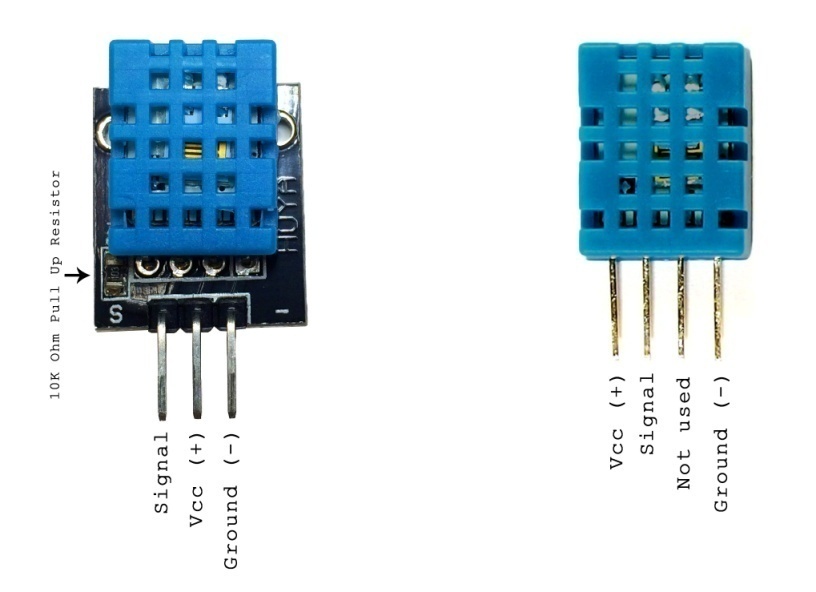
Jako další krok v mé ročníkové práci jsem se rozhodla použít senzor, jenž bude měřit i vlhkost. Jako velice cenově dostupný a snadný na zapojení jsem vyhodnotila senzor DHT11.

## 5.1 Senzor DHT11

DHT11 je velice jednoduchý senzor teploty a relativní vlhkosti. Jeho největší přednosti jsou malé rozměry a nízká cena. Je složen ze termistoru a hydrometru. Je také v plastovém pouzdře, takže nemusíme řešit obal. Je již upraven na zapojení k mikrokontroleru jako je Arduino nebo Raspberry Pi. Já jsem s ním začala pracovat v prosinci a nejdříve mi to vůbec nešlo, senzor neposílal žádná data, ale nakonec jsem zjistila, že mi pořádně neseděly kontakty, jelikož jsem používala nepájivé kontaktní pole. To ale nebyl jediný problém, kterému jsem musela čelit. V kódu, který jsem původně chtěla použít a který byl také v dokumentaci k tomuto senzoru byla novější verze knihovny na základě níž ten senzor vypočítá tu vlastní teplotu, než jsem měla ve svém prostředí. Tedy nefungoval mi program už napsaný výrobcem.

Vlastnosti:

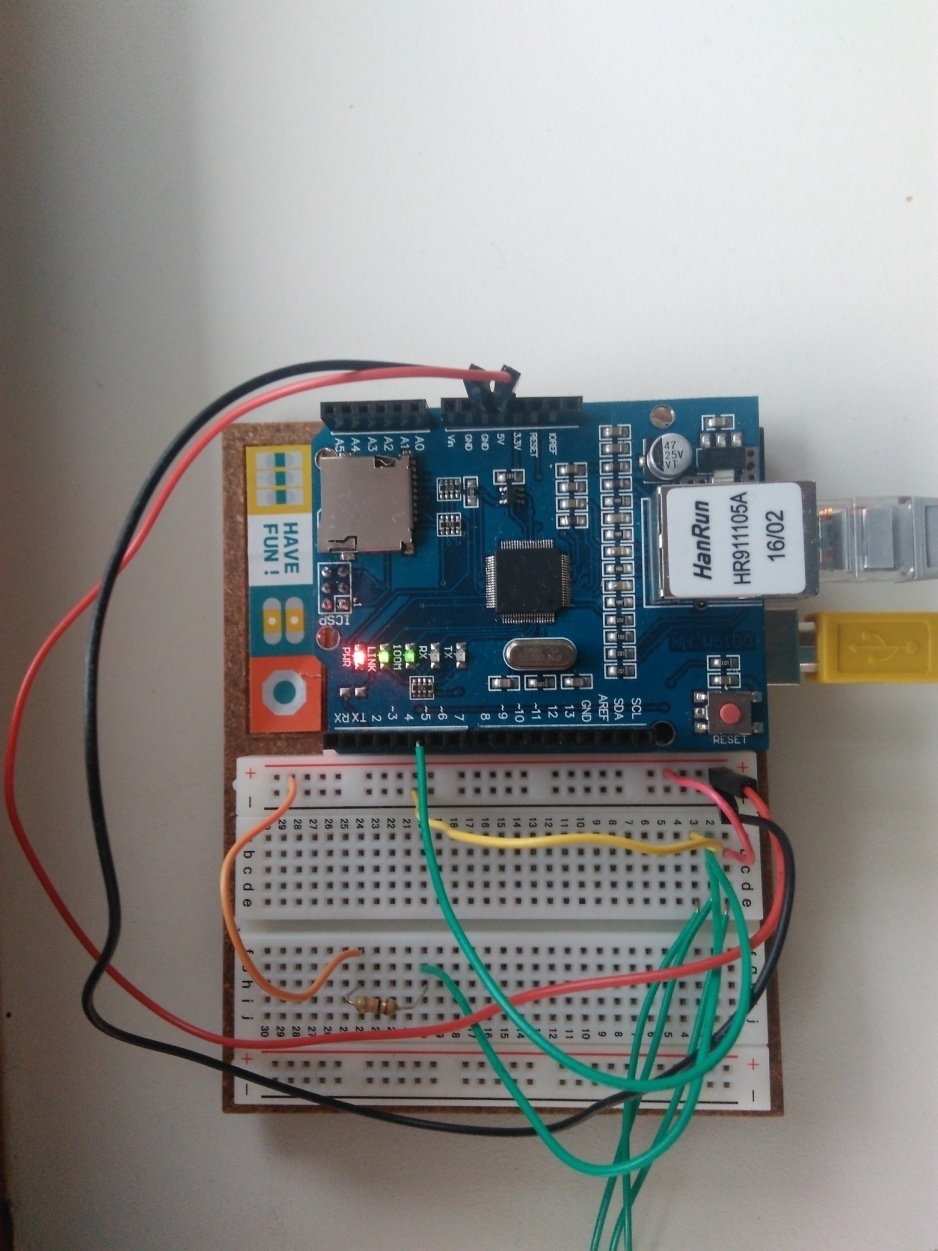
|  |  |
| --- | --- |
| Napájecí napětí | 3–5,5 V |
| Rozsah měření vlhkosti | 20% ~ 90% |
| Přesnost měření vlhkosti | ± 5.0 % RH |
| Rozsah měření teploty | 0 až +50 ℃ |
| Přesnost měření teploty | ± 2.0 ℃ |
| Odezva senzoru | < 5 s |
| Rozměry (mm) | 23 x 12 x 5 |



Senzor DHT11, na pravé straně bez rezistoru a vlevo s již napájeným rezistorem, v meteostanici jsem používala variantu vlevo, ale variantu bez rezistoru jsem používala také.

## 5.2 Zapojení a konstrukce

## Arduino-and-DHT11_bb.png

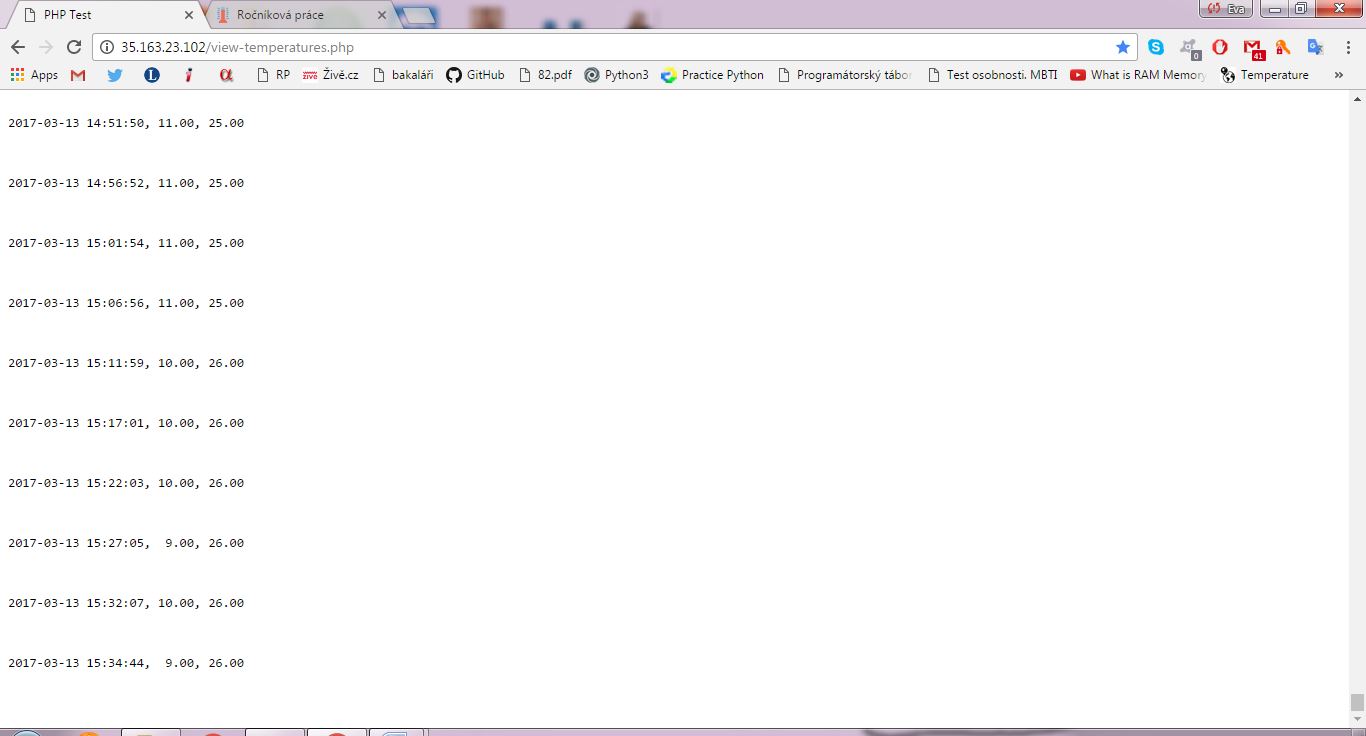




## 5.3 Sběr a vizualizace dat

Hlavní problém této ročníkové práce nebyl až tak získávat data o atmosférických jevech, nýbrž získat také časový údaj o tom, kdy byl tento jev naměřen. Přicházely v úvahu v zásadě dvě možnosti řešení. Ta první, kterou jsem nakonec nezvolila, byl modul reálného času (už komplikovanější součástka připravená k zapojení k Arduinu). Ta funguje tak, že se nejprve nastaví programem přesný čas a poté už modul sám pořád odpočítává čas, dokud mu nedojde už zabudovaná  baterka. Nevýhodou tohoto řešení je jeho krátká výdrž (baterka vydrží jen několik měsíců) a hlavně že to řeší jenom čas a nezálohuje data ve srovnání s druhou variantou. Ta druhá možnost spočívá v použitím internetu. Nejdříve jsem pouze posílala data na server na AWS cloud a odtud na jednoduchou webovou stránku. Stránka tedy sloužila jen jako úložiště naměřených hodnot a kontrola, zda mi stanice funguje. Na analýzu bych data musela exportovat do Excelu a poté s nimi pracovat. Takto jsem původně plánovala mojí ročníkovou práci, ale v diskuzi pod článkem na živě.cz jsem našla službu, která mi velice významně ušetřila práci.

Takto vypadala moje první webová stránka s hodnotami:

****

Data o počasí tedy získávám z senzoru na balkóně a přes Ethernet shield a Arduino jdou data do našeho modemu a odtud se posílají přes knihovnu na službu  [tmep.cz](http://tmep.cz/)  na webovou vizualizaci na stránku <http://dejvice-balkon.tmep.cz/> . Zde jsou velice pěkné statistiky o měření teploty a vlhkosti a z nich vypočítaný rosný bod. Z fyzikálního hlediska je zde  zajímavá možnost přepnout stránku jednak do cizích jazyků, ale hlavně mít hodnoty v jiných méně používaných stupnicích, například Kelvin, Fahrenheit, Reaumur ad. Zdrojový kód se nachází pod tímto odkazem na můj Github: https://github.com/evahoralikova/RP-arduino/blob/master/server/tmep-client.c/tmep-client.c.ino

Nejnovější statistiky:



Nejnovější graf teploty a vlhkosti:



Obsáhlejší graf za poslední tři dny:

## 5.4 Filtrace dat

Čištění dat provádím hlavně z důvodů celkem velkých výkyvů a skoků, které profesionální stanice nevykazují. Z části si to vysvětluji odchylkou senzoru o dva stupně a dále také tím,že po část doby měření byl senzor nezakrytý od slunce a proto ten velký skok.

Na mé doméně www.dejvice-balkon.tmep.cz se sice vytváří grafy a nějaké základní statistiky, ale ze všech naměřených dat, nefiltrují tedy žádné anomální údaje. To provádím prostřednictvím mého programu v jazyce python. Ten bere vždy pět hodnot před a pět hodnot po každé celé hodině. Takto jsem to nastavila hlavně kvůli tomu, aby šlo pěkně porovnat mé měření s měřeními ze stránky meteocentrum.cz (místo měření se nachází několik km od mého a je to nejbližší profesionální stanice). Zde je možné data exportovat do formátu csv a dále s nimi pracovat. Následně je dám do jednoho grafu s mými hodnotami.

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

# 

# 6. Závěr

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

## 6.1 Resumé v AJ

For this year project in Physics I chose to build a amateurish weather station based on Arduino microcontroler. I have made two generations of my station, the first just measured temperature. To make the second one I used a analog sensor of temperature and humidity to measure the atmosphere with this station located on our balcony in Prague Dejvice. In total I have made more than 40000 measurements of temperature in Celsius degrees and humidity in percent of the air. The data gets send to my server and then to my domiain at [www.dejvice-balkon.tmep.cz](http://www.dejvice-balkon.tmep.cz), where simple graphs and analytics are made. The main output of the project is

# 7. Prameny

* Vondráček, Vladimír; Minimum o počasí; nakl. Olympia; Praha 2000
* Československá meteorologická společnost; Meteorologický slovník výkladový a terminologický; Ministerstvo životního prostředí, Praha 1993

# 8. Zdrojové kódy

Zdojové kódy mám zálohované na Githubu, ale až se budu za nějaký čas vracet k mé ročníkové práci, jež jsem napsala v 16 letech a budu se chtít podívat na moje programy, tak třeba už nenajdu ty kódy na serveru a nebo se mi jakýmkoliv způsobem ztratí a neměla bych k nim přístup a to se mi zdá jako škoda.

Na psaní jsem používala Arduino IDE na práci s arduinem nebo Atom IDE a online IDE repl.it na python.

## 8.1 První verze

|  |
| --- |
| #include <SPI.h> |
|  | #include <SD.h> |
|  |  |
|  | File myFile; |
|  |  |
|  |  |
|  | // named constant for the pin the sensor is connected to |
|  | const int sensorPin = A0; |
|  | // room temperature in Celcius |
|  | const float baselineTemp = 20.0; |
|  |  |
|  |  |
|  | void setup() { |
|  | // open a serial connection to display values |
|  | Serial.begin(9600); |
|  | // set the LED pins as outputs |
|  | // the for() loop saves some extra coding |
|  | Serial.print("Initializing SD card..."); |
|  |  |
|  | if (!SD.begin(4)) { |
|  | Serial.println("initialization failed!"); |
|  | return; |
|  | } |
|  | Serial.println("initialization done."); |
|  | myFile = SD.open("teploty.csv", FILE\_WRITE); |
|  |  |
|  | } |
|  |  |
|  | void loop() { |
|  | // read the value on AnalogIn pin 0 |
|  | // and store it in a variable |
|  | int sensorVal = analogRead(sensorPin); |
|  |  |
|  | // send the 10-bit sensor value out the serial port |
|  | Serial.print("sensor Value: "); |
|  | Serial.print(sensorVal); |
|  |  |
|  | // convert the ADC reading to voltage |
|  | float voltage = (sensorVal / 1024.0) \* 5.0; |
|  |  |
|  | // Send the voltage level out the Serial port |
|  | Serial.print(", Volts: "); |
|  | Serial.print(voltage); |
|  |  |
|  | // convert the voltage to temperature in degrees C |
|  | // the sensor changes 10 mV per degree |
|  | // the datasheet says there's a 500 mV offset |
|  | // ((volatge - 500mV) times 100) |
|  | Serial.print(", degrees C: "); |
|  | float temperature = (voltage - .5) \* 100; |
|  | Serial.println(temperature); |
|  |  |
|  |  |
|  | } |

## 8.2 Druhá verze

|  |
| --- |
| #include <SPI.h> |
|  | #include <Ethernet.h> |
|  | #include <DHT.h> |
|  |  |
|  | #define DHTPIN 2 // Data wire is plugged into port 2 on the Arduino |
|  | #define typDHT11 DHT11 // DHT 11 |
|  | //#define typDHT22 DHT22 // DHT 22 (AM2302) |
|  |  |
|  |  |
|  | #define pinDHT 5 |
|  | #define typDHT11 DHT11 // DHT 11 |
|  | DHT mojeDHT(pinDHT, typDHT11); |
|  |  |
|  | // Uncomment whatever type you're using! |
|  | //#define DHTTYPE DHT11 // DHT 11 |
|  | #define DHTTYPE DHT22 // DHT 22 (AM2302), AM2321 |
|  | //#define DHTTYPE DHT21 // DHT 21 (AM2301) |
|  |  |
|  | // initialize DHT sensor. |
|  | DHT dht(DHTPIN, DHTTYPE); |
|  |  |
|  | // assign a MAC address for the ethernet controller. |
|  | // fill in your address here: |
|  | byte mac[] = { |
|  | 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED |
|  | }; |
|  |  |
|  | // initialize the library instance: |
|  | EthernetClient client; |
|  |  |
|  | //char server[] = "tst.tmep.cz"; // domain.tmep.cz |
|  | char server[] = "dejvice-balkon.tmep.cz"; |
|  | char guid[] = "tep"; // guid |
|  |  |
|  | unsigned long lastConnectionTime = 0; // last time you connected to the server, in milliseconds |
|  | const unsigned long postingInterval = 60L \* 1000L; // delay between updates, in milliseconds |
|  | // the "L" is needed to use long type numbers |
|  |  |
|  | void setup() { |
|  | // start serial port: |
|  | Serial.begin(9600); |
|  | while (!Serial) { |
|  | ; // wait for serial port to connect. Needed for native USB port only |
|  | } |
|  |  |
|  | // give the ethernet module time to boot up: |
|  | delay(1000); |
|  | // start the Ethernet connection: |
|  | if (Ethernet.begin(mac) == 0) { |
|  | Serial.println("Failed to configure Ethernet using DHCP"); |
|  | // no point in carrying on, so do nothing forevermore: |
|  | for (;;) |
|  | ; |
|  | } |
|  | printIPAddress(); |
|  | dht.begin(); |
|  | mojeDHT.begin(); |
|  |  |
|  |  |
|  | } |
|  |  |
|  | void loop() { |
|  | float tep = mojeDHT.readTemperature(); |
|  | float vlh = mojeDHT.readHumidity(); |
|  | char teplotaStr[10]; |
|  | if (tep<10){ |
|  | dtostrf(tep,1,0,teplotaStr); |
|  | }else { |
|  | dtostrf(tep,2,0,teplotaStr); |
|  | } |
|  |  |
|  | char vlhkostStr[10]; |
|  | dtostrf(vlh,2,0,vlhkostStr); |
|  | // kontrola, jestli jsou načtené hodnoty čísla pomocí funkce isnan |
|  | Serial.println(teplotaStr); |
|  | Serial.println(vlhkostStr); |
|  | if (isnan(tep) || isnan(vlh)) { |
|  | // při chybném čtení vypiš hlášku |
|  | Serial.println("Chyba při čtení z DHT senzoru!"); |
|  | } else { |
|  | // pokud jsou hodnoty v pořádku, |
|  | // vypiš je po sériové lince |
|  | Serial.print("Teplota: "); |
|  | Serial.print(tep); |
|  | Serial.print(" stupnu Celsia, "); |
|  | Serial.print("vlhkost: "); |
|  | Serial.print(vlh); |
|  | Serial.println(" %"); |
|  | } |
|  |  |
|  | switch (Ethernet.maintain()) |
|  | { |
|  | case 1: |
|  | //renewed fail |
|  | Serial.println("Error: renewed fail"); |
|  | break; |
|  |  |
|  | case 2: |
|  | //renewed success |
|  | Serial.println("Renewed success"); |
|  |  |
|  | //print your local IP address: |
|  | printIPAddress(); |
|  | break; |
|  |  |
|  | case 3: |
|  | //rebind fail |
|  | Serial.println("Error: rebind fail"); |
|  | break; |
|  |  |
|  | case 4: |
|  | //rebind success |
|  | Serial.println("Rebind success"); |
|  |  |
|  | //print your local IP address: |
|  | printIPAddress(); |
|  | break; |
|  |  |
|  | default: |
|  | //nothing happened |
|  | break; |
|  | } |
|  |  |
|  |  |
|  |  |
|  | // if there's incoming data from the net connection. |
|  | // send it out the serial port. This is for debugging |
|  | // purposes only: |
|  | if (client.available()) { |
|  | char c = client.read(); |
|  | Serial.write(c); |
|  | } |
|  |  |
|  |  |
|  | // if ten seconds have passed since your last connection, |
|  | // then connect again and send data: |
|  | if (millis() - lastConnectionTime > postingInterval) { |
|  | httpRequest(teplotaStr,vlhkostStr); |
|  | } |
|  | for(int i=0;i<90 |
|  | ;i++){ |
|  | delay(1000); |
|  | } |
|  | } |
|  |  |
|  | // this method makes a HTTP connection to the server: |
|  | void httpRequest(char\* teplotaStr, char\* vlhkostStr) { |
|  | // close any connection before send a new request. |
|  | // This will free the socket on the WiFi shield |
|  | client.stop(); |
|  |  |
|  | // Read temperature |
|  | /\* float h = dht.readHumidity(); |
|  | // Read temperature as Celsius (the default) |
|  | float t = dht.readTemperature(); |
|  |  |
|  | // Check if any reads failed and exit early (to try again). |
|  | if (isnan(h) || isnan(t)) { |
|  | Serial.println("Failed to read from DHT sensor!"); |
|  | return; |
|  | } |
|  | Serial.print("Temperature: "); |
|  | Serial.print(t); |
|  | Serial.print("\t Humidity: "); |
|  | Serial.println(h); \*/ |
|  |  |
|  | // if there's a successful connection: |
|  | if (client.connect(server, 80)) { |
|  | Serial.print("connecting..."); |
|  | // send the HTTP GET request: |
|  | client.print("GET /index.php?"); |
|  | client.print(guid); |
|  | client.print("="); |
|  | //client.print(t); |
|  | client.print(teplotaStr); |
|  | client.print("&humV="); |
|  | //client.print(h); |
|  | client.print(vlhkostStr); |
|  | client.println(" HTTP/1.1"); |
|  | client.print("Host: "); |
|  | client.println(server); |
|  | client.println("User-Agent: arduino-ethernet"); |
|  | client.println("Connection: close"); |
|  | client.println(); |
|  |  |
|  | Serial.println(" done."); |
|  | // note the time that the connection was made: |
|  | lastConnectionTime = millis(); |
|  | } else { |
|  | // if you couldn't make a connection: |
|  | Serial.println(" connection failed"); |
|  | } |
|  | } |
|  |  |
|  | void printIPAddress() |
|  | { |
|  | Serial.print("My IP address: "); |
|  | for (byte thisByte = 0; thisByte < 4; thisByte++) { |
|  | // print the value of each byte of the IP address: |
|  | Serial.print(Ethernet.localIP()[thisByte], DEC); |
|  | Serial.print("."); |
|  | } |
|  |  |
|  | Serial.println(); |
|  | } |

## 8.3 Filtrace dat

|  |
| --- |
| vstupTeploty=[] |
|  | file = open("vstup.csv", "r") |
|  | file.readline() |
|  | for line in file: |
|  | print(line) |
|  | novaTeplota=line.split(",")[1] |
|  | novaTeplotaInt=int(novaTeplota) |
|  | novyCas=line.split(",")[0] |
|  |  |
|  | novaDvojice=[] |
|  | novaDvojice.append(novyCas) |
|  | novaDvojice.append(novaTeplotaInt) |
|  |  |
|  | vstupTeploty.append(novaDvojice) |
|  |  |
|  | #print(vstupTeploty) |
|  |  |
|  | # |
|  | # 2017-04-08 08:48:48,15,23 |
|  |  |
|  | #vstupTeploty=[['2017-04-08 13:35:45',14],['2017-04-08 13:56:34',25],['2017-04-08 13:58:23',17], |
|  | # ['2017-04-08 13:59:12',14],['2017-04-08 14:00:56',15], |
|  | # ['2017-04-08 14:02:09',14],['2017-04-08 14:07:45',15],['2017-04-08 14:35:45',14], |
|  | # ['2017-04-08 14:56:34',25],['2017-04-08 14:58:03',17], |
|  | # ['2017-04-08 14:58:12',14],['2017-04-08 15:01:56',12],['2017-04-08 15:02:09',14],['2017-04-08 15:05:45',11]] |
|  | for hodina in range(0,24): |
|  | filtrovane=[] |
|  | for aktualni in vstupTeploty: |
|  | cas=aktualni[0] |
|  | aktualniHodina=cas[11:13] |
|  | minuty=cas[14:16] |
|  | if (hodina==int(aktualniHodina)+1 and int(minuty) > 55) or (hodina==int(aktualniHodina) and int(minuty) < 5): |
|  | filtrovane.append(aktualni) |
|  |  |
|  | print(filtrovane) |
|  | pocet=len(filtrovane) |
|  | soucet=0 |
|  | prumer=0 |
|  | for aktualni in filtrovane: |
|  | teplota=aktualni[1] |
|  | soucet=soucet+teplota |
|  | prumer=soucet/float(pocet) |
|  | print ("hodina: {} prumer: {}".format(hodina, prumer)) |