|  |  |  |
| --- | --- | --- |
| Logo__SSPU_2016_Barva | | |
| **Závěrečná studijní práce**  **dokumentace** | | |
| **WiFi Meteo stanice** | | |
| Lukáš Fojtík | | |
|  | | |
|  | |  |
| **Obor:** | 18-20-M/01 INFORMAČNÍ TECHNOLOGIE  se zaměřením na počítačové sítě a programování | |
| **Třída:**  **Školní rok:** | IT4  2016/2017 | |

Poděkování

*Velmi rád bych poděkoval všem, kteří mi pomohli s vytvořením tohoto projektu, zejména pak panu učiteli Ing. Petru Grussmannovi, který mi pomohl, nejen s výběrem projektu, ale i s jeho realizací. Dále pak děkuji panu učiteli Mgr. Marku Lučnému, za pomoc při psaní dokumentace, Mgr. Marcelovi Godovskému a samozřejmě i spolužákům a rodině za jejich věcné připomínky.*

Prohlašuji, že jsem závěrečnou práci vypracoval samostatně a uvedl veškeré použité   
informační zdroje.

Souhlasím, aby tato studijní práce byla použita k výukovým účelům na Střední průmyslové   
a umělecké škole v Opavě, Praskova 399/8.

V Opavě 31. 12. 2016

*podpis autora práce*

**ANOTACE**

Cílem projektu bylo vytvoření meteo stanice využívající WiFi přístupu k internetu, kterou může podle jednoduchého návodu nakonfigurovat i běžný uživatel. Zařízenívyužívající mikročip ESP8266-201 a senzor DHT22 umožňuje měření teploty a vlhkosti vzduchu. Získané údaje jsou zobrazeny na displeji současně s aktuálním datem a časem. Prostřednictvím WiFi jsou rovněž posílány na webový server thingspeak.com a mohou být v přehledné podobě sledovány na webové stránce.

OBSAH

[Úvod 4](#_Toc470888293)

[1 Využité technologie 5](#_Toc470888294)

[1.1 Mikrokontrolér 5](#_Toc470888295)

[1.2 Použitý software 6](#_Toc470888296)

[1.2.1 Arduino IDE 6](#_Toc470888297)

[1.2.2 Fritzing 7](#_Toc470888298)

[2 Způsoby řešení a použité postupy 8](#_Toc470888299)

[2.1 Seznam součástek 8](#_Toc470888300)

[2.2 Schématické zapojení 8](#_Toc470888301)

[2.3 DHT22 9](#_Toc470888302)

[2.4 Webový server a webový klient 9](#_Toc470888303)

[2.5 Zhasínání a zapínání displeje 9](#_Toc470888304)

[2.6 Přerušení procesoru 9](#_Toc470888305)

[2.7 Displej 9](#_Toc470888306)

[2.8 Tvorba vlastních znaků 10](#_Toc470888307)

[2.9 Thingspeak 10](#_Toc470888308)

[2.9.1 Konfigurace kanálu 10](#_Toc470888309)

[2.9.2 Posílání dat na Thingspeak 10](#_Toc470888310)

[2.10 NTP reálný čas a datum 10](#_Toc470888311)

[2.11 Ukázka kódu 11](#_Toc470888312)

[3 Provoz meteo stanice 12](#_Toc470888313)

[3.1 Připojení k WiFi 12](#_Toc470888314)

[3.2 Po připojení k WiFi 13](#_Toc470888315)

[3.3 Rozhraní s aktuální teplotou 13](#_Toc470888316)

[3.4 Displej 14](#_Toc470888317)

[3.5 Vytvořené grafy 14](#_Toc470888318)

[Závěr 15](#_Toc470888319)

[Seznam použitýCH INFORMAČNÍCH ZDROJů 16](#_Toc470888320)

[Seznam příloh 17](#_Toc470888321)

Úvod

Cílem projektu bylo vytvoření WiFi meteo stanice určené k sledování teploty a vlhkosti vzduchu. Součástí jeho realizace mělo být funkční zapojení a naprogramování mikrokontroléru, vytvoření uživatelského přístupu k datům teploty a vlhkosti pomocí jednoduchého webového rozhraní, zprovoznění komunikace se serverem thingspeak.com včetně správné konfigurace. Mým dalším záměrem bylo také zprovoznění LCD displeje tak, aby zobrazoval údaje o teplotě i vlhkosti vzduchu společně s aktuálním časem a datem.

Na začátku projektu stálo velmi důležité rozhodnutí, které technologie budou použity. Nejdříve bylo nutné si uvědomit, jak bude celé zařízení pracovat a jaké technologické nároky na něj budou kladeny. Po uvážení jsem se rozhodl pro mikrokontrolér ESP8266-201, protože nejlépe odpovídal mým požadavkům. Při volbě displeje jsem zase musel zohlednit charakter a množství posílaných údajů; nakonec jsem použil dvouřádkový a šestnáctiznakový displej se sběrnicí I2C.

Dalším krokem bylo navržení zapojení všech součástek na nepájivém poli. Problém byl s různým napájením mikrokontroléru a LCD displeje, který jsem vyřešil zakoupením napájecího modulu nepájivého pole. Pro vyzkoušení funkčnosti zapojení jsem použil program ESPEasy, který je díky svému jednoduchému nahrávání a uživatelsky přívětivému webovému rozhraní výborným nástrojem pro ověření zapojení.

Pro tvorbu aplikace jsem se snažil zvolit takový programovací jazyk, který by umožňoval co nejjednodušší prací s mikrokontroléry a který by odpovídal úrovni mých programátorských znalostí. Po zohlednění všech požadavků jsem zvolil jazyk wiring, který je používán zejména k programování mikrokontrolérů Arduino a navazuje na jazyk C/C++, se kterým jsem již pracoval.

Po dokončení programovaní a vyladění chyb, na které jsem narazil, jsem meteo stanici nainstaloval v pokoji a nechal jsem ji několik dní spuštěnou, abych si ověřil funkčnost při dlouhodobém používání.

První kapitolu své dokumentace věnuji použitým technologiím, následně popisuji svůj postup při řešení problému a v závěrečné kapitole se stručně zmiňuji o tom, jak hotová meteo stanice i navazující webová aplikace fungují.

# Využité technologie

## Mikrokontrolér

Jádrem projektu je ESP8266 Wireless WiFi Module Develop Board, která obsahuje mikrokontrolér ESP201. K tomuto mikrokontroléru není třeba programátor a díky develop boardu stačí pouze mini USB kabel jak pro napájení, tak i pro přenos dat.

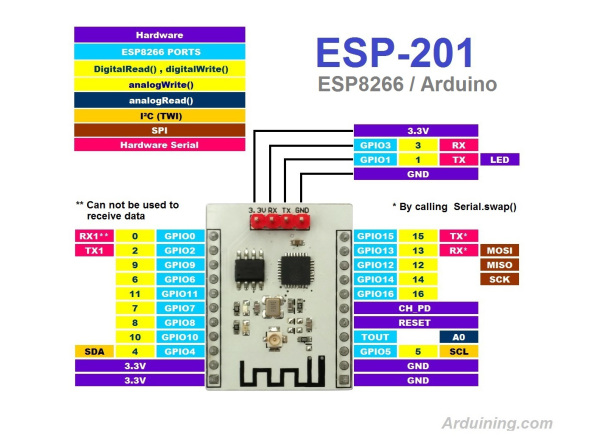
Mikrokontrolér ESP201 má 26 pinů, z pro programování lze použít 18 pinů.

Parametry ESP201:

* 3,3 V operační napětí
* 512 KB (4 Mb) flash paměti
* 18 využitelných portů
* WiFi b/g/n

Na Develop Boardu dále najdeme:

* RGB diodu
* Bzučák
* Relé
* Dvě tlačítka



Obr. 2.1 Pinout mikrokontroléru

## Použitý software

### Arduino IDE

Open-source vývojové prostředí firmy Arduino je určené sice hlavně pro mikrokontroléry Adruino, ale po přizpůsobení se dá použít i pro programování ESP mikrokontrolérů. Program používá jazyk Wiring, který je odnoží jazyka C/C++. Arduino IDE obsahuje mnoho funkcí, které ulehčují programování. Mezi nejpoužívanější patří nahrávání do paměti, kdy nemusíte řešit žádné programátory, stačí pouze nastavit typ mikrokontroléru a správný port. Další užitečná funkce je sériová linka, která může být použita k debugování programu.



Obr. 2.2.1 Arduino IDE

### Fritzing

Fritzing je open-source program pro tvorbu schémat zapojení. Obsahuje velké množství elektronických komponentů, se kterými můžeme pracovat. Pokud v programu nějaká součástka není, je možné ji do knihovny programu přidat. Součástí je také navrhování fyzického rozmístění komponentů na plošném spoji. Výsledné schéma můžeme uložit jako obrázek nebo jako šablonu pro vyleptání plošného spoje.



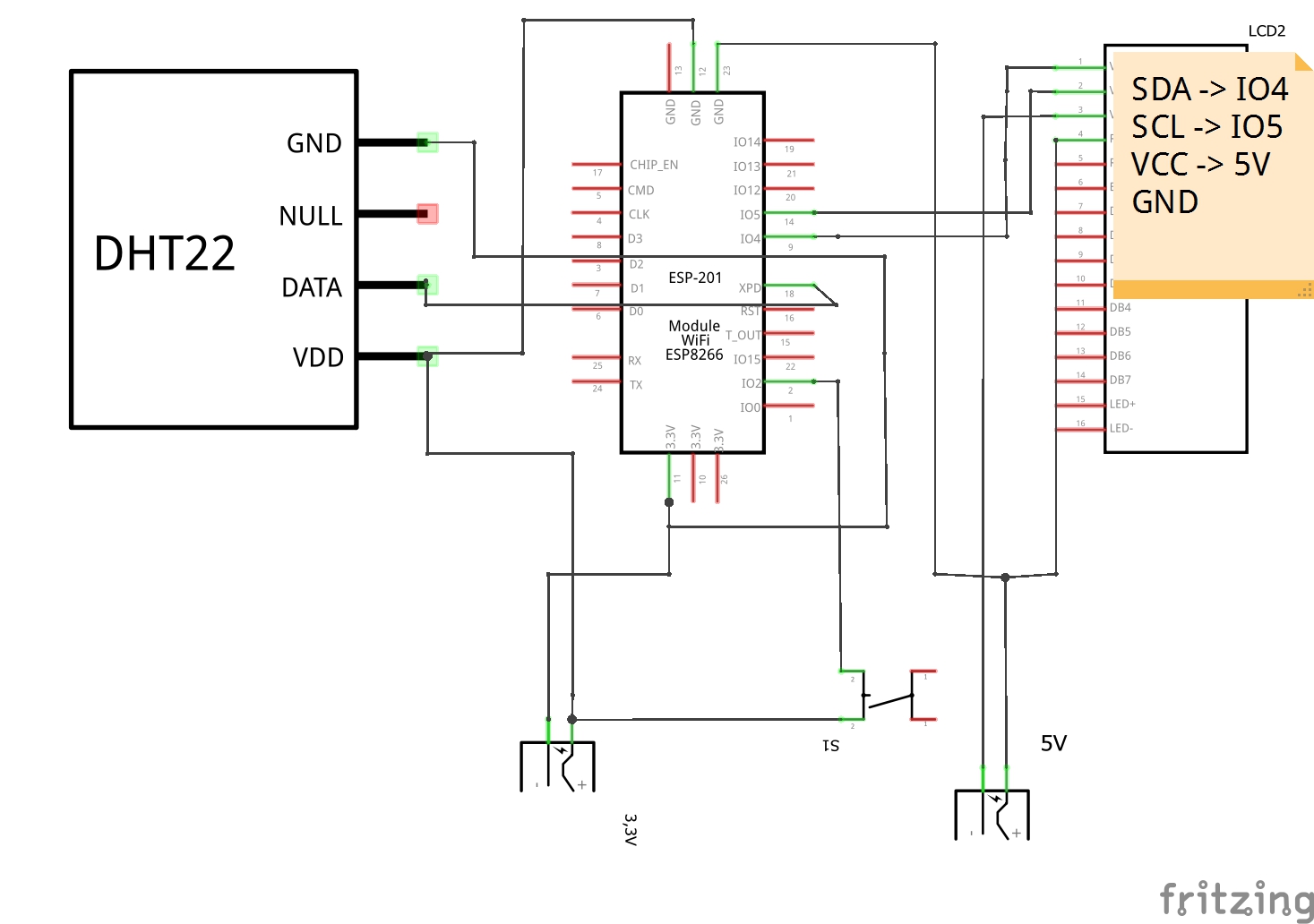
Obr. 2.2.2 Fritzing

# Způsoby řešení a použité postupy

## Seznam součástek

* Mikrokontrolér ESP-201
* SDK Develop Board
* 16x2 I2C LCD display
* Napájecí modul nepájivého pole
* Nepájivé pole
* Senzor DHT22
* 11 V DC zdroj

## Schématické zapojení



Obr. 3.2 Schéma zapojení

## DHT22

DHT22 je senzor sloužící k získávání údajů o teplotě a vlhkosti. Oproti své levnější verzi DHT11 má vyšší přesnost a větší rozsah měřených hodnot, zatímco rozdíl cen není tak velký. To jsou hlavní důvody, proč jsem se rozhodl vybrat tento senzor.

## Webový server a webový klient

Webová stránka je jednou z částí mého projektu. Použil jsem jednoduchý webserver, kdy je do proměnné typu string uložena celá webová stránka a ta je poté serverem posílána klientovi, který ji pomocí webového prohlížeče zobrazí. Během spouštění aplikace musí dojít ke spuštění webserveru. Poté už pouze dochází k obsluze klienta ve dvouvteřinových intervalech.

## Zhasínání a zapínání displeje

Zapínání a vypínání podsvícení displeje je zajištěno pomocí tlačítka, které funguje pod přerušením. Kdykoliv dojde k přerušení stisknutím tlačítka, procesor zastaví činnost, provede obsluhu přerušení, kdy zapne nebo vypne podsvícení. Po ukončení obsluhy přerušení se opět vrátí do hlavní smyčky a pokračuje v činnosti.

## Přerušení procesoru

Pro správnou činnost přerušení musíme nastavit tlačítka do módu pro vstup. Poté stačí vyvoláním funkce nastavit, že stisk tlačítka vyvolá přerušení procesoru. Také musíme nastavit, kdy má program reagovat na stisk a kterou námi předem definovanou funkci má vyvolat. Funkce, která slouží jako obsluha přerušení, má datový typ void a měla by být definována před hlavní funkcí setup.

## Displej

Pro vypisování na displej je potřeba nejdříve nastavit několik parametrů. První krokem je volba správné knihovny. Poté musí dojít k vytvoření objektu, kterému je nutné nastavit správnou adresu I2C sběrnice, a tu lze zjistit I2C skenerem. Po dosazení správné adresy je potřeba ještě definovat počet znaků v řádku a počet řádků. Poté je displej připraven k práci.

## Tvorba vlastních znaků

V mém projektu pro zpřehlednění textu na displeji používám ikony teploměru a kapky vody pro symbolizaci teploty a vlhkosti. Pro vytvoření těchto ikon je potřeba definovat, které bity jednotlivých znakových polí budou výrazné, a které ne. Tato definice probíhá pomocí funkce s návratovým typem byte. Takto připravené funkce se následně převedou do podoby LCD znaků a mohou být použity na displeji.

## Thingspeak

ThingSpeak je IoT analytická platforma, která umožňuje sbírat, vizualizovat a analyzovat data uložená v cloudu. Jeho základní činností je samotné sbírání dat a okamžité vykreslování grafu.

### Konfigurace kanálu

Práce s platformou Thingspeak je jednoduchá. Po vytvoření uživatelského účtu stačí vytvořit nový kanál a přidat jednotlivá pole teploty a vlhkosti, kde každé pole musí mít stejné jméno jako proměnná, kterou do ní budeme posílat. Thingspeak nabízí další možnosti specifikace sbíraných dat, jako jsou název kanálu, jeho popis, tagy, lokaci a mnoho dalšího.

### Posílání dat na Thingspeak

Data jsou na Thingspeak posílána pomocí URL. Jako identifikátor slouží jedinečný write API key. Za něj jsou vloženy jednotlivé údaje o teplotě a vlhkosti. Data musí mít stejné pořadí, jako mají na webu Thingspeak.

## NTP reálný čas a datum

NTP (network time protokol) je protokol, získávající čas a datum z internetových NTP serverů. Jeho použití je jednoduché. Při volbě správné knihovny se pouze vytvoří objekt s parametrem, kterým je časové pásmo, a poté se jen vyvolává funkce pro získání aktuálních dat. Pro tvar data DD.MM.RRRR, bylo potřeba ve třídě knihovny upravit pořadí dat.

## Ukázka kódu



Obr. 2.11 Ukázka kódu ukládání konfigurace WiFi a parametrů

Na obrázku 2.11 je zachycena část celého kódu, která se stará o ukládání informací o WiFi a zadaných parametrů, jako je adresa serveru a port – v mém případě write API key. Ostatní části kódu se starají o výpisy na sériovou linku a displej, čtení ze senzoru, ukládání do webové stránky a aktualizaci času a datumu.

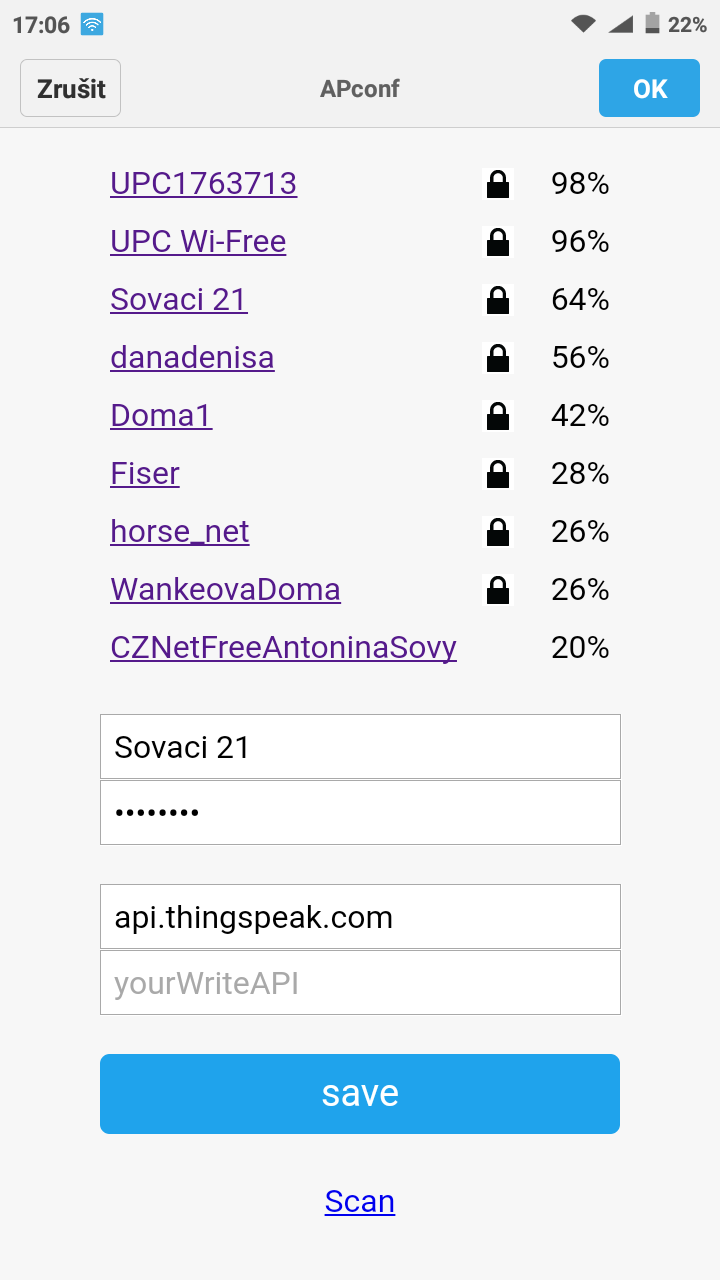
# Provoz meteo stanice

## Připojení k WiFi

Pokud se aplikaci nepodaří připojit na poslední uloženou WiFi, nebo žádná uložená není, změní se mikrokontrolér v přístupový bod, přes který se nastaví potřebné parametry jako jsou jméno a heslo k WiFi, server a write API key.



Obr 3.1 a) Informace o přístupovém bodu na displeji



Obr. 3.1 b) Uživatelské rozhraní pro nastavení připojení

## Po připojení k WiFi

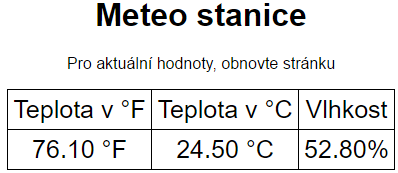
Po kliknutí na požadovanou WiFi síť dojde k uložení údajů o SSID, hesle a write API key. Na displeji se zobrazí informace o úspěšném připojení k WiFi a IP adresa, přes kterou se následně dá připojit na webové rozhraní meteo stanice pro zobrazení údajů ze senzoru.



Obr. 3.2 IP adresa stanice a informace o úspěšném připojení

## Rozhraní s aktuální teplotou

Součástí aplikace je také jednoduché webové rozhraní s aktuálními údaji o teplotě a vlhkosti. Přistup na webové rozhraní je přes webový prohlížeč, kdy stačí zadat IP adresu stanice.



Obr. 4.3 Ukázka webového rozhraní s aktuálními údaji

## Displej

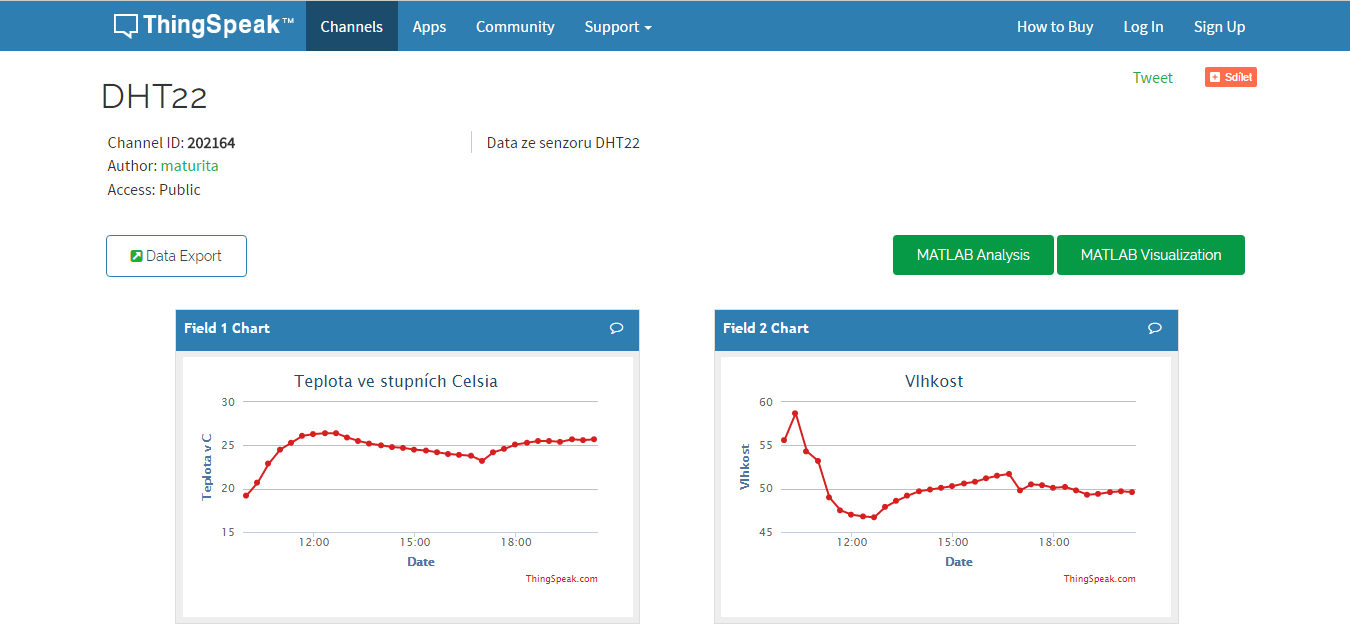
Po dokončení připojení k WiFi se na displeji zobrazí aktuální čas a datum společně s teplotou a vlhkostí. Pomocí dvou tlačítek lze libovolně vypínat a zapínat podsvícení displeje.



Obr. 3.4 Zobrazení teploty, vlhkosti, času a datumu na displeji

## Vytvořené grafy

Na obrázku můžete vidět, jak vypadá výsledný graf sesbíraných hodnot ze senzoru. Data se vykreslují automaticky a není třeba žádného dalšího zásahu. Jednotlivé grafy je možné pomocí HTML kódu vkládat na své vlastní webové stránky. Dále si můžeme nastavit popisek jednotlivých os, jejich měřítko, popisek, typ grafu a mnoho dalšího. Grafy hodnot z mé stanice jako příklad najdete na [této adrese](https://thingspeak.com/channels/202164).



Obr. 3.5 Pohled na webové rozhraní ThingSpeak

# Závěr

Cílem tohoto projektu bylo vytvoření funkční meteo stanice využívající WiFi. Požadavkem bylo rovněž uživatelsky co nejpřívětivější webové rozhraní pro konfiguraci připojení k WiFi. Mezi klíčové součásti také patří zobrazování dat získaných ze senzoru na displeji, na jednoduché webové stránce a následné vytvoření grafů změny teploty a vlhkosti v čase.

Do budoucna by bylo možné použít vlastní MQTT server, který by shromažďoval data a následně vytvářel grafy. Další možností by bylo využít SPIFF i pro webovou stránku, což by umožňovalo jednodušší využití kaskádových stylů nebo javascriptových efektů připojením externího souboru, popř. využití frameworku Bootstrap.

Další možnou úpravou by bylo vytvořit vhodnou krabičku, která by chránila stanici proti mechanickému poškození a umožňovala by i venkovní použití.

Tuto meteo stanici jsem vytvořil pro své vlastní účely a používám ji doma.

Ceny podobných meteo stanic se pohybují v řádech tisíců korun, zatímco náklady na výrobu mé stanice byly zlomkem této ceny. V tomto ohledu můj projekt vyčnívá nad konkurencí.

Seznam použitýCH INFORMAČNÍCH ZDROJů

[] TZAPU. Github. tzapu/WiFiManager. [online]. 1.1.2016 [cit. 2016-12-25]. Dostupné z: https://github.com/tzapu/WiFiManager

[2] IoT with ESP8266. . [online]. 4.3.2015 [cit. 2016-12-25]. Dostupné z: <https://myesp8266.blogspot.cz/2015/03/other-interesting-board-is-this-board.html>

[3] ARDUNAUT. Arduining. ESP-201 an ESP8266 breadboard friend. [online]. 31.12.2015 [cit. 2016-12-25]. Dostupné z: https://arduining.com/2015/12/31/esp-201-an-esp8266-breadboard-friend/

[4] arduino-info [online]. [cit. 2016-12-25]. Dostupné z: https://arduino-info.wikispaces.com/LCD-Blue-I2C

Seznam příloh

č. 1 Aplikace WiFi meteo stanice

č. 2 Schéma zapojení

č. 3 Fotodokumentace stanice