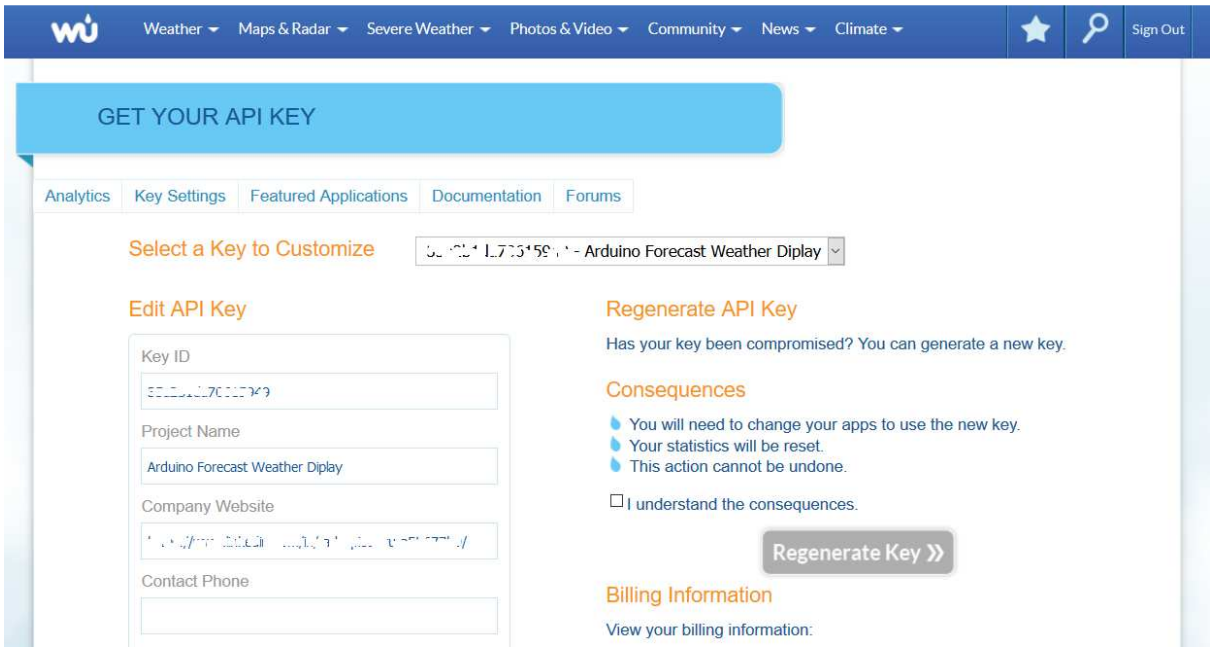
Cum putem utiliza prognoza meteo oferită de Weather Underground

Așa cum am arătat și în lecția precedentă serviciul Weather Undergroud (1) este un serviciu independent (nu este un serviciu al unei țări sau strucutură oficială) de predicție a vremii. Acest serviciu Internet permite realizarea de prognoze locale pe baza informațiilor provenite de la o stații meteo personale (PWS – Personal Wether Station) sau corelând informațiile de la mai multe stații meteo personale aflate într-o anumită zonă.

Dacă în lecția precedetă am arătat cum să construim o mini stație meteo care să raporteze date către Weather Underground în lecția de față vom arăta cum putem utiliza datele de prognoză calcultate de Weather Underground. Pentru acest lucru este necesar să solicităm o cheie API (API Key) de conectare de la adresa (2):

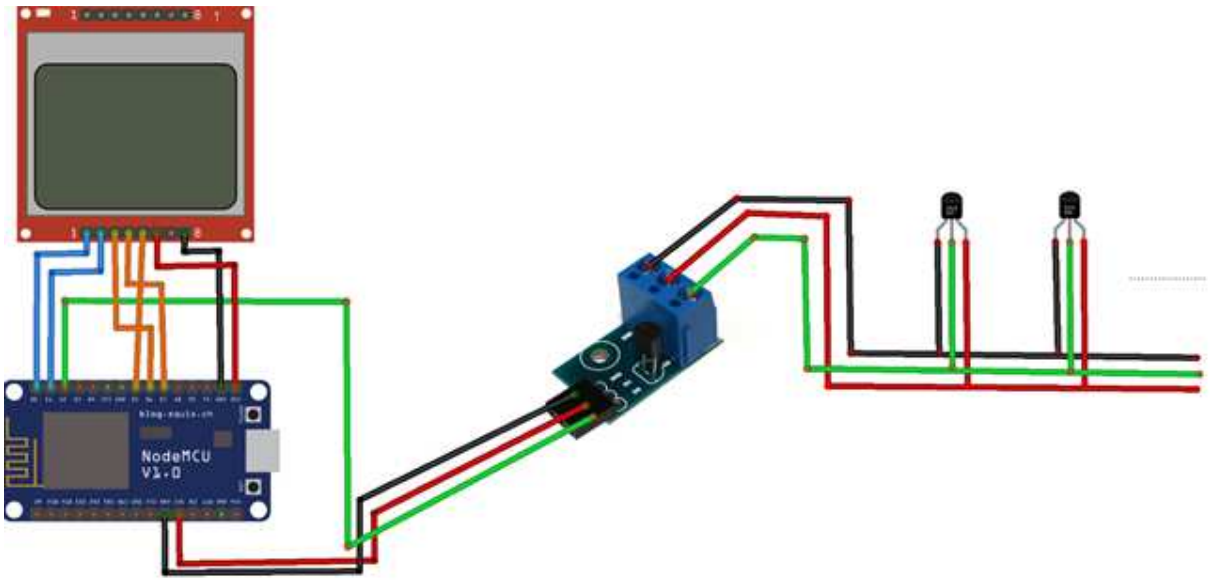


Furnizarea prognozei meteo este un serviciu comercial (se plătește) dar pentru dezvoltatori (developers) există un plan tarifar gratuit (Stratus Plan) ce permite efectuarea de 500 de interogări pe zi (maxim 10 interogări pe minut) – limitări ce nu afectează sistemul prezentat în această lecție.

How much will you use our service?			
	Monthly Pricing	Calls Per Day	Calls Per Minute
(•) Developer	$0	500	10
( ) Drizzle	$20	5000	100
( ) Shower	$200	100,000	1000
( ) Downpour	Get in touch for more than 100,000 calls per day.		

Your Selected Plan: Stratus Developer				Purchase Key »
Monthly Pricing**	Calls Per Day	Calls Per Minute	+ History	TOTAL
$0	500	10	Not Included	$0 USD per month

Odată obținută cheia de conectare la platforma Weather Underground putem scrie aplicații proprii care să prezinte prognoza meteo fără a fi necesară accesarea site-ului Weather Underground (1) sau utilizarea aplicației mobile Weather Underground (3). Mai multe informații se pot consulta în pagina de suport a platformei (4) unde sunt prezentate exemple pentru mai multe limbaje de programare. Funcțiile API puse la dispoziție de platforma Weatgher Underground se pot accesa și de pe un sistem embedded simplu. Pentru exemplificarea acestui lucru vom utiliza sistemul descris în lecția „Ceas IoT cu programare OTA” format dintr-o placă NodeMCU (5) și un ecran LCD grafic 84x48 pixeli (6):



Sistemul va păstra funcționalitatea de ceas și senzor de temperatură IoT dar va afișa și prognoza meteo pentru ziua curentă și ziua următoare.

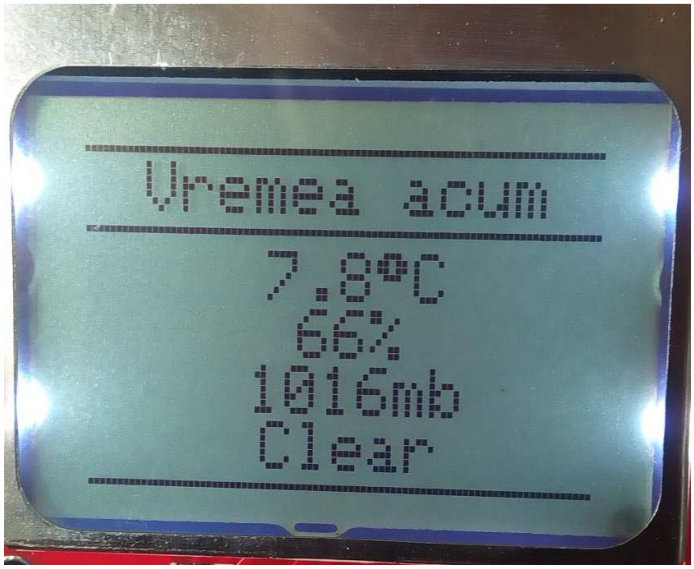
Vom pleca de la programul scris pentru sistem Ceas IoT. Programul a fost dezvoltat și testat utilizând Arduino IDE 1.8.3 având instalată extensia ESP8266 Community 2.3.0. Pentru accesarea funcțiilor API Weather Underground vom utiliza bibliotecile Json Streaming Parser 1.0.5 (7) și ESP8266 Weather Station 1.3.2 (8). Programul Ceas IoT se va completa cu următoarele declarații inițiale (este necesară completarea cheii API – constanta *WUNDERGROUND_API_KEY* și, dacă este cazul, modificarea localității pentru care se dorește obținerea prognozei meteo – constantele *WUNDERGROUND_ZMW_CODE* și *WUNDERGR_UND_CITY*):

```
#include <JsonListener.h>
#include "WundergroundConditions.h"
#include "WundergroundForecast.h"
const String  WUNDERGRROUND_API_KEY = "…";
const boolean IS_METRIC = true;
const boolean USE_PM = false;
const String WUNDERGROUND_ZMW_CODE = "00000.26.WLRBS";
const String  WUNDERGRROUND_LANGUAGE = "EN";
const String  WUNDERGR_UND_STATE_OR_COUNTRY = "RO";
const String  WUNDERGR_UND_CITY = "Bucharest";
WundergroundConditions wunderground(IS_METRIC);
WundergroundForecast wundergroundf(IS_METRIC);
WGConditions conditions;
WGForecast forecasts[3];
```

La sfârșitul secțiunii *setup()* se vor adăuga următoarele două instrucțiuni:

```
wunderground.updateConditions(&conditions,
WUNDERGRROUND_API_KEY, WUNDERGRROUND_LANGUAGE,
WUNDERGROUND_ZMW_CODE);
wundergroundf.updateForecast(forecasts, 3,
WUNDERGRROUND_API_KEY, WUNDERGRROUND_LANGUAGE,
WUNDERGR_UND_STATE_OR_COUNTRY, WUNDERGR_UND_CITY);
```

În cadrul secțiunii *loop()* se va modifica partea de afișare astfel încât să avem patru stări de afișare (patru ecrane diferite) care se vor succeda la un interval de 15 secunde (fiecare ecran va rămâne afișat timp de 15 secunde): primul ecran va afișa ora și data (la fel ca și în lecția precedentă), al doilea ecran va afișa temperatura interioară și statusul conexiunii IoT (la fel ca și în lecția precedentă), al treilea ecran va afișa prognoza pentru ziua curentă iar al patrulea pentru ziua următoare.



Din păcate, chiar dacă funcțiile API Weather Underground au implementată traducerea informațiilor în limba română, vom utiliza în program prognoza în limba engleză deoarece setul de caracter a ecranului LCD nu include și diacritice. În capturile anterioare se poate observa că pentru ziua curentă avem cer senin (Clear) și pentru prognoza zilei următoare se preconizează cer noros și câteva picături de ploaie.

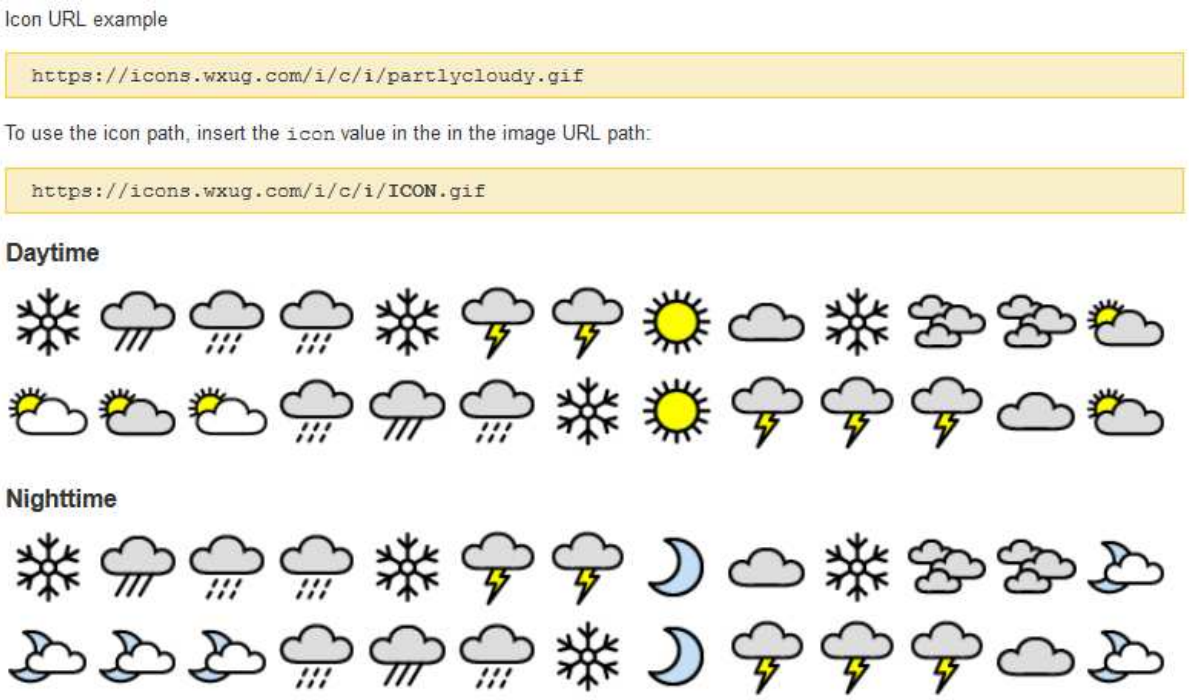
```
if (second()<15) {
      // codul rămâne neschimbat față de lecția anterioară
    }
else if (second()<30) {
      // codul rămâne neschimbat față de lecția anterioară
    }
else if (second()<45) {
      display.drawLine(0,0,83,0,BLACK);
      display.drawLine(0,47,83,47,BLACK);
      display.setTextColor(BLACK);
      display.setTextSize(1);
      display.setCursor(10,2);
      display.print("Vremea acum");
      display.drawLine(0,11,83,11,BLACK);
      display.setCursor(30,14);
      display.print(conditions.currentTemp);
      display.print((char)247);
      display.print("C");
      display.setCursor(35,22);
      display.print(conditions.humidity);
      display.setCursor(27,30);
      display.print(conditions.pressure);
      display.setCursor((84-
        (5*conditions.weatherText.length()))/2,38);
      display.print(conditions.weatherText);
      display.display();
    }
    else {
      display.fillScreen(BLACK);
      display.drawLine(0,1,83,1,WHITE);
      display.drawLine(0,46,83,46,WHITE);
      display.setTextColor(WHITE);
```

```
      display.setTextSize(1);
      display.setCursor(5,3);
      display.print("Vremea maine");
      display.drawLine(0,12,83,12,WHITE);
      display.setCursor(20,14);
      display.print("Min:");
      display.print(forecasts[2].forecastLowTemp);
      display.print((char)247);
      display.print("C");
      display.setCursor(20,22);
      display.print("Max:");
      display.print(forecasts[2].forecastHighTemp);
      display.print((char)247);
      display.print("C");
      String temp = forecasts[2].forecastText;
      int prim = temp.indexOf('.');
      display.setCursor((84-(5*prim))/2,30);
      display.print(temp.substring(0,prim));
      display.setCursor(0,38);
     display.print(temp.substring(prim+2,
                 temp.indexOf('.',prim+2)));
      display.display()     }

if (millis() - lastConnectionTime > postingInterval) {
      wunderground.updateConditions(&conditions,
        WUNDERGRROUND_API_KEY, WUNDERGRROUND_LANGUAGE,
        WUNDERGROUND_ZMW_CODE);
      wundergroundf.updateForecast(forecasts, 3,
        WUNDERGRROUND_API_KEY, WUNDERGRROUND_LANGUAGE,
        WUNDERGR_UND_STATE_OR_COUNTRY,
        WUNDERGR_UND_CITY);
      // codul rămâne neschimbat față de lecția anterioară
    }
```

Prognoza meteo oferită de Weather Underground are o acoperire de zece zile. În cadrul sistemului prezentat nu am utilizat decât ziua curentă și ziua următoare dar programul poate fi ușor modificat pentru afișarea unei prognoze pe mai multe zile. În plus există mai multe facilități ce nu au fost utilizate în exemplul nostru dar pot constitui dezvoltări ulterioare interesante: avertizări de vreme extremă (a se vedea exemplul *WundergroundAlertsDemo* din cadrul bibliotecii ESP8266 Weather Station) sau elemente de evoluție astronomică (a se vedea exemplul *WundergroundAstronomyDemo*).

O altă posibilă direcție de îmbunătățire posibilă a sistemului de predicție meteo este utilizarea simbolurilor grafice oferite de Weather Underground pentru o reprezentare sugestivă a evoluției meteo.



Un exemplu de proiect ce utilizează aceste simboluri este „ESP8266 Weather Widget” (9), proiect ce utilizează un ecran grafic OLED de rezoluție mai mare.

Referințe on-line

(1) Weather Forecast & Reports - Long Range & Local | Weather Underground
https://www.wunderground.com/

(2) API | Weather Underground
https://www.wunderground.com/api/

(3) Google Play – Weather Underground
https://play.google.com/store/apps/details?id=com.wunderground.android.weather

(4) Weather API: Introduction
https://www.wunderground.com/weather/api/d/docs

(5) NodeMCU v2
https://www.robofun.ro/nodemcu-v2-lua-based-esp8266-development-kit?utm_source=newsletter&utm_medium=email&utm_content=productLink&utm_campaign=CURS_EMAIL

(6) Graphic LCD 84x48 - Nokia 5110
https://www.robofun.ro/graphic-lcd-84x48-nokia-5110?utm_source=newsletter&utm_medium=email&utm_content=productLink&utm_campaign=CURS_EMAIL

(7) squix78/json-streaming-parser
https://github.com/squix78/json-streaming-parser

(8) squix78/esp8266-weather-station
https://github.com/squix78/esp8266-weather-station

(9) ESP8266 Weather Widget
http://www.instructables.com/id/ESP8266-Weather-Widget/