

VAIA



- È stato un evento meteorologico estremo che ha interessato il nord-est italiano (in particolare l'area delle Dolomiti e delle Prealpi Venete) dal 26 al 30 ottobre 2018. L'evento si è originato a seguito di una perturbazione di origine atlantica.
- I venti hanno raggiunto le velocità "uragano" (grado 12), che comunemente si originano solo su acque tropicali o subtropicali del pianeta.

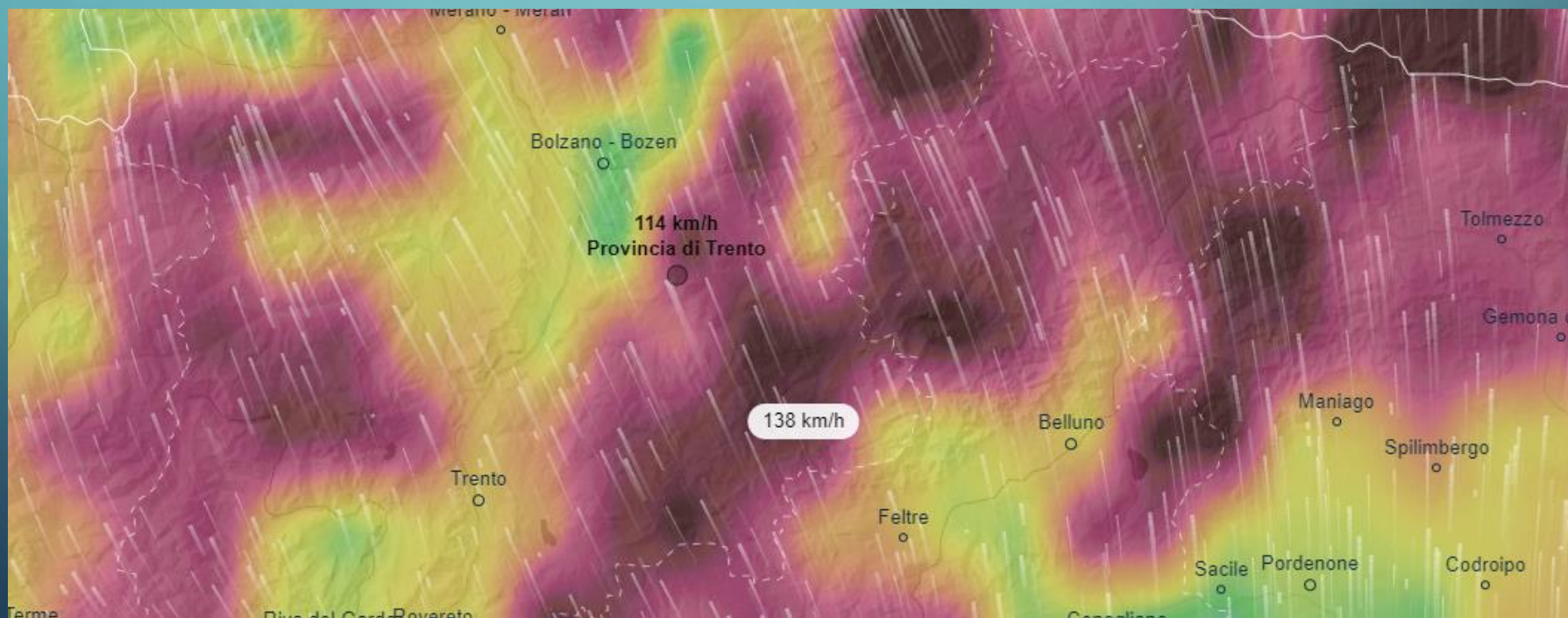


- Il fortissimo vento di scirocco, ha soffiato tra i 100 e i 200 km/h per diverse ore, e ha provocato la distruzione di migliaia di ettari di foreste alpine di conifere.
- Caddero al suolo in soli tre giorni (27, 28 e 29 ottobre) in Veneto e del Trentino fino a 715,8 mm e ben 870 mm a Forni di Sotto in Friuli di pioggia, e i fiumi Piave e Brenta sono straripati; in montagna stessa sorte hanno subito dei torrenti come il Rio Cadino.



Il vento di scirocco ha toccato il 29 ottobre raffiche in Veneto di 192,24 km/h registrate dalla stazione Arpa del monte Cesen e di 166,68 km/h dall'omologa stazione del monte Verena mentre in Carnia si sono toccati i 200 km/h e i 170 km/h rispettivamente nelle stazioni Osmer di Cima Rest e di Col Gallina, e in Trentino, sul Passo Rolle, il vento ha raggiunto i 217,3 km/h.

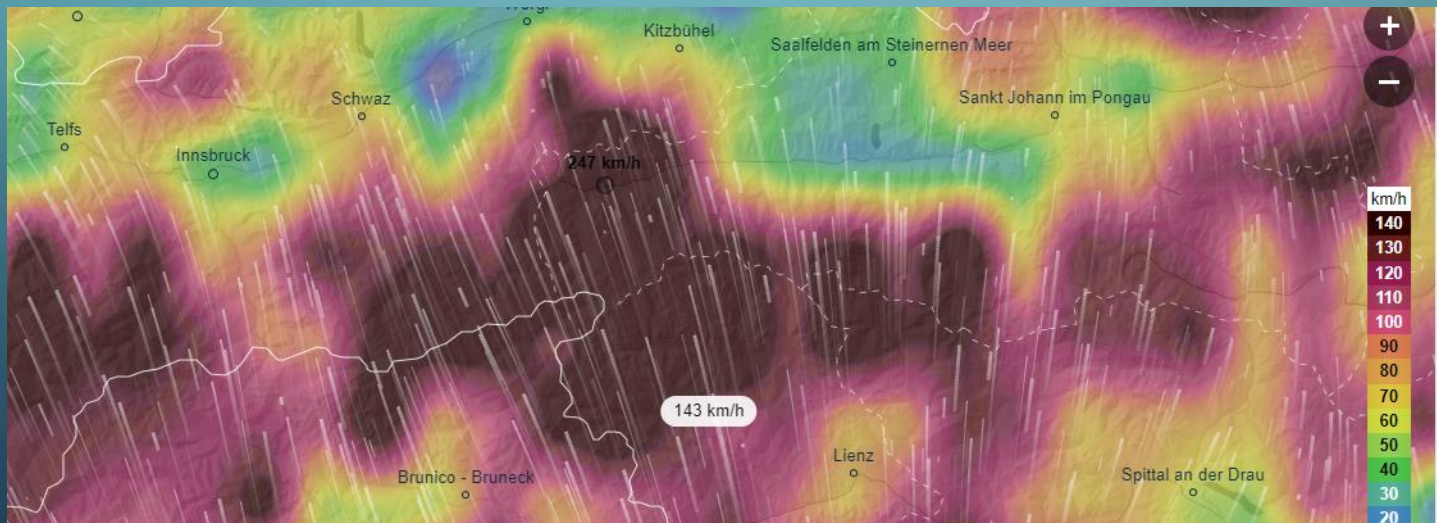
Tra le zone più colpite da esso ci sono state l'intera Valle di San Lucano, il Lago di Carezza, le Valli di Fassa e di Fiemme l'Altopiano di Piné e del Lagorai. Anche parte della Regione Lombardia (soprattutto la Valcamonica) è stata interessata dagli eventi.



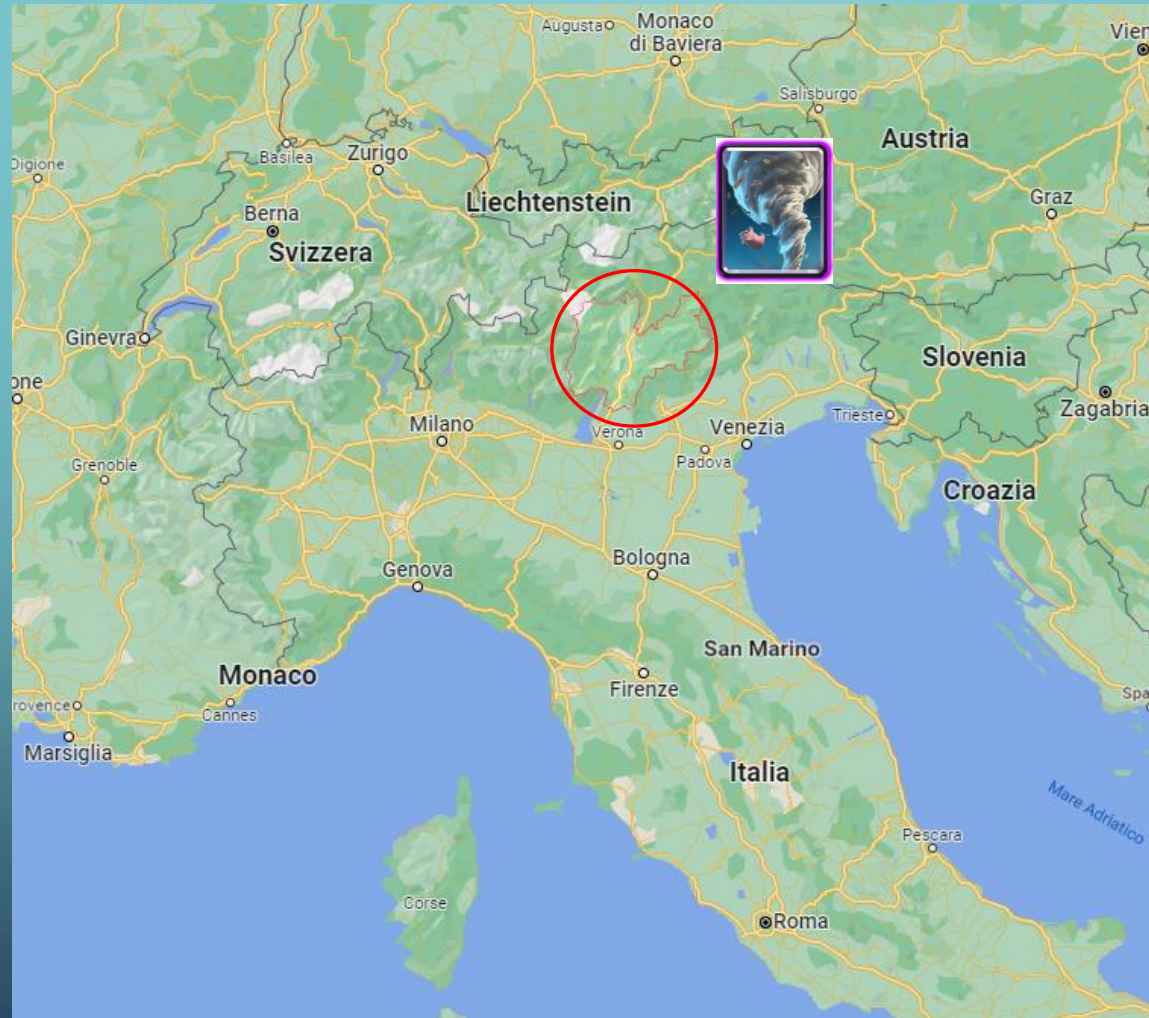
Il 29 ottobre si sono verificati due eventi di alta marea eccezionale che hanno interessato il centro storico di Venezia.

La stima dei danni in Friuli Venezia Giulia, secondo la protezione civile regionale, ammonta a 615 milioni di euro. In Veneto i danni sono stati valutati in 1,769 miliardi di euro mentre le stime per il Trentino sono sui 250-300 milioni di euro e 85,4 milioni in Alto Adige. In Lombardia le stime sono sui 40 milioni di euro di danni. Per quanto concerne le foreste, a causa del vento secondo le stime sono stati abbattuti 42 milioni di alberi su una superficie di 41 000 ettari.

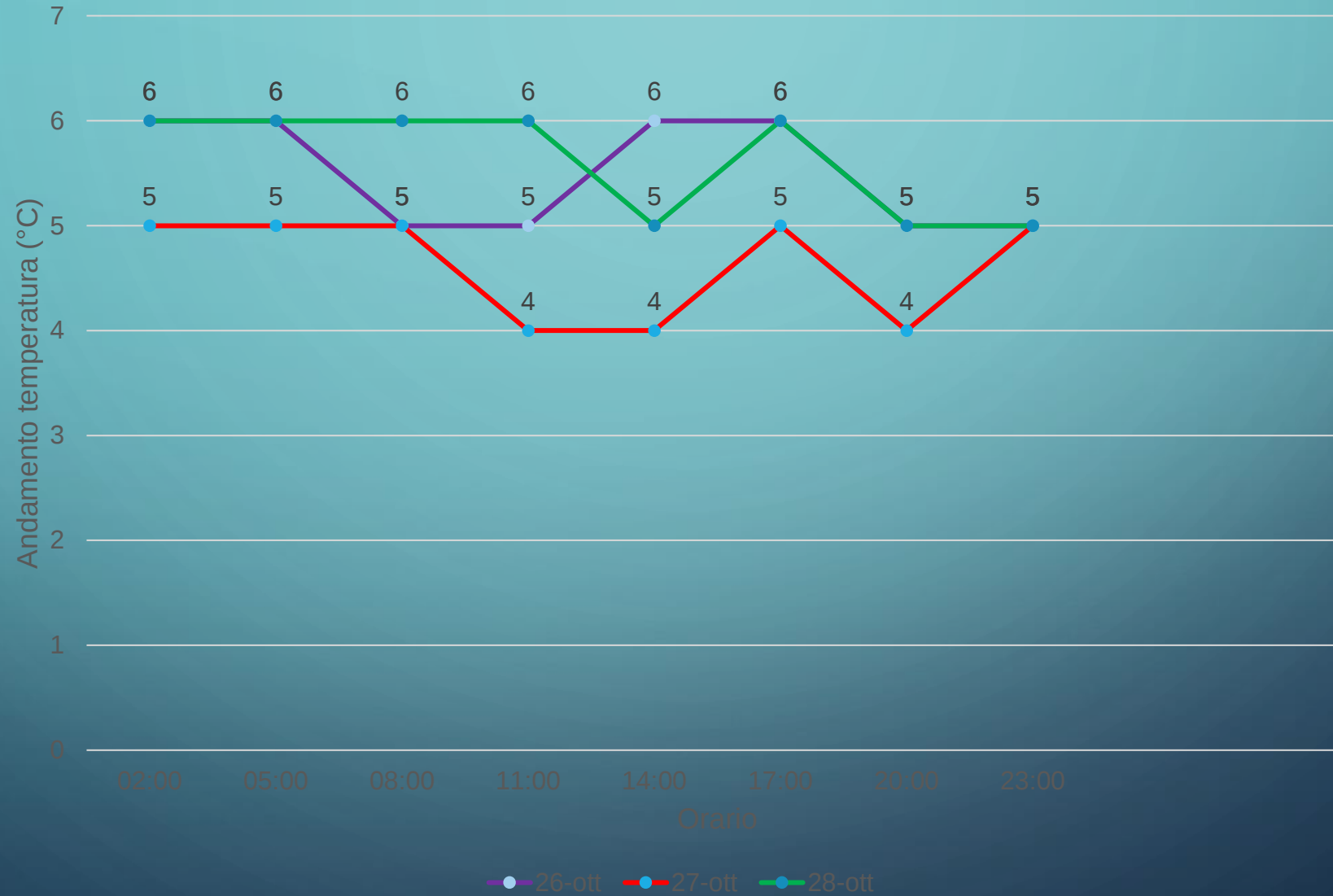
Tra le zone più colpite dal vento ci sono state l'intera Valle di San Lucano, il Lago di Carezza, le Valli di Fassa e di Fiemme l'Altopiano di Piné e del Lagorai. Anche parte della Regione Lombardia (soprattutto la Valcamonica) è stata interessata dagli eventi.



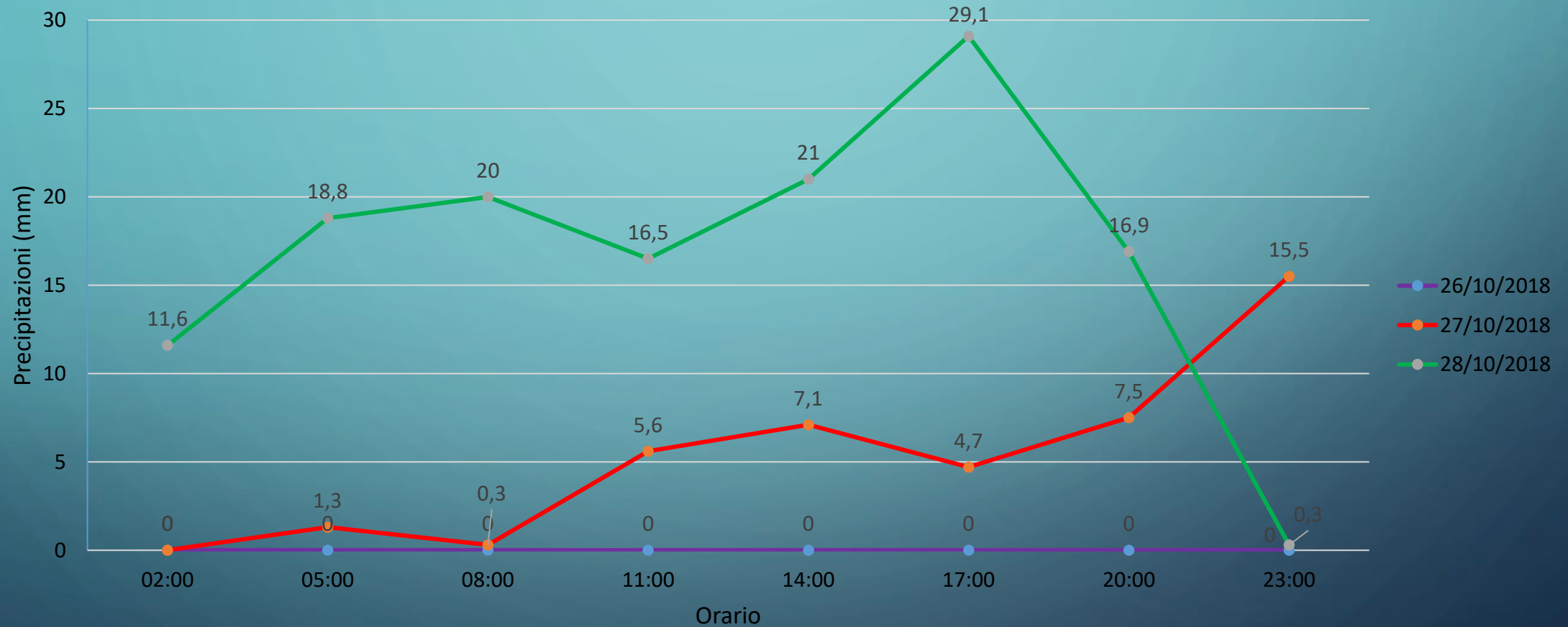
HITBOX TRENTINO



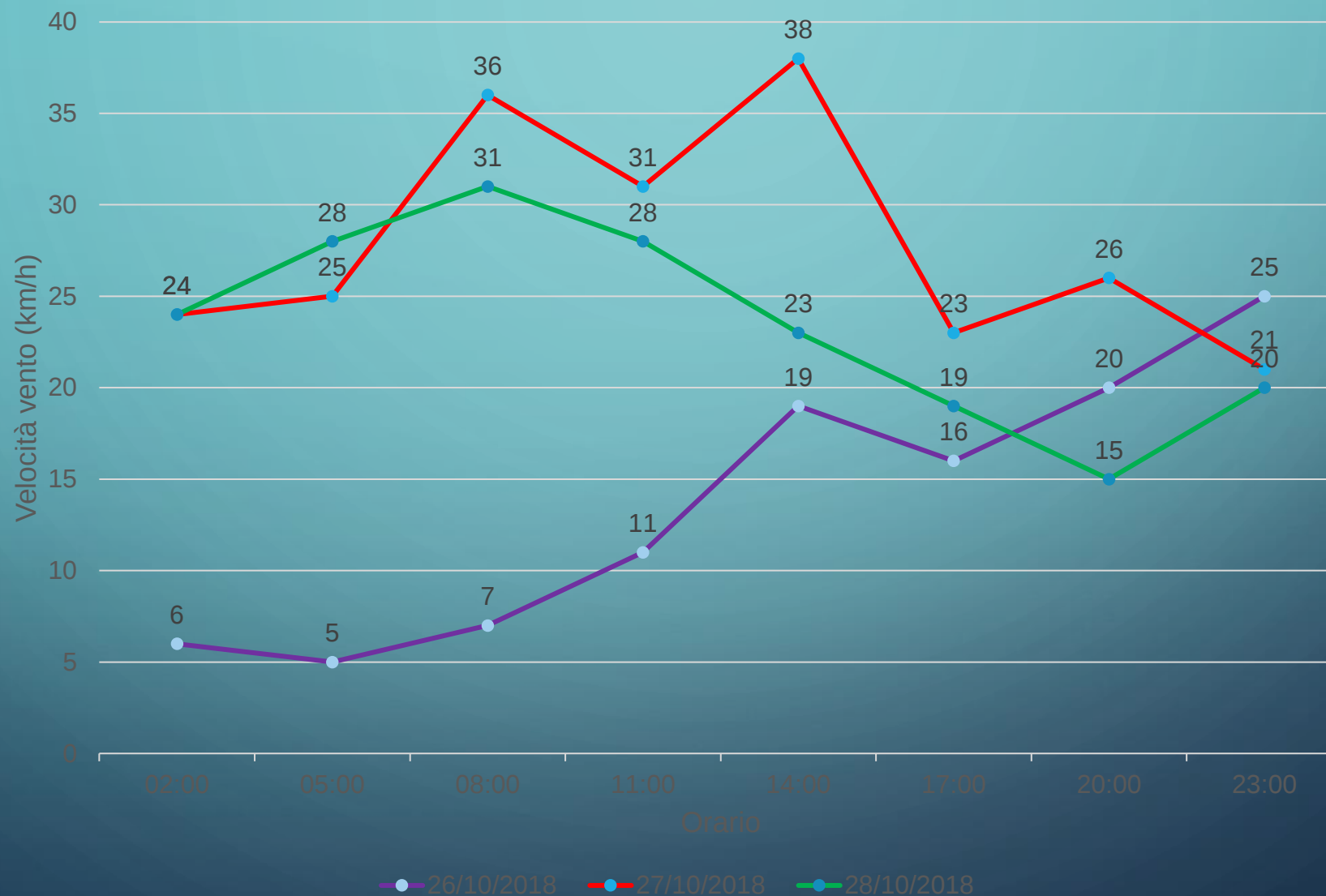
ANDAMENTO TEMPERATURA 26-27-28/10/2018



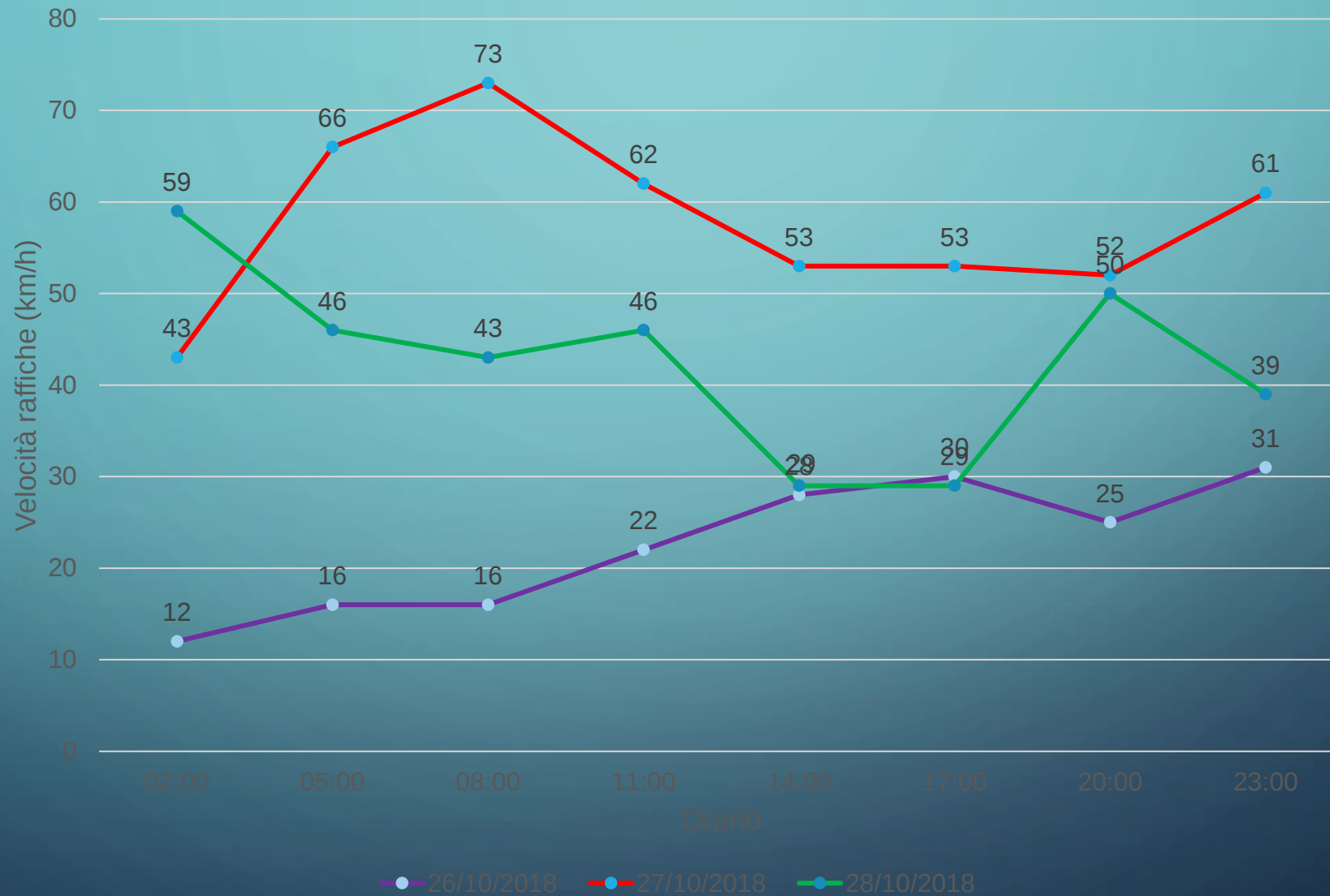
ANDAMENTO PRECIPITAZIONI 26-27-28/10/2018



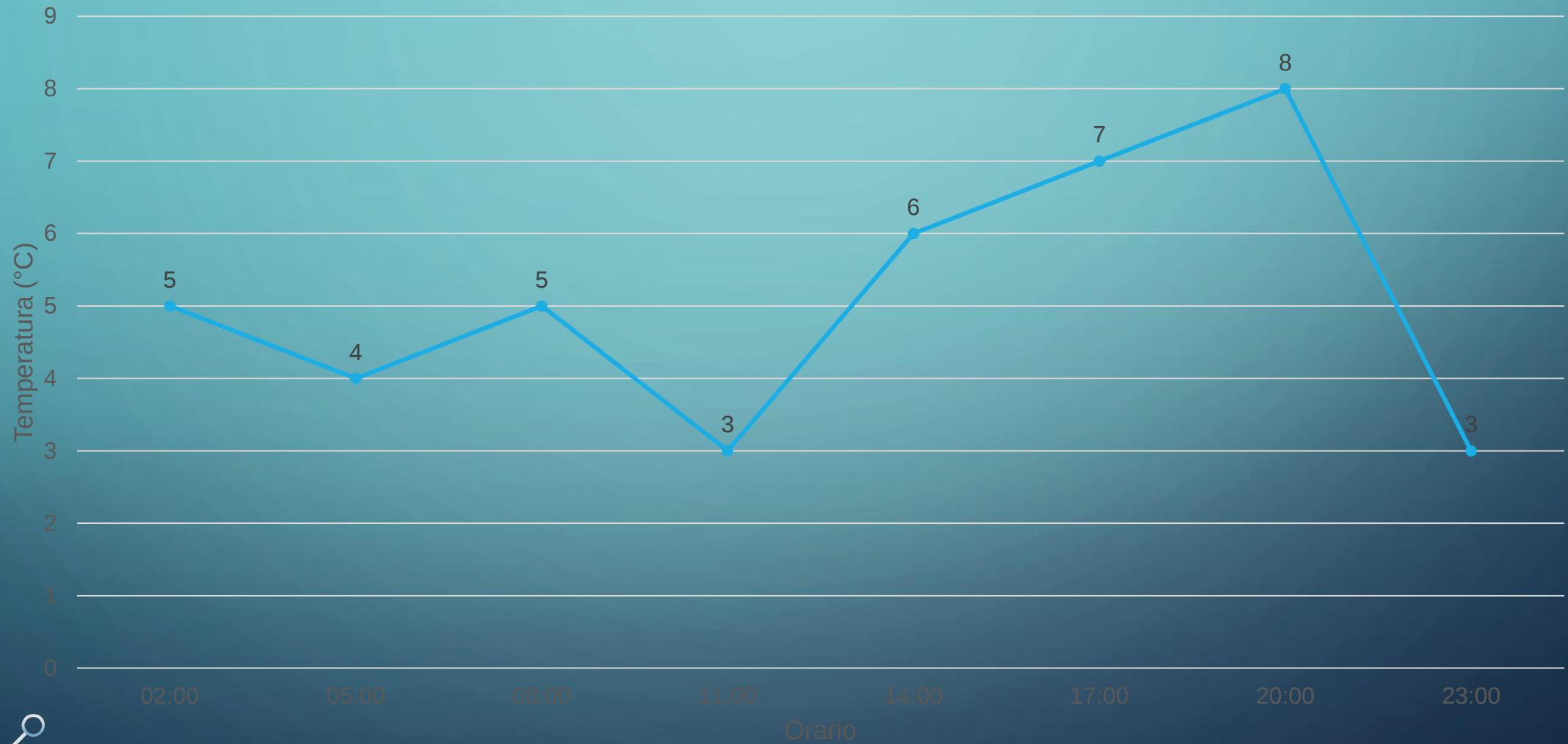
ANDAMENTO VENTO 26-27-28/10/2018



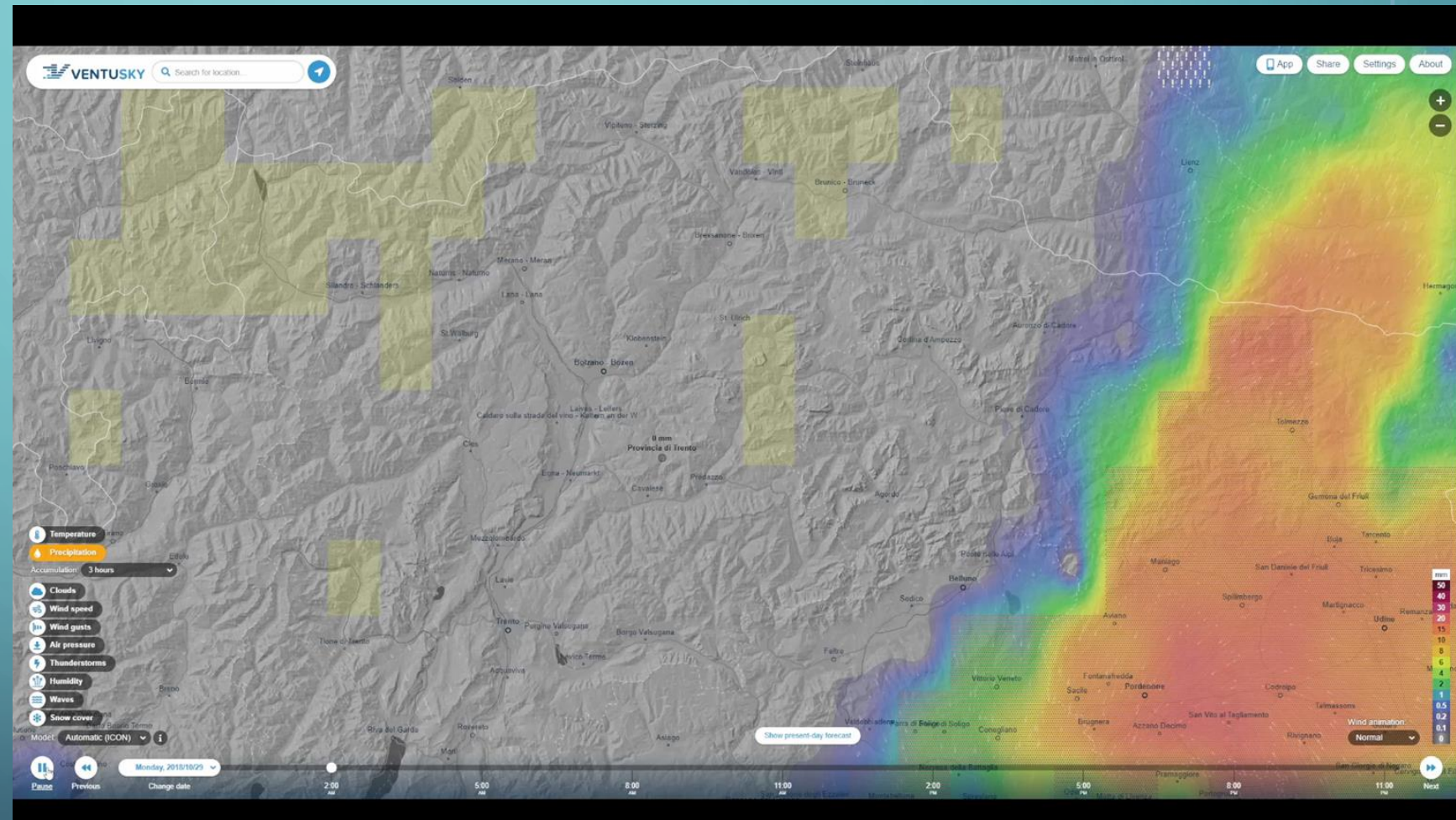
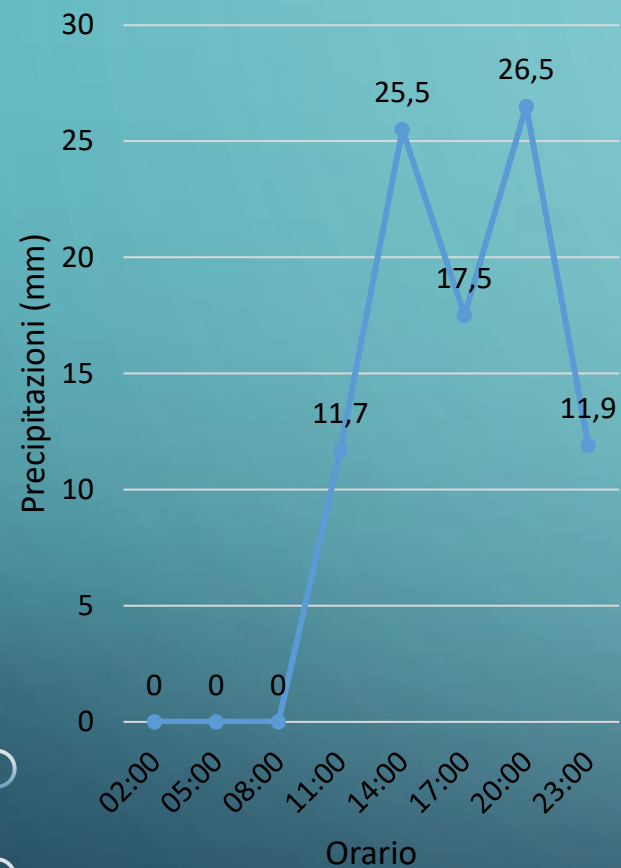
ANDAMENTO RAFFICHE 26-27-28/10/2018



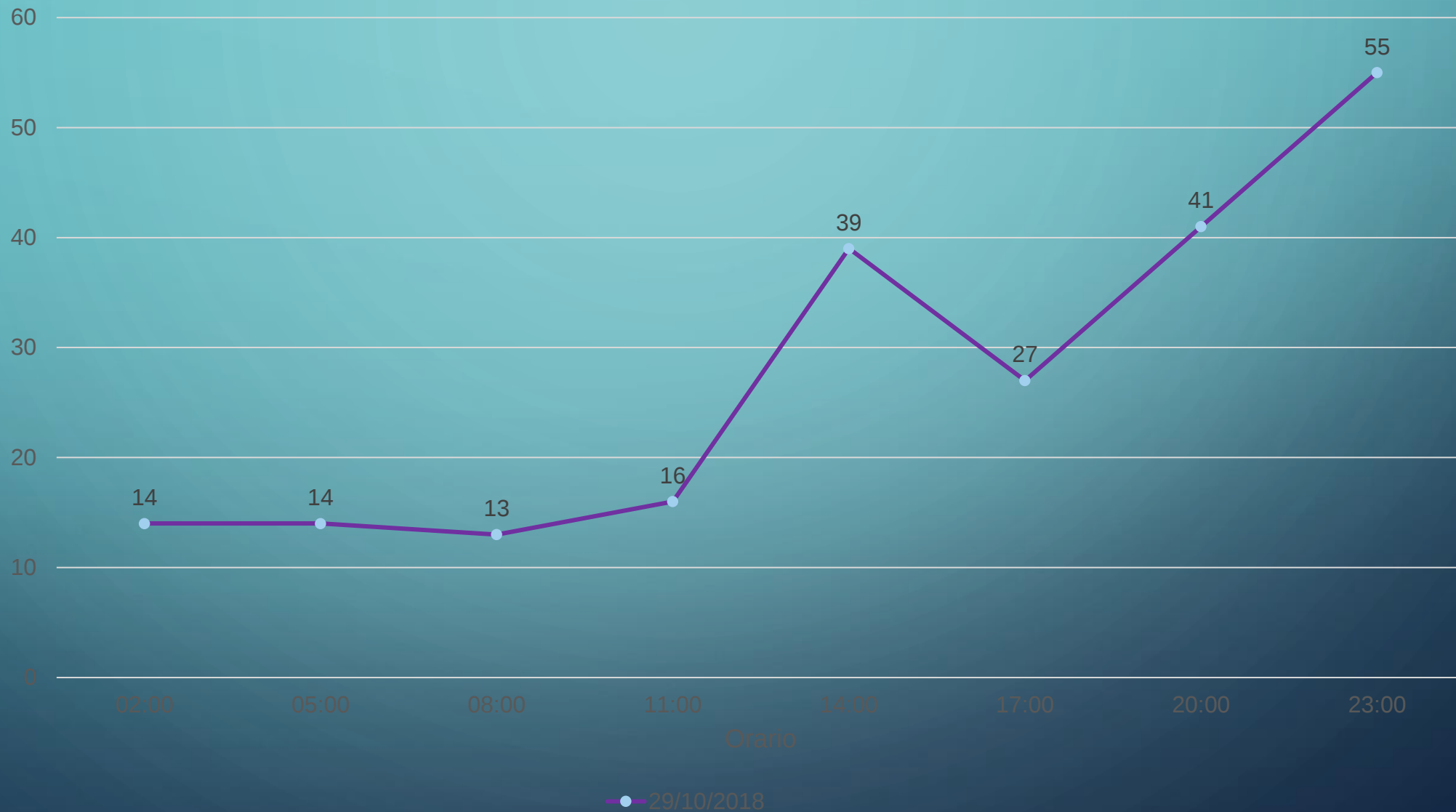
29 OTTOBRE 2018 TEMPERATURA



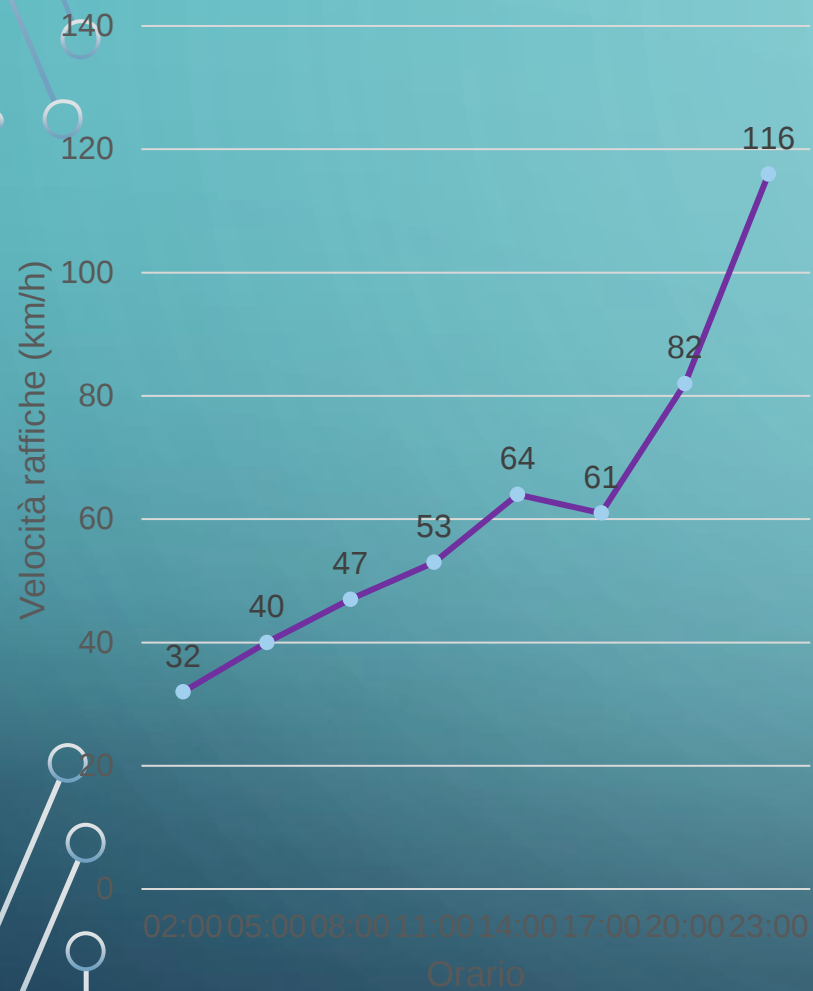
29 OTTOBRE 2018 PRECIPITAZIONI



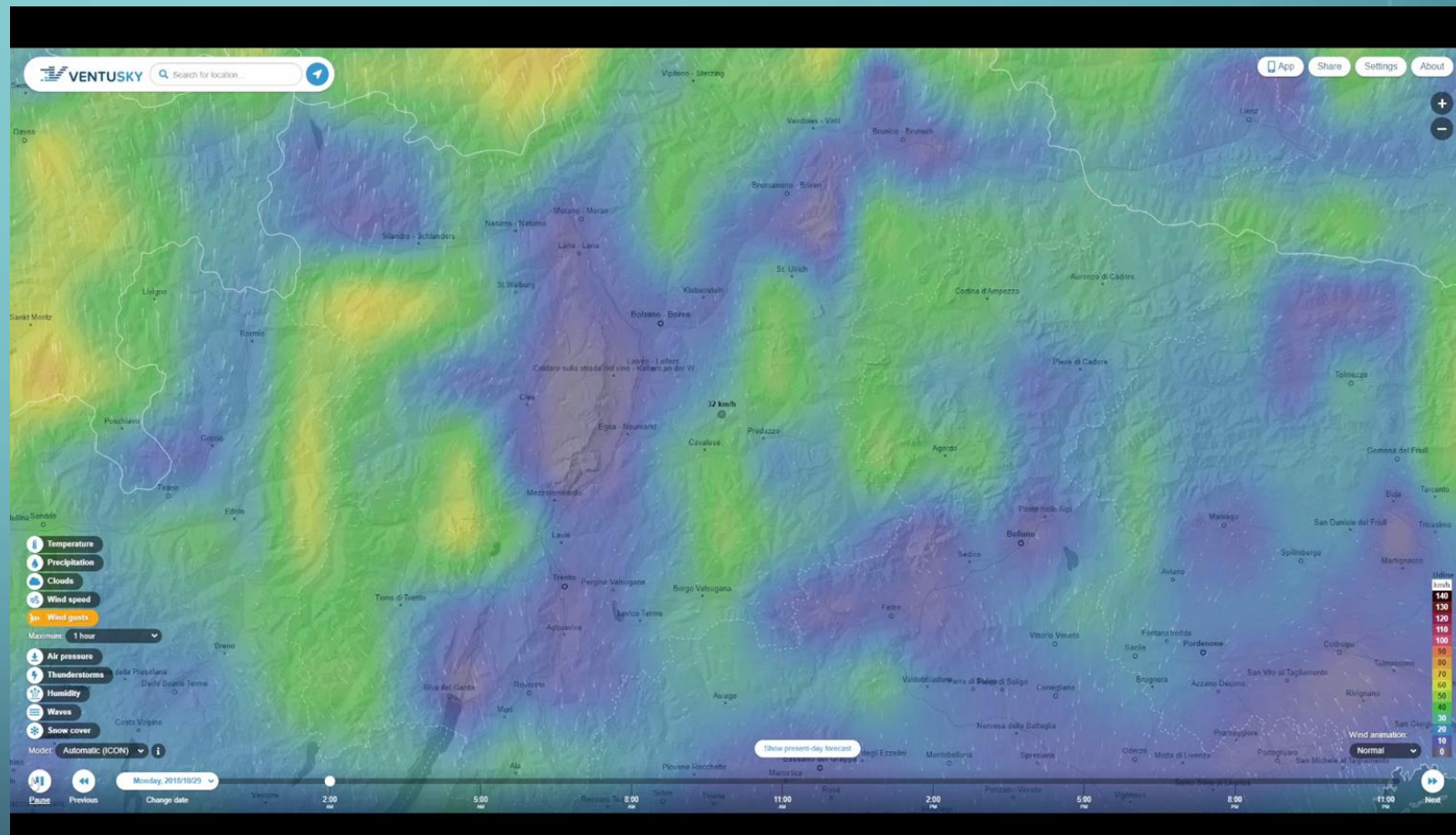
29 OTTOBRE 2018 VENTO



29 OTTOBRE 2018 RAFFICHE



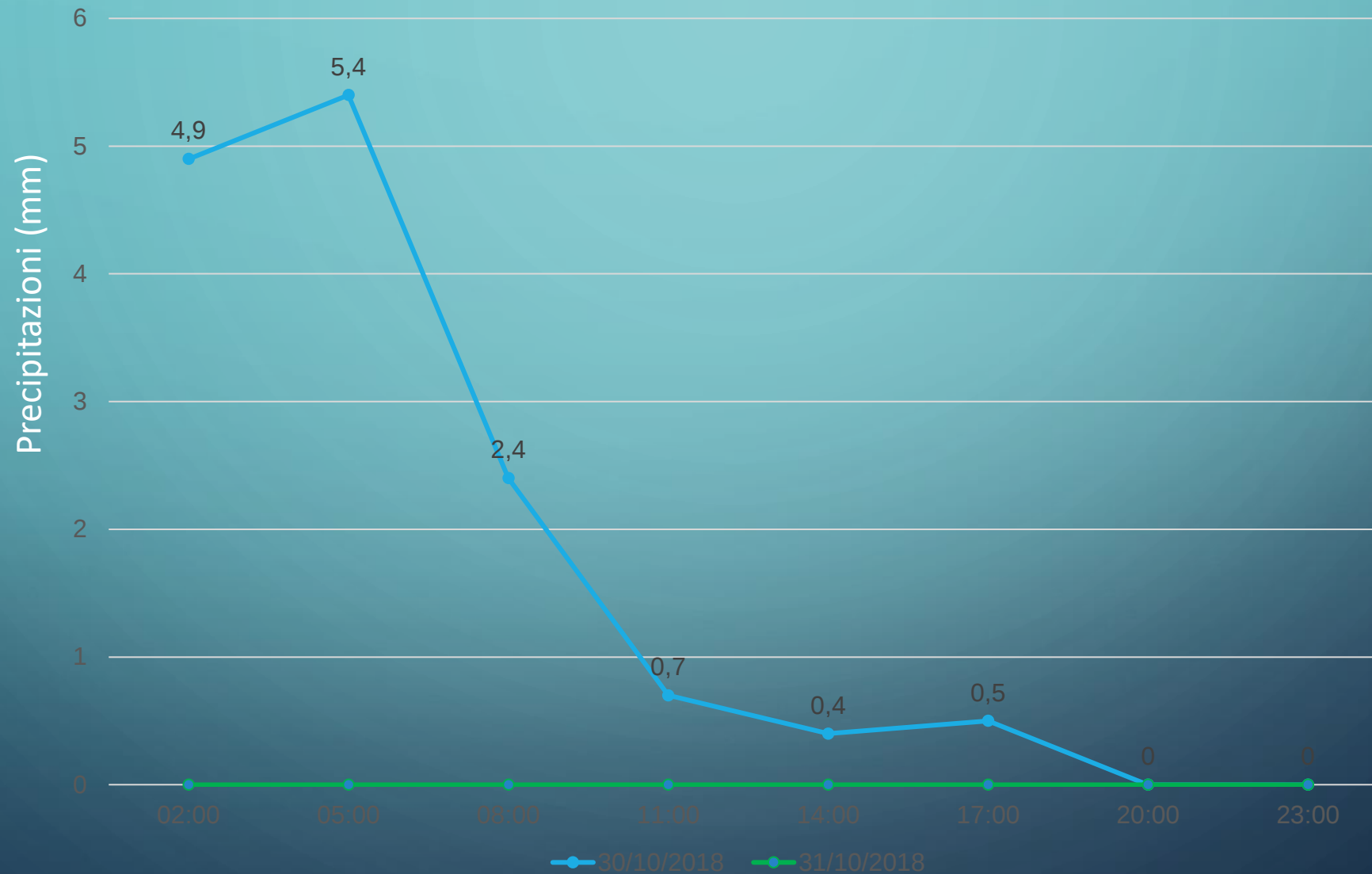
● 29/10/2018



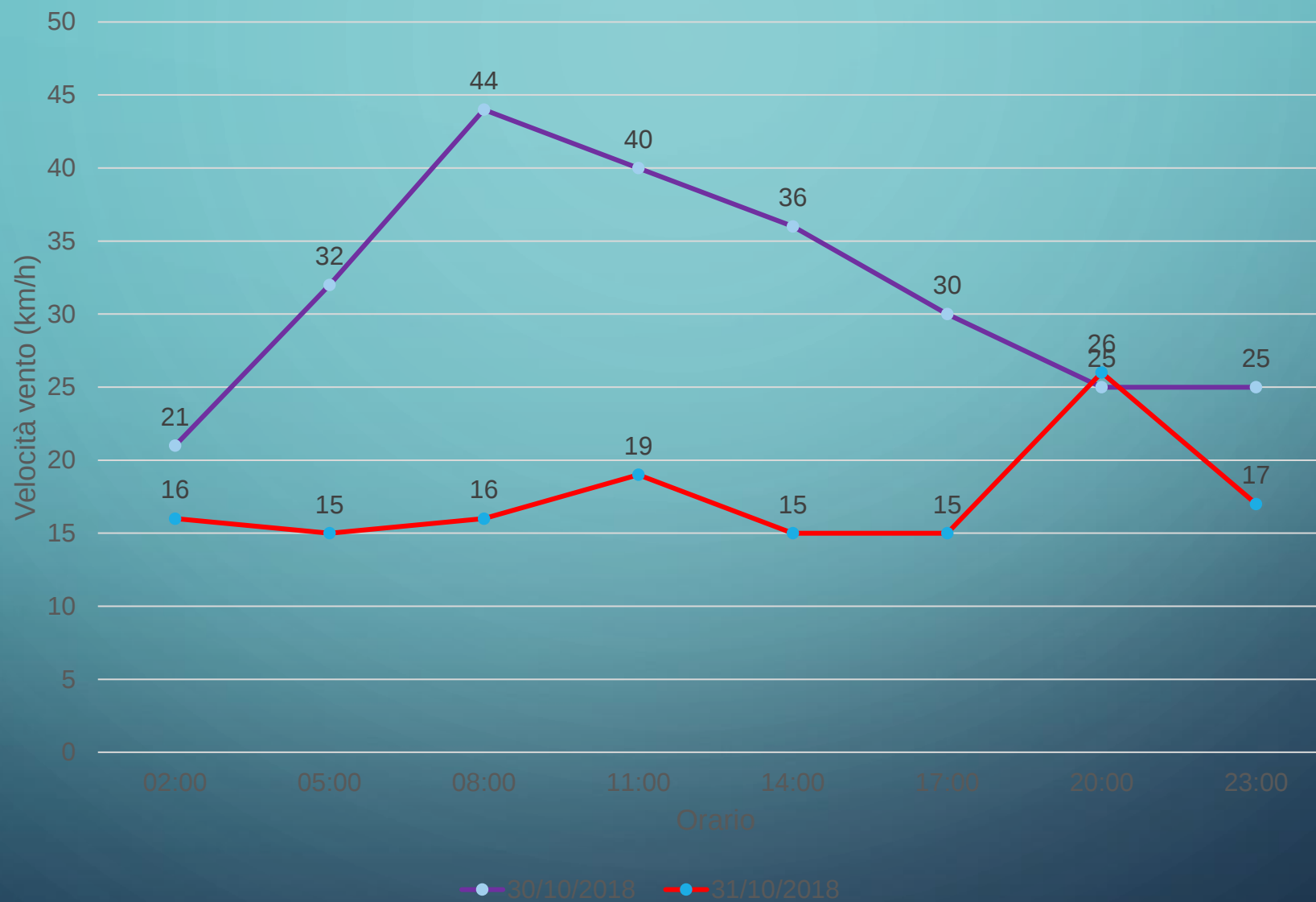
ANDAMENTO TEMPERATURA 30-31/10/2018



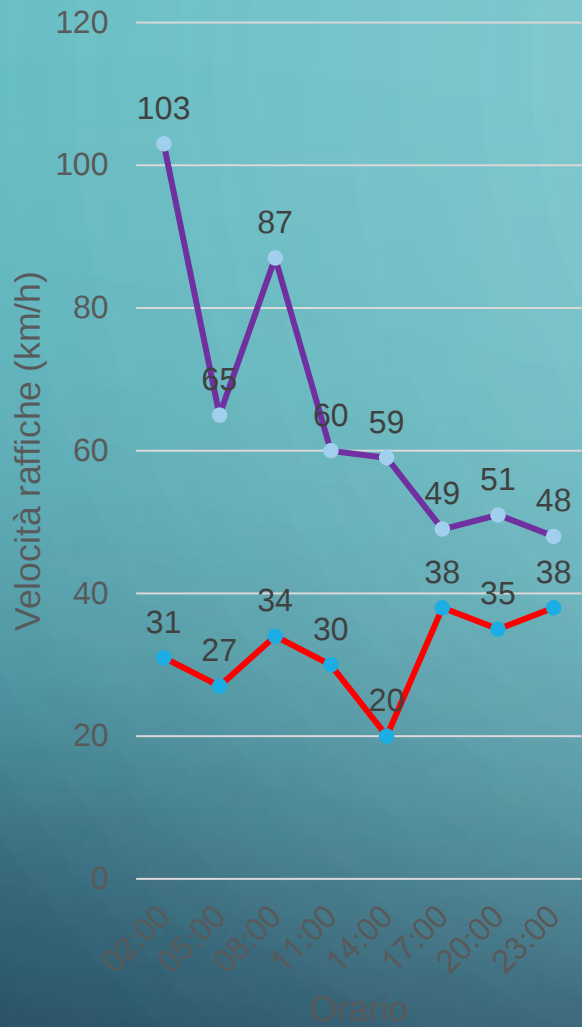
ANDAMENTO PRECIPITAZIONI 30-31/10/2018



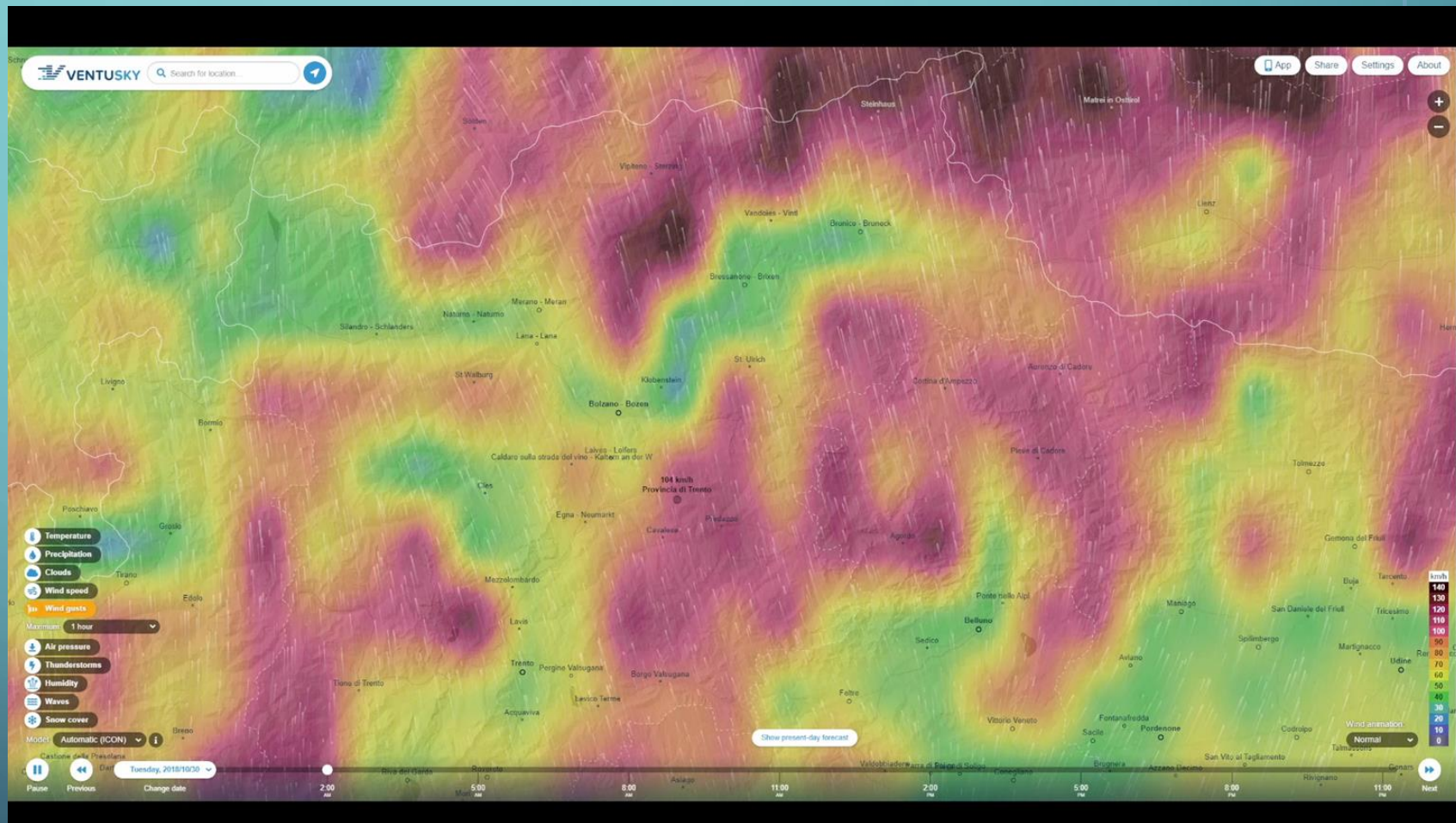
ANDAMENTO VENTO 30-31/10/2018



ANDAMENTO RAFFICHE 30-31/10/2018



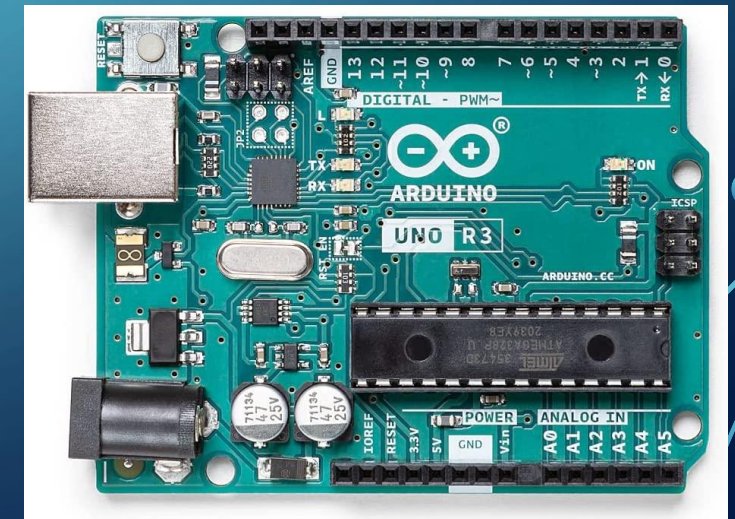
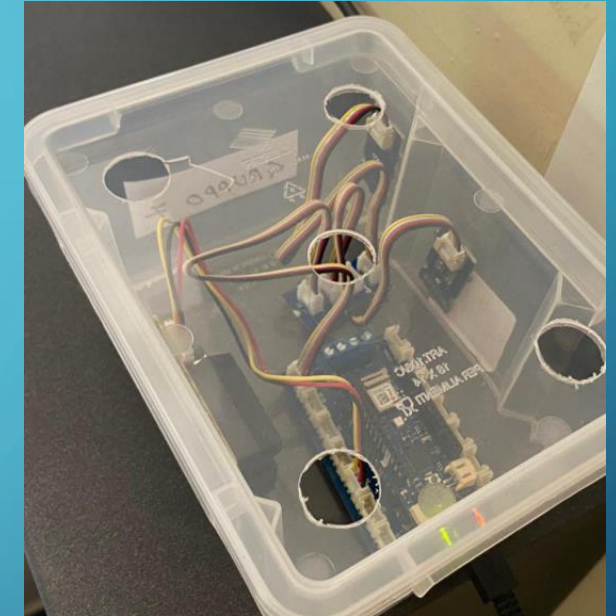
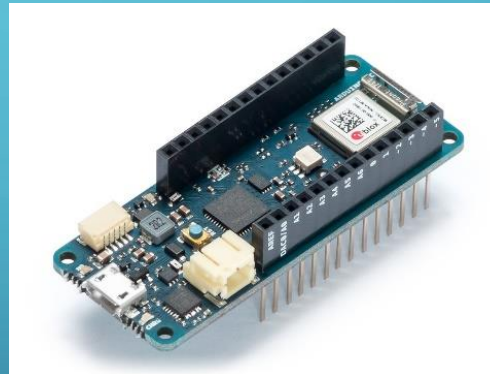
● 30/10/2018 ● 31/10/2018

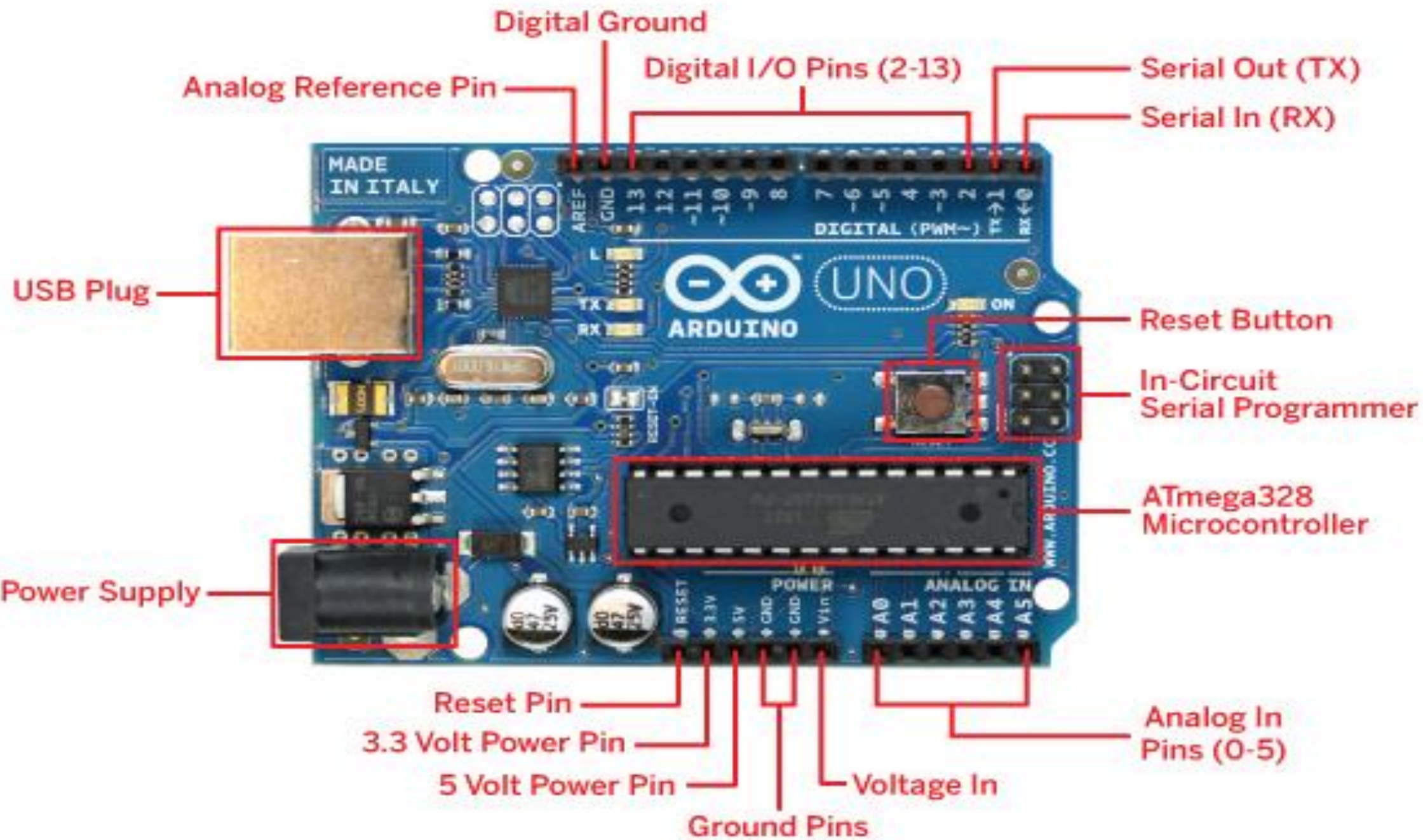


IL NOSTRO PROGETTO VEDI

Arduino è una piattaforma hardware composta da una serie di schede elettroniche dotate di un microcontrollore (il cervello). È stata ideata e sviluppata nel 2005 da alcuni membri dell'Interaction Design Institute di Ivrea (TO) come strumento per la prototipazione rapida, con scopi sia didattici che professionali. Con Arduino si possono realizzare controllori di luci, sensori.

Noi abbiamo collegato al nostro arduino diversi sensori per monitorare i parametri ambientali all'interno della nostra scuola (temperatura, pressione, umidità, concentrazione di CO₂, polvere); per capire se una semplice serra potrebbe cambiare la situazione di un determinato luogo come la nostra scuola





```

1 #include "Seeed_SHT35.h"
2 #define SECRET_SSID "TassoniWiFi"
3 #define SECRET_PASS "ospitewifi"
4
5 #include <WiFiNINA.h>
6 WiFiClient client;
7 char ssid[] = SECRET_SSID; // SSID (name)
8 char pass[] = SECRET_PASS;
9
10
11 int status = WL_IDLE_STATUS; // the Wifi radio's status
12 #include <ArduinoJson.h>
13 //vedi https://arduinojson.org/v5/assistant/
14 const size_t json_capacity = JSON_OBJECT_SIZE(7) + 100;
15 StaticJsonDocument<json_capacity> doc;
16
17
18 char dboard_server[] = "10.0.50.10"; // Dashboard Server
19 int dboard_port = 8000;
20
21 /*SAMD core*/
22 #ifdef ARDUINO_SAMD_VARIANT_COMPLIANCE
23 #define SDAPIN 20
24 #define SCLPIN 21
25 #define RSTPIN 7
26 #define SERIAL SerialUSB
27 #else
28 #define SDAPIN A4
29 #define SCLPIN A5
30 #define RSTPIN 2
31 #define SERIAL Serial
32 #endif
33
34 SHT35 sensor(SCLPIN);
35
36
37 #include "Seeed_BME280.h"
38 #include <Wire.h>
39
40 BME280 bme280;
41
42
43 int pin = 2;
44 unsigned long duration;
45 unsigned long starttime;
46 unsigned long sampletime_ms = 30000; //sampe 30s ;
47 unsigned long lowpulseoccupancy = 0;
48 float ratio = 0;
49 float concentration = 0;
50
51 void setup() {
52   Serial.begin(9600);
53   if(!bme280.init()){
54     Serial.println("Device error!");
55   }
56
57   delay(10);
58   SERIAL.println("serial start!!");
59   if(sensor.init()){
60     {
61       SERIAL.println("sensor init failed!!!");
62     }
63
64     pinMode(pin, INPUT);
65
66     // attempt to connect to Wifi network:
67     while (status != WL_CONNECTED) {
68       Serial.print("Attempting to connect to network: ");
69       Serial.println(ssid);
70
71       // Connect to WPA/WPA2 network:
72       status = WiFi.begin(ssid, pass);
73
74       // wait 10 seconds for connection:
75       delay(10000);
76     }
77
78     | delay(1000);
79   }
80 }
81
82 void loop() {
83   // put your main code here, to run repeatedly:
84
85   u16 value=0;
86   u8 data[6]={0};
87   float temp,hum;
88   if(NO_ERROR!=sensor.read_meas_data_single_shot(HIGH_REP_WITH_STRCH,&temp,&hum))
89   {
90     SERIAL.println("read temp failed!!");
91     SERIAL.println(" ");
92     SERIAL.println(" ");
93     SERIAL.println(" ");
94   }
95   else
96   {
97     SERIAL.println("result=====");
98     SERIAL.print("temperature =");
99     SERIAL.println(temp);
100
101     SERIAL.print("humidity =");
102     SERIAL.println(hum);
103
104     SERIAL.println(" ");
105     SERIAL.println(" ");
106     SERIAL.println(" ");
107   }
108
109   float press_b, temp_b, hum_b;
110
111   //get and print temperatures
112   Serial.print("Temp: ");
113   temp_b = bme280.getTemperature();
114   Serial.print(temp_b);
115   Serial.println("C");//The unit for Celsius because original arduino don't support speical symbols
116
117   //get and print atmospheric pressure data
118   Serial.print("Pressure: ");
119   press_b = bme280.getPressure();
120   Serial.print(press_b);
121   Serial.println("Pa");
122
123   //get and print humidity data
124   Serial.print("Humidity: ");
125   hum_b = bme280.getHumidity();
126   Serial.print(hum_b);
127   Serial.println("%");
128
129
130
131   starttime = millis();//get the current time;
132
133   Serial.println("Starting Dust Sampling");
134   while ((millis()-starttime) < sampletime_ms) {
135     duration = pulseIn(pin, LOW);
136     lowpulseoccupancy = lowpulseoccupancy+duration;
137   }
138
139   ratio = lowpulseoccupancy/(sampletime_ms*10.0); // Integer percentage 0=>100
140   concentration = 1.1*pow(ratio,3)-3.8*pow(ratio,2)+520*ratio+0.62; // using spec sheet curve
141   Serial.print(lowpulseoccupancy);
142   Serial.print(",");
143   Serial.print(ratio);
144   Serial.print(",");
145   Serial.println(concentration);
146   lowpulseoccupancy = 0;
147
148
149
150   doc["temp_aria"] = temp_b;
151   doc["umid_aria"] = hum_b;
152   doc["press_aria"] = press_b;
153   doc["ha_temp"] = temp;
154   doc["ha_hum"] = hum;
155   doc["pm_conc"] = concentration;
156
157   client.stop();
158
159   Serial.println("\nConnessione al server IoT");
160   if (client.connect(dboard_server, dboard_port))
161   {
162     Serial.println("Connesso al Cloud IoT");
163     // Make a HTTP request:
164     client.println("POST /api/v1/ex8GRGFLq9oIj0WUt0a/telemetry HTTP/1.1");
165     client.println("Host: ");
166     client.println("Connection: close");
167     client.print("Content-Length: ");
168     client.println(measureJson(doc));
169     client.println("Content-Type: application/json");
170     // Terminate headers with a blank line
171     client.println();
172     // Send JSON document in body
173
174     serializeJson(doc, client);
175     serializeJson(doc, Serial);
176
177     // note the time that the connection was made:
178   }
179
180   delay(300000);
181 }
182
183
184 void printData() {
185   Serial.println("Board Information:");
186   // print your board's IP address:
187   IPAddress ip = WiFi.localIP();
188   Serial.print("IP Address: ");
189   Serial.println(ip);
190
191   Serial.println();
192   Serial.println("Network Information:");
193   Serial.print("SSID: ");
194   Serial.println(WiFi.SSID());
195
196   // print the received signal strength:
197   long rssi = WiFi.RSSI();
198   Serial.print("signal strength (RSSI):");
199   Serial.println(rssi);
200
201 }
202

```