





FUEL PRICE MONITOR

Controllo attraverso ESP32 di un pannello touch interattivo che espone i prezzi dei distributori di carburante in una determinata zona.

di DANIEL SCAINI

I continui rincari delle materie prime che colpiscono la nostra vita quotidiana cominciano ad essere un problema non trascurabile. Alcuni di questi sono inevitabili, mentre per altri riusciamo ad avere un minimo di scelta e a far fronte trovando alternative. Quello che da alcuni mesi ci preoccupa molto è il rincaro dei carburanti, tanto che è cresciuta la sensibilità circa la ricerca del distributore che esponga il prezzo migliore. Ma quante volte subito dopo aver appena fatto il pieno, magari in una zona che frequentiamo d'abitudine, notiamo lungo la strada un distributore di gran lunga più conveniente? La soluzione potrebbe essere un dispositivo che ci aiuti in queste decisioni e ci consigli il migliore. In questo articolo proveremo a sviluppare un assistente a cui, passati in ingresso una città oppure le sue coordinate, ci proponga a video tutti i distributori e ci comunichi in tempo reale quale o quali siano i migliori dove poter fare rifornimento. Per far ciò utilizzeremo, lato software, un'app di messaggistica già nota a tutti: Telegram. La quale potrà inviare messaggi di testo oppure condividere la propria posizione. Lato hardware invece utilizzeremo una delle schede più conosciute ed utilizzate, ovvero la ESP32. Quest'ultima sarà all'interno di un circuito più complesso che prevede al suo fianco un utilissimo display touch LCD, in modo da poter mostrare a video i risultati e da poter interagire con essi attraverso un'interfaccia semplicissima, facilmen-



Fig. 1
Interfaccia
utente.

te consultabile anche da chi è poco avvezzo alla tecnica (**Fig. 1**).

I SERVIZI

Come accennato durante l'introduzione l'algoritmo si basa su latitudine e longitudine e sfrutta due servizi per poter estrarre i dati di cui abbiamo bisogno.

Il primo servizio si occupa di risalire, dall'inserimento di un campo testuale, alle coordinate vere e proprie. Su Internet oggi giorno vengono offerti numerosi servizi che si occupano di geo tracking o di reverse geo tracking.

Fra i più famosi vi sono senza dubbio quelli offerti da società come Google (<https://developers.google.com/maps/documentation/geocoding/overview>), Here, uno dei leader nel campo dei navigatori (<https://developer.here.com/tutorials/learn-the-reverse-geocoding-using-postman/>) oppure OpenWeather (<https://openweathermap.org/api/geocoding-api>). Nel nostro caso sfrutteremo un quarto fornitore free. La nostra scelta ricade infatti su Geocode.xyz. Questo ci offre, previa iscrizione, la possibilità di fare diverse richieste gratis a patto che non superiamo la frequenza di 1 per secondo (<https://geocode.xyz/pricing>).

Una volta registrati sul sito, attraverso l'inserimento di una username valida ed una password e ricevuta la e-mail di conferma, viene rilasciata API_KEY personale per poter invocare le richieste. L'engagement con l'API avviene attraverso una chiamata GET al seguente URL: https://geocode.xyz/Milano?json=1&auth=API_KEY.

Una volta completata la richiesta, abbiamo accesso al seguente codice JSON che riporta, oltre a latitudine e longitudine, anche la precisione del dato di ritorno:

```
{
  "standard": {
    "address": {},
    "statename": {},
    "city": "Milano",
    "prov": "IT",
    "countryname": "Italy",
    "postal": {},
    "confidence": "0.90"
  },
  "longt": "9.18023",
  "alt": {
    "loc": [
      {
        "longt": "9.16903501162404",
        "prov": "LOMBARDIA",
        "city": "MILANO",
        "postal": "20161",
        "score": "3011",
        "latt": "45.5186721687148"
      },
      {
        "longt": "9.20334553448277",
        "prov": "IT",
        "city": "Milano",
        "postal": "20124",
        "score": "580",
        "latt": "45.4855161034483"
      },
      {
        "longt": "9.16904",
        "prov": "IT",
        "city": "Milano",
        "countryname": "Italy",
        "postal": "20161",
        "region": {},
        "latt": "45.51867"
      }
    ]
  },
  "elevation": {},
  "remaining_credits": "-7",
```

Listato 1

```
{
  "success": true,
  "center": {
    "lat": 44.90938025959227,
    "lng": 10.66000690845997
  },
  "results": [
    {
      "id": "Tamoil Gualtieri",
      "name": "8976",
      "fuels": [
        {
          "id": 47742999,
          "price": 1.819,
          "name": "Benzina",
          "fuelId": 1,
          "isSelf": true
        },
        {
          "id": 47742998,
          "price": 1.859,
          "name": "Gasolio",
          "fuelId": 2,
          "isSelf": true
        }
      ],
      "location": {
        "lat": 44.889557326884855,
        "lng": 10.617287009954453
      },
      "insertDate": "2023-01-17T11:16:11+01:00",
      "address": "VARIANTE S.S. 62 9 42044 - GUALTIERI RE",
      "brand": "Tamoil"
    },.....
  ]
}
```

```
"latt": "45.47547"
}
```

Il secondo servizio, una volta impostati la posizione ed un raggio intorno ad essa, ci restituisce tutti i distributori della zona interessata.

Per ottenere questa lista sfruttiamo un API pubblica presente anche all'interno del sito del Ministero delle Imprese e del Made in Italy (<https://carburanti.mise.gov.it/ospzSearch/zona>).

All'interno del portale è possibile, attraverso un'interfaccia grafica basilare, scegliere il proprio comune di residenza sulla mappa, impostare un raggio di azione e ottenere così la lista completa di tutti i distributori di benzina nelle vicinanze.

A proposito del raggio, va precisato che il portale del MISE consente di impostare un'area di ricerca limitata, quindi se intendete modificare il raggio ricordate che il raggio impostato nel nostro firmware è 5 km e se andate oltre è possibile che i risultati restituiti contengano un errore; nessun problema,

invece, se impostate un raggio inferiore.

La riga di codice interessata è riportata nel paragrafo qui sotto e il parametro passato è "radius": 5.

L'API alla base di questa piattaforma, a differenza della precedente, è una chiamata POST, accessibile anche dall'esterno all'indirizzo: <https://carburanti.mise.gov.it/ospzApi/search/zona>. Trattandosi di una funzione POST, vi è un dato in entrata e un codice JSON che conterrà tutto il necessario; questa è la stringa corrispondente:

```
{"points": [{"lat": "44.917653509784884", "lng": "10.65221077251436"}], "radius": 5}
```

In risposta ad essa avremo accesso ad un oggetto con la struttura proposta dal **Listato 1**.

L'HARDWARE DEL PROGETTO

L'hardware su cui ci basiamo, che potete trovare nello store on-line di Futura Elettronica (www.futurashop.it), è un Esp32-S3 con display





Fig. 2
Il display con
Esp32 utilizzato.

TFT da 3,5". Il core di questa scheda è un processore operante a 2,4 GHz di frequenza di clock, con interfacce I²C-Bus, I²S, SPI, PWM, UART, USB; dispone inoltre di due porte USB di tipo C, uno slot per SD-Card ed è dotato connettività WiFi e Bluetooth. Nella **Fig. 2** vedete la nostra board da entrambe le facce.

Un altro aspetto non trascurabile è sicuramente la memoria, che può essere una Flash, RAM oppure EEPROM.

Sfrutteremo quest'ultima, Electronically Erasable Programmable Read Only Memory, per poter immagazzinare informazioni capaci di sopravvivere al riavvio della macchina stessa.

Il display ha un touchscreen capacitivo e sfrutta il controller grafico ILI9488 e interfacciamento SPI. Quest'ultima caratteristica lo rende decisamente migliore di altri display dotati di interfaccia di comunicazione parallela.

Arduino IDE ci mette a disposizione una libreria molto importante per poter interagire con questo

schermo; pertanto, una volta aperto il nostro IDE entriamo nella finestra di dialogo del Gestore Librerie (comando di menù Strumenti>Gestione Librerie) e da essa andiamo ad installare la versione 1.1.2 di LovyanGFX.

L'uso di questa libreria è veramente elementare: si crea un oggetto LGFX, si inizializza, si settano alcuni parametri, che vedremo dopo, quindi questo diventa la "tela" su cui possiamo disegnare o scrivere. Un esempio può essere il codice contenuto in HowToUse→simple_use.ino che si trova nella libreria LovyanGFX. Ci sono anche altre entità, come per esempio le Sprite, ma servono in caso di utilizzi più avanzati.

IL BOT DI TELEGRAM

Telegram è un'applicazione di messaggistica istantanea, tra le più utilizzate in Italia insieme con Facebook Messenger, WhatsApp, Signal e Instagram Direct.

Nasce dall'idea di Pavel Durov, giovane imprenditore russo ed ex studente dell'Università statale di San Pietroburgo, insieme a suo fratello Nikolai. Tra le altre cose, i fratelli Durov hanno fondato anche il social network russo VK (VKontakte, che in italiano significa letteralmente "in contatto"). L'anno di nascita di Telegram risale al 2013, quattro anni più tardi rispetto al suo diretto concorrente WhatsApp. Pavel Durov sapeva che per ottenere successo avrebbe dovuto proporre qualcosa di diverso, come effettivamente ha fatto.

Nel corso degli ultimi anni, numerosi utenti si sono avvicinati a Telegram perché delusi dai "down" che hanno colpito l'app concorrente.

Un altro fattore importante da considerare nella vicenda è la sicurezza di WhatsApp, più volte finito sotto attacco dai suoi detrattori, i quali spingono molto sulle polemiche legate alla composizione dell'impero Zuckerberg (WhatsApp, Instagram, Facebook).

Una delle peculiarità più interessanti di questo servizio di messaggistica è insita nei gruppi: questi possono raggiungere dimensioni enormi, anche oltre i 100.000 membri.

Inoltre ci dà la possibilità di creare dei veri e propri canali di trasmissione che possono avere un numero illimitato di utenti e possono essere sia privati che pubblici.

Tra quelli privati ci sono i bot ed è proprio quest'ultima caratteristica che possiamo sfruttare per creare un filo diretto fra noi e la nostra scheda con ESP32.

Un bot è un "utente" di Telegram non umano, fatto

Mensile di elettronica applicata, attualità scientifica, novità tecnologiche.

Elettronica In

www.elettronica.in.it

oltre l'elettronica