

Leggere la generazione DPF con un Arduino (ESP32)

Questa descritta è la procedura per poter leggere tramite OBDII senza l'utilizzo di uno smartphone il messaggio di rigenerazione del DPF.

Cosa ci serve?

Un lettore OBD che invia i dati tramite porta bluetooth. Uno compatibile è sicuramente questo

https://www.amazon.it/dp/B09ZYGVG98?ref=ppx_yo2ov_dt_b_fed_asin_title

Una scheda Arduino avanzata che si chiama ESP32. Se si vuole poterla inserire in una scatolina stampata in 3D comprate questa versione

https://www.amazon.it/dp/B08BTWJGFX?ref=ppx_yo2ov_dt_b_fed_asin_title

Due led da 3.5mm (blu e rosso) e due resistenze da 330 ohm

Un saldatore e po' di manualità.

Se avete una stampante 3D o conoscete qualcuno che ve la può stampare una contenitore per inserire la scheda. In alternativa potete trovarne una già fatta nei negozi di elettronica.

Scaricare i file necessari a questo indirizzo

https://github.com/bottosac/ESP32_OBDII_Scenic_DPFMode_LED

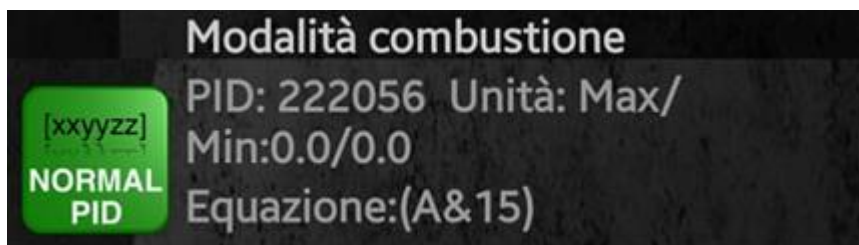
Come funziona?

Sulla porta OBD che trovate in basso a sinistra della vostra Scenic protetta da un coperchio vengono trasmesse tutte le informazioni che la centralina trasmette sul bus CAN di sistema.

Su questo OBD vengono inviati una serie di messaggi standard chiamati PID ([OBD-II PIDs - Wikipedia](#)).

Oltre ai messaggi standard ogni costruttore invia una serie di messaggi personalizzati tramite un comando apposito che può essere inviato alla centralina.

Il messaggio che a noi interessa intercettare è il 222056



Quando chiediamo alla centralina questo dato ci risponde con un numero che ha i seguenti valori:

- 1: normale
- 2: avvio rigenerazione
- 3: rigenerazione in corso
- 4: raffreddamento
- 5: rilascio NOX

- 6: rilascio SOX
- 7: off

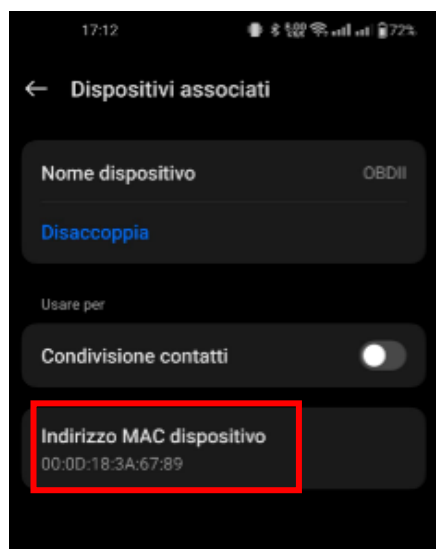
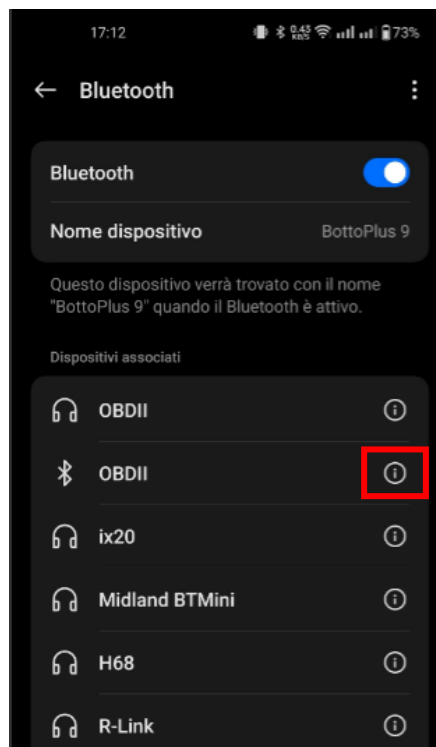
La nostra scheda ESP32 leggerà questo valore e accenderà un led rosso fisso quando il valore è diverso da 1

Procedimento

Collegare l'interfaccia OBD

Innanzitutto colleghiamo la nostra presa OBD all'auto e colleghiamoci con il cellulare attivando il bluetooth e ricercando i dispositivi. Troverete una scheda di nome ODBII. Collegatevi e si chiede il pin inserite 1234

A questo punto ci interessa sapere l'indirizzo della scheda che potete ottenere dalle informazioni del dispositivo:



Ci salviamo questo indirizzo facendo uno screenshot oppure appuntandocelo da qualche parte.

Scaricare il software Arduino

Ora ci serve il software per programmare la scheda ESP32. Andate a questo indirizzo

<https://www.arduino.cc/en/software> e scaricate la versione per il vostro sistema operativo. Io ho Windows quindi vi illustro il procedimento secondo Windows 11.

Installate il programma seguendo la procedura di installazione. Il software è già configurato per le schede Arduino ma a noi serve programmare la ESP32.

A questo punto per usare la nostra scheda ESP32 seguite alcuni semplici passi. Prima di impostare l'Arduino IDE, bisogna installare il driver per la comunicazione USB-seriale.

Collegate la scheda Arduino al PC e verificate se tra i dispositivi USB vedete correttamente la scheda ESP32 Dev Kit.

Se il driver non viene installato automaticamente, c'è una pagina di supporto che contiene i driver per Windows/Mac o Linux e si può scegliere a seconda di quale si usa. I driver possono essere scaricati dal seguente link [driver CP2102](#)

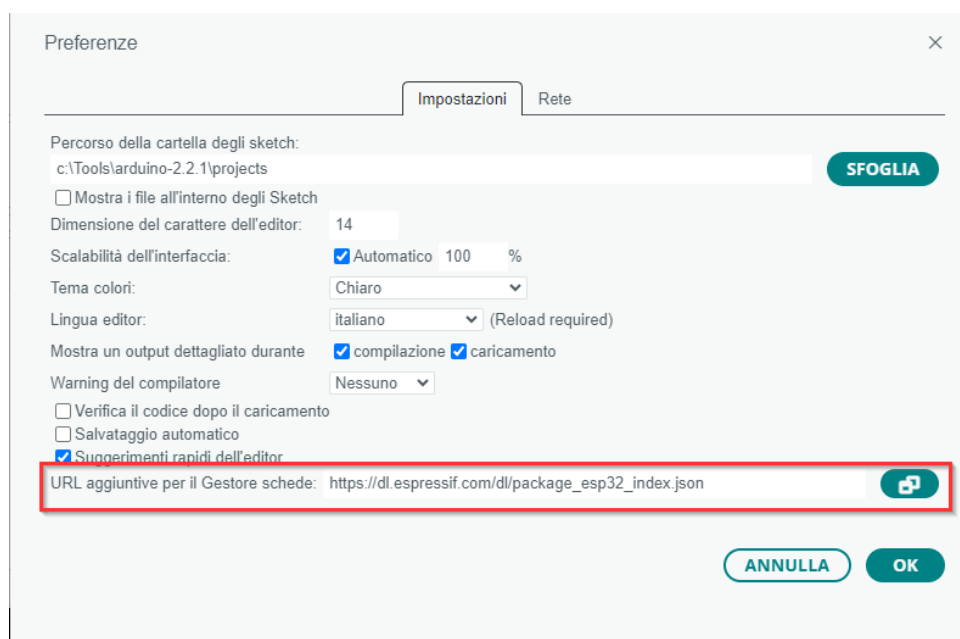
Scollegate la scheda, installate i driver e dopo provate a ricollegarla e vedere se compare tra i dispositivi.

A questo punto dobbiamo installare la scheda ESP32 dentro al software Arduino IDE.

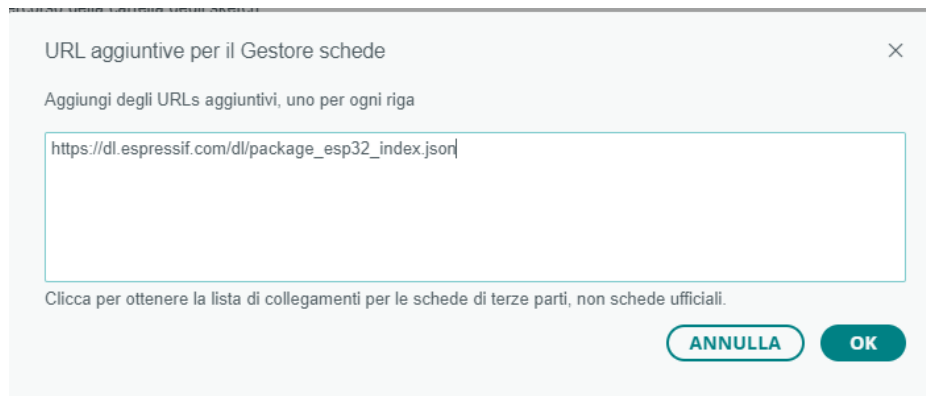
Aprire Arduino IDE e andare su: File > Preferenze, e trovare il campo URL aggiuntivi.

Poi copiare il seguente URL:

https://dl.espressif.com/dl/package_esp32_index.json

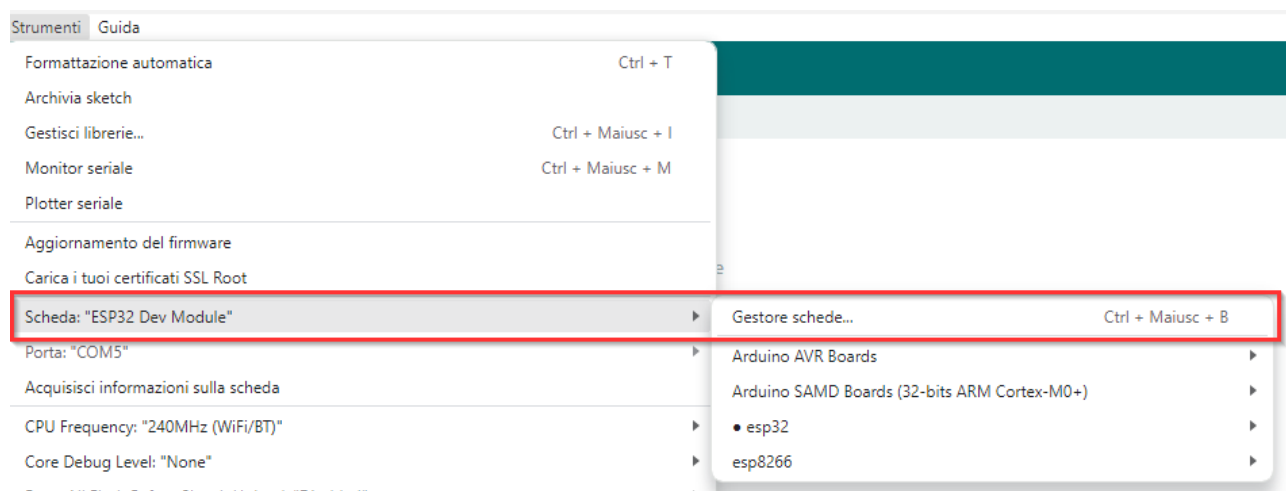


Incolla il link nel campo URL aggiuntivi. Se in questo campo ci sono uno o più link, basta aggiungere una virgola dopo l'ultimo link, incollare il nuovo link dopo la virgola e cliccare sul pulsante OK.

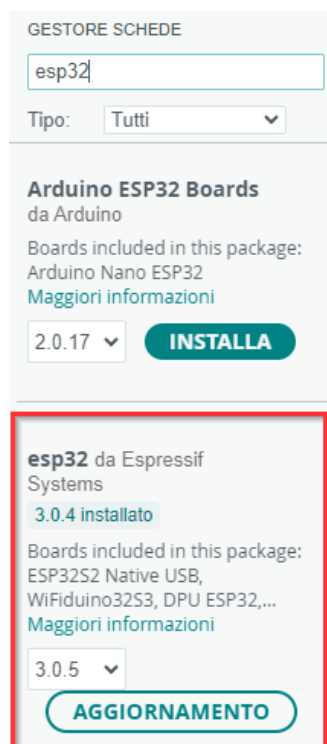


Aprire nuovamente l'Arduino IDE e andare su:

Strumenti > Scheda > Gestione Schede

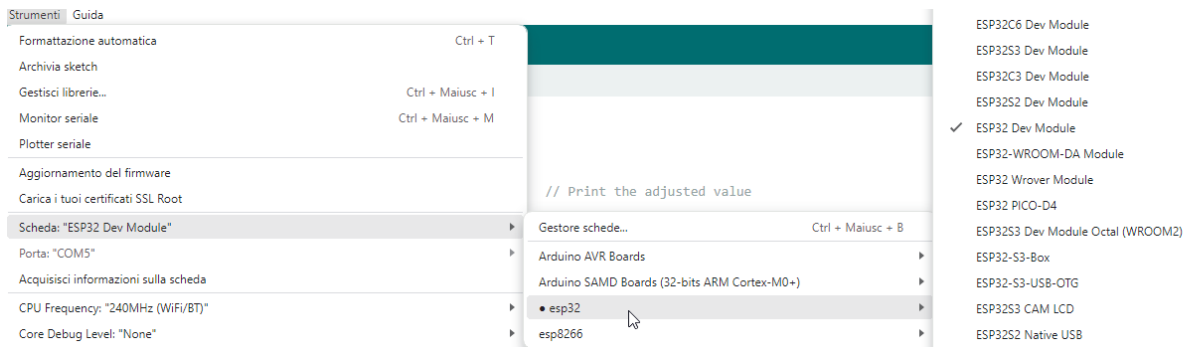


Quando si apre una nuova finestra, digitate *esp32* nella casella di ricerca e installate la scheda chiamata *esp32* prodotta da *Espressif Systems*, come mostrato nella seguente immagine:



Per selezionare la scheda ESP32, andare su:

Strumenti > Scheda > ESP32 Arduino > ESP32 Dev Module



Per caricare il codice dello sketch sulla scheda ESP32, selezionate prima la porta alla quale avete collegato la scheda. Andate a: **Strumenti > Porta > {nome porta}**

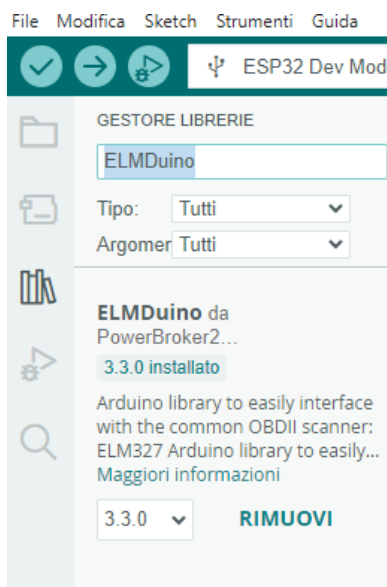
Se avete difficoltà potete seguire questo tutorial su YouTube

<https://www.youtube.com/watch?v=VDIzH0lxRKc>

Aggiungere le librerie

Per poter caricare correttamente il programma vi serviranno alcune librerie. Per installarle andate su **Strumento>Gestisci Librerie**

Ora inserite il testo **ELMDuino** e vi troverà la seguente libreria



Cliccate su "INSTALLA". Nell'immagine vedere "RIMUOVI" perché da me è già installata.

Caricare il programma (sketch)

Ora andate a questo [indirizzo](#) e scaricate il codice tramite il pulsante "download raw file"

bottosac / ESP32_ODBII_Scenic_DPFMode_LED Public

Notifications Fork 0 Star 0

Code Issues Pull requests Actions Projects Security Insights

main ESP32_ODBII_Scenic_DPFMode_LED / ESP32_ODBII_Scenic_DPFMode_LED.ino

bottosac Collegamento via mac address ee84411 · last week

125 lines (97 loc) · 3.18 KB

Code Blame Raw Copy Download Edit View

```
1 //06/09/2024: v1 lettura messaggio 22 2056 DPF Mode (1=normale, 5=rigenerazione, 7=attesa) e gestione tramite led
2
3 #include "BluetoothSerial.h"
4 #include "ELMDuino.h"
5
6 BluetoothSerial SerialBT;
7 #define ELM_PORT SerialBT
8 #define DEBUG_PORT Serial
9
10 ELM327 myELM327;
11
12 uint32_t rpm = 0;
13 bool dpfON = false;
14
15 int nb_query_state = SEND_COMMAND; // Set the initial query state ready to send a command
```

Cliccate sul file scaricato oppure nel ARDUINO IDE cliccate su File>Apri...

Ora dobbiamo modificare l'indirizzo della scheda OBD con quello che avete voi. Andate alla riga 24 del programma e modificate i numeri che vedete con quelli che vi siete copiati. Dovete inserire i numeri tra i ":"

```
// ***** PINS *****
#define GPIO_BT 19
#define GPIO_DPF 18

uint8_t address[6] = {0x00, 0x0D, 0x18, 0x3A, 0x67, 0x89};
//00:0D:18:3A:67:89 Vecchio OBDII

void setup()
{
```

A questo punto salvate il progetto dando il nome che preferite.

Ora collegate la scheda al PC e premete il tasto "Carica"

```
1 //06/09/2024: v1 lettura messaggio 22 2056 DPF Mode (1=normale, 5=rigenerazione, 7=attesa) e gesti
2
3 #include "BluetoothSerial.h"
4 #include "ELMdino.h"
5
6 BluetoothSerial SerialBT;
7 #define ELM_PORT SerialBT
8 #define DEBUG_PORT Serial
9
10 ELM327 myELM327;
11
12 uint32_t rpm = 0;
13 bool dpfON = false;
14
15 int nb_query_state = SEND_COMMAND; // Set the initial query state ready to send a command
16 int dpfstate = 0;
17
18 String msg;
19
20 // ***** PINS *****
21 #define GPIO_BT 19
22 #define GPIO_DPF 18
23
24 uint8_t address[6] = {0x00, 0x0D, 0x18, 0x3A, 0x67, 0x89};
25 //00:0D:18:3A:67:89 Vecchio OBDII
26
```

Il programma prima compilerà il file e se non ci sono errori lo caricherà sulla scheda. Alla fine vedrete un messaggio che ha completato il caricamento al 100%.

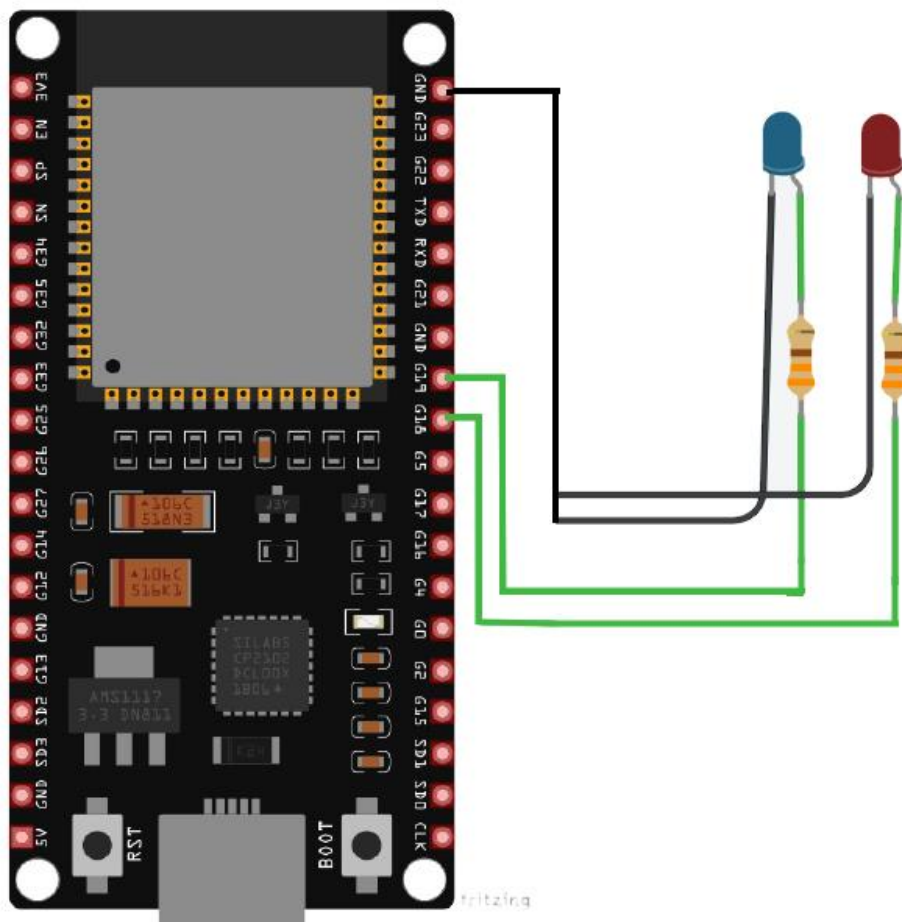
Saldare il led sulla scheda

Ora ci serve poter capire quando la scheda è collegata all'interfaccia OBD e quando sta comunicando correttamente.

Per fare ciò ci serve saldare due led ai pin 18 e 19 della scheda tramite le due resistenze da 330ohm

Qui sotto avete lo schema elettrico

Attenzione a rispettare la polarità del led (il pin più lungo è il positivo/anodo)



Funzionamento

Collegate la scheda alla presa USB accendisigari o a quella del bracciolo. All'avvio la scheda accenderà per 2" i led blu e rosso per un check di funzionamento poi li spegnerà.

Accendete l'auto, quando la scheda trova l'interfaccia OBD mostrerà alcuni lampeggi del led blu. Se il collegamento sarà attivo manderà un'altra serie di lampeggi e poi quando la comunicazione è attiva diventerà blu fisso. Dopo 30" per non disturbare il led blu si spegnerà ma la comunicazione è comunque attiva.

Ogni 2" la scheda richiederà alla centralina lo stato della rigenerazione. Durante la richiesta lampeggia brevemente il led rosso. Se il valore ricevuto è 1 il led rosso si spegne. Se il valore è diverso da 1 significa che l'auto non è in modalità normale e probabilmente sta facendo una rigenerazione (circa ogni 20km fa un rilascio dei Nox e quindi per alcuni secondi va in stato 5) e il led rosso rimarrà acceso fino a quando non tornerà a 1 o quando spegnete l'auto. Nel caso in cui il led rosso è acceso fisso è consigliabile non spegnere il motore per terminare correttamente la rigenerazione.

Nota: Se collegate la scheda al bracciolo quando spegnete l'auto rimarrà acceso il led rosso perché il messaggio CAN passerà allo stato 7. Se invece è collegato all'accendisigari allo spegnimento dell'auto si spegnerà la scheda.

Nella tabella vengono indicati gli stati dei led:

| Led | Stato | Funzionamento |
|---------|--------------|---|
| Led Blu | Acceso fisso | La scheda è collegata all'interfaccia OBD |
| | Lampeggio | La scheda ha riconosciuto l'interfaccia OBD e sta cercando di connettersi |

| | | |
|------------------|--------------|---|
| | Spento | Dopo una serie di lampeggi: collegamento non riuscito con l'interfaccia OBD Dopo 30" dalla accensione: collegamento riuscito con l'interfaccia ODB |
| Led Rosso | Acceso fisso | La scheda ha rilevato una rigenerazione in atto o uno stato non normale |
| | Lampeggio | La scheda sta richiedendo alla centralina il valore dello stato di rigenerazione |
| | Spento | Nessuna rigenerazione in atto o stato normale |

Contenitore

Se avete una stampante 3D o avete qualcuno che può stamparvela nel progetto su github trovate il progetto .stl da caricare in uno slicer tipo Cura. Il progetto è basato sulla scheda comprata da AZDelivery, per le schede di altri produttori potrebbero non andare bene le dimensioni.

[Link al file stl](#)

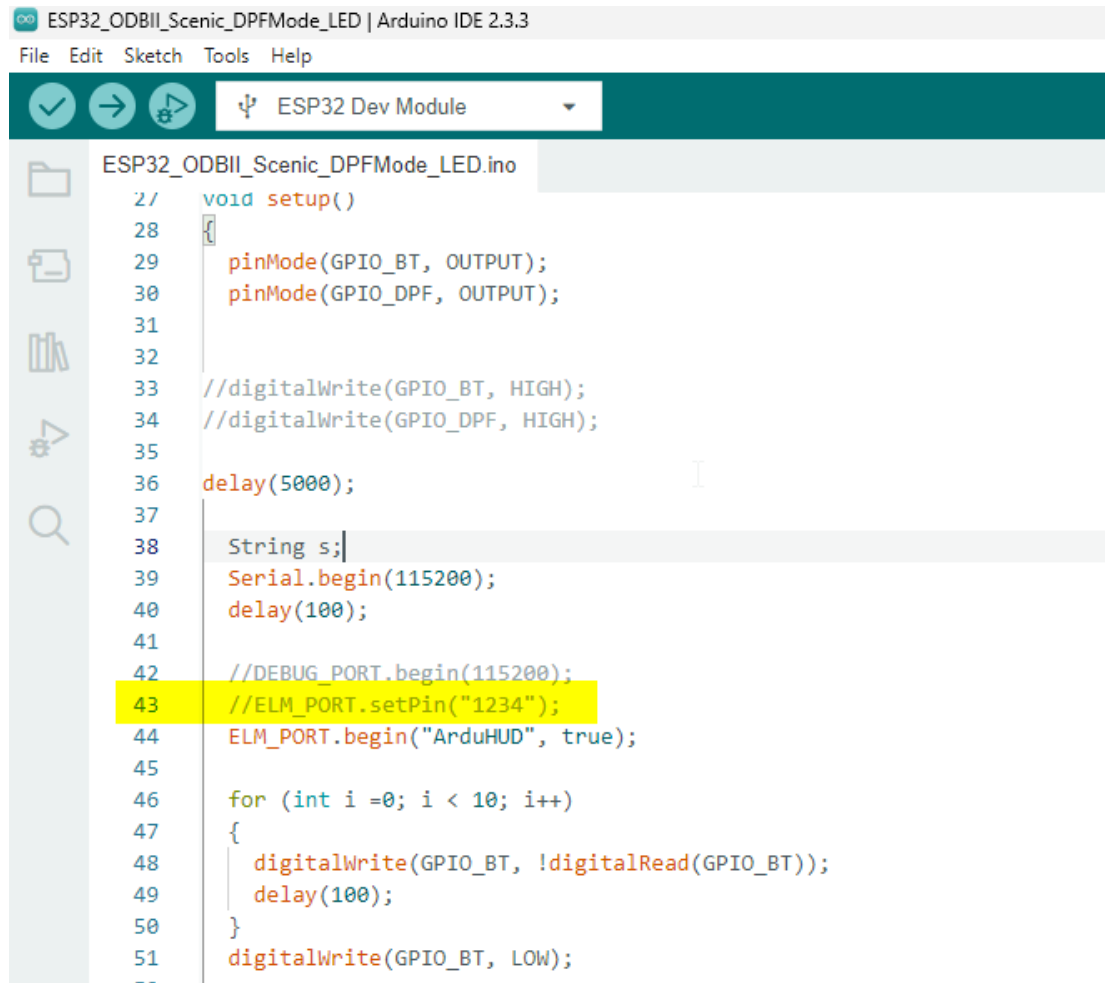
Problemi di connessione all'interfaccia OBD

Alcune interfacce OBD vogliono un pin per potersi collegare via bluetooth. In questo caso potrebbe succedere che al primo collegamento del ESP32 alla OBD venga richiesto l'inserimento del pin (di solito è 1234).

Ovviamente non possiamo inserirlo noi quindi è necessario predisporre l'ESP32 che invii il pin almeno la prima volta. Successivamente non sarà più necessario.

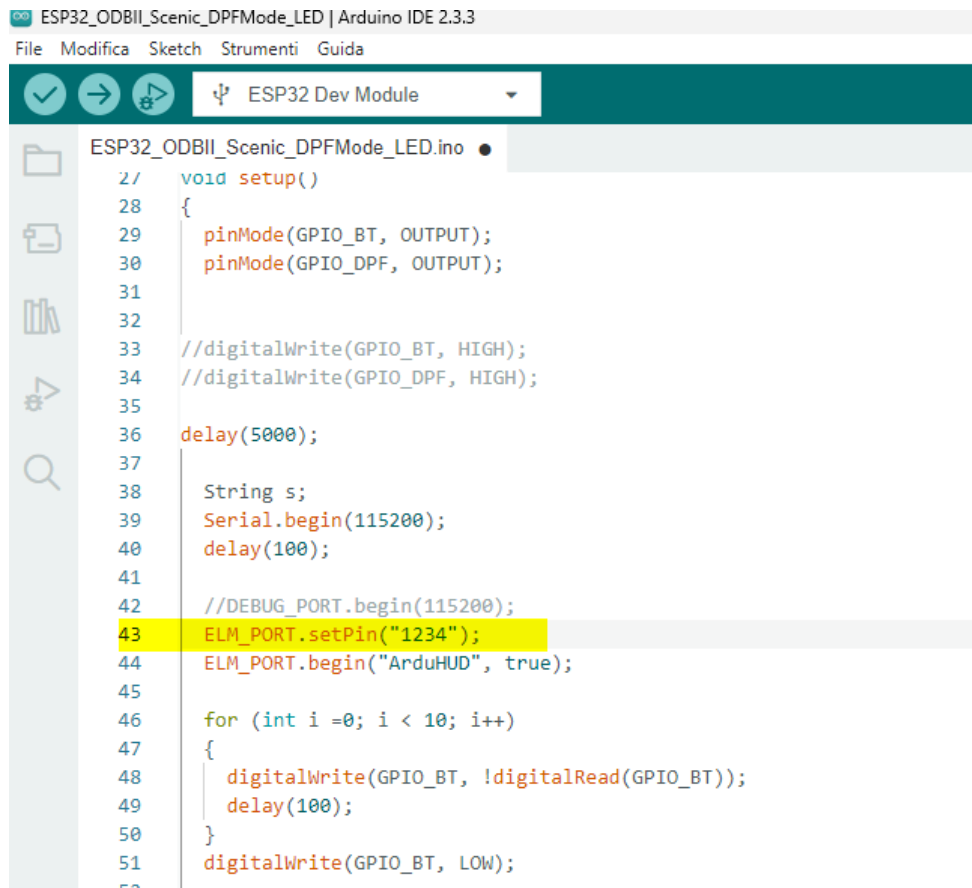
Per fare ciò dobbiamo modificare lo sketch del progetto e abilitare una riga del codice.

Apriamo l'Arduino IDE e apriamo il nostro progetto



```
ESP32_ODBII_Scenic_DPFMode_LED | Arduino IDE 2.3.3
File Edit Sketch Tools Help
ESP32 Dev Module
ESP32_ODBII_Scenic_DPFMode_LED.ino
27 void setup()
28 {
29   pinMode(GPIO_BT, OUTPUT);
30   pinMode(GPIO_DPF, OUTPUT);
31
32
33   //digitalWrite(GPIO_BT, HIGH);
34   //digitalWrite(GPIO_DPF, HIGH);
35
36   delay(5000);
37
38   String s;
39   Serial.begin(115200);
40   delay(100);
41
42   //DEBUG_PORT.begin(115200);
43   //ELM_PORT.setPin("1234");
44   ELM_PORT.begin("ArduHUD", true);
45
46   for (int i =0; i < 10; i++)
47   {
48     digitalWrite(GPIO_BT, !digitalRead(GPIO_BT));
49     delay(100);
50   }
51   digitalWrite(GPIO_BT, LOW);
52 }
```

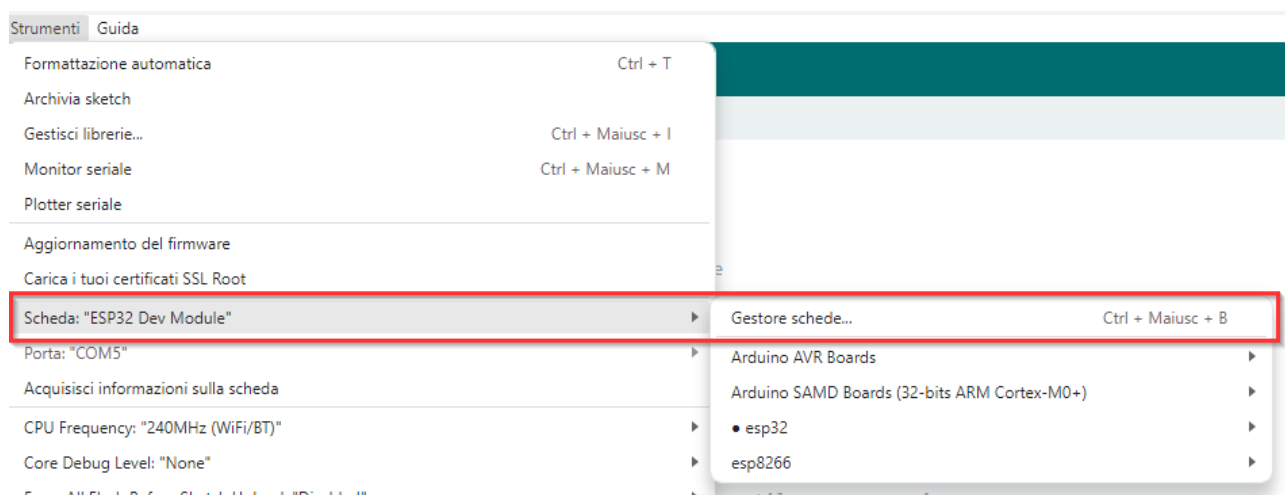
Andiamo alla riga 43 evidenziata e cancelliamo le due // in modo da abilitare l'istruzione.



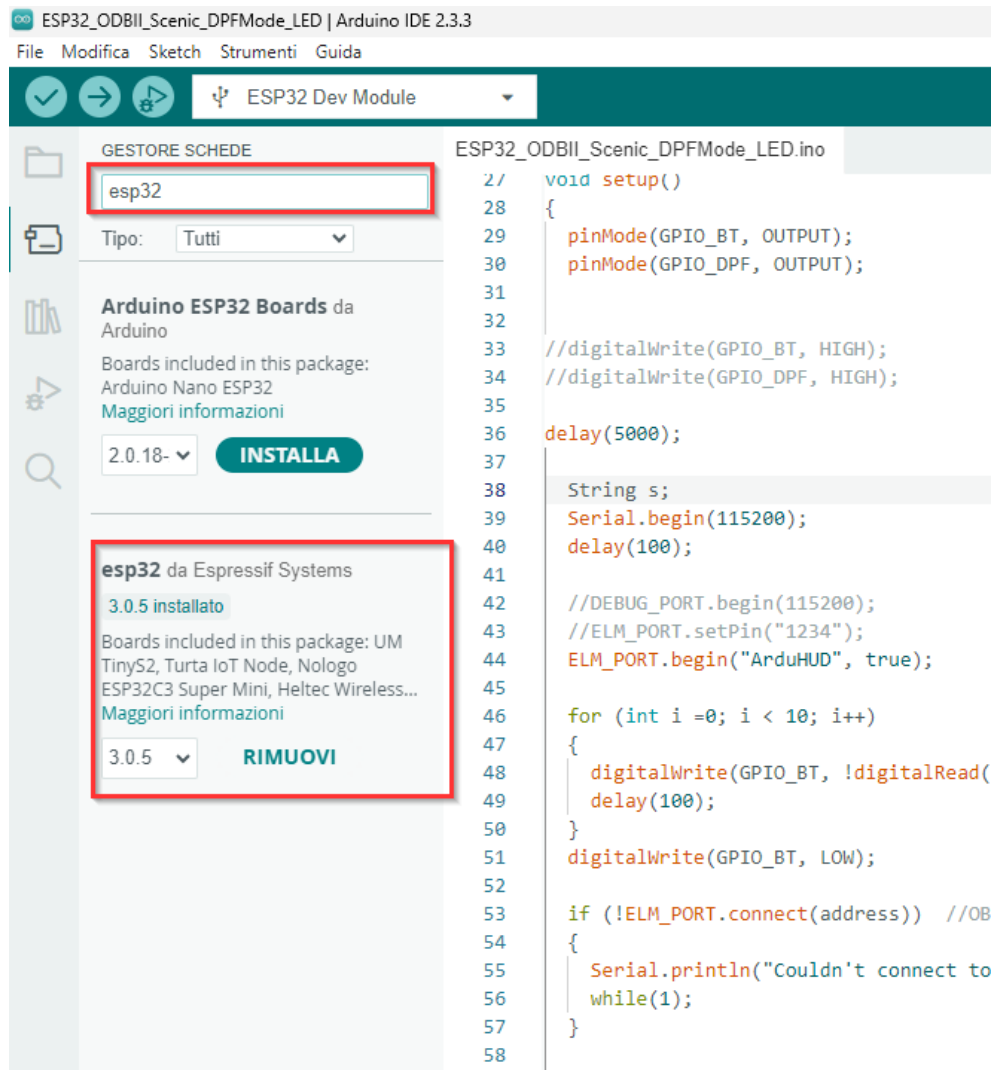
```
27 void setup()
28 {
29   pinMode(GPIO_BT, OUTPUT);
30   pinMode(GPIO_DPF, OUTPUT);
31
32
33   //digitalWrite(GPIO_BT, HIGH);
34   //digitalWrite(GPIO_DPF, HIGH);
35
36   delay(5000);
37
38   String s;
39   Serial.begin(115200);
40   delay(100);
41
42   //DEBUG_PORT.begin(115200);
43   ELM_PORT.setPin("1234");
44   ELM_PORT.begin("ArduHUD", true);
45
46   for (int i =0; i < 10; i++)
47   {
48     digitalWrite(GPIO_BT, !digitalRead(GPIO_BT));
49     delay(100);
50   }
51   digitalWrite(GPIO_BT, LOW);
52 }
```

Ora se proviamo a compilare il progetto per caricarlo otterremmo un errore perché l'istruzione setPin è stata rimossa nelle ultime versioni di Arduino.

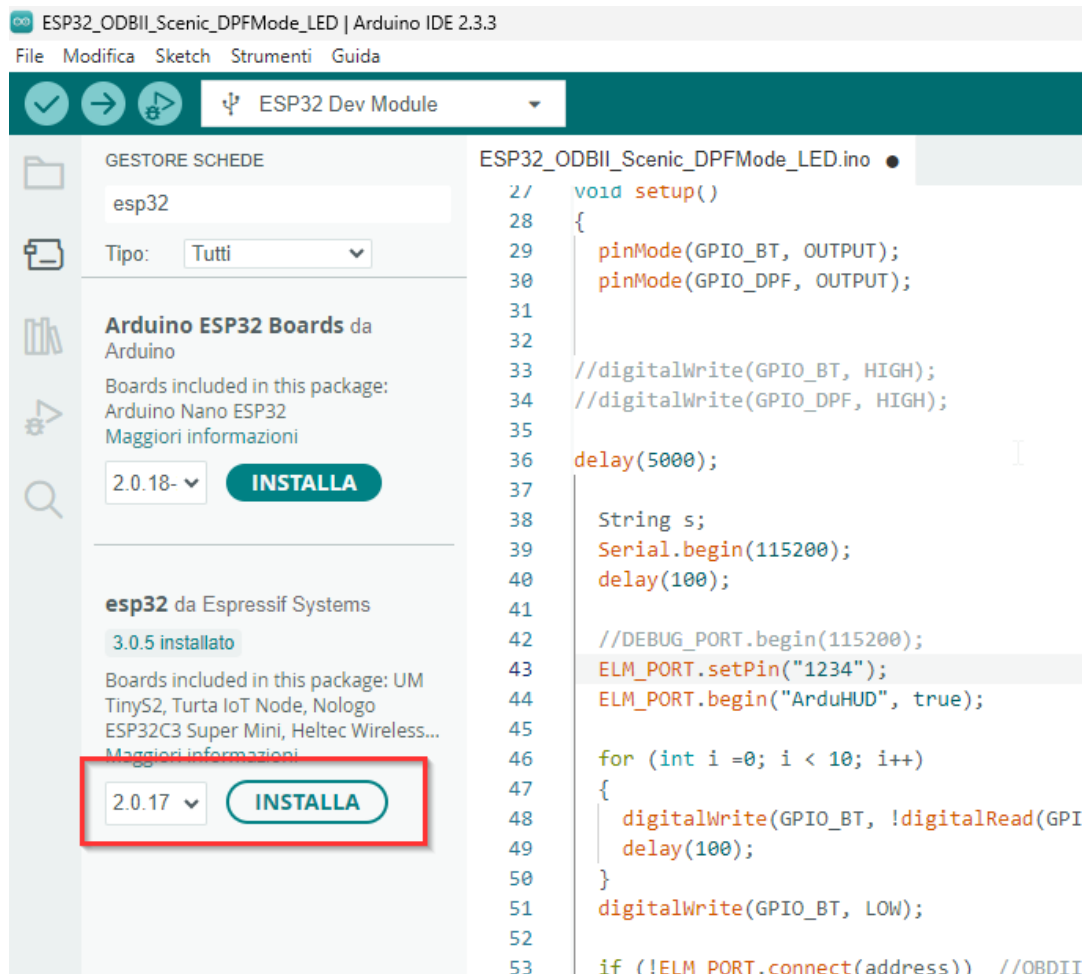
Per risolvere questo problema dobbiamo andare *Strumenti>Scheda: ESP32>Gestore schede*



Nel campo filtra scriviamo ESP32 e identifichiamo il box "esp32 da Espressif Systems"



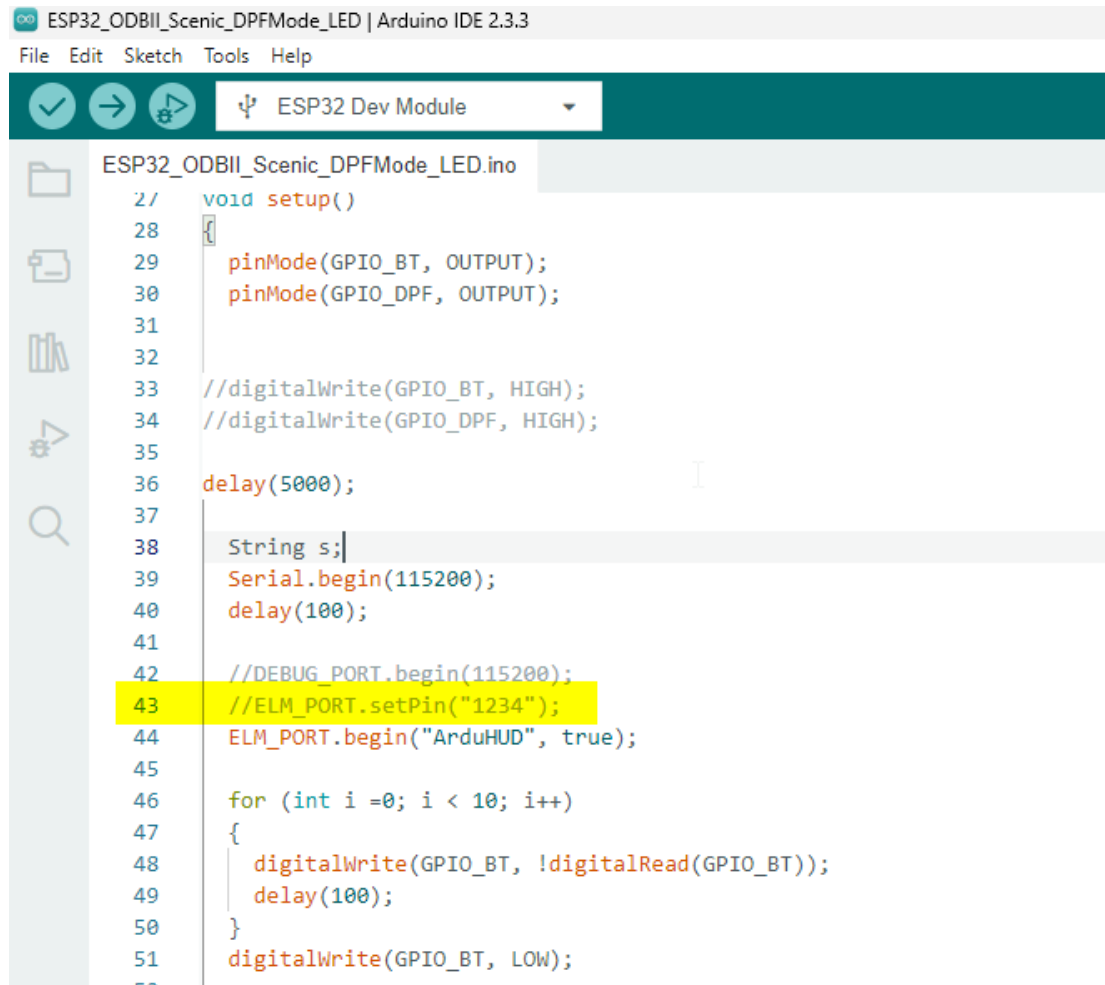
A questo punto dobbiamo scegliere la versione 2.0.17 e cliccare "INSTALLA"



Attendete che Arduino IDE installi la vecchia versione delle librerie. Quando ha terminato potete connettere la scheda, cliccare nuovamente il bottone in alto per caricare lo sketch.



Ora provate a connettere la scheda ESP32 al OBD e dovreste potervi connettere. A questo punto se la connessione viene stabilita correttamente non sarà più necessario inserire il pin quindi potete tornare all'ultima versione della scheda "esp32 da Espressif Systems" che attualmente è la 3.0.5 e commentare nuovamente la riga setPin



```
ESP32_ODBII_Scenic_DPFMode_LED | Arduino IDE 2.3.3
File Edit Sketch Tools Help
ESP32 Dev Module
ESP32_ODBII_Scenic_DPFMode_LED.ino
27 void setup()
28 {
29   pinMode(GPIO_BT, OUTPUT);
30   pinMode(GPIO_DPF, OUTPUT);
31
32
33   //digitalWrite(GPIO_BT, HIGH);
34   //digitalWrite(GPIO_DPF, HIGH);
35
36   delay(5000);
37
38   String s;
39   Serial.begin(115200);
40   delay(100);
41
42   //DEBUG_PORT.begin(115200);
43   //ELM_PORT.setPin("1234");
44   ELM_PORT.begin("ArduHUD", true);
45
46   for (int i =0; i < 10; i++)
47   {
48     digitalWrite(GPIO_BT, !digitalRead(GPIO_BT));
49     delay(100);
50   }
51   digitalWrite(GPIO_BT, LOW);
```