

---

---

ELABORATO ASM  
ARCHITETTURA DEGLI ELABORATORI

---

---

AUTORI

ANGELICO ENRICO: VR476760  
PORTARO RICCARDO: VR471482

*Università degli studi di Verona*

MAGGIO 2022

# Indice

<b>1</b>	<b>Descrizione del progetto</b>	<b>2</b>
<b>2</b>	<b>Descrizione variabili</b>	<b>3</b>
<b>3</b>	<b>Modalità di passaggio/restituzione dei valori delle funzioni</b>	<b>4</b>
<b>4</b>	<b>Diagramma di flusso</b>	<b>5</b>
4.1	Funzione Trova pilota . . . . .	5
4.2	Funzione Confronto ID . . . . .	6
4.3	Funzione WtoFile . . . . .	6
4.4	Funzione Stampa . . . . .	7
4.5	Telemetry . . . . .	8
<b>5</b>	<b>Scelte progettuali</b>	<b>9</b>

# 1 Descrizione del progetto

Sviluppare un programma che simuli il sistema di telemetria del videogame F1. Il sistema riceve in input i dati del pilota in ogni istante di tempo nel seguente formato:

$\langle \text{tempo} \rangle, \langle \text{id\_pilota} \rangle, \langle \text{velocità} \rangle, \langle \text{rpm} \rangle, \langle \text{temperatura} \rangle$

Dove:

- **Tempo:** indica l'istante di tempo in cui vengono registrati i valori;
- **Id\_pilota:** rappresenta un valore numerico che determina univocamente un pilota della lista seguente all'elenco puntato;
- **Velocità:** indica la velocità del pilota in quel determinato istante di tempo;
- **Rpm:** indica i giri per minuto nell'istante di tempo registrato;
- **Temperatura:** indica la temperatura del motore al tempo registrato;

ID	Nome
0	Pierre Gasly
1	Charles Leclerc
2	Max Verstappen
3	Lando Norris
4	Sebastian Vettel
5	Daniel Ricciardo
6	Lance Stroll
7	Carlos Sainz
8	Antonio Giovinazzi
9	Kevin Magnussen
10	Alexander Albon
11	Nicholas Latifi
12	Lewis Hamilton
13	Romain Grosjean
14	George Russell
15	Sergio Perez
16	Daniil Kvyat
17	Kimi Raikkonen
18	Esteban Ocon
19	Valtteri Bottas

Il programma dovrà restituire in output i dati relativi al solo pilota selezionato in ogni istante di tempo che dovranno rispettare le seguenti soglie:

- **Giri motore:**
  - LOW:  $rpm \leq 5000$
  - MEDIUM:  $5000 < rpm \leq 10000$
  - HIGH:  $rpm > 10000$
- **Temperatura:**
  - LOW:  $temp \leq 90$
  - MEDIUM:  $90 < temp \leq 110$
  - HIGH:  $temp > 110$

- **Velocità:**
  - LOW:  $speed \leq 100$
  - MEDIUM:  $100 < speed \leq 250$
  - HIGH:  $speed > 250$

Il file di output dovrà riportare le soglie per ogni istante di tempo in cui il pilota viene monitorato e dovrà seguire la seguente struttura:

<tempo>,<livello\_rpm>,<livello\_temperatura>,<livello\_velocità>

Inoltre alla fine del file di output sarà presente una nuova riga dove verranno indicati i dati medi di ogni livello durante tutti gli istanti registrati secondo la seguente struttura:

<rpm max>,<temp max>,<velocità max>,<velocità media>

Nel caso in cui ad inizio file è presente un id\_pilota non corretto verrà stampato nel file di output : "Invalid".

## 2 Descrizione variabili

Nel file **telemetry.s** sono state utilizzate le seguenti variabili:

- **pilot\_idpilota\_str**: stringa contenente i nomi di ogni pilota al variare dell'id in correlazione alla tabella precedente;
- **invalid\_pilot\_str**: stringa usata per gestire il caso in cui l'id pilota non è nella tabella sopracitata;
- **count\_char**: contatore della posizione per la stringa di output;
- **count**: variabile utilizzata per due scopi:
  1. scorre tutti gli id dei piloti per verificare la corrispondenza tra il nome del pilota inserito nella stringa di input e quelli delle variabili (pilot\_idpilota\_str);
  2. conta le virgole per sapere a che punto della riga del file di input ci troviamo, utile per assegnare i caratteri letti nelle variabili di monitoraggio del pilota;
- **freq\_pilota**: utilizzata per contare quante volte il pilota con quel determinato id è stato registrato nella telemetria. Variabile utile per calcolare la velocità media a fine programma;
- **flag**: utilizzata per terminare il programma nel caso in cui non è stata trovata nessun riscontro tra l'id in input e gli id pilota presenti nella tabella;
- **id**: salva l'id del pilota inizialmente inizializzato a -1, valore non presente all'interno della tabella, per richiamare la funzione trova\_pilota;
- **virgola**: usata per scrivere nel file di output la virgola di separazione tra i dati del pilota;
- **invio**: usata per andare a capo nel file di output a seguito di ogni istante di tempo;
- **velMax**: variabile per memorizzare la velocità massima che andrà poi scritta nell'ultima riga del file di output
- **velMedia**: variabile per memorizzare la velocità media, questa verrà sommata con tutte le velocità del pilota registrate e verrà successivamente divisa per la freq\_pilota
- **rpmMax**: variabile che memorizza i giri per minuto massimi registrati dal pilota;
- **temperaturaMax**: variabile che memorizza la temperatura massima del motore registrati sull'auto del pilota;
- **flagid**: viene posta a zero se l'id letto non corrisponde a quello del pilota cercato. In tal caso la flag salta la fase di registrazione dei valori del pilota;
- **IntToStr\_Len**: variabile che tiene conto della lunghezza di "intToStr" e usata nella funzione itoa per salvare ogni carattere nella variabile "intToStr";

- **rpm\_long**: memorizza i giri del motore corrente del pilota;
- **temperatura\_long**: memorizza la temperatura del motore corrente del pilota;
- **vel\_long**: memorizza la velocità corrente del pilota;
- **id\_long**: memorizza l'id corrente del pilota
- **LOW, MEDIUM, HIGH**: stringhe usate per stampare in output le soglie dei giri del motore, temperatura e velocità;
- **intToStr**: variabile utilizzata per salvare il numero trasformato in stringa / carattere
- **tempo\_len**: memorizza la lunghezza della stringa tempo;
- **tempo\_str**: stringa del tempo registrato;

### 3 Modalità di passaggio/restituzione dei valori delle funzioni

Di seguito elenchiamo tutte le funzioni utilizzate nel file:

- **Trova\_pilota**: La funzione scorre la prima riga fino al raggiungimento del ritorno di carrello confrontando la stringa letta con le variabili "*pilot\_idpilot\_str*", salvando in "*id*" il numero esatto del pilota, altrimenti scrive sul file di output "*invalid*", attraverso la funzione "*WtoFile*" e in tal caso imposta la variabile "*flag*" a zero (utilizzata nel main per terminare il programma);
- **Confronto\_id**: confronta l'id del pilota di riferimento con l'id trovato nella riga delle telemetrie, se l'id confrontato è uguale porremo il "*flag\_id*" a uno (questo permetterà successivamente di andare avanti con l'operazione di scansione dei vari parametri), altrimenti verrà impostato a zero e "salteremo" al ritorno di carrello della riga corrente. In questo caso verrà resettato "*tempo\_len*", "*id\_long*", e impostata la variabile "*count*" a zero
- **Stampa**: questa funzione setta per tempo, rpm, temperatura e velocità in *ecx* l'indirizzo della stringa, in *edx* la sua lunghezza e richiama la funzione "*WtoFile*" che scriverà le informazioni in output. Inoltre questa funzione verifica i parametri finali richiesti(rpm massimi, velocità massima, temperatura massima e velocità media);
- **WtoFile**: prende in input i valori dei registri *ecx*, *edx*, *edi*; scrivendo in *edi*, attraverso un ciclo, ogni carattere della stringa contenuta in *ecx* (della lunghezza di *edx*) tenendo conto attraverso la variabile "*count\_char*" in quale posizione della stringa di output ci troviamo;
- **itoa**: la funzione converte un intero passatogli dal registro *EAX* restituendo una stringa che viene inserito nella variabile "*IntToString*" e la propria lunghezza nella variabile "*IntToStr\_len*".

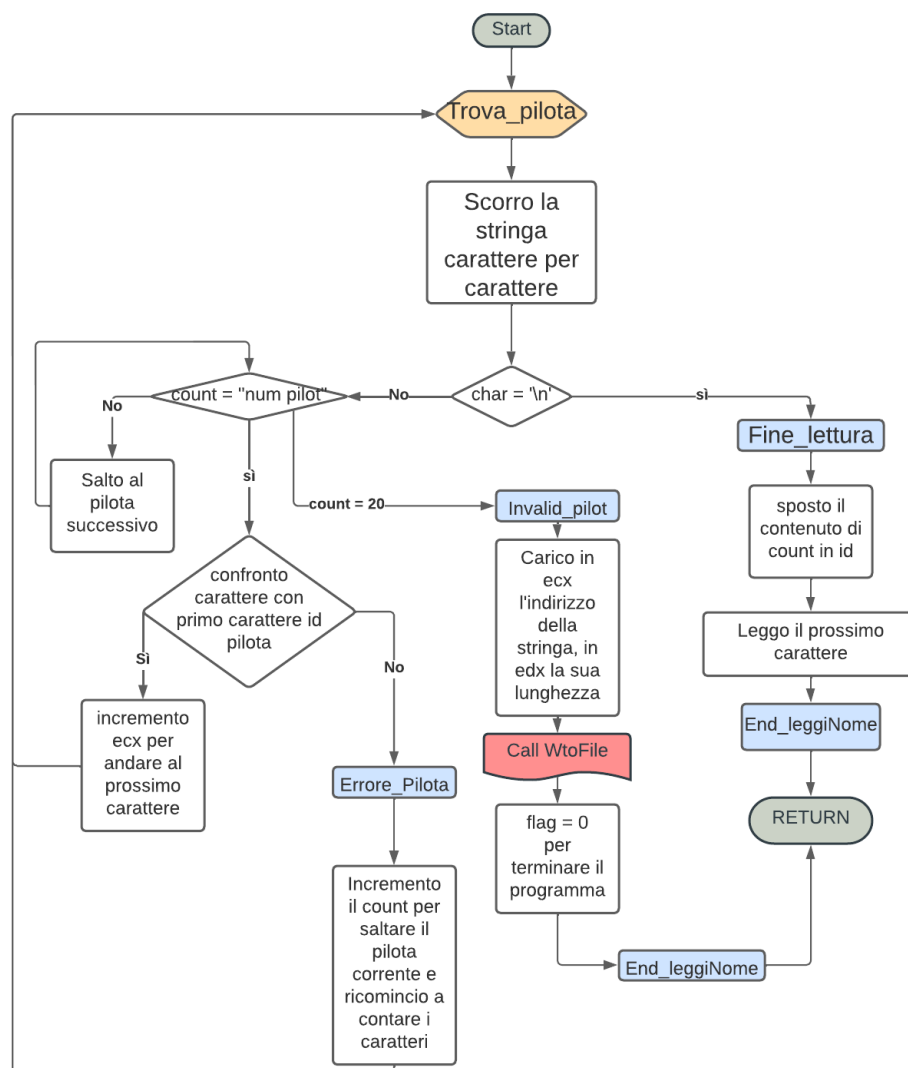
## 4 Diagramma di flusso

Il diagramma di flusso segue la seguente legenda:

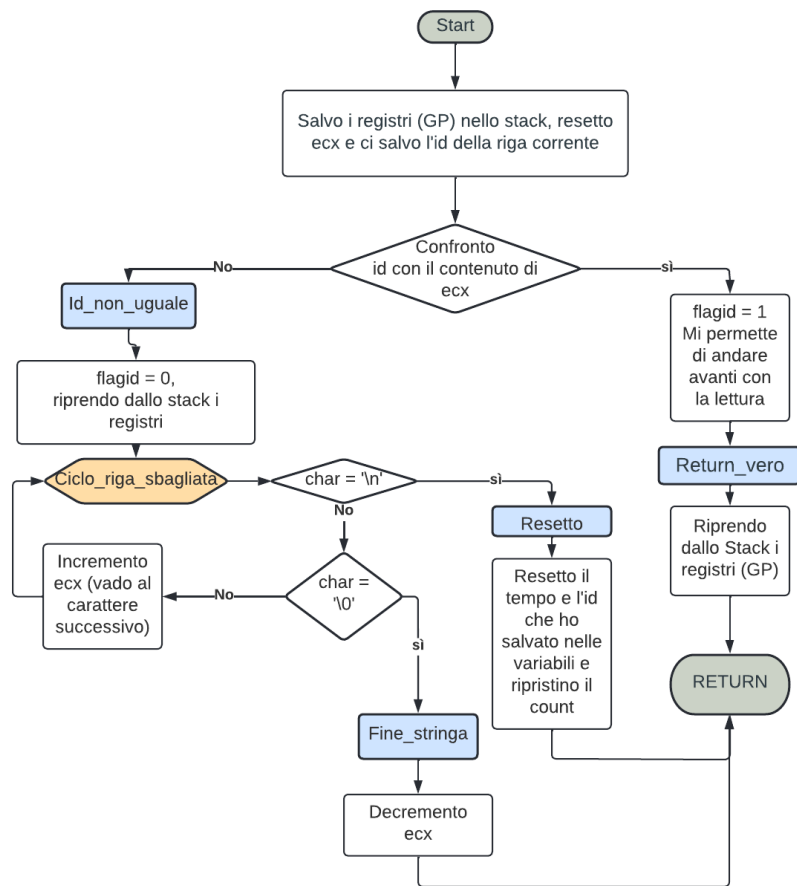


Riportate di seguito il Diagramma di flusso delle funzioni utilizzate e della principale funzione "Telemetry"

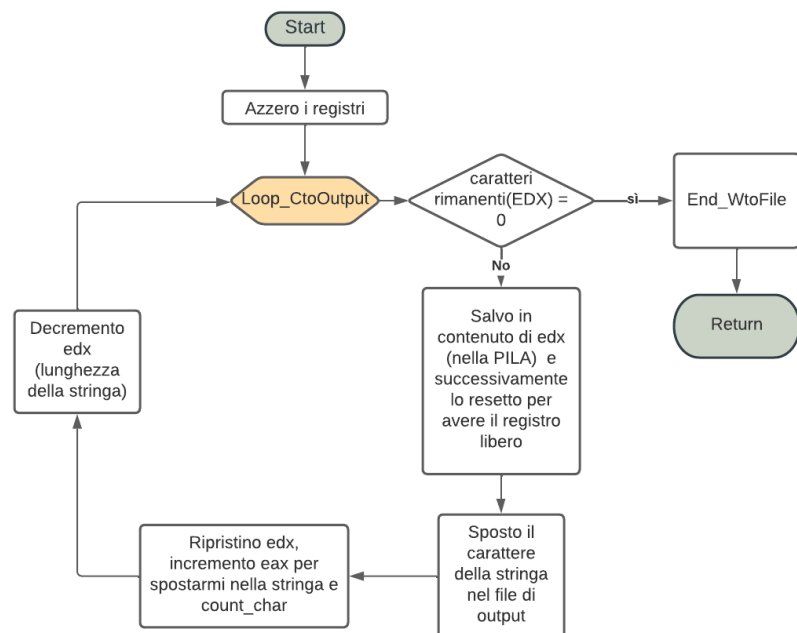
### 4.1 Funzione Trova pilota



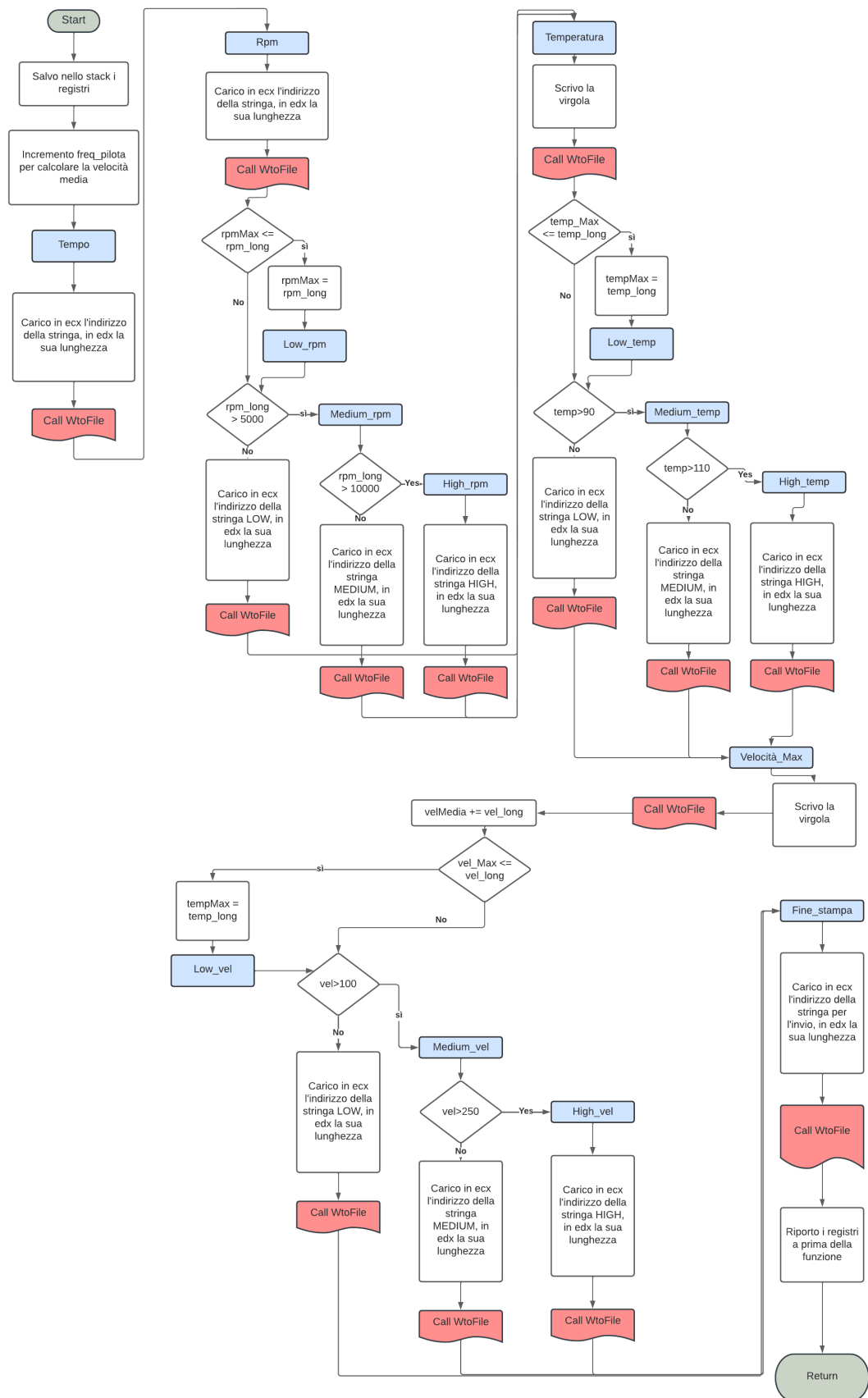
## 4.2 Funzione Confronto ID



## 4.3 Funzione WtoFile

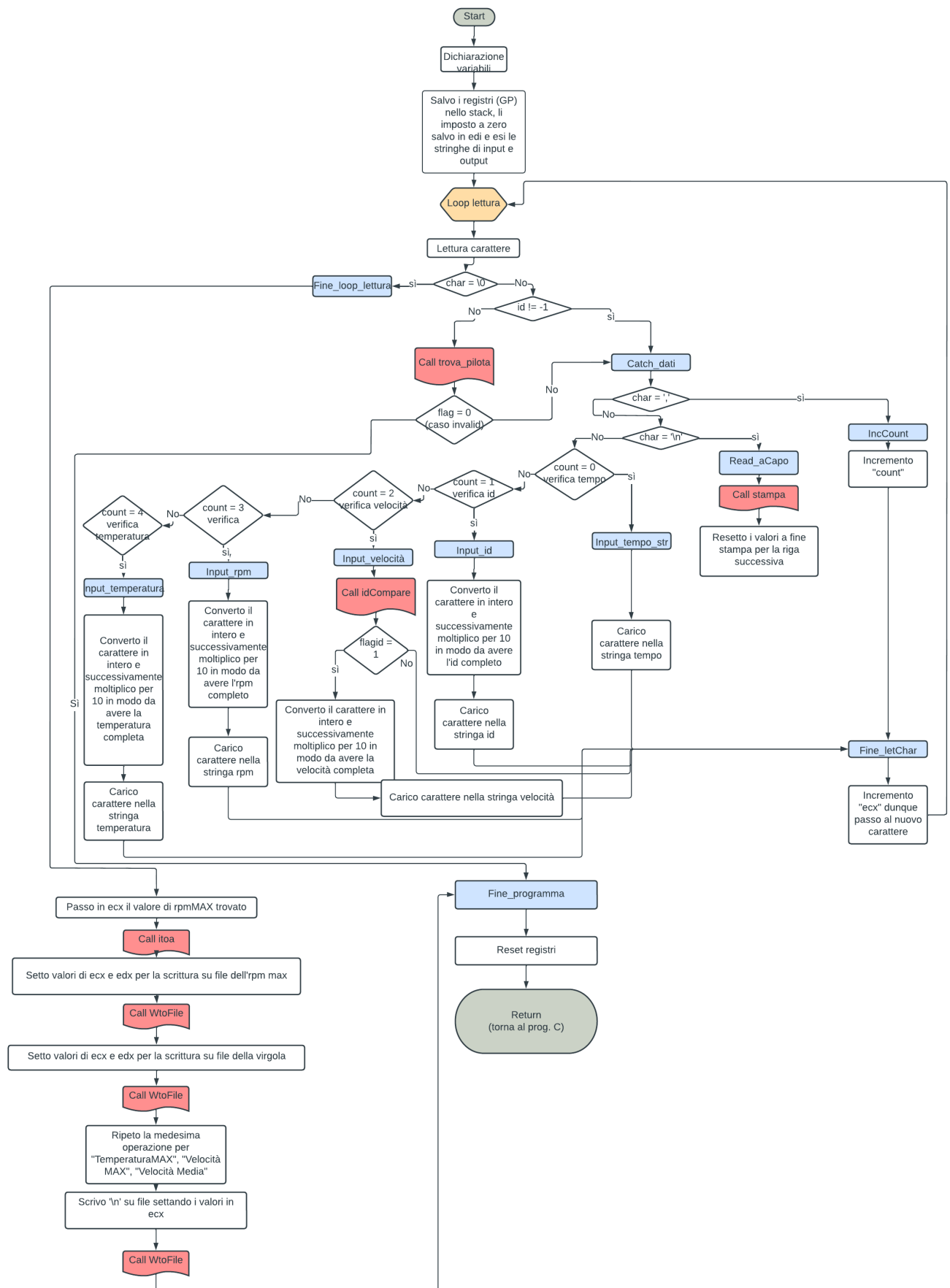


## 4.4 Funzione Stampa





## 4.5 Telemetry



## 5 Scelte progettuali

Durante la fase di scrittura del progetto abbiamo fatto le seguenti scelte progettuali:

- Nella fase di lettura della riga abbiamo preferito procedere con la comparazione dell'id pilota con l'id di riferimento prima del completamento della riga, per rendere più ottimizzato il nostro programma, ovvero non salvare nelle variabili di riferimento valori che non avremmo usato ma saltare direttamente alla telemetria successiva qualora l'id fosse errato senza perdere ulteriore tempo;
- Un'altra implementazione è stata fatta durante l'acquisizione delle variabili id, velocità, rpm e temperatura. Tale implementazione consente di memorizzare le variabili come interi senza l'uso di variabili ausiliarie di tipo stringa e di una funzione di trasformazione da stringa a intero, questo avviene tramite la lettura del carattere a cui verrà sottratto "48" (primo carattere numerico della tabella ascii) e il risultato andrà sommato al registro EAX (contenente l'indirizzo della variabile) il quale ad ogni ciclo verrà moltiplicato \*10 ( $EAX * EBX$   $EBX=10$ ) per lasciare spazio al carattere;
- Per sapere quando verificare l'id e quale parametro stessimo leggendo abbiamo deciso di utilizzare la variabile "*count*" che andrà a contare le virgole. In questo modo leggeremo prima l'id con il valore del count delle virgole ad 1, il programma verifica se l'id corrisponde al pilota dato in input e nel caso continuerà successivamente ad incrementare le virgole e leggere parametri diversi. Ad ogni riga il count verrà resettato e il conteggio ripartirà.