

ALMA MATER STUDIORUM - UNIVERSITA' DI BOLOGNA

SCUOLA DI INGEGNERIA E ARCHITETTURA - Sede di Forlì

CORSO DI LAUREA MAGISTRALE IN INGEGNERIA MECCANICA  
(Classe: LM-33 Codice: 8771)

TESI DI LAUREA  
in  
Metodi di progetto per l'ingegneria industriale

***Progettazione, ottimizzazione e sviluppo di un sistema  
WOPI (Water Out Pressure In) per airbox***

CANDIDATO  
Alessandro Cicognani

RELATORE  
Prof.ssa Francesca De Crescenzo

CORRELATORE  
Prof.ssa Beatrice Pulvirenti

Anno Accademico 2017/2018







# Indice

<b>Capitolo 1. Premessa</b>	pag.	1
<b>Capitolo 2. Introduzione e metodo di sviluppo del progetto</b>	pag.	3
2.1 Mission Statement	pag.	3
2.2 Pianificazione	pag.	4
2.3 Progettazione concettuale	pag.	5
2.4 Progettazione del sistema	pag.	12
<b>Capitolo 3. Simulazioni fluidodinamiche</b>	pag.	19
3.1 Il disegno 3D	pag.	19
3.2 La mesh	pag.	22
3.3 Simulazioni	pag.	23
3.3.1 Risultati per modello senza tubo di flusso	pag.	28
3.3.2 Risultati per modello con tubo di flusso	pag.	38
3.3.3 Analisi dei risultati	pag.	48
3.3.4 Grid sensitivity	pag.	53
<b>Capitolo 4. ECU e WOPI</b>	pag.	57
4.1 Software	pag.	57
4.2 Hardware	pag.	59
4.2.1 Schema dello shield del master (ECU)	pag.	61
4.2.2 Schema dello shield dello slave (WOPI)	pag.	66
4.2.3 Valvola	pag.	71

<b>Capitolo 5. Conclusioni</b>	pag.	85
<b>Bibliografia e Ringraziamenti.</b>	pag.	87
<b>Allegato. Codice Arduino</b>	pag.	91

# 1. Premessa

Il seguente elaborato finale è stato sviluppato in collaborazione e presso la BMC s.r.l, azienda leader nel settore dello sviluppo e produzione di filtri ed airbox per il settore automotive, durante un periodo di tirocinio per tesi. Esso tratta la progettazione, l'ottimizzazione e lo sviluppo di un modello che utilizzerà il sistema WOPI (Water Out Pressure In, brevetto del dott. Gaetano Bergami, titolare della BMC) da applicare ad un airbox. In un motore a combustione interna, affinché la combustione sviluppi il maggior quantitativo di coppia possibile, uno degli aspetti da ottimizzare riguarda il quantitativo di aria in ingresso, la sua pressione e la velocità con cui questa entra in camera di combustione.

Per aumentare la portata di aria massima aspirabile dal motore a combustione interna, la presa d'aria dell'airbox viene posta in corrispondenza della sezione frontale dell'automobile. In questa configurazione, la pressione dell'aria, in ingresso al box, permette di generare una sovrappressione tanto maggiore quanto è maggiore la velocità del veicolo. Questo permette di riempire maggiormente i cilindri del motore, incrementandone il rendimento volumetrico e quindi la potenza che sarà in grado di generare.

Un airbox, in definitiva, deve fornire al motore sia aria pura (ovvero priva di particelle estranee, motivo per cui al suo interno viene collocato un elemento filtrante) che priva di gocce d'acqua (possibile presenza all'interno sia a causa della condensazione dell'umidità dell'aria aspirata, sia, in volume anche maggiore, in caso di pioggia). Questo secondo aspetto sarà quello su cui si focalizzerà questo elaborato.

Attualmente, per eliminare la presenza di queste particelle d'acqua, vengono praticati dei fori, più o meno grandi, sulla parete dell'airbox. Questa semplice tecnica, permette di eliminarla, ma causa anche una depressurizzazione.

Per contrastare questo fenomeno, è già noto l'utilizzo di condotti elastici, normalmente chiusi o semichiusi che, nel frattempo, permettono la lenta discesa gravitazionale dell'acqua.

Questo sistema, però, non è in grado di sigillare il foro di scolo, provocando quindi anch'esso un calo della pressurizzazione del box. Inoltre, potrebbe intasarsi, rendendo inefficace lo scolo. In più, nel tempo potrebbe sopraggiungere una perdita di elasticità, causando un peggioramento della propria funzionalità.

Scopo del WOPI è quello di eliminare tutti i precedenti inconvenienti nel modo più efficiente possibile. Il sistema, come si vedrà alla fine del prossimo capitolo, verrà concepito composto da una centralina secondaria comunicante con quella principale del veicolo che, grazie al segnale dei potenziometri dell'acceleratore e dell'interruttore del tergicristallo (o del sensore pioggia, nel caso di applicazioni più lussuose), comanderà l'apertura di una valvola normalmente chiusa posta lateralmente all'airbox. In questo modo, si potrà scaricare l'acqua presente al suo interno quando essa sarà presente, mentre sigillerà lo scolo in sua assenza, evitando questo già citato calo di pressurizzazione quando si procede col veicolo.

Nel capitolo seguente verrà quindi dapprima descritto il processo di sviluppo concettuale del WOPI e di simulazione della centralina del veicolo (ECU, Engine Control Unit).

Successivamente, nel terzo capitolo, si presenta l'attività di verifica della variazione fluidodinamica delle prestazioni, legata all'introduzione del sistema WOPI in specifiche condizioni di marcia del veicolo tramite CFD (Computational Fluid Dynamics).

Infine nel capitolo 4, si approfondisce il lavoro svolto sui componenti hardware e software relativi allo sviluppo e l'automazione del sistema. Verrà quindi implementata una centralina secondaria al modello, realizzata utilizzando il software di compilazione "Arduino" (versione 1.8.5) e componenti hardware (relè, interruttori, ecc.) che verranno collegati ad una scheda microcontrollore compatibile con la Scheda "Arduino Uno R3", realizzando così il circuito che comanderà l'apertura della valvola in funzione dei segnali che riceverà dalla centralina primaria. Verrà inoltre scelta una valvola con caratteristiche di funzionamento idonee a questo progetto, che dovrà avere dimensioni ridotte, poiché si ipotizza di utilizzare questo sistema anche nell'ambito racing, dove non sempre si ha disponibilità sufficienti di spazio sotto il cofano, dove sono già presenti molti elementi che spesso non possono essere posizionati altrove.

## 2. Introduzione e metodo di sviluppo del progetto

L'obiettivo del progetto, come introdotto nella premessa, riguarda lo sviluppo di un sistema in grado di eliminare l'umidità ed eventuale pioggia dall'airbox, senza però avere perdite di pressione al suo interno che limiterebbero le prestazioni del veicolo (eliminando il bisogno dei classici fori di scarico normalmente presenti). Vediamo di seguito il processo comparativo seguito per confrontare le due configurazioni (classica e innovativa) dell'airbox che ha portato alla soluzione che mostreremo nei capitoli successivi e allo sviluppo di un modello che ne replichi le funzionalità.

Tale processo è ottenuto adattando i metodi di processo di sviluppo prodotto proposti da Ulrich Eppinger al caso specifico al fine di adottare un approccio sistematico per le scelte progettuali. Le diverse fasi sono descritte nel dettaglio nei paragrafi seguenti.

### 2.1 Mission Statement

Il mission statement o dichiarazione di intenti è uno schema atto a riassumere i punti principali della proposta di sviluppo prodotto. Nel caso del sistema di scarico dell'airbox è utile ad avere una visione immediata dell'obiettivo progettuale e degli aspetti aziendali e di mercato di principale importanza.

#### **Breve descrizione del prodotto:**

Sistema di scarico dell'umidità/acqua all'interno dell'airbox.

#### **Obiettivi principali:**

Creare un prodotto che sia il più possibile performante per avere minor caduta di pressione tra ingresso ed uscita dell'airbox.

#### **Mercato obiettivo del prodotto:**

Produttori di auto e moto da competizione e di lusso (primario);

Produttori di auto e moto da strada e utilitarie (secondario).

#### **Ipotesi e vincoli di guida per l'attività di sviluppo:**

Dimensioni limite;

Minimizzazione del costo fabbricazione;

Minimizzazione del numero di componenti del prodotto.

**Interessati, stakeholder:**

Produttori di materie prime, industria automobilistica, produttori di valvole, produttori di circuiti integrati e schede elettroniche e produttori di airbox.

Ora che le voci e gli obiettivi principali del progetto sono state definite si può procedere con la pianificazione delle possibili soluzioni.

## 2.2 Pianificazione

Verranno utilizzati di seguito due metodi, “SWOT Analysis” e “Pros & Cons”, per la proposta progettuale dello scarico dell’airbox. Entrambi i metodi consistono nella creazione di uno schema per la raccolta sistematica delle informazioni. L’obiettivo di questi schemi è quello di individuare i punti di forza e le debolezze della proposta, in modo da rafforzarne il valore dell’opportunità e individuare degli aspetti strategici per il successivo sviluppo.

### Swot Analysis

<b>Punti di forza</b> <ul style="list-style-type: none"><li>• Possibilità di accedere a utilizzatori potenziali</li><li>• Industrializzazione/intercambiabilità dei pezzi</li><li>• Ridotto volume del prodotto</li><li>• Possibilità di personalizzazione e sostituzione dei componenti</li><li>• Design ad alte performance</li><li>• Capacità progettuali airbox già presenti nel gruppo</li></ul>	<b>Debolezze</b> <ul style="list-style-type: none"><li>• Maggior costo di produzione</li><li>• Maggiori costi e tempi di manutenzione del veicolo</li><li>• Difficoltà di montaggio e tempo di assemblaggio</li></ul>
<b>Opportunità</b> <ul style="list-style-type: none"><li>• Mercato automobilistico</li><li>• Mercato motociclistico</li></ul>	<b>Minacce</b> <ul style="list-style-type: none"><li>• Possibile risposta negativa del mercato</li></ul>

**Tabella 2.1:** Swot Analysis

## Pros & Cons

Pros	Cons
<ul style="list-style-type: none"><li>• Componenti di facile reperibilità</li><li>• Ripetibilità</li><li>• Mass Customization</li></ul>	<ul style="list-style-type: none"><li>• Regole di progetto</li><li>• Spazi ridotti</li></ul>

**Tabella 2.2:** Pros & Cons

Dai “pros” si possono distinguere quindi il fatto di poter trovare tutti i componenti già sul mercato, in quanto standardizzati, la possibilità di ripetere lo stesso procedimento per ogni WOPI prodotto se le pressioni in gioco tra ingresso ed uscita in esso rientrano nello stesso range, ma nel caso queste fossero diverse si punta al soddisfacimento delle necessità individuali di ogni cliente sostituendo al massimo un solo componente (Mass Customization). Dal lato dei “Cons” invece ci sono gli stessi limiti che si affrontano nella progettazione dell’airbox, in particolare sarà necessario ridurre le dimensioni del WOPI il più possibile per non sottrarre spazio agli altri elementi presenti sotto il cofano dell’auto.

Possiamo procedere ora alla pianificazione concettuale, dati i notevoli benefici appena riscontrati dai due metodi.

## 2.3 Progettazione concettuale

Per prima cosa bisogna riconoscere e organizzare i bisogni del cliente. Attraverso una breve analisi del prodotto, si è convenuto che questa sia la “scaletta” che più fedelmente li rispecchia:

- Efficienza
- Facilità di montaggio/produzione
- Dimensioni limitate
- Design ottimale
- Facilità di manutenzione

Ora veniamo alla definizione delle specifiche tecniche in gioco nel progetto (in rosso verranno identificate le variabili dipendenti e tra parentesi quadre l'unità di misura):

- Forma e volume airbox [m<sup>3</sup>] (dipendente da lunghezza, altezza, profondità e forma prestabilite è la variabile principale per la progettazione di un airbox, da cui poi dipenderanno tutte le altre)
- Caduta di pressione [Pascal]
- Peso airbox [kg] (funzione ovviamente del suo volume e del materiale utilizzato)
- Costo [€]

Ora che abbiamo definito sia i bisogni del cliente che le specifiche, si procede correlandole tra loro con la “casa della qualità” (da notare che non saranno presenti le caselle relative ai risultati della concorrenza, in quanto stiamo parlando di un prodotto innovativo):

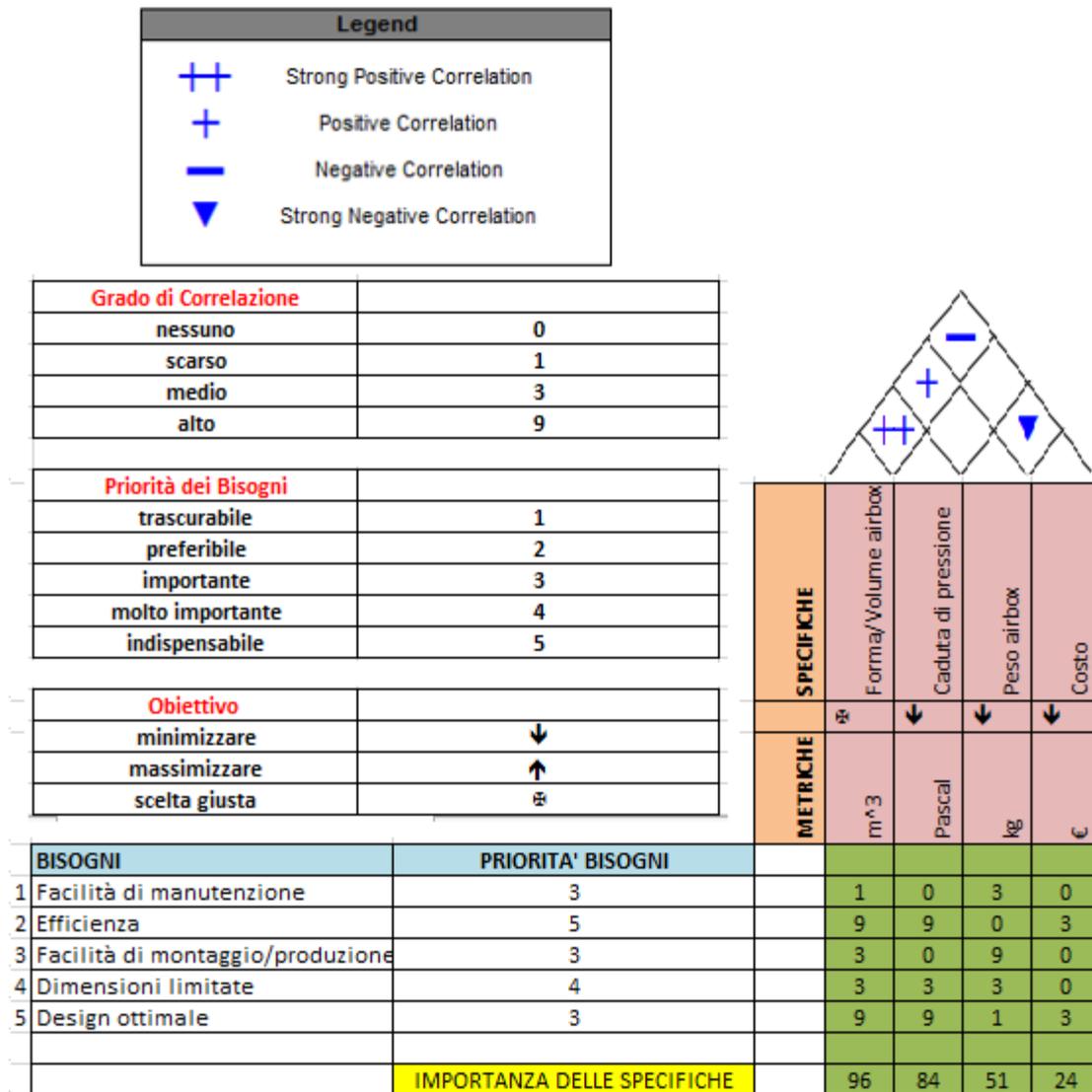
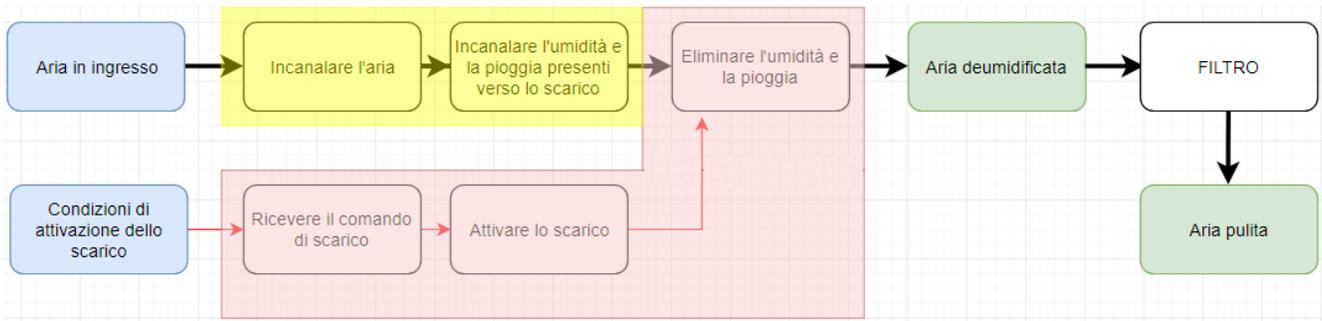


Tabella 2.3: House of Quality

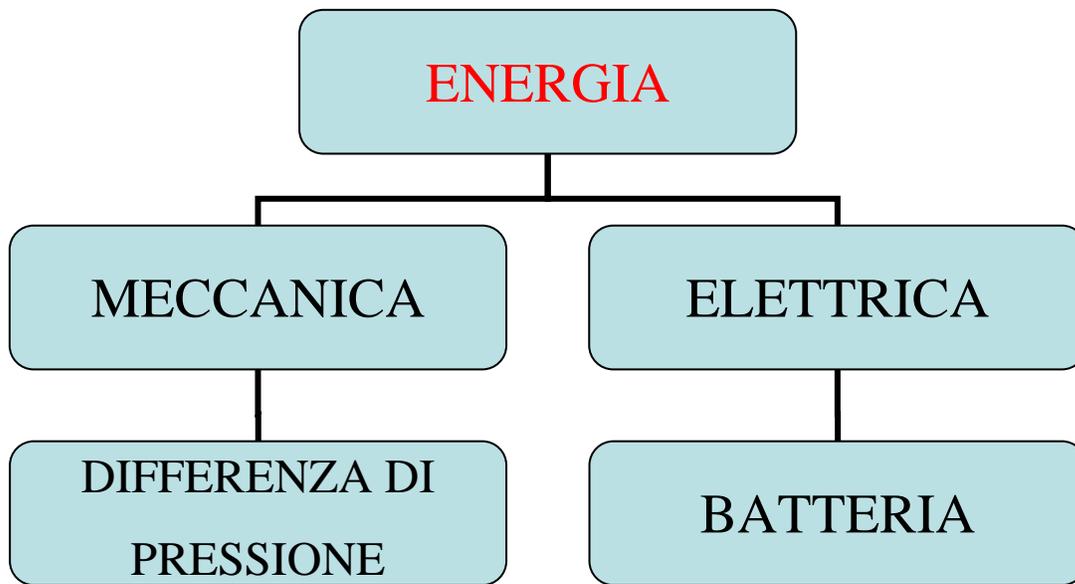
La caduta di pressione tra ingresso ed uscita è il parametro più importante da tenere in conto per lo sviluppo del progetto, ma sarà in relazione solo col design dell'airbox, per niente invece col suo peso e costo. Cosa si intende per il design dell'airbox? Il percorso che esso farà seguire al flusso d'aria in arrivo dall'ingresso, che meno penalizza l'inevitabile caduta di pressione.

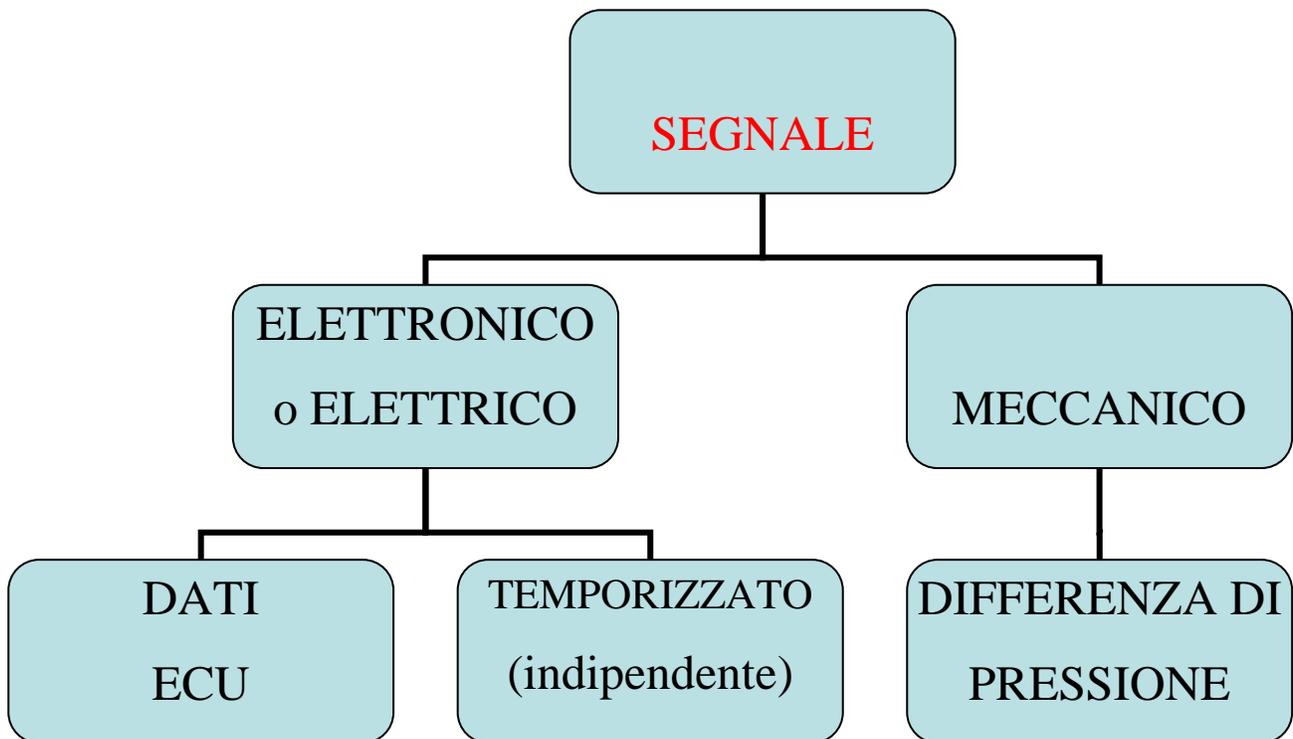
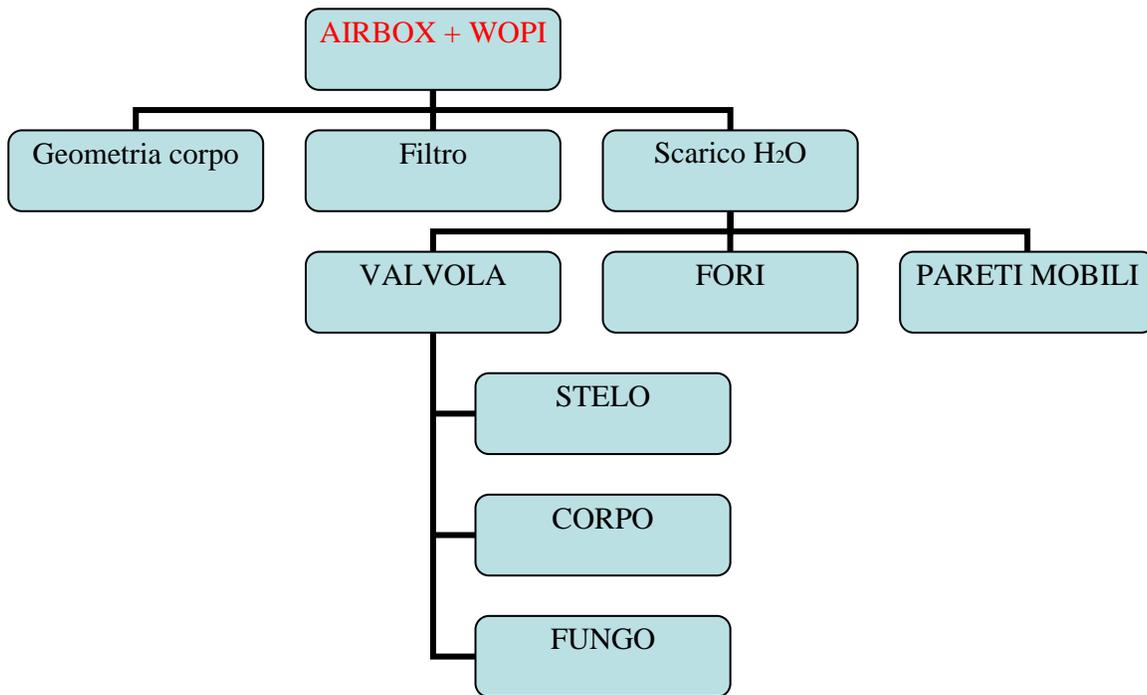
Per semplificare la ricerca dei vari tipi di soluzione è opportuno ricorrere ora alla scomposizione funzionale del prodotto:



**Diagramma 2.4:** Scomposizione funzionale del prodotto

Avendo scomposto tutto il processo di filtraggio e deumidificazione possiamo ora procedere con la ricerca sistematica delle soluzioni per ottenere il nostro scopo:





Diagrammi 2.5: Ricerca sistematica delle soluzioni

Guardando questi diagrammi risulta immediato che è possibile ottenere potenzialmente una discreta varietà di prodotti, utilizzando vari modi di energizzare il sistema (totalmente ad energia meccanica, oppure un mix di energia elettrica e meccanica), di scaricare l'acqua presente (valvola, fori, circuito interno od esterno) e come segnalare al sistema di scaricarla, per esempio tramite la differenza di pressione tra airbox e ambiente, tramite segnale dalla ECU (Engine Control Unit, verrà ampiamente definita nel capitolo 4) o comandato indipendentemente da un circuito elettrico secondario. Andiamo ora a selezionare il concept finale del nostro prodotto. Dalla generazione dei concept sono venute fuori molteplici soluzioni per definire la struttura degli elementi interni, molte delle quali però, per praticità e costi sono state subito scartate.

Tra questi concept di sviluppo del prodotto, si è preso in considerazione lo sviluppo di un sistema di scarico a comando con valvola, in alternativa ad uno di scarico automatico che utilizza il precarico delle molle e la differenza di pressione per aprirsi. Il primo di questi poi è stato suddiviso in due possibili soluzioni: l'elettrovalvola viene comandata autonomamente (caso di circuito indipendente che, per esempio, la apre e chiude dopo tot lassi temporali, indipendentemente dal fatto che piova o meno e che possa o meno portar via pressione quando viene richiesta o no maggior potenza dal motore), oppure, sotto appositi segnali ricevuti dall'ECU si aprirà solo nei momenti più opportuni. Viene quindi eseguito un "Concept Scoring" (metodo delle matrici decisionali) tra queste tre configurazioni:

		A – Scarico Automatico (Precarico molle)		B – Scarico con e.valvola e circuito indipendente		C – Scarico con e.valvola e circuito comandato da ECU	
	PESO	Classifica	Punteggio Pesato	Classifica	Punteggio Pesato	Classifica	Punteggio Pesato
EFFICIENZA	0,6	1	0,6	3	1,8	5	3
FACILITA' DI PRODUZIONE	0,3	5	1,5	3	0,9	1	0,3
FACILITA' MANUTENZIONE	0,1	5	0,5	3	0,3	2	0,2
PUNTEGGIO TOTALE	1	2,6		3		3,5	
CLASSIFICA		3		2		1	
CONTINUARE?		NO		NO		SI'	

**Tabella 2.6:** Concept Scoring delle configurazioni del WOPI

Risulta evidente che il sistema che utilizza una valvola per lo scarico comandata direttamente dall'ECU è un concept più appetibile rispetto ai suoi concorrente, anche se il punteggio ottenuto con tale metodo (quantitativo) non fornisce una differenza considerevole di valori rispetto a quello con circuito indipendente.

Per quanto riguarda il design e le dimensioni dell'airbox e del suo filtro, si è proceduto semplicemente nel disegnare un prototipo, che verrà mostrato nella sezione "layout approssimativo", inoltre qui non viene trattato poiché non influisce sullo sviluppo del concept.

## 2.4 Progettazione del sistema

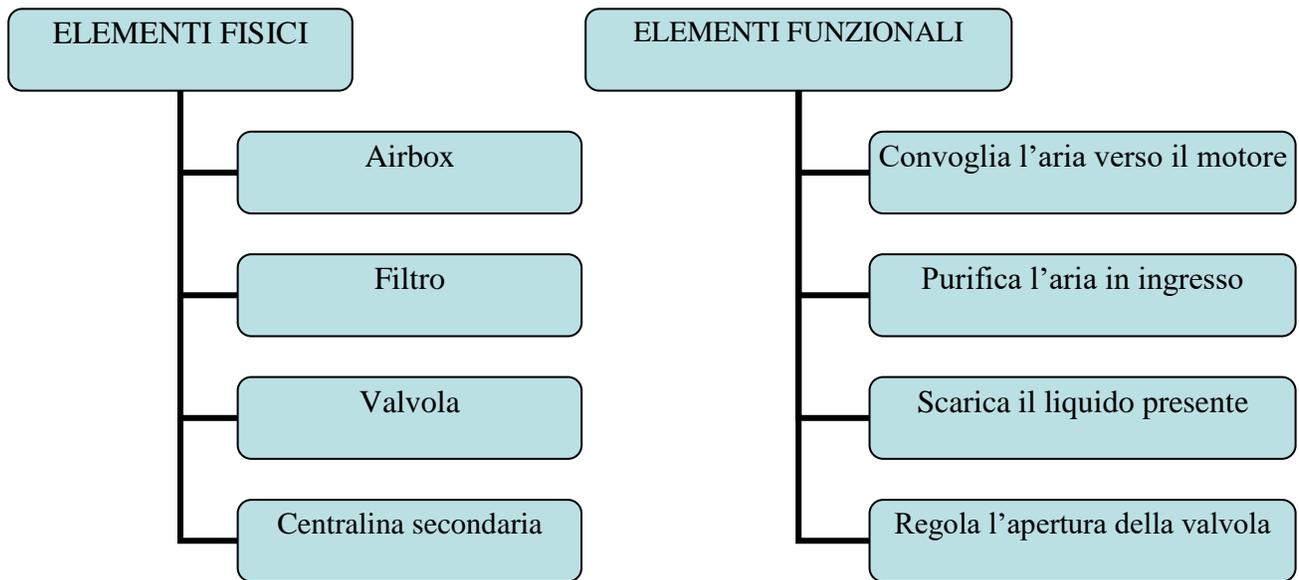
Le funzioni che dovrà assolvere il sistema WOPI, oltre ad eliminare l'acqua e l'umidità interne all'airbox, saranno:

- Mantenere in pressione l'airbox
- Scaricare i liquidi solo quando non compromette la performance del motore
- Non avere perdite di pressione durante la fase di chiusura
- Convogliare i liquidi presenti nella parte inferiore dell'airbox per facilitarne la fuoriuscita

Adesso si va avanti riconducendo le precedenti scomposizioni funzionali nei seguenti blocchi fisici:

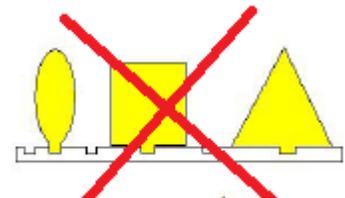
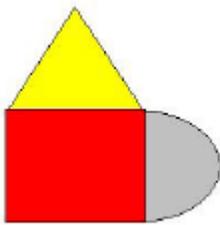
- Airbox
- Filtro
- Valvola
- Centralina secondaria

Questo prodotto è stato pensato per avere un'architettura modulare, le cui interazioni tra i vari blocchi sono ben definite e sono generalmente fondamentali per le funzioni principali del prodotto.



**Diagramma 2.7:** Architettura del prodotto

È stata scelta un'architettura modulare “a sezioni”, in questo modo ho assenza di un blocco comune (detto bus) dove tutti gli elementi andrebbero collegati (questo nel caso avessi adottato un'architettura modulare “a slot” oppure “a bus”).



Questo tipo di modularità può essere sfruttata per poter implementare il sistema WOPI a diversi modelli di airbox, dove il sarà “sufficiente” progettare il circuito di scarico e il relativo attacco per la valvola (ipotizzando che venga usata sempre la stessa).

Questa modularità si converte in una (limitata) difficoltà di assemblaggio ogni volta del sistema, al contrario con un architettura di tipo integrale, ovvero avendo prodotto il tutto come un “blocco unico” avrebbe eliminato la questione. Però l’architettura di tipo modulare ha diversi vantaggi sull’integrale, ovvero:

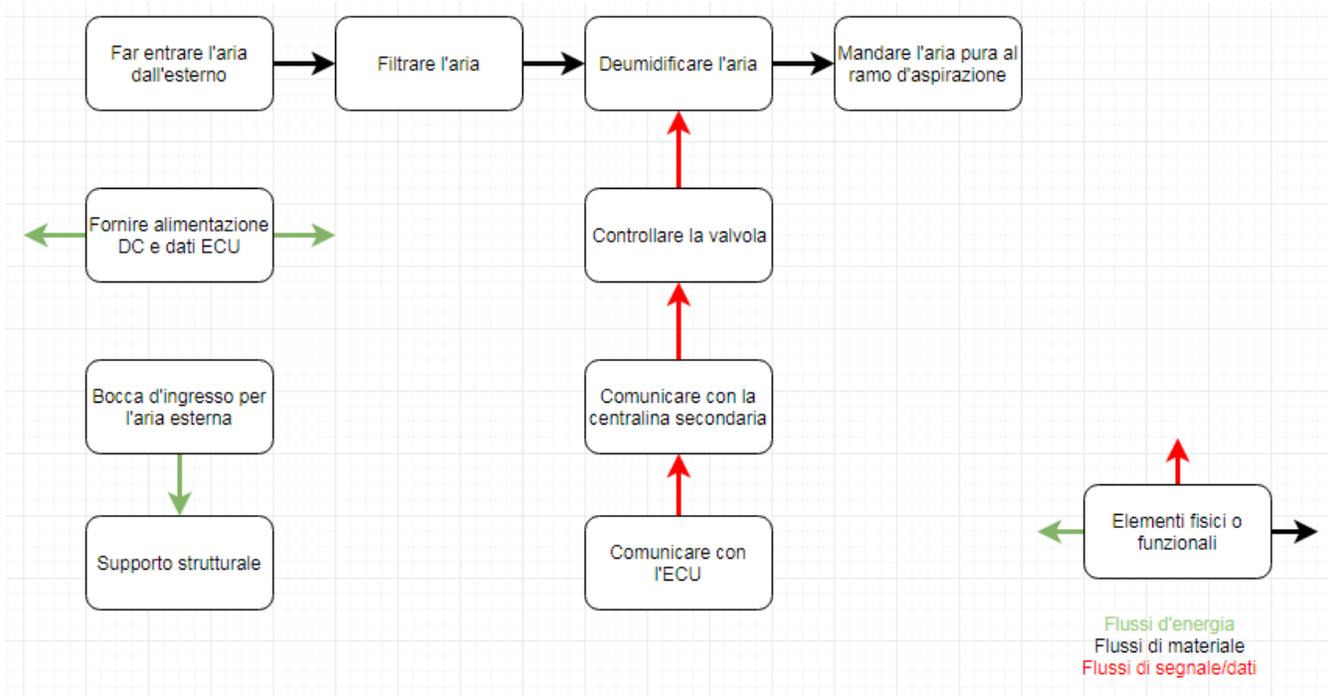
- C’è elevata standardizzazione, ovvero con ogni probabilità si utilizzerà la stessa valvola in ogni prodotto (relativa ad diversi modelli di airbox) in quanto la pressione interna e le quantità di umidità e pioggia saranno sempre molto simili
- I componenti standard hanno maggiore affidabilità (una valvola prodotta da un’azienda specializzata lo è certamente di più di una costruita ad hoc
- I tempi di sviluppo vengono ridotti grazie a questa standardizzazione

Stesso discorso vale per la centralina secondaria, almeno inizialmente, per la quale andrebbe assunto personale qualificato e comprate macchine per l’incisione di PCB (printed circuit board o circuito stampato). Questo comporta diversi benefici:

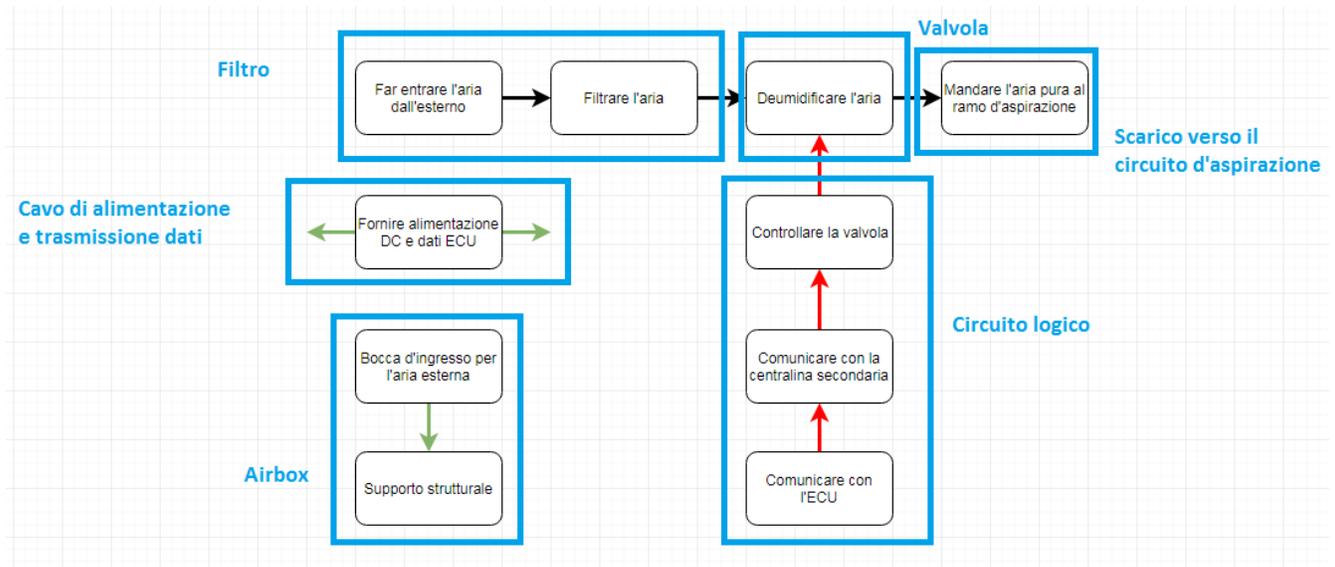
- Minimizzando gli investimenti su questi componenti l’azienda potrà concentrarsi solo sui nuovi componenti, ovvero il design e la produzione sui propri airbox di un adeguato circuito di scarico che convogli l’acqua verso la valvola. Questo comporta quindi la focalizzazione dell’azienda sui componenti chiave in cui è più esperta e preparata sia in termini di progettazione che produzione.
- Applicazione dell’economia di scala se, come ipotizzato in precedenza, i componenti acquistati verranno utilizzati su più di una versione del prodotto

Possiamo passare ora allo sviluppo vero e proprio dell’architettura di prodotto, che è composta da quattro fasi:

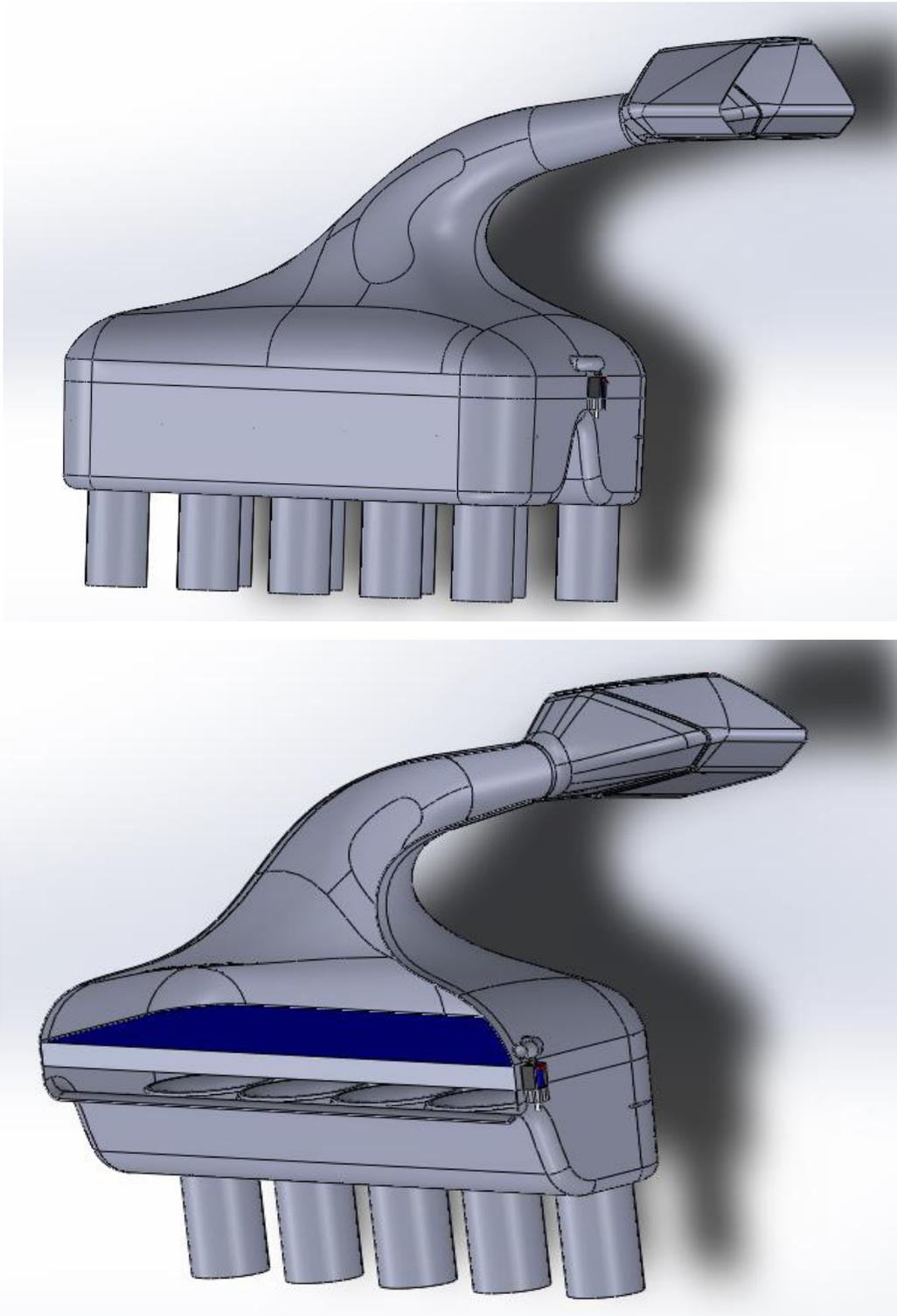
### 1. Schema del prodotto



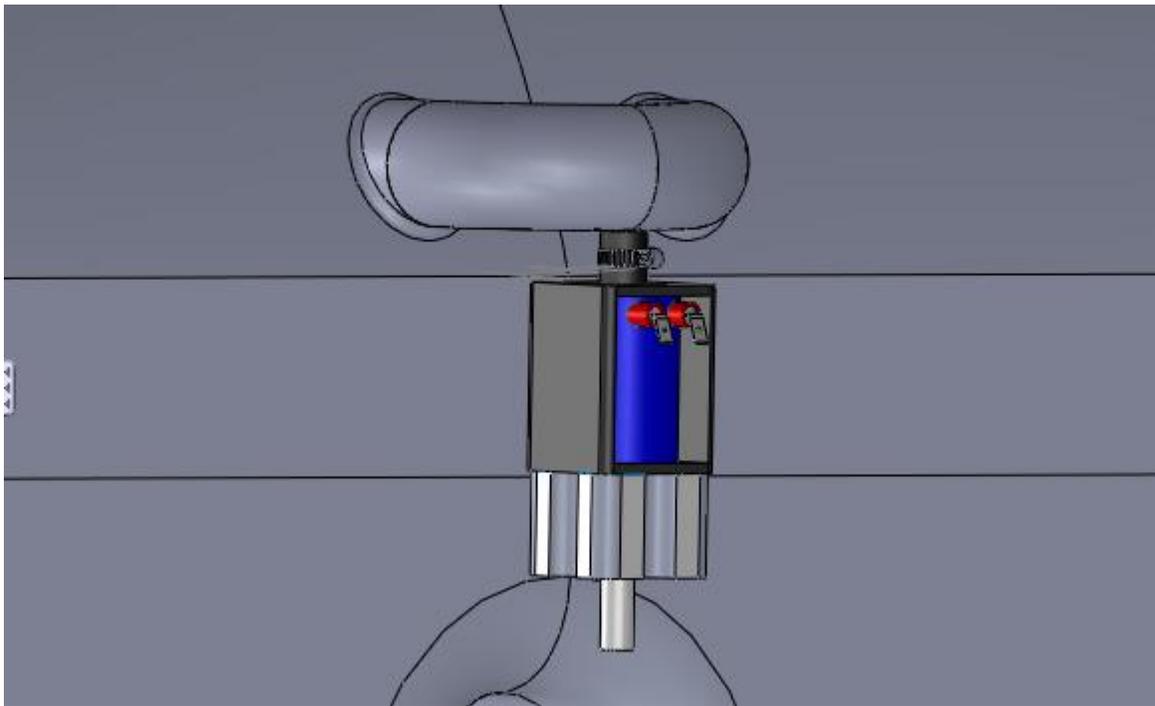
### 2. Raggruppamento degli elementi dello schema



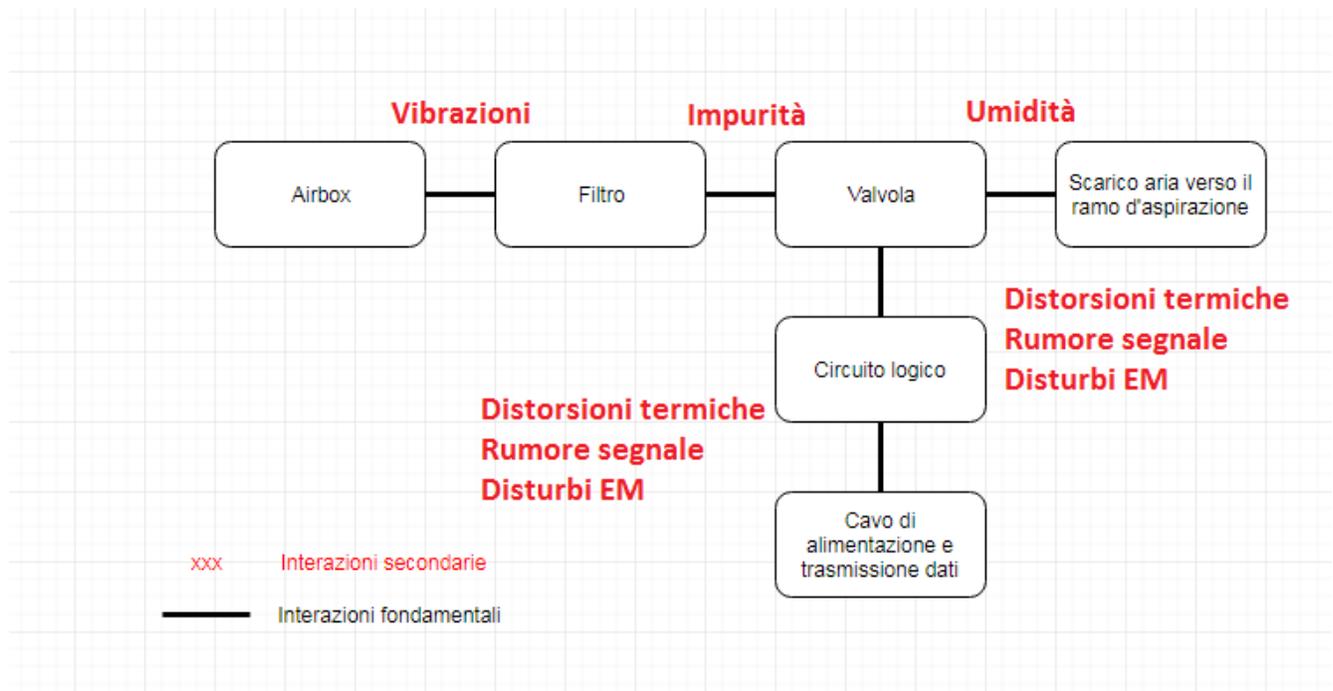
### 3. Layout Approssimativo



**Disegno 2.7:** Layout approssimativo del WOPI installato su un airbox (quest'ultimo gentile concessione della BMC srl)



#### 4. Interazioni fondamentali ed accidentali



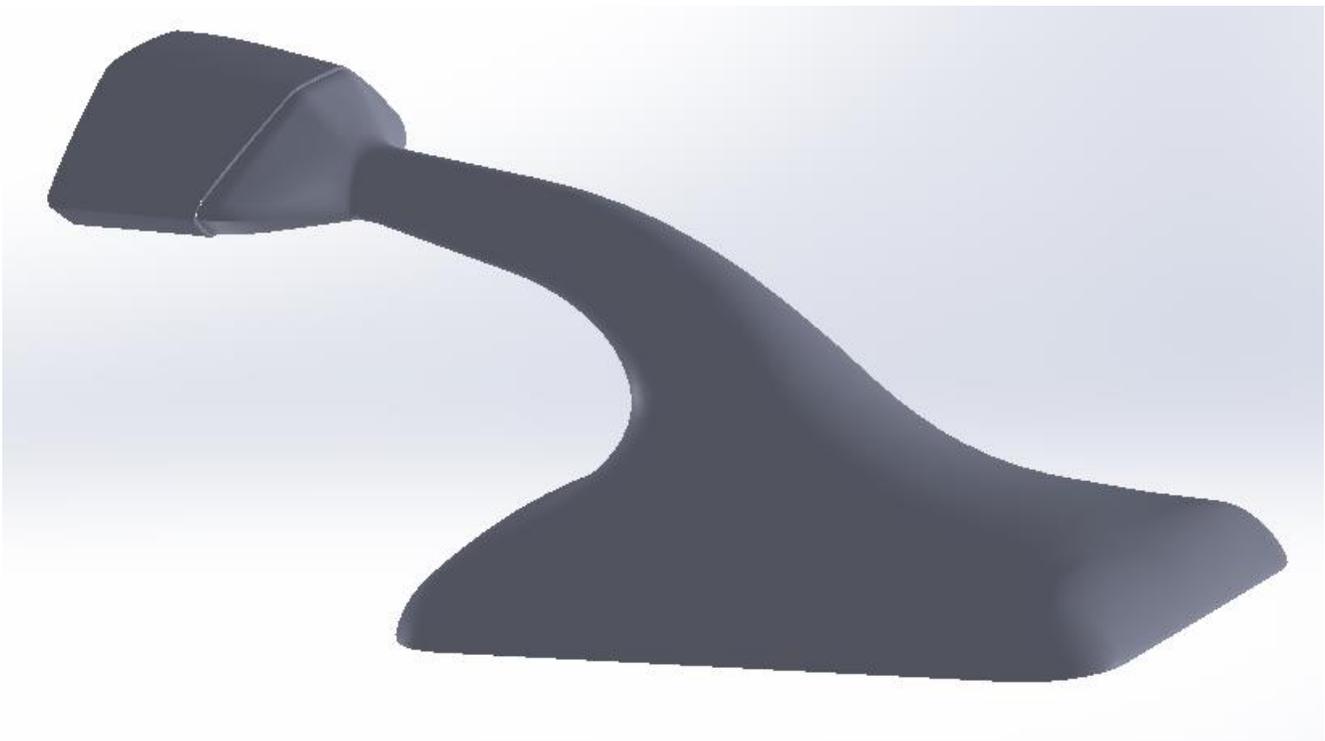
Ora abbiamo tutti gli elementi necessari per procedere nei capitoli successivi allo sviluppo del sistema WOPI e valutare se valga la pena produrlo, in termini di performance. Verrà inoltre prodotto un prototipo funzionante per simularne l'uso, completo di tutti i componenti principali.

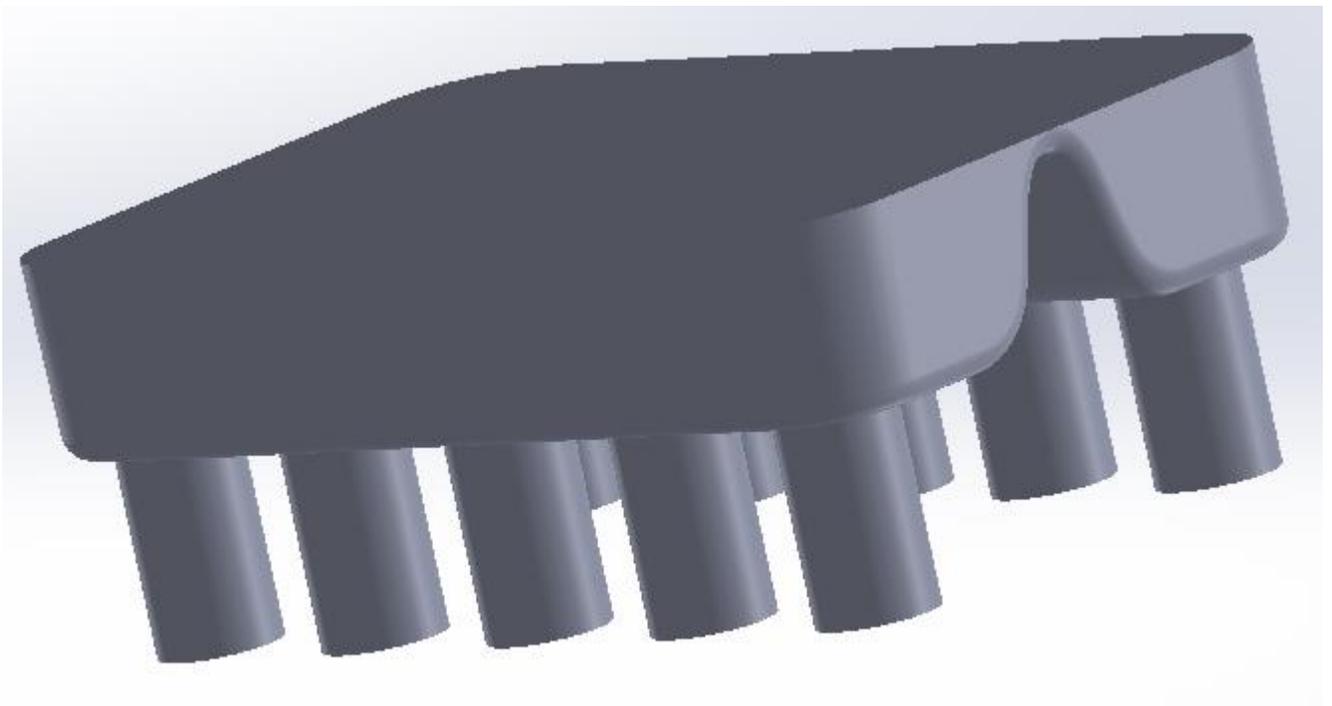
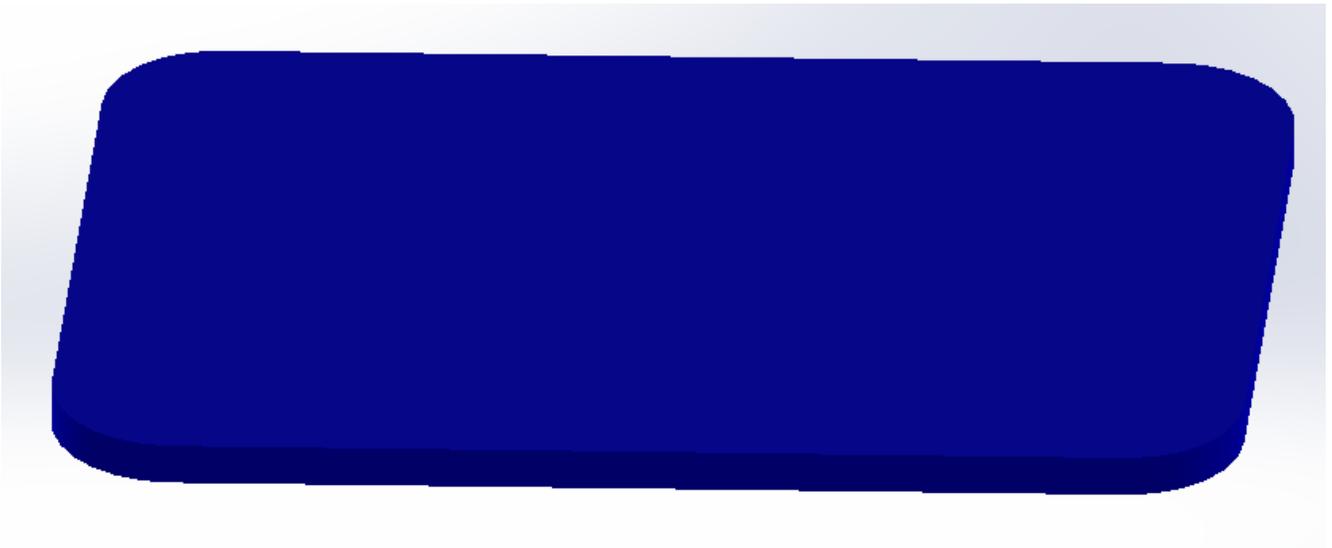
## 3. Simulazioni fluidodinamiche

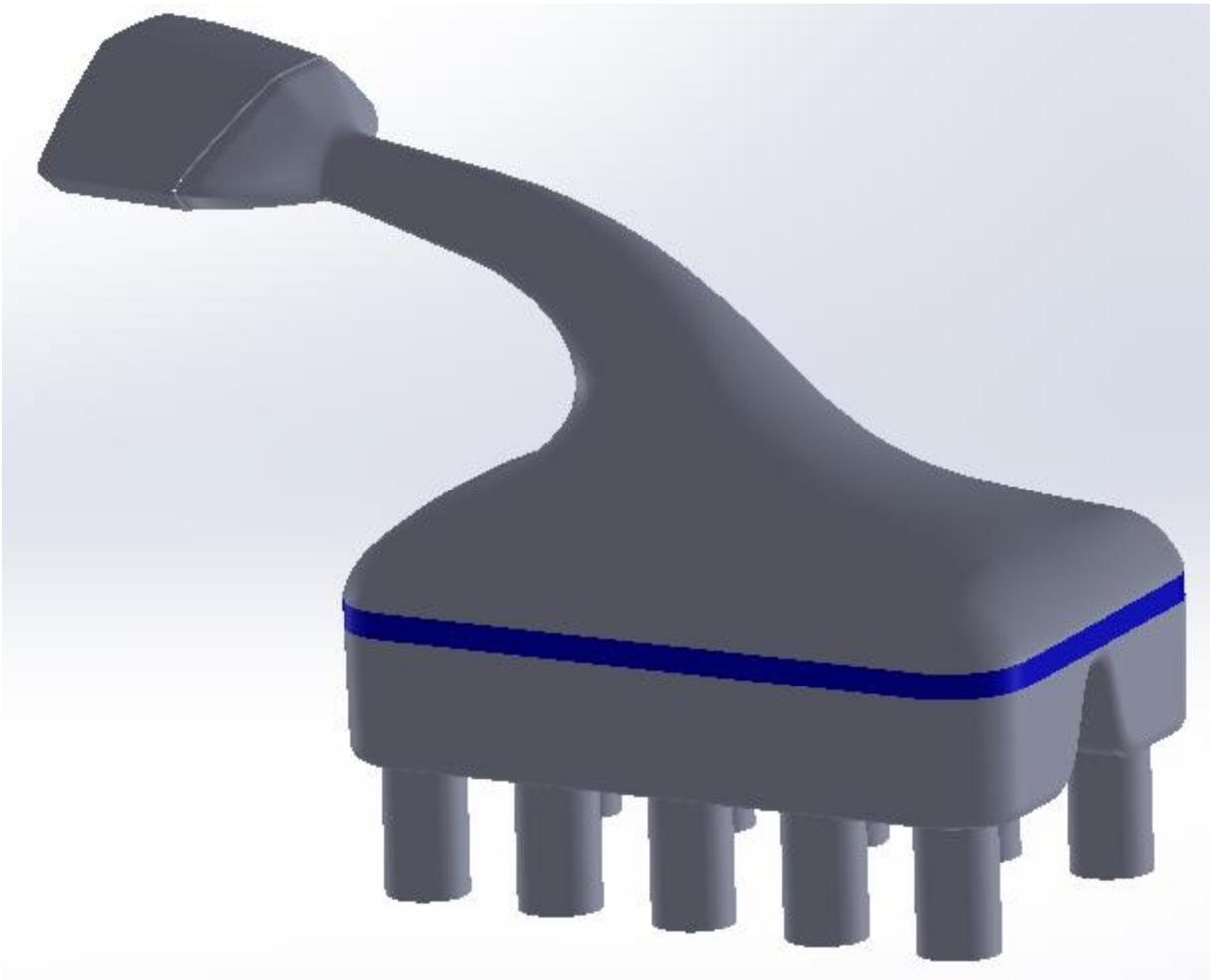
È necessario, prima di costruire un modello fisico che rispecchi fedelmente il funzionamento reale del prototipo, verificare che a livello teorico il dispositivo funzioni nel modo desiderato. Inoltre coi dati ottenuti, sarà possibile anche effettuare migliorie al progetto, come ad esempio scegliere il tipo di valvola più adatta allo scopo o in che posizione collocarla, in modo che funzioni propriamente durante la marcia del veicolo.

### 3.1 Il disegno 3D

Riproduciamo il modello di un airbox che verrà verosimilmente prodotto ed installato su un veicolo esistente. Esso sarà composto principalmente da tre parti: inlet, filtro ed outlet. Dal disegno progettuale mostrato nel secondo capitolo, si possono ricavare i negativi di ognuna di queste parti, necessari per poter eseguire la simulazione, che verranno mostrati col medesimo ordine nelle righe sottostanti, con in coda anche l'assieme finale:

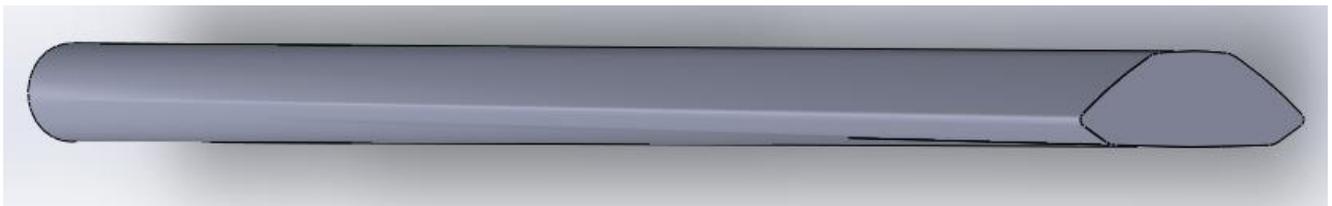






**Disegni 3.1:** Negativi del progetto dell'airbox

Inoltre in una seconda versione delle simulazioni verrà adottato un quarto componente, il cosiddetto “tubo di flusso”, il cui utilizzo e scopo sarà illustrato nel paragrafo 3.3:



**Disegno 3.2:** Tubo di flusso



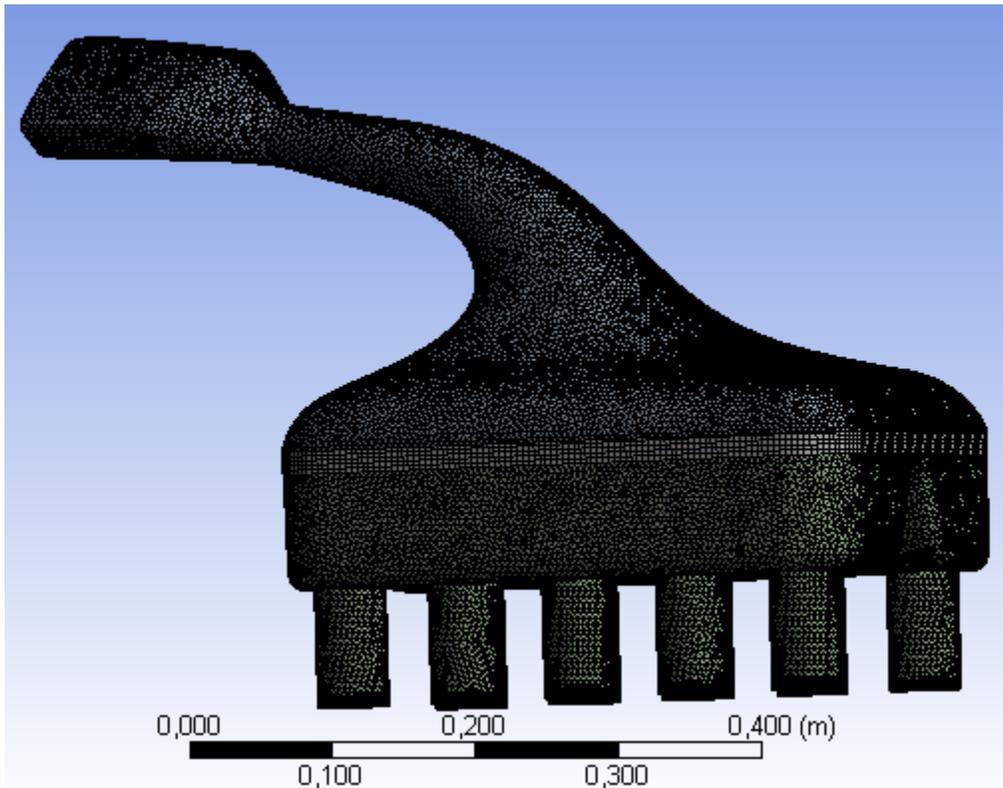
**Disegno 3.3:** Complessivo

### 3.2 Mesh

Letteralmente “maglia”, utilizzando il disegno 3D realizzato possiamo rappresentare il modello con un reticolo composto da molte facce poligonali. Grazie a questo è possibile eseguire la simulazione ed ottenere le informazioni fisiche desiderate durante l’utilizzo dell’airbox.

Ovviamente un modello più è fitto (ha un maggior numero di celle), migliore sarà la precisione del risultato. Per verificare inoltre che i risultati ottenuti siano ripetibili (ovvero che la simulazione sia stabile e non si sia giunti alla soluzione in modo fortuito) le simulazioni sono state eseguite con tre diversi numeri di celle: 125'000, 250'000 e 500'000. Per ottenere queste diverse dimensioni sarà necessario variare volta per volta le dimensioni massime e minime delle celle; inoltre, per avere una mesh di qualità bisogna che le loro skewness (asimmetria) ed ortogonalità siano inferiori rispettivamente a 0,4 e 60°. Questo perché in un modello a volumi finiti, tutte le equazioni vengono discretizzate usando la distanza tra i centri delle celle adiacenti (ipotizzando che i vettori unenti questi centri siano perpendicolari alla faccia comune). Quando questo criterio non viene soddisfatto, il flusso calcolato non è più normale a questa faccia e questo porta a valori sbagliati delle variabili nel volume di controllo, tutto questo impedisce alla soluzione di convergere, se non addirittura portare a divergenza!

Di seguito viene mostrata la mesh di 500'000 celle senza tubo di flusso posto all'ingresso, quella con invece non verrà mostrata in quanto essendo lungo ben 2 metri, se preso per intero, data la finezza degli elementi, l'immagine farà vedere solo un disegno 3D del WOPI di color nero.



**Disegno 3.4:** Mesh del negativo. "Images used courtesy of ANSYS, Inc."

### 3.3 Simulazioni

Fatti i passaggi precedenti si possono far partire le simulazioni vere e proprie. Possono essere condotte seguendo tre strade:

- 1) In aria libera. Ovvero, il suo dominio non sarà solo l'airbox, ma verrà inserito in un volume d'aria. In sostanza, va creato un "involucro" sufficientemente grande intorno all'airbox per poter ridurre il campo di pressioni esterno. Questo metodo richiede però troppe celle per essere utilizzato e in queste condizioni non è realizzabile
- 2) Come nel caso 1) ma senza il dominio d'aria esterno, quindi solo utilizzando l'airbox

- 3) Simulazione forzata. È un'alternativa alla risoluzione in aria libera, dove le simulazioni sono state eseguite, oltre che con queste 3 parti (inlet, filtro e outlet), anche con l'ausilio di un'ulteriore componente già visto in precedenza chiamato "tubo di flusso", che funge da divergente ed aumenta la pressione dell'aria in ingresso. In questo caso si parla di simulazione con "effetto RAM" (o presa dinamica, per simulare la vettura in movimento).

Quindi, i metodi 2 e 3 saranno quelli adottati per verificare l'effettiva utilità del WOPI. Saranno entrambi veritieri ma, non daranno mai gli stessi risultati.

Le simulazioni verranno eseguite in condizioni statiche con velocità dell'aria in condizioni di numero di Mach pari a 0,6 e 1. Il numero di Mach ( $M$ ) è il rapporto tra la velocità del fluido e la velocità del suono all'interno del fluido in questione. In caso di  $M < 1$  si sarà di fronte ad un moto subsonico, sonico se  $= 1$ , infine supersonico se  $> 1$ . Bisogna ora determinare la portata d'aria in funzione di questo numero in entrambi i casi. Per farlo, si fa riferimento all'equazione di moto adiabatico dei fluidi comprimibili e all'equazione di stato dei gas perfetti, che danno forma alla seguente equazione semplificata:

$$\dot{m} = S \frac{p_t}{\sqrt{RT}} \frac{\sqrt{\gamma} M}{\left(1 + \frac{\gamma - 1}{2} M^2\right)^{\frac{\gamma + 1}{\gamma - 1}}}$$

Con  $S$  sezione di passaggio del fluido (visto prima essere  $0,014439 \text{ m}^2$ ),  $p_t$  la sua pressione totale,  $R$  la costante dei gas perfetti (che per gas biatomici come l'aria, vale  $287 \text{ J/kgK}$ ),  $T$  è la temperatura dell'aria ( $20^\circ\text{C}$ , che nella formula vengono convertiti in  $293 \text{ K}$ ), infine  $\gamma$  è il rapporto tra il calore specifico dell'aria a pressione costante e quello a volume costante ( $1,4$  per l'aria). Troveremo quindi come portate in massa, per  $M$  pari a  $0,6$  e  $1$  rispettivamente  $0,3098$  e  $0,3534 \text{ kg/s}$ .

Ora, note le portate d'aria in gioco, bisogna trovare le velocità con cui questa entrerà nell'inlet dell'airbox. Questo si ricava facilmente dall'equazione di continuità della massa, data:

$$\dot{m} = \rho A v$$

Dove abbiamo rispettivamente: densità dell'aria (in kg/m<sup>3</sup>), sezione di passaggio (in m<sup>2</sup>) e velocità del fluido. Dividendo la portata per densità e sezione si ricavano le velocità per M pari a 1 (20 m/s) e 0,6 (17,53 m/s).

Nella tabella successiva verranno riportati tutti i 4 casi che verranno simulati ed analizzati per entrambi i metodi con tutti e 3 i tipi di mesh considerati, per un totale di 24 simulazioni:

CASO A	CASO A BIS	CASO B	CASO B BIS
Portata: 0,3534 kg/s	Portata: 0,3534 kg/s	Portata: 0,3098 kg/s	Portata: 0,3098 kg/s
WOPI: chiuso	WOPI: aperto	WOPI: chiuso	WOPI: aperto

**Tabella 3.5:** Elenco casi di simulazione

Di seguito viene illustrato il set di impostazioni utilizzato per l'esecuzione:

- **Metodo risolutivo:** eddy-viscosity (vorticosità-viscosità) a due equazioni: K- $\Omega$  tipo SST (shear stress transport).

I modelli turbolenti a due equazioni sono quelli più usati, specialmente il K- $\Omega$ , ormai diventato uno standard nell'industria (si ricorda, per definizione, che la base di tutti i metodi a due equazioni parte dall'ipotesi di viscosità vorticoso di Boussinesq). Questo metodo si comporta generalmente molto bene, ma specialmente quando si ha a che fare in condizioni avverse dei gradienti di pressione e flusso separato. Si possono presentare elevate turbolenze in regioni che presentano stagnazione del flusso e con forti accelerazioni, ma comunque molto ridotte rispetto ad altri metodi, come il K- $\epsilon$ . K rappresenta la turbolenza dell'energia cinetica. Misurata in m<sup>2</sup>/s<sup>2</sup>, rappresenta l'energia cinetica sull'unità di massa delle fluttuazioni turbolente in un flusso del medesimo tipo e per definizione vale:

$$K = \frac{3}{2}(UI)^2$$

U rappresenta la velocità media del flusso d'aria, I invece l'intensità della turbolenza, che può essere stimata con la seguente equazione:

$$I = 0,16 Re_{dh}^{-1/8}$$

Dove  $Re_{dh}$  è il numero di Reynolds calcolato sul diametro idraulico dell'ingresso all'airbox:

$$Re_{dh} = U \frac{Dh}{\nu} = 313373$$

$$Dh = \frac{4S}{2p} = 0,111919 \text{ m}$$

S rappresenta la sezione di passaggio del condotto ( $0,014439 \text{ m}^2$ , misurata con Solidworks),  $\nu$  la viscosità cinematica dell'aria ( $0,000015 \text{ m}^2/\text{s}$  per l'aria) mentre  $2p$  il perimetro "bagnato" dal fluido ( $0,51605 \text{ m}$ , determinato come abbiamo fatto con la sezione). Ottenuti questi dati siamo ora in grado di ricavare  $I = 0,03609$  e quindi anche  $K = 0,78$ .

Invece  $\Omega$ , rappresenta la dissipazione turbolenta specifica (misurata in  $1/\text{s}$ ), cioè il rapporto al quale l'energia cinetica turbolenta viene convertita in energia termica interna per l'unità di volume e tempo. Per definizione:

$$\Omega = \frac{\sqrt{K}}{l}$$

Tuttavia per alcuni codici CFD, si adotta una correzione che porta  $\Omega$  ad essere espressa come:

$$\Omega = \frac{C_\mu^{-\frac{1}{4}} \sqrt{K}}{l}$$

$C_\mu$  è una costante del modello, spesso posta pari a  $0,09$ .  $l$  invece rappresenta la scala di lunghezza della turbolenza e, per semplicità, può essere stimata grazie al diametro idraulico:

$$l = 0,038Dh = 0,0042529 \text{ m}$$

Perciò  $\Omega$  alla fine varrà circa  $430 \text{ 1/s}$ . Nel solutore fluidodinamico, alla fine sono stati adottati delle approssimazioni, ovvero ponendo  $K = 1$ , aumenterà di conseguenza anche  $\Omega$ , che verrà settata a  $500$ . Infine come base per la simulazione è stata scelta la pressione e statica nel tempo.

- **Densità dell'aria:**  $1,225 \text{ Kg/m}^3$ , fluido comprimibile.
- **Simulazione del filtro (mezzo poroso):** il flusso passante attraverso un mezzo poroso può essere ipotizzato seguendo l'equazione di Blake-Kozeny:

$$\frac{|\Delta P|}{L} = \frac{150\mu}{Dp^2} * \frac{(1 - \epsilon)^2}{\epsilon^2} * v_\infty$$

Dove  $\mu$  rappresenta la viscosità del fluido,  $Dp$  il diametro medio delle particelle,  $L$  lo spessore del filtro ed  $\epsilon$  la frazione di vuoto del filtro.

Questa equazione è quindi composta da due variabili, chiamate rispettivamente resistenza inerziale  $\alpha$  e resistenza viscosa  $\beta$ :

$$\alpha = \frac{Dp^2}{150} \varepsilon^2$$
$$\beta = \frac{3,5(1-\varepsilon)}{Dp} \frac{1}{\varepsilon^2}$$

Altrimenti, si può utilizzare la legge di Darcy-Forchheimer, tipica dei moti turbolenti:

$$\frac{\Delta P}{\Delta x} = \alpha V^2 + \beta V$$

Dove  $\alpha$  rappresenta il coefficiente inerziale, mentre  $\beta$  il rapporto tra la viscosità dell'aria e il prodotto tra la sua densità e la permeabilità del filtro. Questi valori sono stati già ricavati in precedenza dall'azienda in seguito a precedenti simulazioni eseguite e valgono, rispettivamente:  $\alpha = 351,17$  e  $\beta = 2113,98$ .

- **Condizioni al contorno:**

Inlet (senza effetto RAM): velocità aria = 20 m/s (casi A e A bis) e 17,53 m/s (casi B e B bis), per poter avere continuità di portata in massa di rispettivamente 0,3534 e 0,3098 kg/s;

Pressione relativa: 0 Pa; Temperatura: 293 K (20°C).

Inlet (con effetto RAM): tutto uguale rispetto al caso senza tubo di flusso, eccetto per la velocità dell'aria = 42 m/s che, grazie alla divergenza del tubo, arriverà in ingresso all'airbox a 20 e 17,53 m/s per avere rispettivamente 0,3534 e 0,3098 kg/s di portata.

Outlet alle trombette (casi A e A bis): portata in massa d'aria in uscita = 0,3534 kg/s; gradiente di velocità e pressione = 0.

Outlet alle trombette (casi B e B bis): portata in massa d'aria in uscita = 0,3098 kg/s; gradiente di velocità e pressione = 0.

Ulteriore condizione all'outlet nei casi con gli scarichi aperti (bis): pressione relativa = 0 Pa; gradiente di velocità = 0 1/s.

- **Inizializzazione:** Standard (qui si può scegliere tra standard e ibrida. La prima permette di specificare manualmente tutte le condizioni iniziali, mentre la seconda utilizza le condizioni al contorno per ricavarle, ma è un'inizializzazione più "grezza" rispetto alla standard).

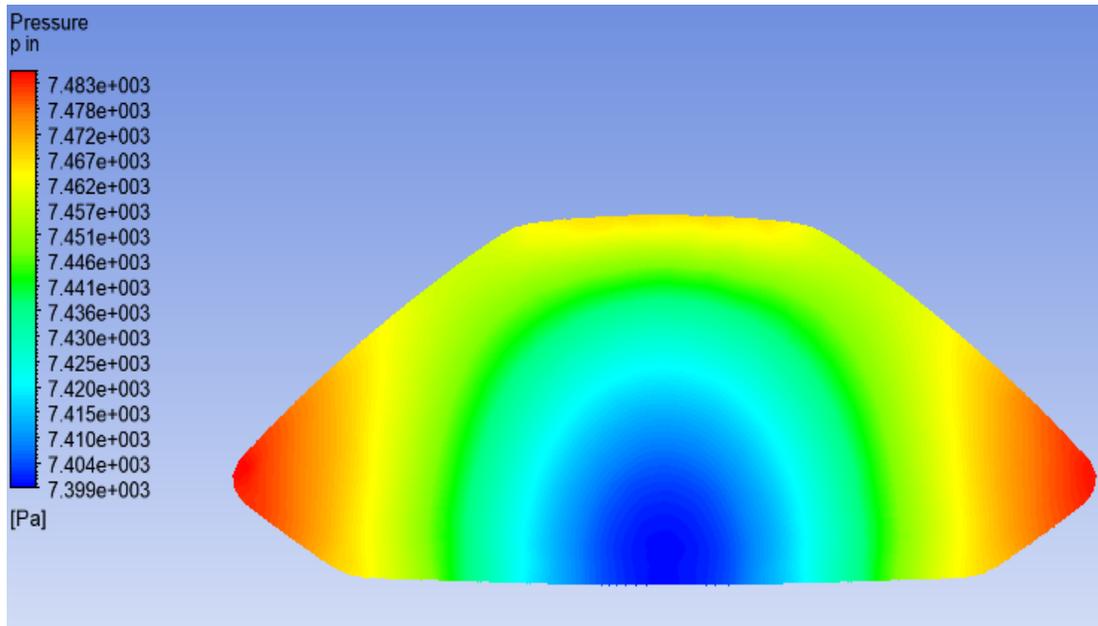
- **Numero di iterazioni:** 300, si vedrà essere abbondantemente sufficiente per avere portata in massa di aria costante sia in ingresso che uscita, nonché un calo significativo dei residui della simulazione (dell'ordine di  $10^{-3}$ )

Eseguite tutte e ventiquattro le simulazioni, si possono ricavare i range (minimi e massimi) ed i valori medi dei campi di pressione e velocità nelle zone più significative atte alla verifica dell'effettiva utilità del WOPI, per questo vengono riportati, per tutti e quattro i casi possibili, quelli di: ingresso e uscita dall'airbox, e poco sopra l'ingresso del filtro. Sarà comunque lampante al lettore vedendo questi campi, come il vantaggio dell'utilizzo o meno del WOPI si potrà conoscere solo conoscendo i valori medi di velocità e pressione, dato che gli andamenti e i valori minimi e massimi saranno molto simili tra loro.

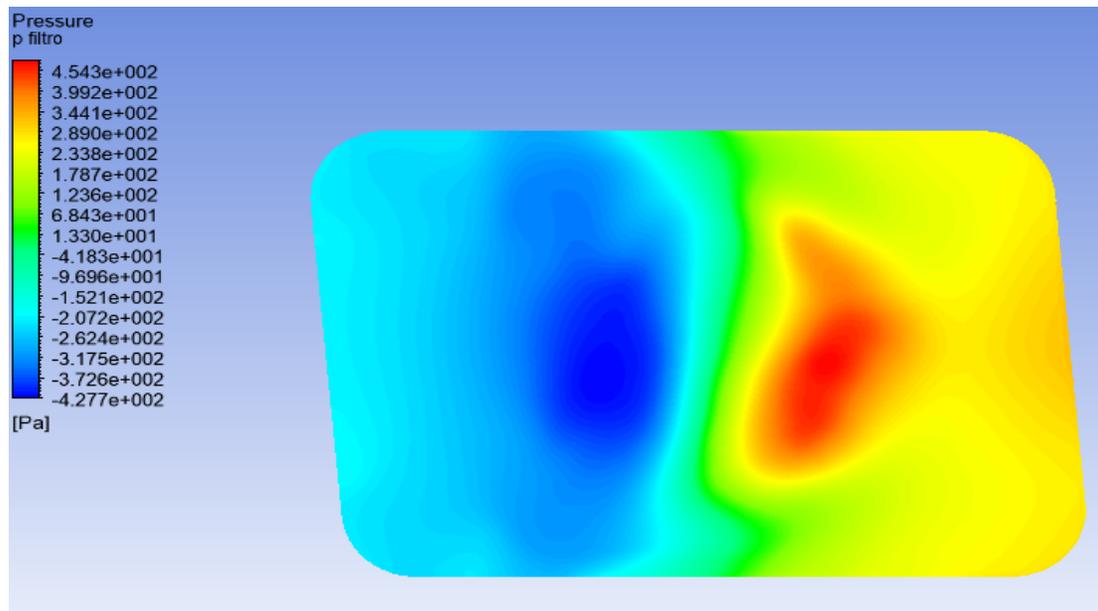
### 3.3.1 RISULTATI PER MODELLO SENZA TUBO DI FLUSSO

#### 3.3.1.1 WOPI CHIUSO CON PORTATA IN MASSA DI 0,3534 Kg/s

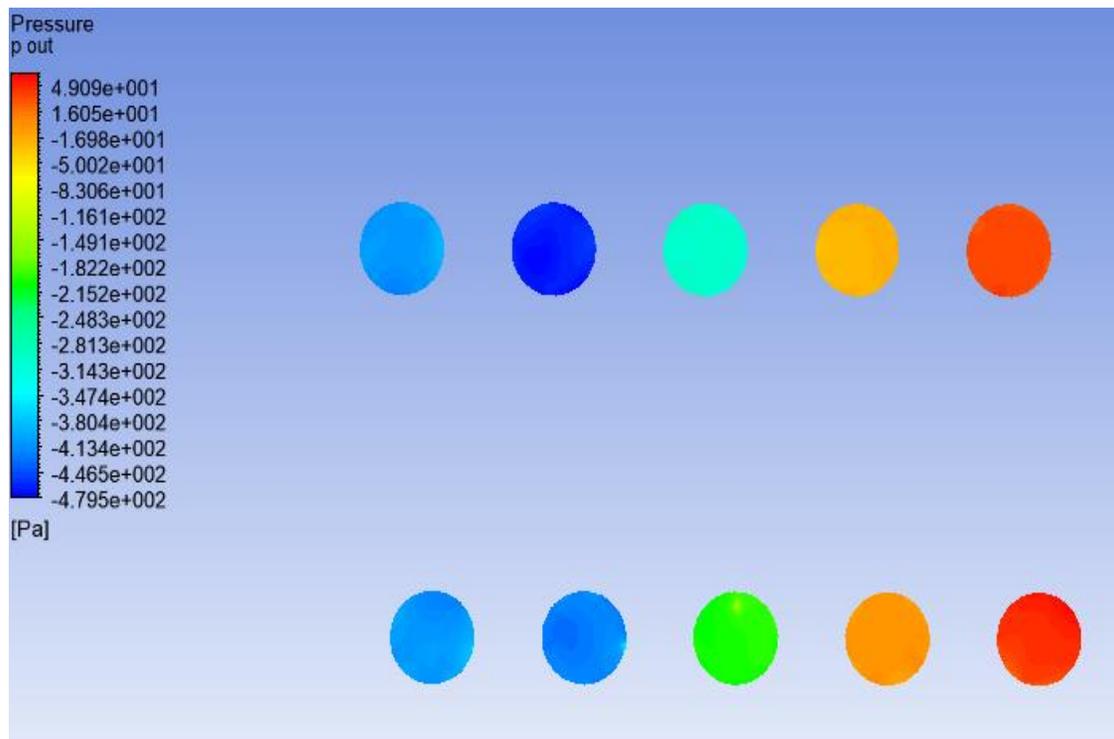
##### Campi di pressione



"Images used courtesy of ANSYS, Inc."

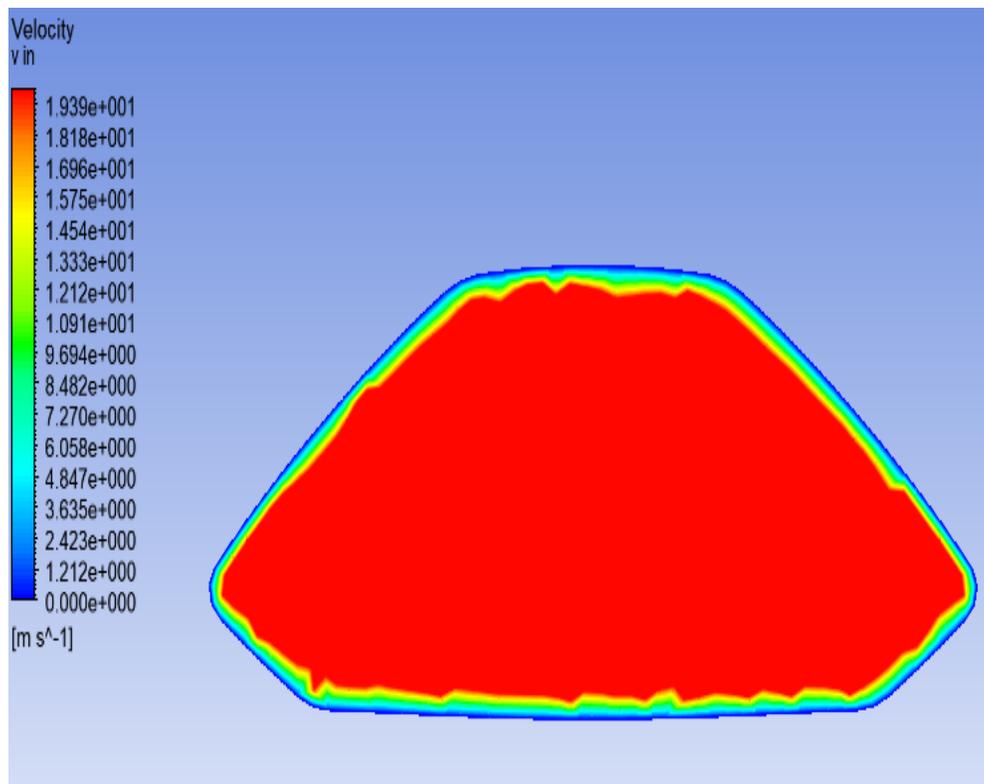


"Images used courtesy of ANSYS, Inc."

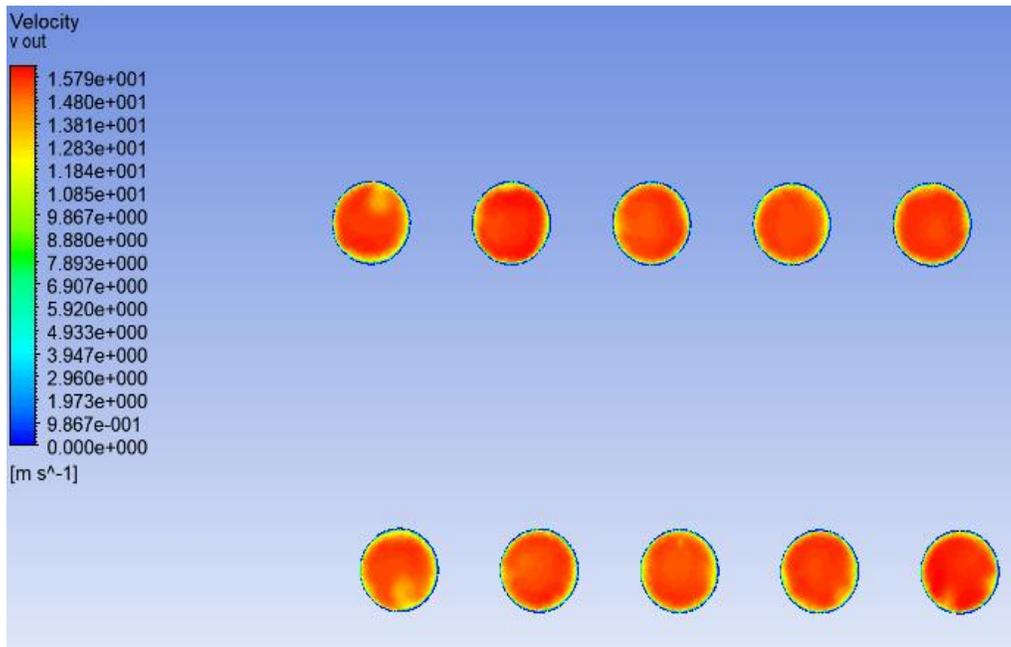


"Images used courtesy of ANSYS, Inc."

### Campi di velocità



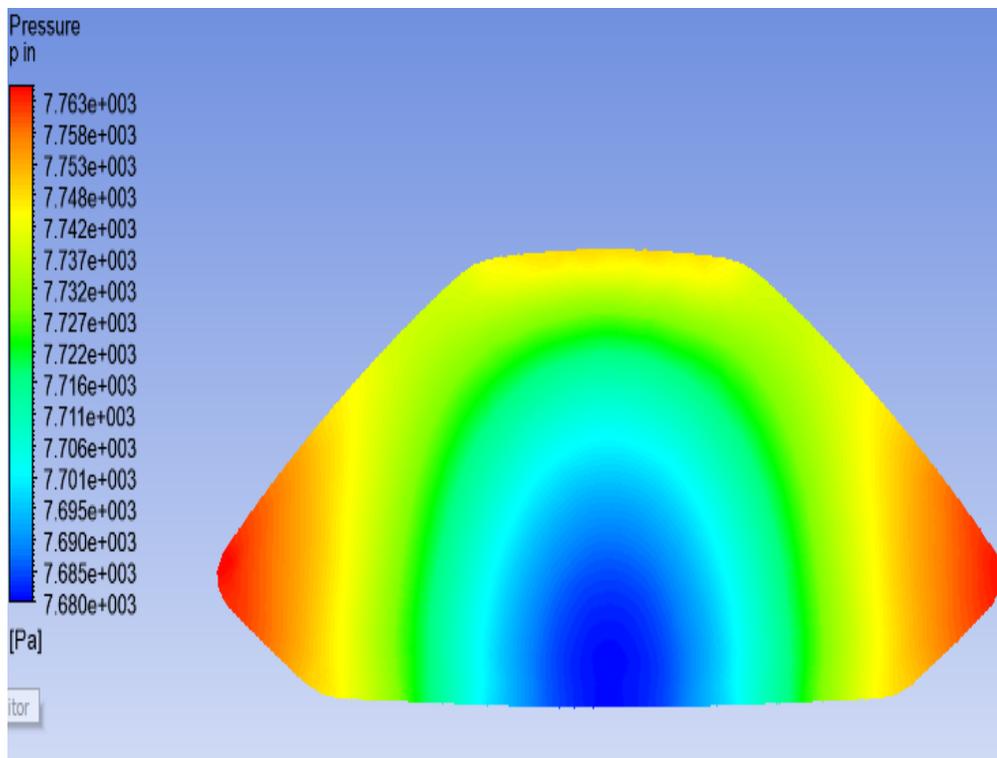
"Images used courtesy of ANSYS, Inc."



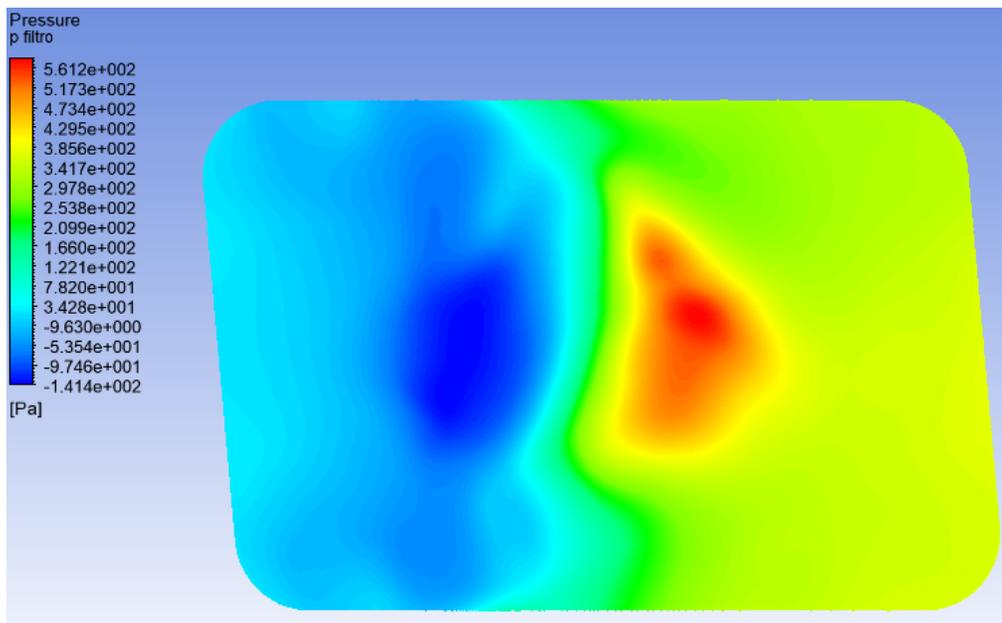
"Images used courtesy of ANSYS, Inc."

### 3.3.1.2 WOPI APERTO CON PORTATA IN MASSA DI 0,3534 Kg/s

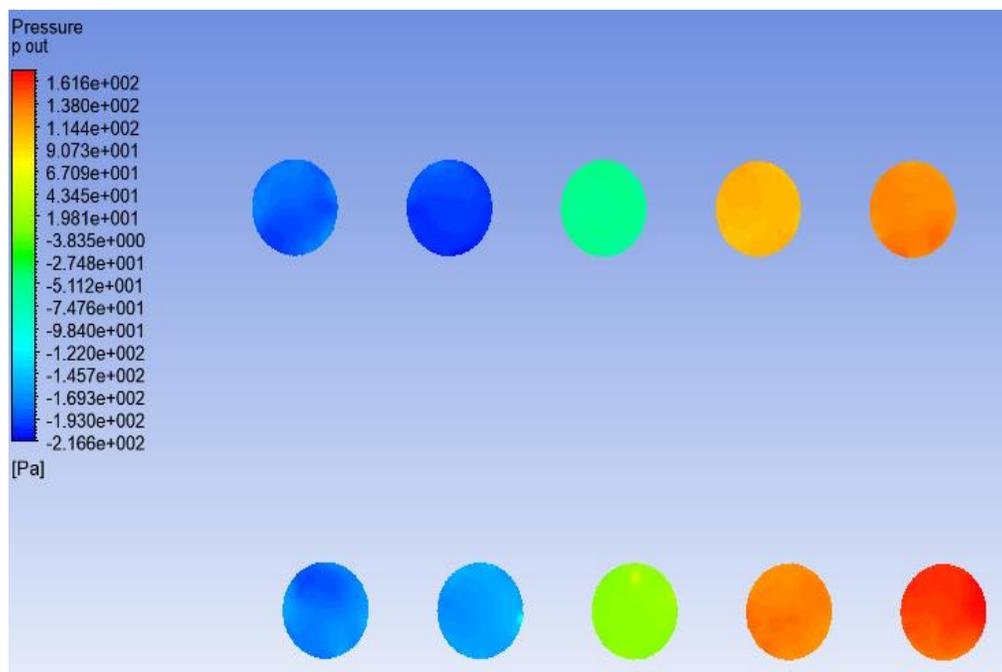
#### Campi di pressione



"Images used courtesy of ANSYS, Inc."

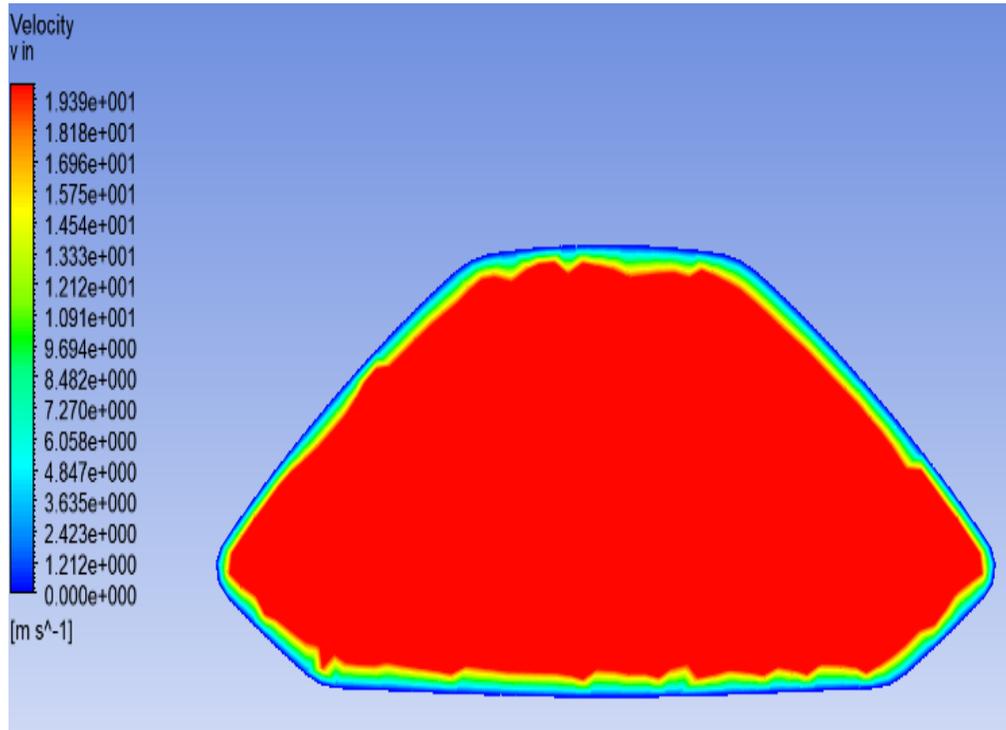


"Images used courtesy of ANSYS, Inc."

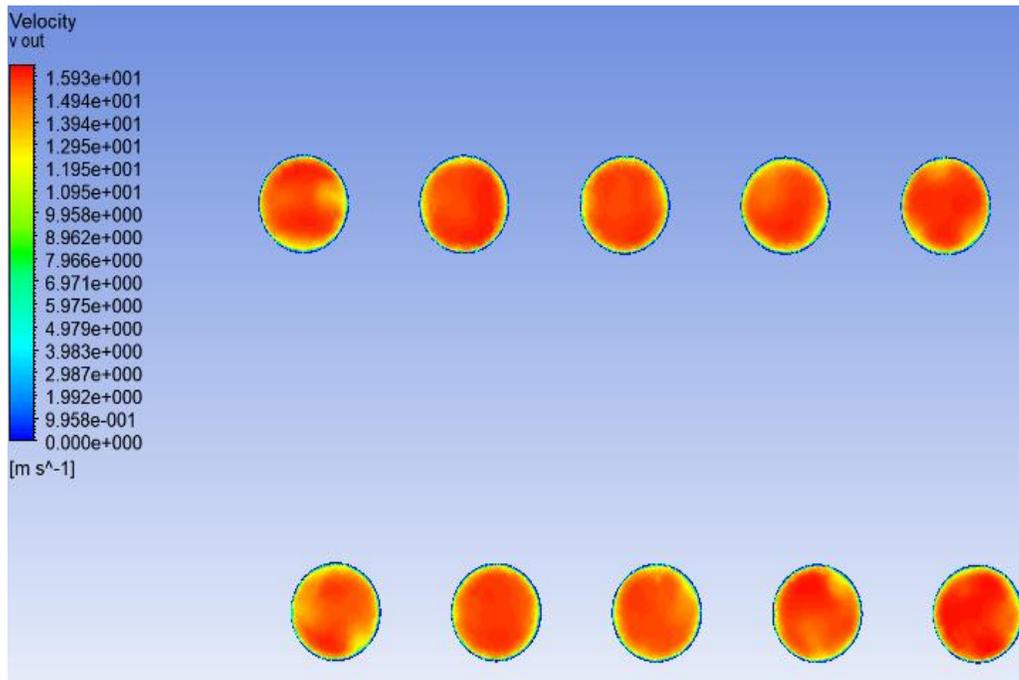


"Images used courtesy of ANSYS, Inc."

### Campi di velocità



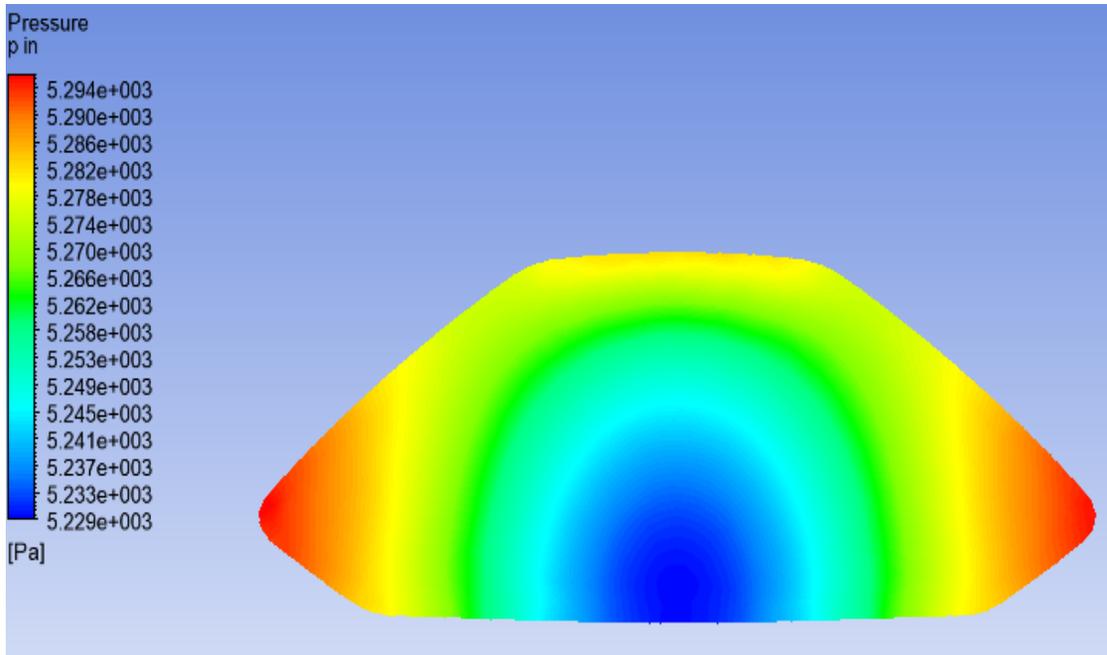
"Images used courtesy of ANSYS, Inc."



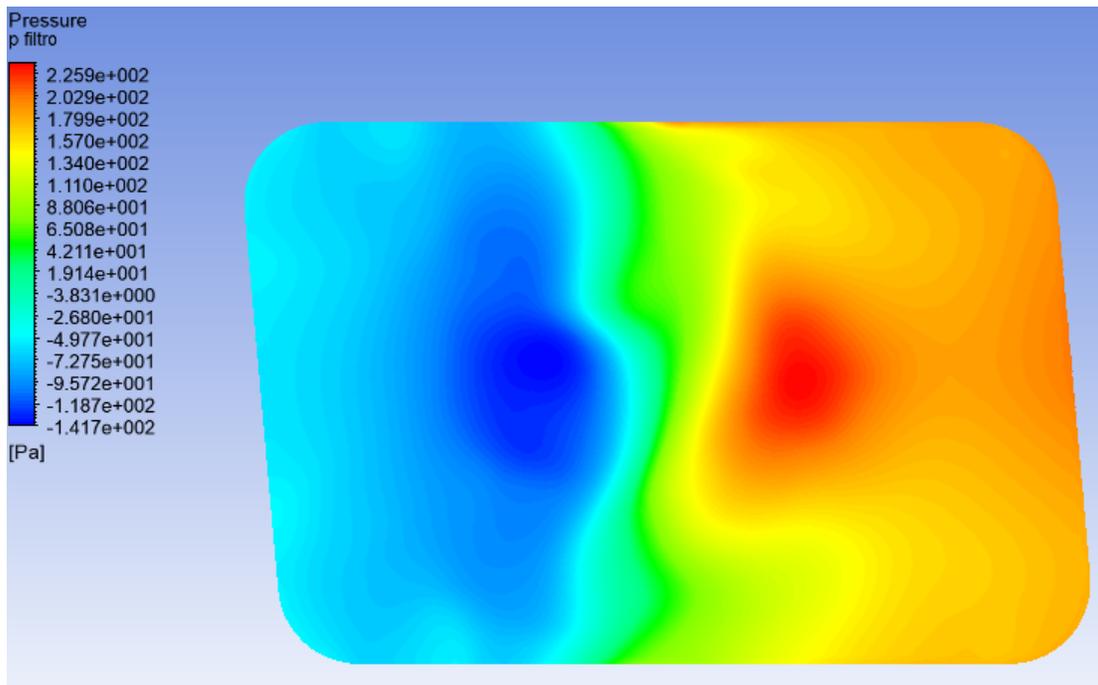
"Images used courtesy of ANSYS, Inc."

### 3.3.1.3 WOPI CHIUSO CON PORTATA IN MASSA DI 0,3098 Kg/s

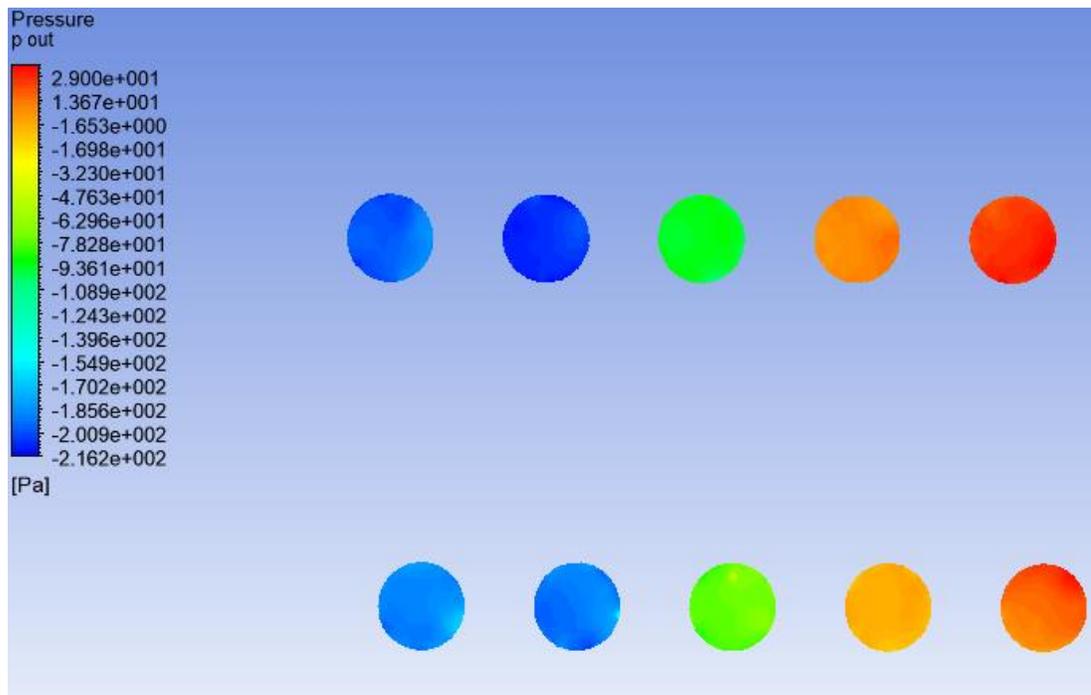
#### Campi di pressione



"Images used courtesy of ANSYS, Inc."

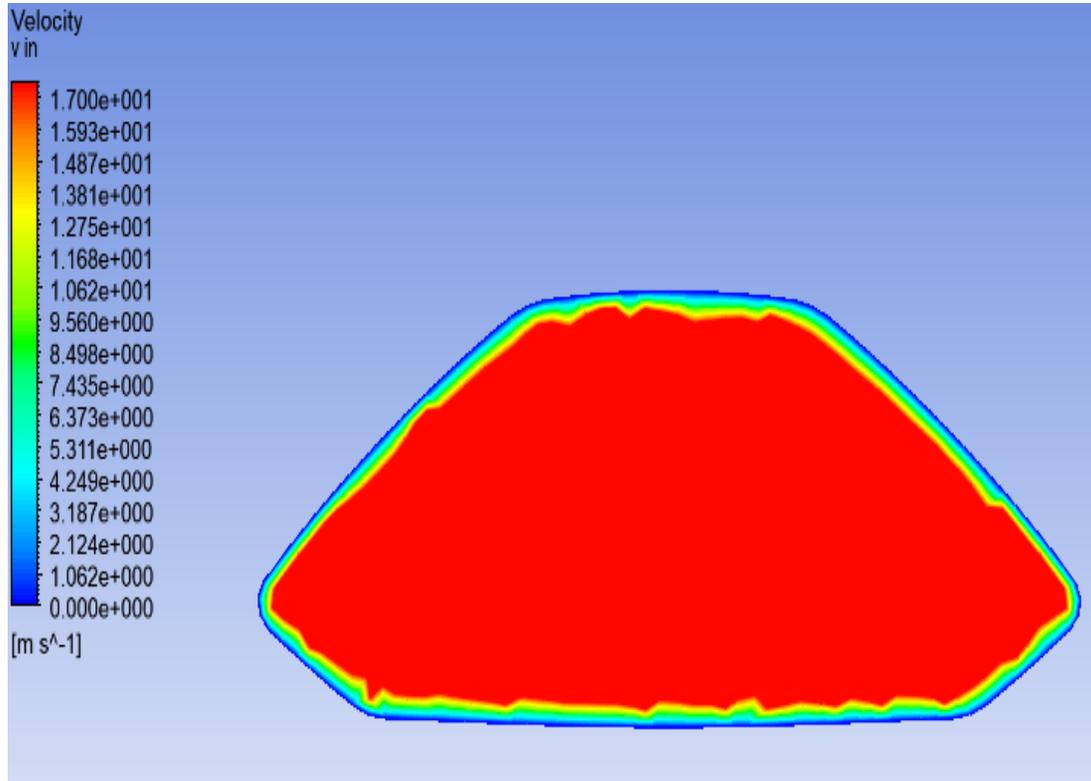


"Images used courtesy of ANSYS, Inc."

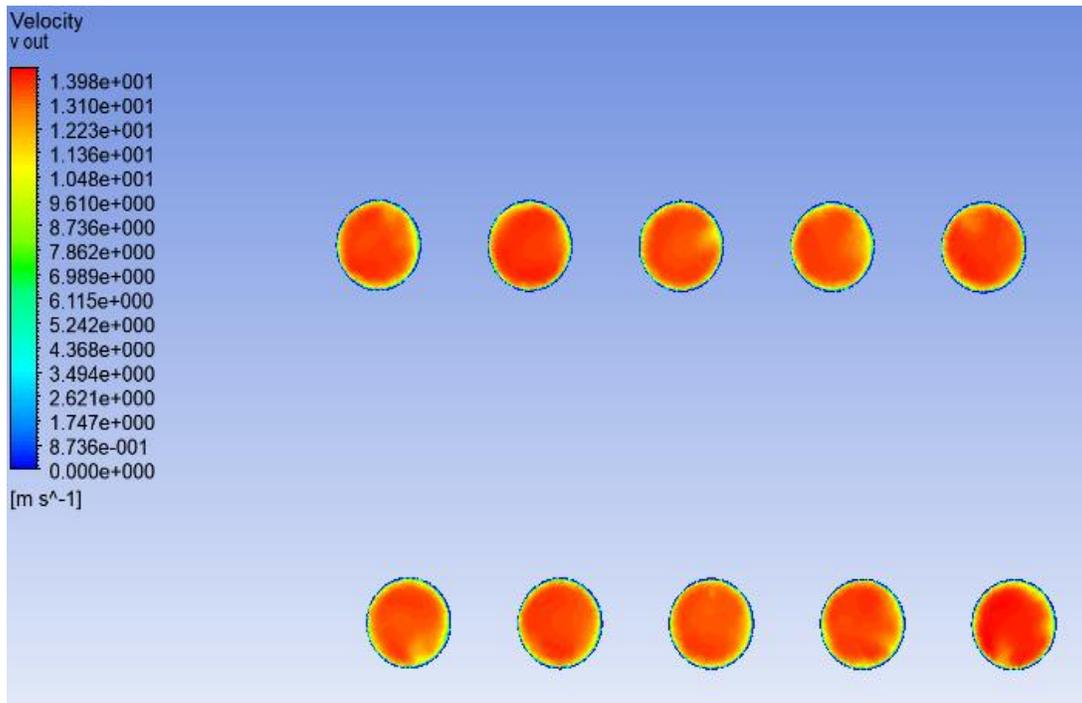


"Images used courtesy of ANSYS, Inc."

### Campi di velocità



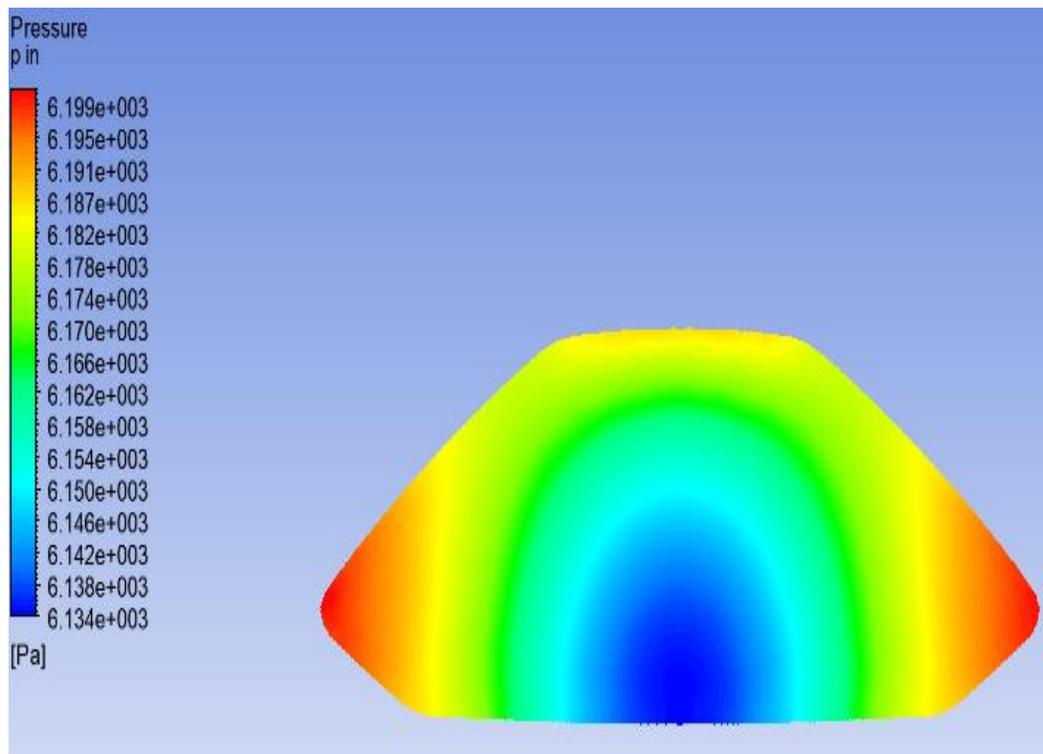
"Images used courtesy of ANSYS, Inc."



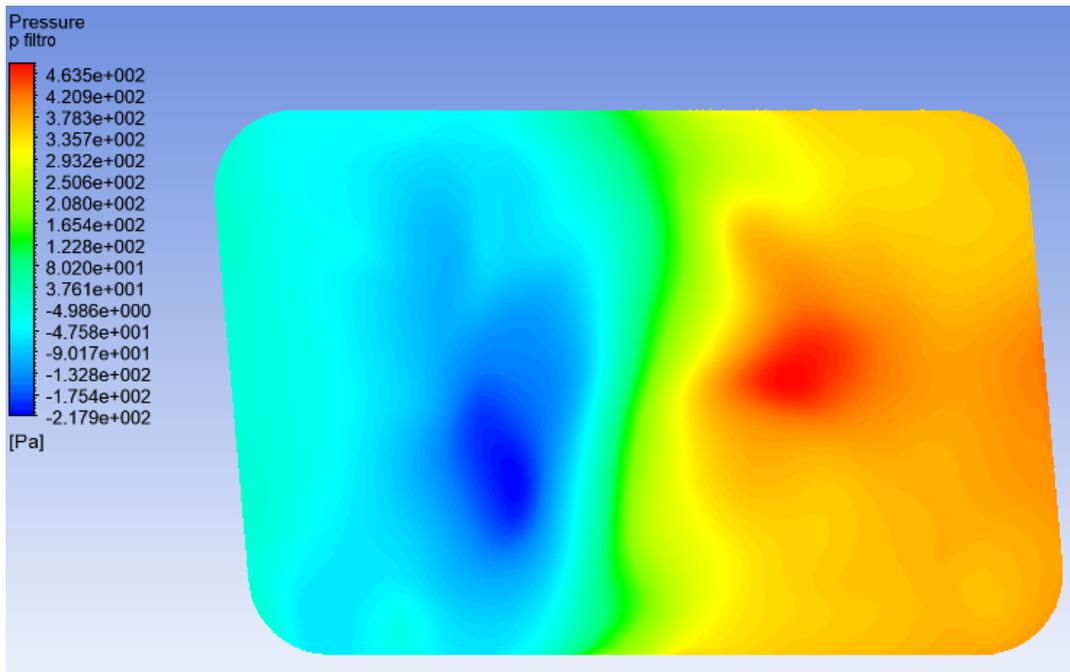
"Images used courtesy of ANSYS, Inc."

### 3.3.1.4 WOPI APERTO CON PORTATA IN MASSA DI 0,3098 Kg/s

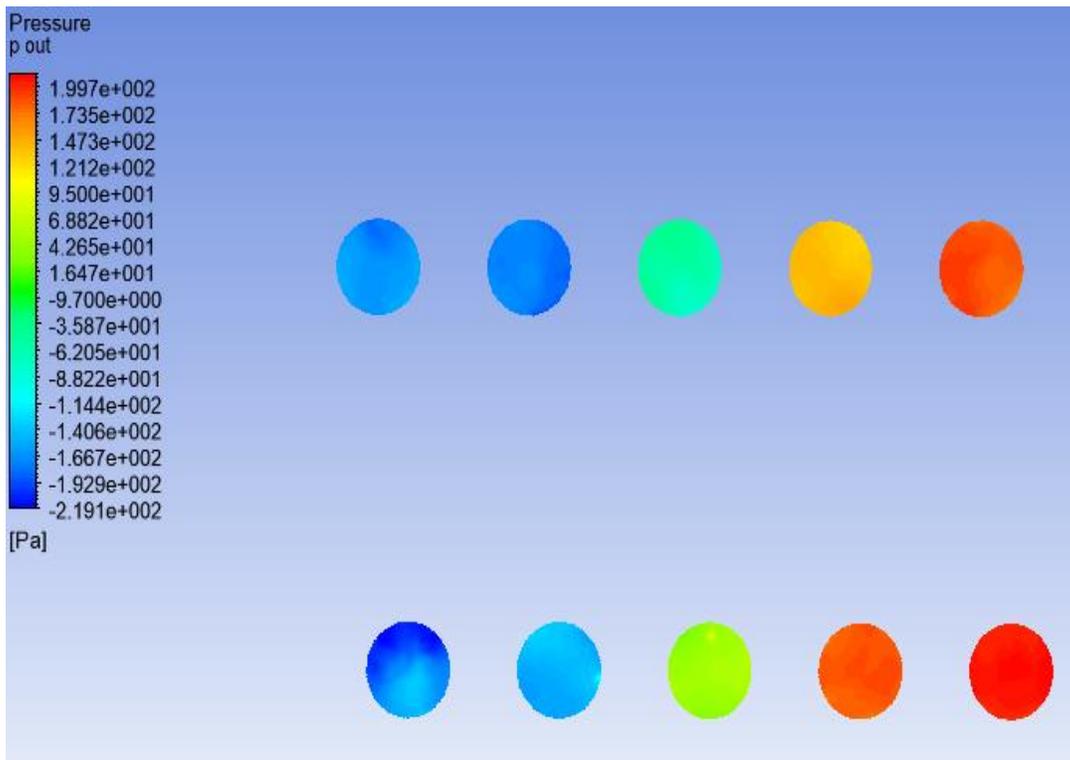
#### Campi di pressione



"Images used courtesy of ANSYS, Inc."

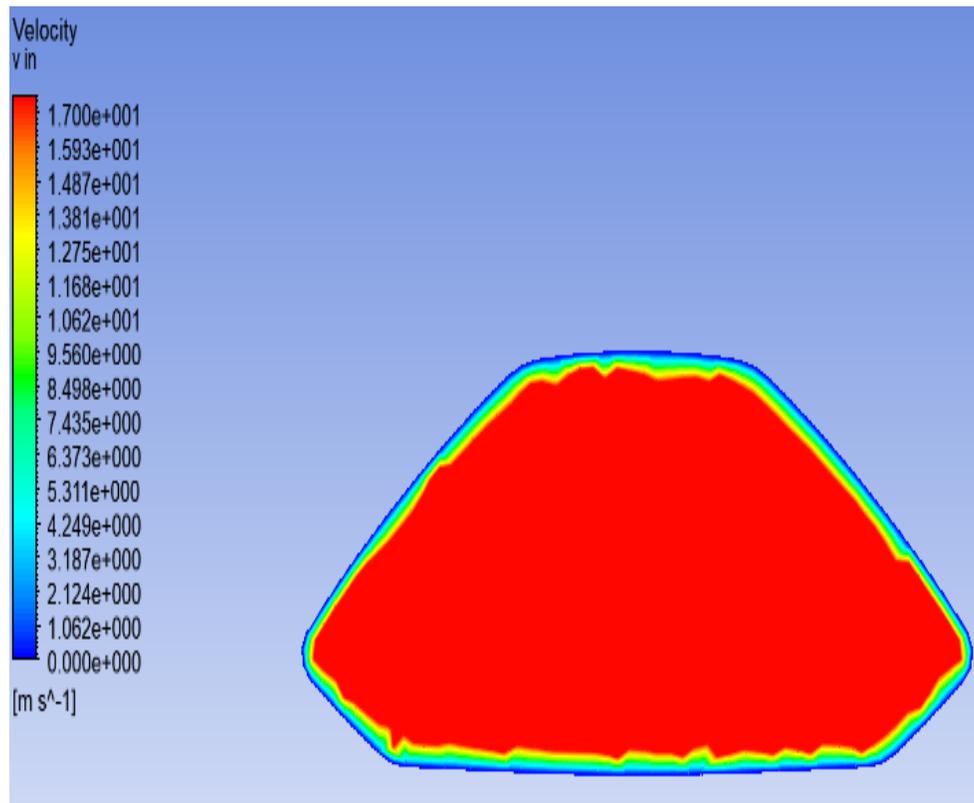


"Images used courtesy of ANSYS, Inc."

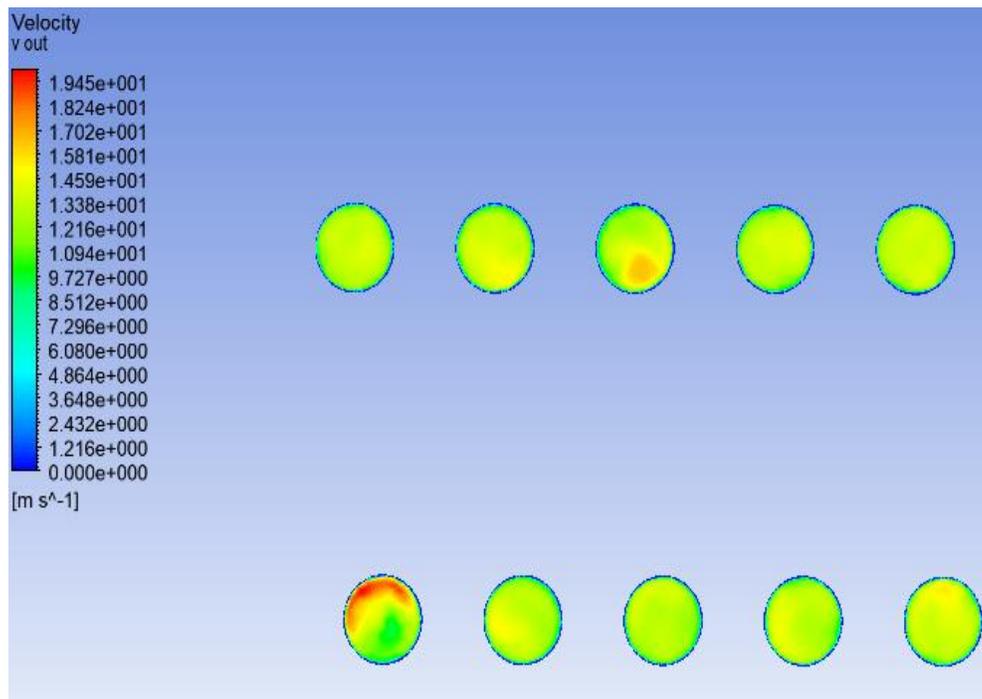


"Images used courtesy of ANSYS, Inc."

### Campi di velocità



"Images used courtesy of ANSYS, Inc."

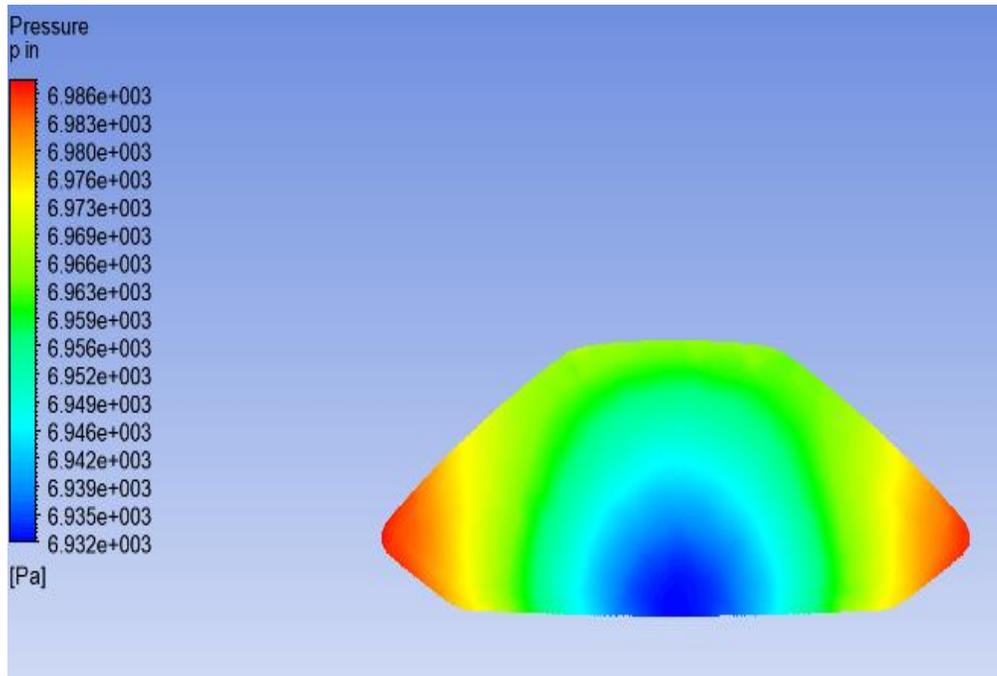


"Images used courtesy of ANSYS, Inc."

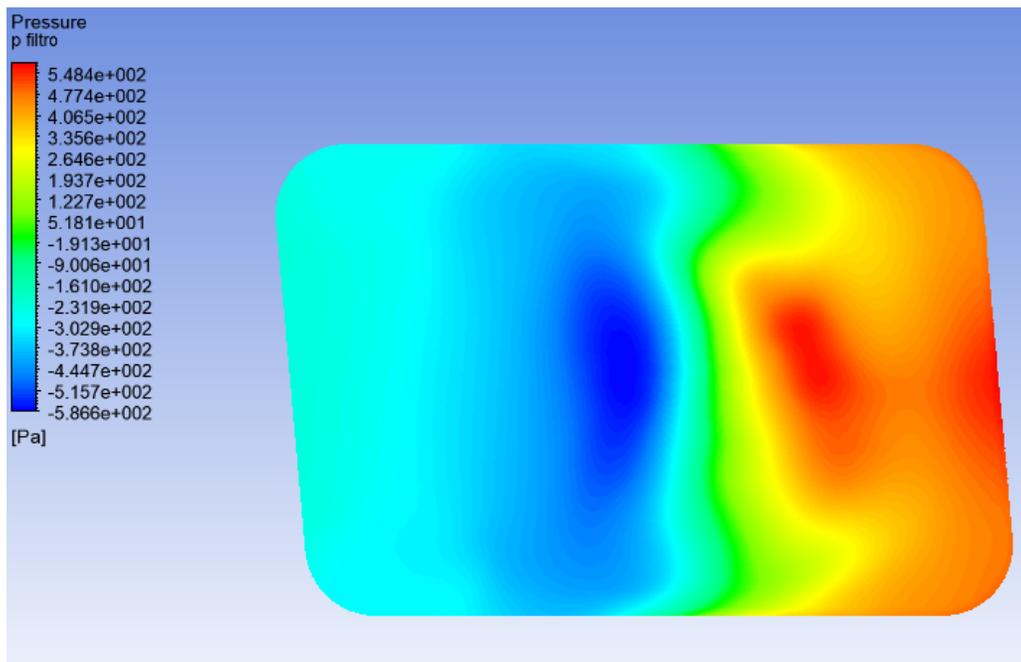
### 3.3.2 RISULTATI PER MODELLO CON TUBO DI FLUSSO (FORZATA)

#### 3.3.2.1 WOPI CHIUSO CON PORTATA IN MASSA DI 0,3534 Kg/s

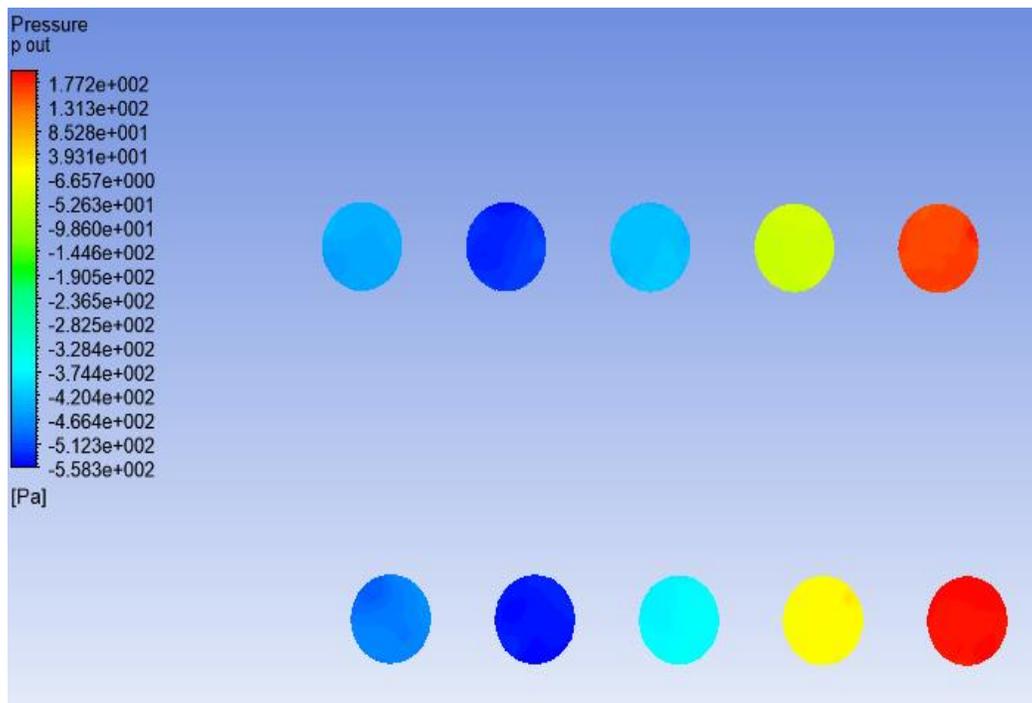
##### Campi di pressione



"Images used courtesy of ANSYS, Inc."

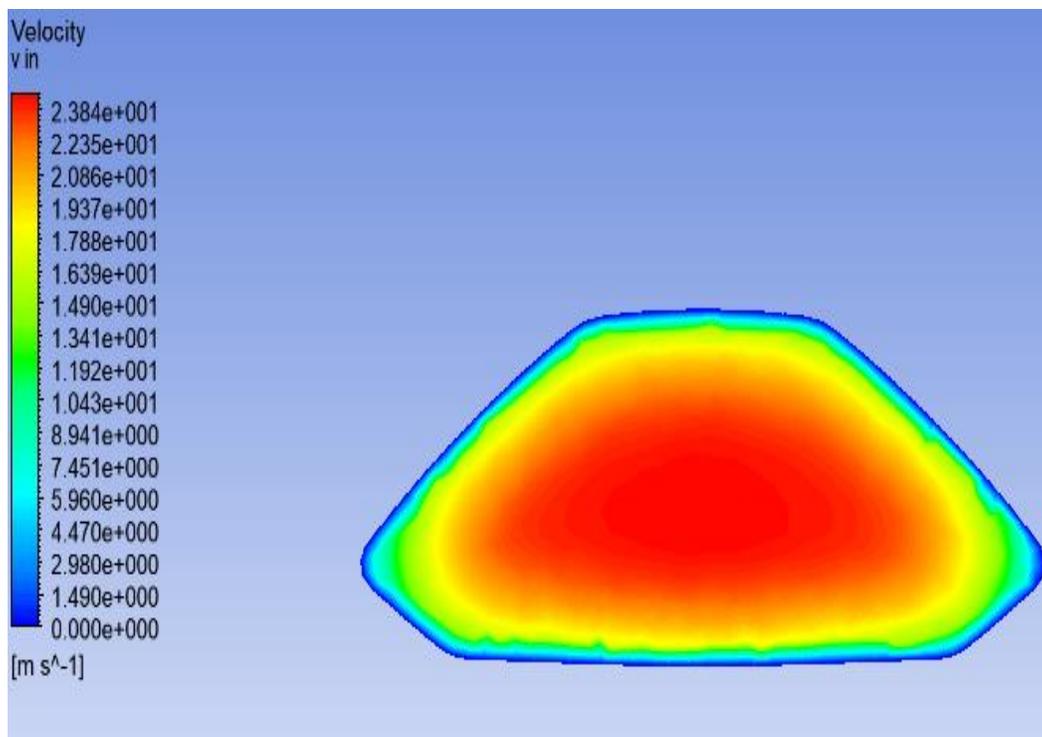


"Images used courtesy of ANSYS, Inc."

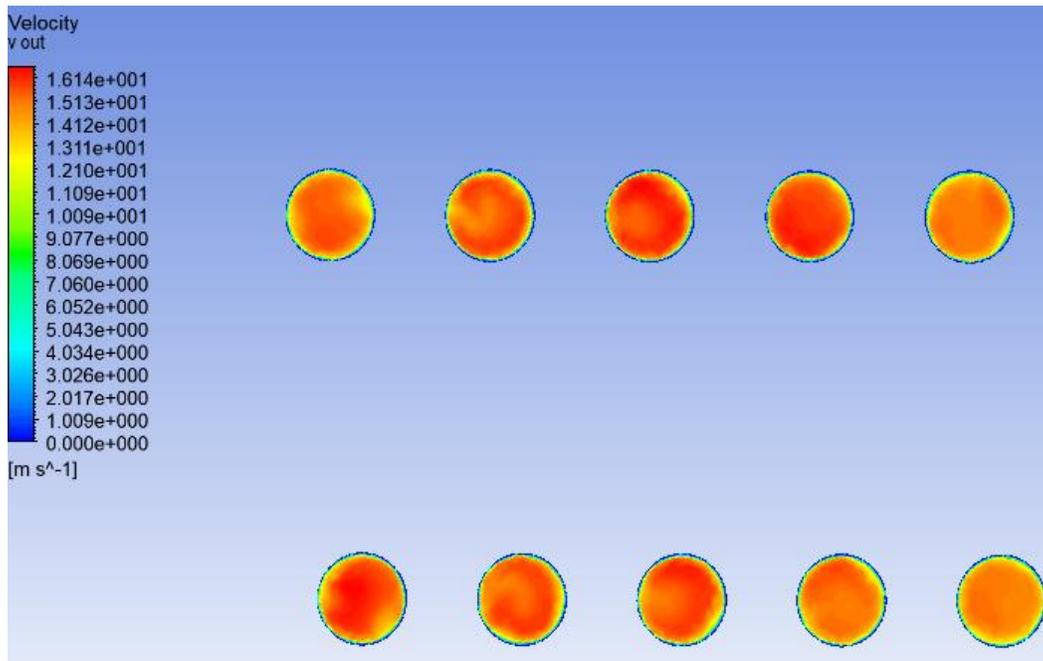


"Images used courtesy of ANSYS, Inc."

### Campi di velocità



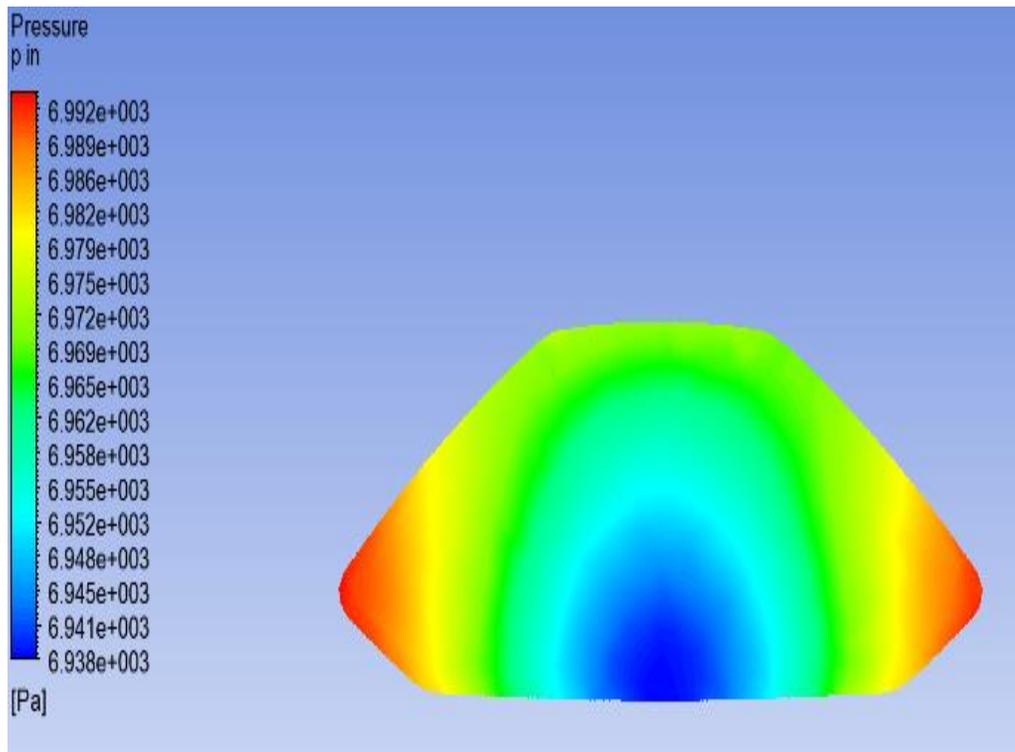
"Images used courtesy of ANSYS, Inc."



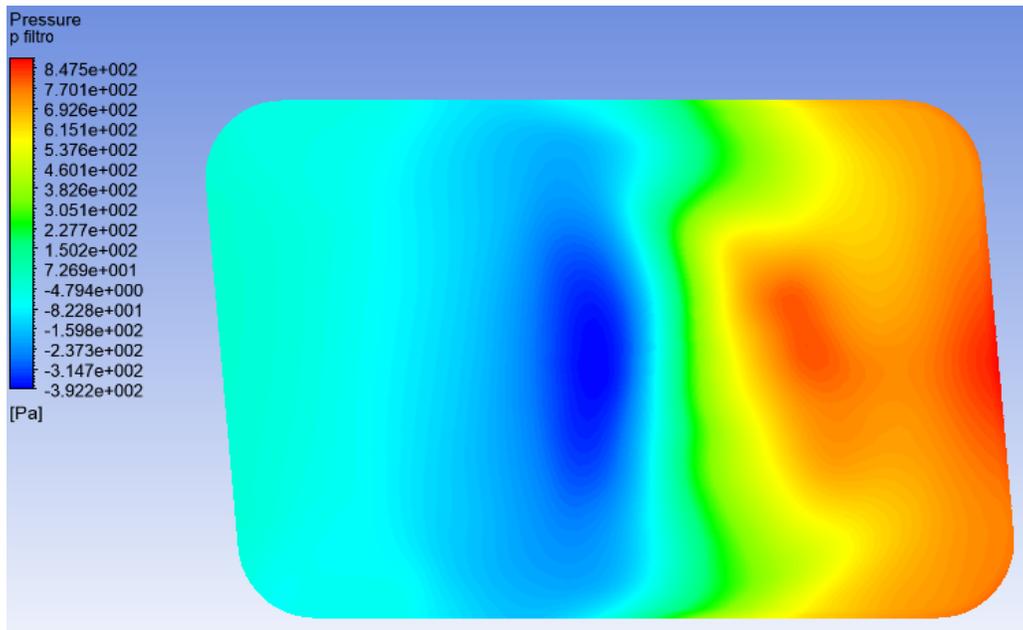
"Images used courtesy of ANSYS, Inc."

### 3.3.2.2 WOPI APERTO CON PORTATA IN MASSA DI 0,3534 Kg/s

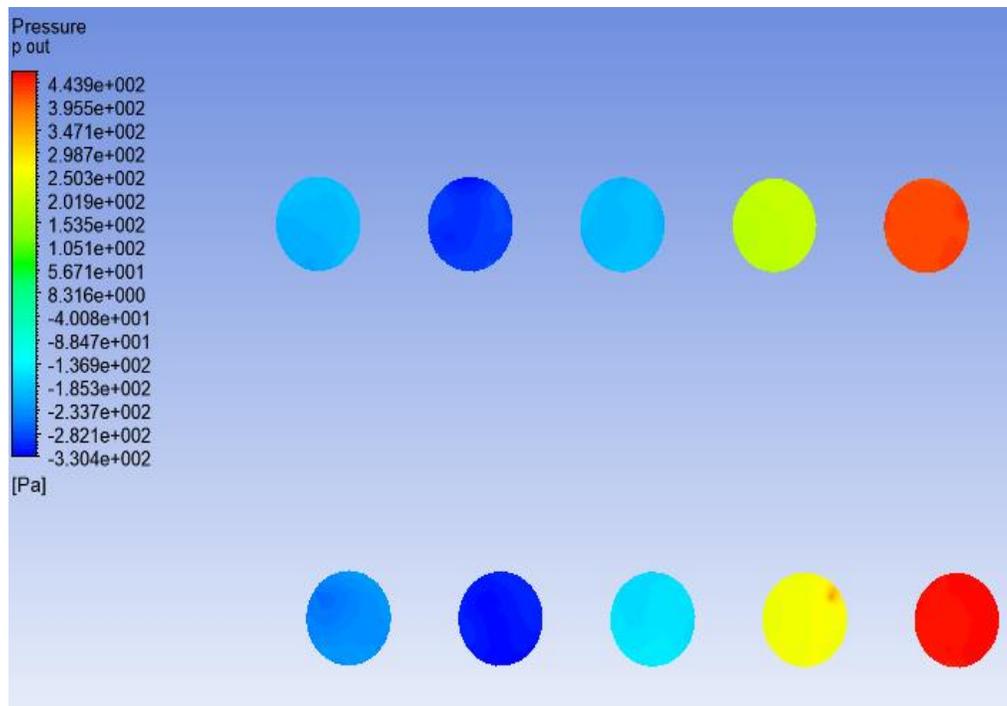
#### Campi di pressione



"Images used courtesy of ANSYS, Inc."

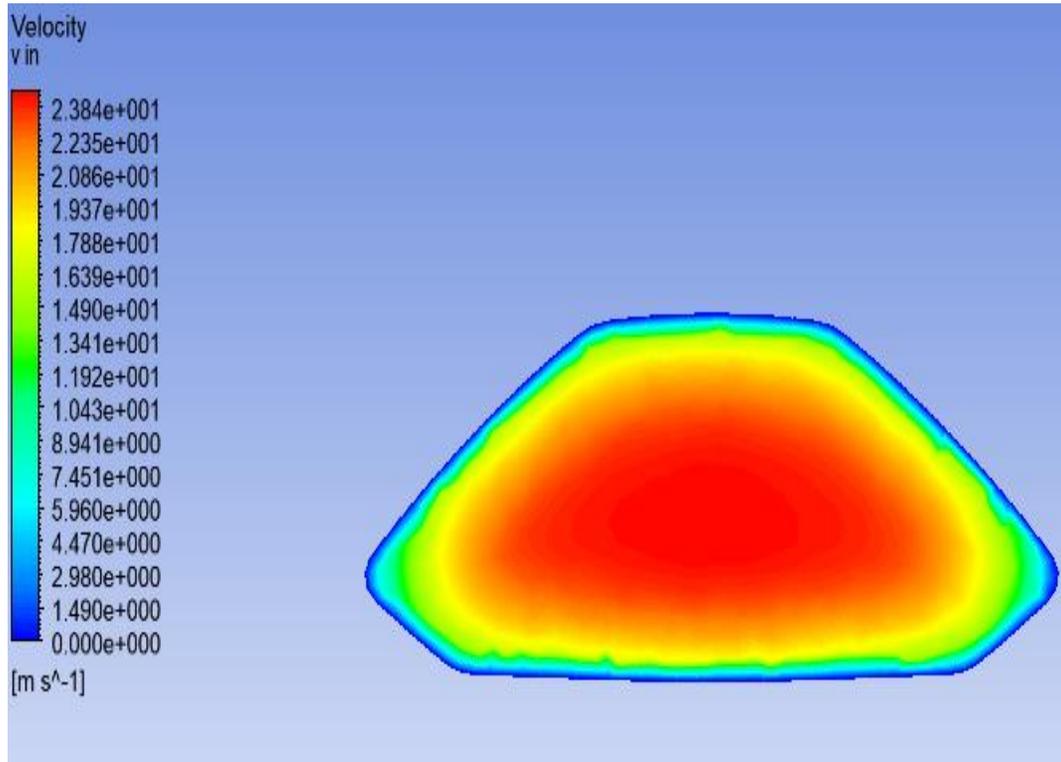


"Images used courtesy of ANSYS, Inc."

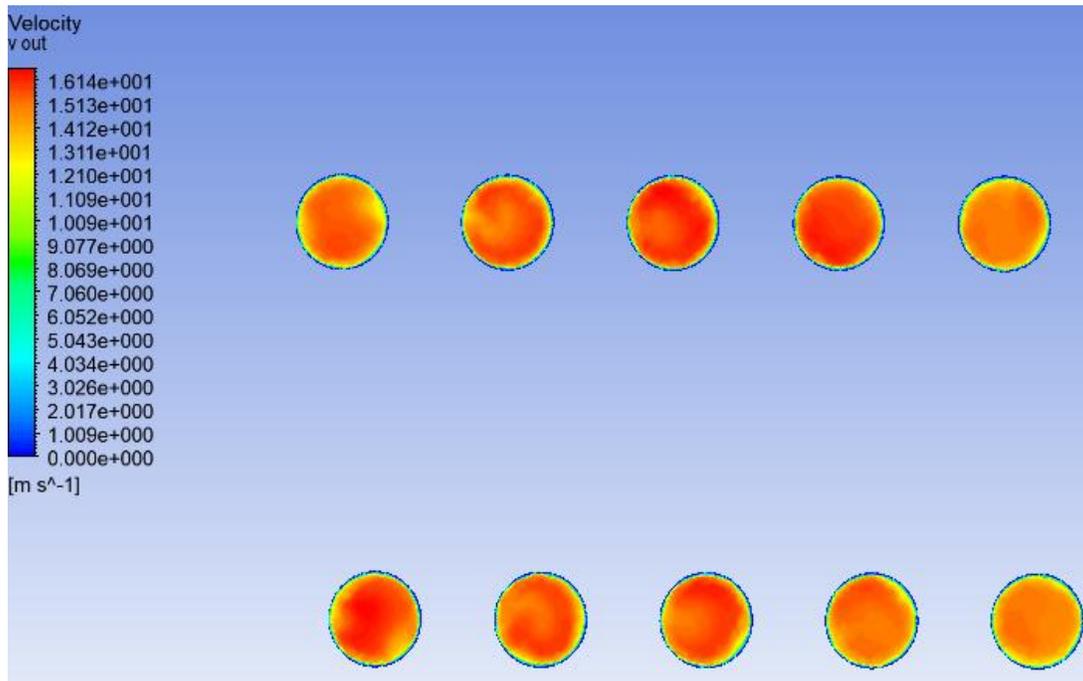


"Images used courtesy of ANSYS, Inc."

### Campi di velocità



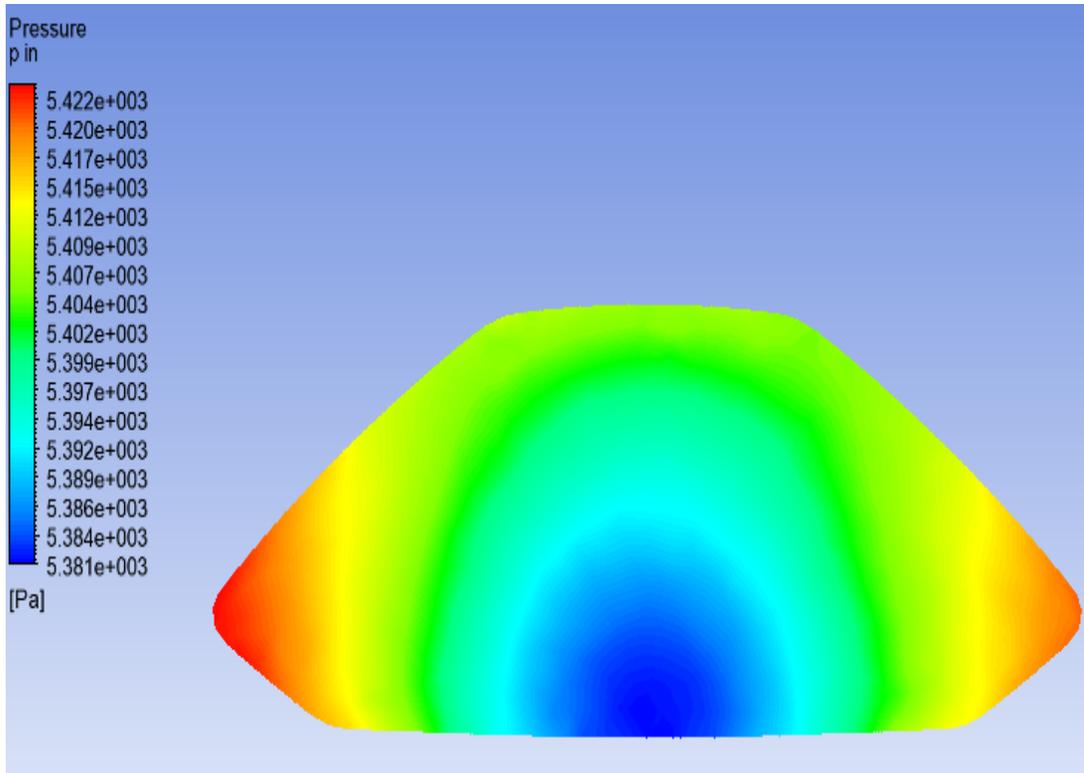
"Images used courtesy of ANSYS, Inc."



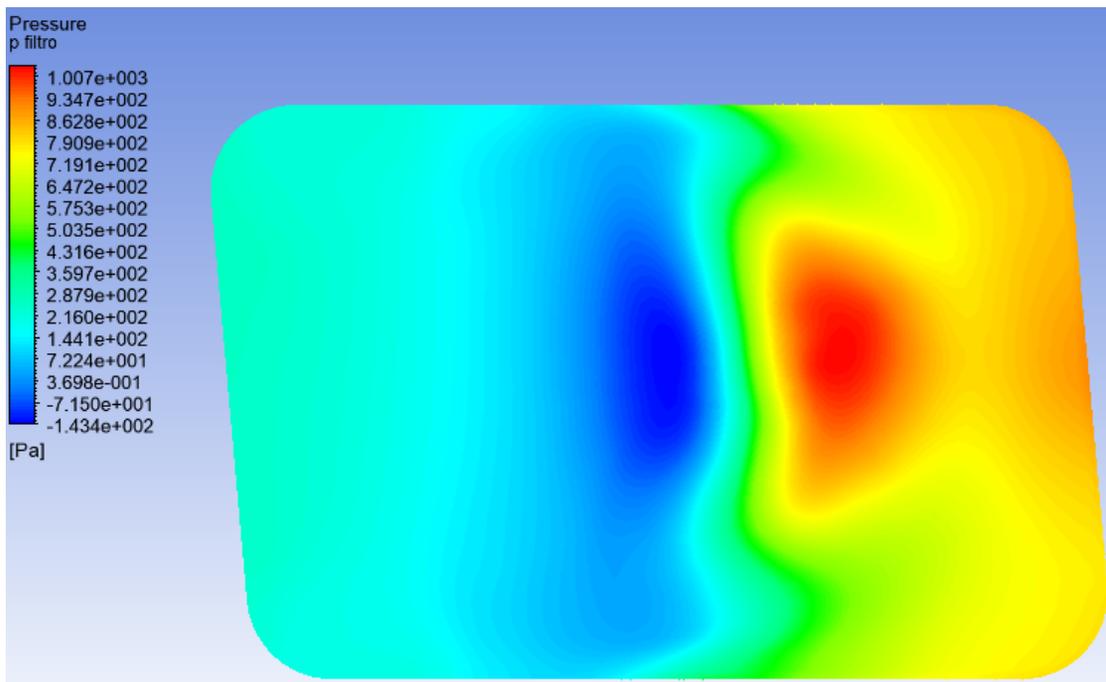
"Images used courtesy of ANSYS, Inc."

### 3.3.2.3 WOPI CHIUSO CON PORTATA IN MASSA DI 0,3098 Kg/s

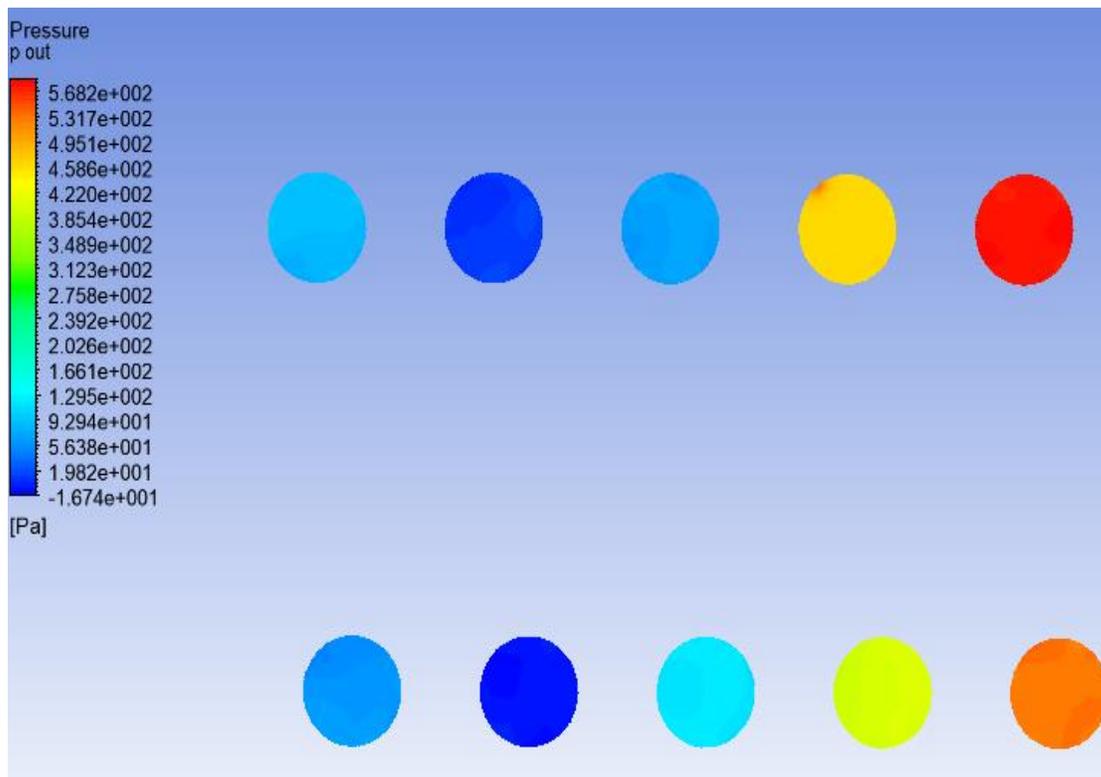
#### Campi di pressione



"Images used courtesy of ANSYS, Inc."

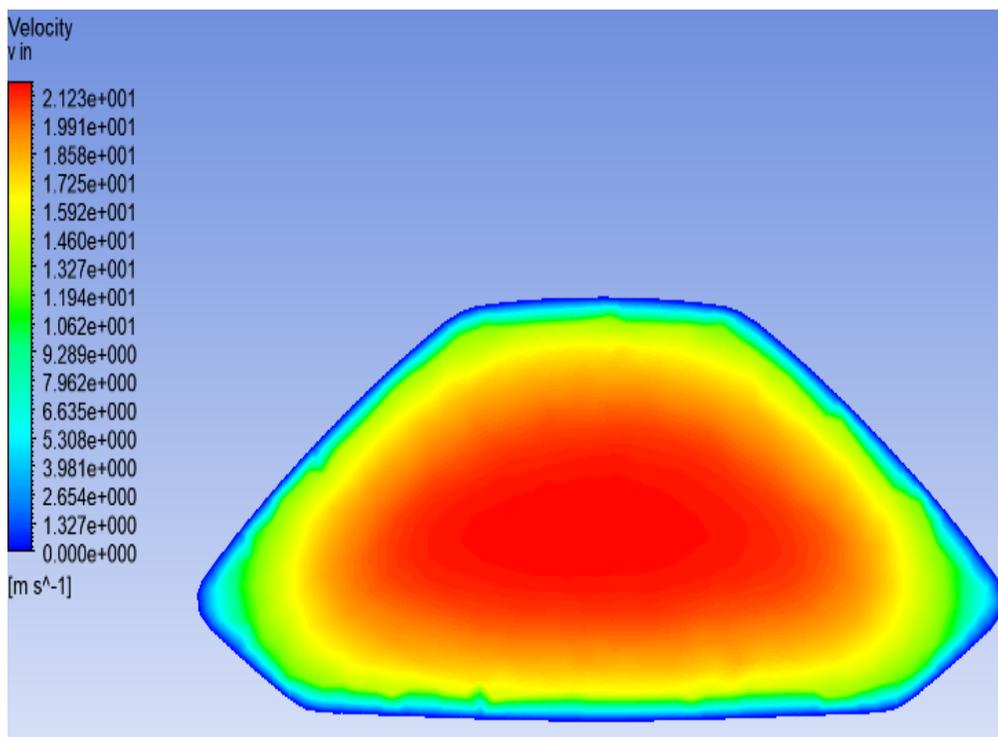


"Images used courtesy of ANSYS, Inc."

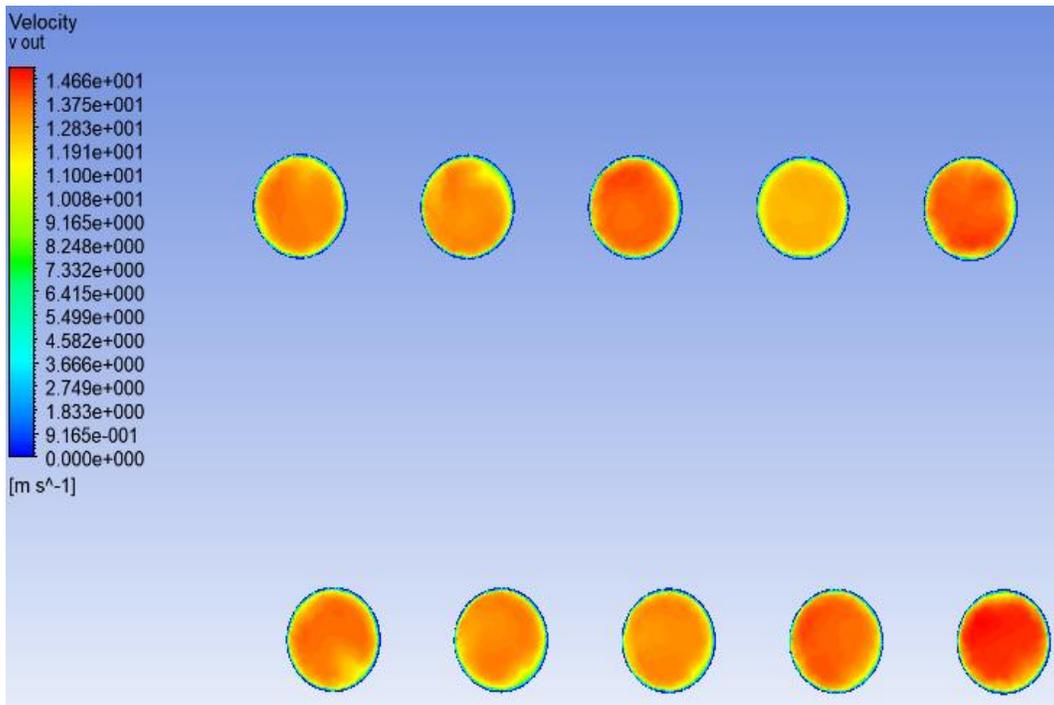


"Images used courtesy of ANSYS, Inc."

### Campi di velocità



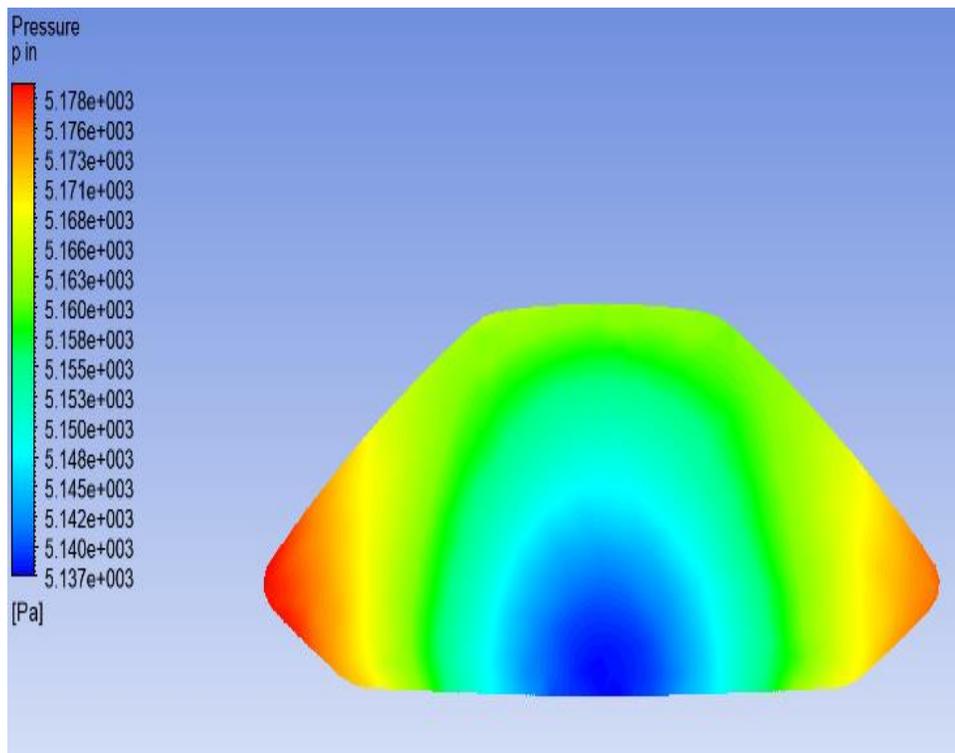
"Images used courtesy of ANSYS, Inc."



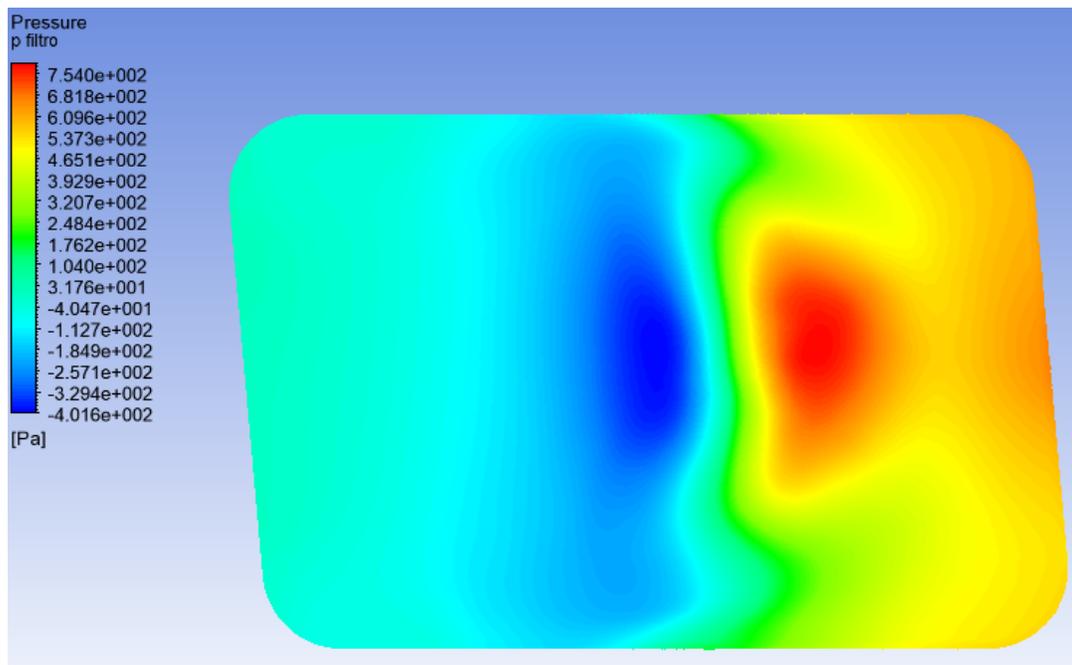
"Images used courtesy of ANSYS, Inc."

### 3.3.2.4 WOPI APERTO CON PORTATA IN MASSA DI 0,3098 Kg/s

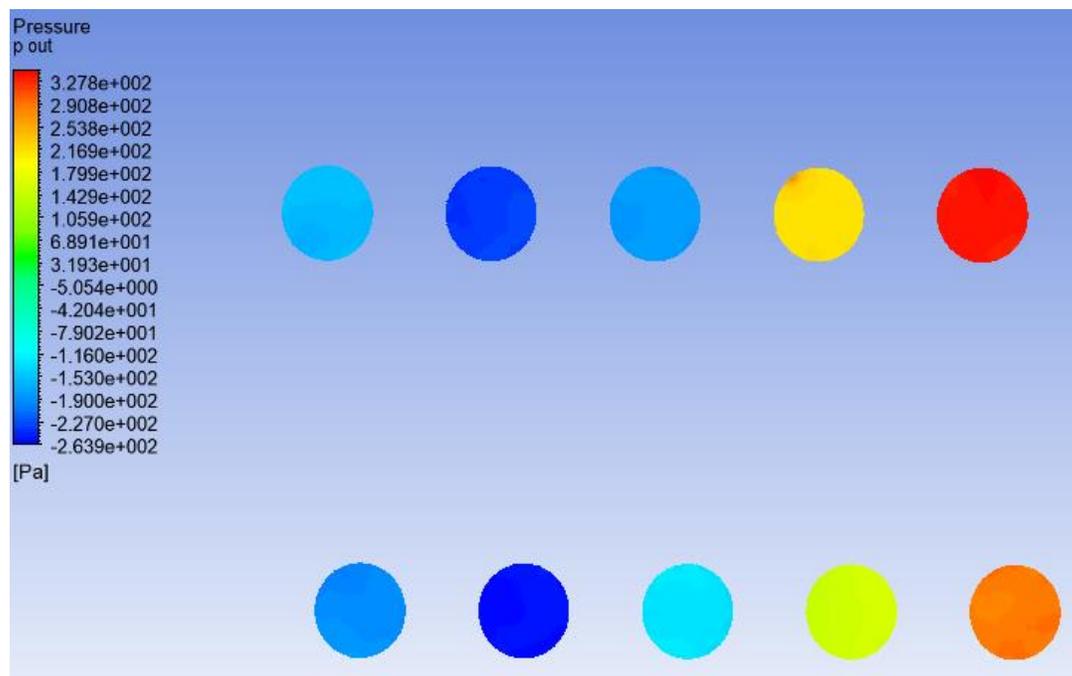
#### Campi di pressione



"Images used courtesy of ANSYS, Inc."

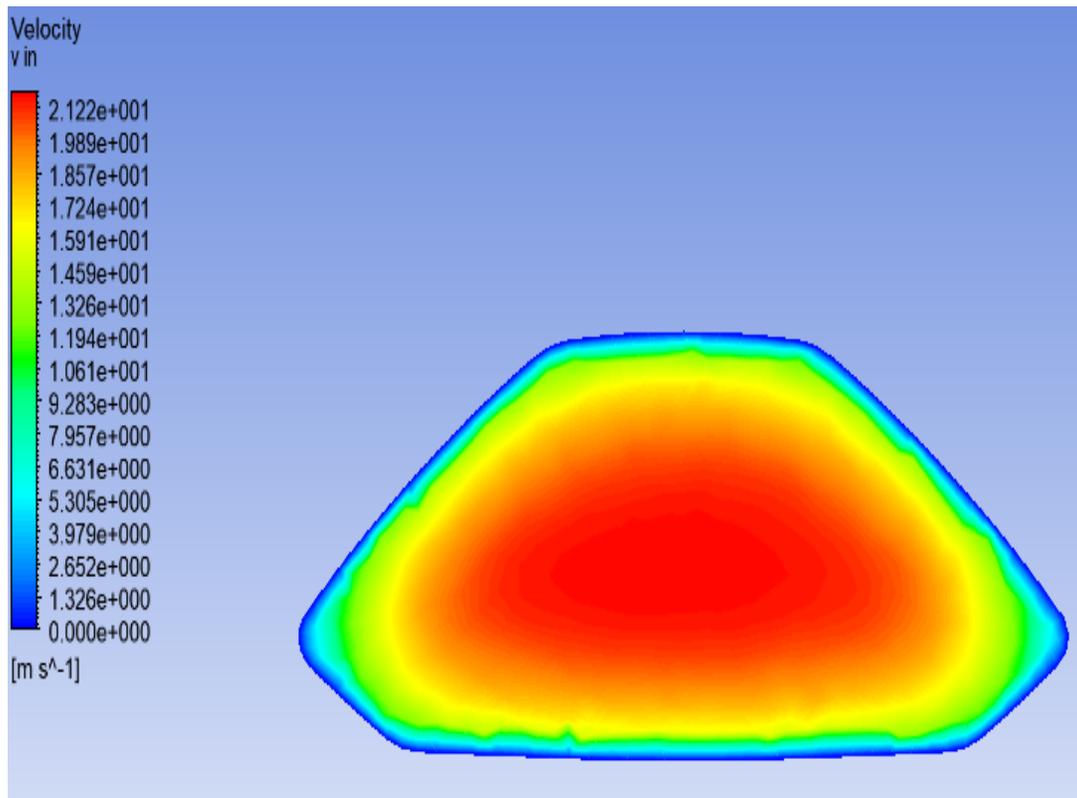


"Images used courtesy of ANSYS, Inc."

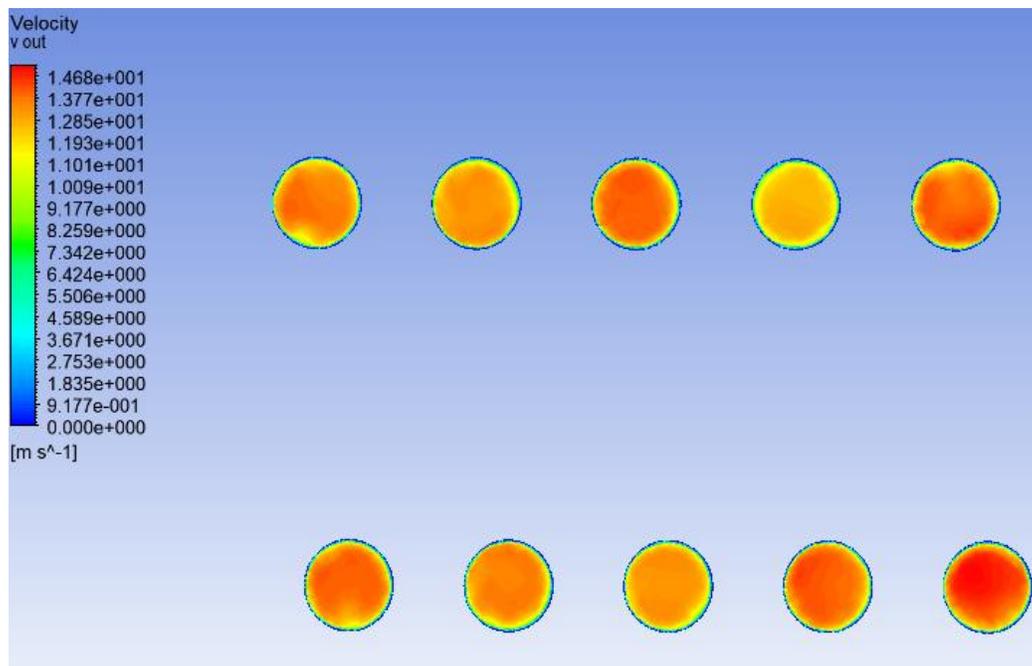


"Images used courtesy of ANSYS, Inc."

### Campi di velocità



"Images used courtesy of ANSYS, Inc."



"Images used courtesy of ANSYS, Inc."

### 3.3.3 ANALISI DEI RISULTATI

Più le mesh sono state infittite, minore è risultato il range delle grandezze misurate. Con 125'000 celle infatti, con andamenti analoghi a quelli visti nei campi con 500'000, si sono raggiunti i valori massimi e minimi dei campi di pressione, ma questo è normale in quanto con l'infittirsi della mesh, i risultati saranno sempre più convergenti ad un unico valore:

	M1	M1	M0,6	M0,6
	WOPI chiuso	WOPI aperto	WOPI chiuso	WOPI aperto
<b>Pressione filtro</b>	7/5109 Pa	-163/4843 Pa	-2/4065 Pa	-130/4184 Pa
<b>Pressione In</b>	8214/8294 Pa	8276/8356 Pa	6503/6565 Pa	6396/6457 Pa
<b>Pressione Out</b>	-261/1502 Pa	-410/1242 Pa	-238/1170 Pa	-310/1072 Pa

**Tabella 3.6:** Esempi dei range ottenuti dei campi di pressione e velocità

Di seguito sono stati riportati i valori medi di ogni grandezza fisica che è stata monitorata nelle varie zone dell'airbox. Infine, nell'ultima riga viene stato rappresentato il valore della caduta di pressione tra ingresso ed uscita, vero metro di misura della convenienza o meno nell'attrezzare un'airbox di un sistema WOPI.

#### 3.3.3.1 VALORI MEDI SIMULAZIONI CON TUBO DI FLUSSO

- **125'000 CELLE**

	M1	M1	M0,6	M0,6
	WOPI chiuso	WOPI aperto	WOPI chiuso	WOPI aperto
<b>Velocità ingresso</b>	20.07 m/s	20.07 m/s	17.61 m/s	17.61 m/s
<b>Velocità uscita</b>	15.25 m/s	15.25 m/s	13.29 m/s	13.27 m/s
<b>Pressione filtro</b>	123 Pa	130 Pa	310 Pa	118 Pa
<b>Pressione In</b>	5981 Pa	6052 Pa	4467 Pa	4349 Pa
<b>Pressione Out</b>	-13 Pa	-12 Pa	117 Pa	-78 Pa
<b><math>\Delta P</math></b>	5994 Pa	6064 Pa	4350 Pa	4427 Pa

**Tabella 3.7:** Valori medi di pressione e velocità

- **250'000 CELLE**

	M1	M1	M0,6	M0,6
	WOPI chiuso	WOPI aperto	WOPI chiuso	WOPI aperto
<b>Velocità ingresso</b>	20.07 m/s	20.07 m/s	17.63 m/s	17.63 m/s
<b>Velocità uscita</b>	15.04 m/s	15.09 m/s	13.19 m/s	13.19 m/s
<b>Pressione filtro</b>	125 Pa	189 Pa	329 Pa	165 Pa
<b>Pressione In</b>	5298 Pa	5497 Pa	4507 Pa	4368 Pa
<b>Pressione Out</b>	-12 Pa	-51 Pa	156 Pa	-5 Pa
<b><math>\Delta P</math></b>	5310 Pa	5548 Pa	4351 Pa	4373 Pa

**Tabella 3.8:** Valori medi di pressione e velocità

- **500'000 CELLE**

	M1	M1	M0,6	M0,6
	WOPI chiuso	WOPI aperto	WOPI chiuso	WOPI aperto
<b>Velocità ingresso</b>	20.02 m/s	20.02m/s	17.55 m/s	17.55 m/s
<b>Velocità uscita</b>	14.78m/s	14.78 m/s	12.96 m/s	12.95 m/s
<b>Pressione filtro</b>	186 Pa	186 Pa	387 Pa	137 Pa
<b>Pressione In</b>	6965 Pa	6975 Pa	5402 Pa	5158 Pa
<b>Pressione Out</b>	-7 Pa	-8 Pa	230 Pa	-17 Pa
<b><math>\Delta P</math></b>	6972 Pa	6983 Pa	5172 Pa	5175 Pa

**Tabella 3.9:** Valori medi di pressione e velocità

### 3.3.3.2 VALORI MEDI SIMULAZIONI SENZA TUBO DI FLUSSO

- 125'000 CELLE

	M1	M1	M0,6	M0,6
	WOPI chiuso	WOPI aperto	WOPI chiuso	WOPI aperto
<b>Velocità ingresso</b>	20 m/s	20 m/s	17.54 m/s	17.54 m/s
<b>Velocità uscita</b>	15.05 m/s	15.12 m/s	13.24 m/s	13.06 m/s
<b>Pressione filtro</b>	452 Pa	264 Pa	351 Pa	200 Pa
<b>Pressione In</b>	8258 Pa	8319 Pa	6536 Pa	6429 Pa
<b>Pressione Out</b>	206 Pa	12 Pa	159 Pa	17 Pa
<b><math>\Delta P</math></b>	8052 Pa	8307 Pa	6377 Pa	6412 Pa

Tabella 3.10: Valori medi di pressione e velocità

- 250'000 CELLE

	M1	M1	M0,6	M0,6
	WOPI chiuso	WOPI aperto	WOPI chiuso	WOPI aperto
<b>Velocità ingresso</b>	20 m/s	20 m/s	17.53 m/s	17.53 m/s
<b>Velocità uscita</b>	14.86 m/s	14.79 m/s	12.97 m/s	13 m/s
<b>Pressione filtro</b>	26 Pa	118 Pa	14 Pa	111 Pa
<b>Pressione In</b>	8165 Pa	8201 Pa	6335 Pa	6753 Pa
<b>Pressione Out</b>	-107 Pa	-197 Pa	-160 Pa	-60 Pa
<b><math>\Delta P</math></b>	8272 Pa	8398 Pa	6495 Pa	6813 Pa

Tabella 3.11: Valori medi di pressione e velocità

- **500'000 CELLE**

	M1	M1	M0,6	M0,6
	WOPI chiuso	WOPI aperto	WOPI chiuso	WOPI aperto
<b>Velocità ingresso</b>	20 m/s	20 m/s	17.53 m/s	17.53 m/s
<b>Velocità uscita</b>	14.75 m/s	14.76 m/s	12.93m/s	13.05 m/s
<b>Pressione filtro</b>	-19 Pa	174 Pa	53 Pa	148 Pa
<b>Pressione In</b>	7443 Pa	7723 Pa	5263 Pa	5909 Pa
<b>Pressione Out</b>	-216 Pa	-24 Pa	-90 Pa	5 Pa
<b>ΔP</b>	7659 Pa	7747 Pa	5353 Pa	5904 Pa

**Tabella 3.12:** Valori medi di pressione e velocità

Si può notare come l'utilizzo o meno del tubo di flusso influisca quasi per nulla sui valori delle velocità, mentre per le pressioni invece sì, in maniera minore sui valori in uscita mentre molto di più su quelli in ingresso. Per questo conviene tirare le conclusioni sull'effettiva convenienza del WOPI coi valori medi di ogni campo di velocità e pressione ottenuti:

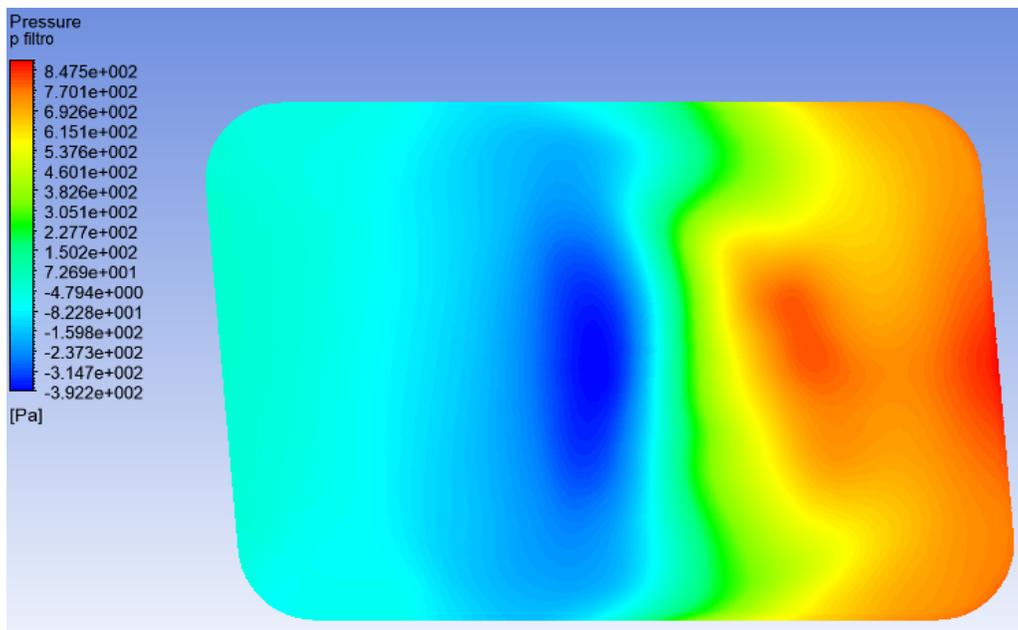
	M1	M1	M0,6	M0,6
	WOPI chiuso	WOPI aperto	WOPI chiuso	WOPI aperto
<b>Velocità ingresso</b>	20,03 m/s	20,03 m/s	17.57 m/s	17.57 m/s
<b>Velocità uscita</b>	14.96 m/s	14.97 m/s	13.10 m/s	13.09 m/s
<b>Pressione filtro</b>	149 Pa	177 Pa	241 Pa	147 Pa
<b>Pressione In</b>	7018 Pa	7128 Pa	5418 Pa	5494 Pa
<b>Pressione Out</b>	-25 Pa	-47 Pa	-69 Pa	-23 Pa
<b>ΔP</b>	7043 Pa	7175 Pa	5350 Pa	5517 Pa

**Tabella 3.13:** Valori medi di pressione e velocità

Quindi dalle simulazioni, si può dedurre che il sistema WOPI influisce positivamente sulla caduta di pressione tra ingresso ed uscita dell'airbox nell'ordine di  $10^2$  Pascal. Non rappresenta perciò un guadagno di pressione così eccezionale come si poteva sperare, ma comunque porta i suoi benefici rispetto ad una scatola dell'aria classica, tutto questo ovviamente in linea teorica.

Infatti, se si volessero ottenere risultati certi, sarà necessario in futuro eseguire dei test fisici (su banco di prova o su un veicolo “prototipo”).

Un’ultima considerazione va fatta sulla posizione della valvola di scarico dell’umidità. Viene riproposta di seguito un’immagine rappresentante il campo di pressione sopra all’ingresso d’aria sul filtro:



"Images used courtesy of ANSYS, Inc."

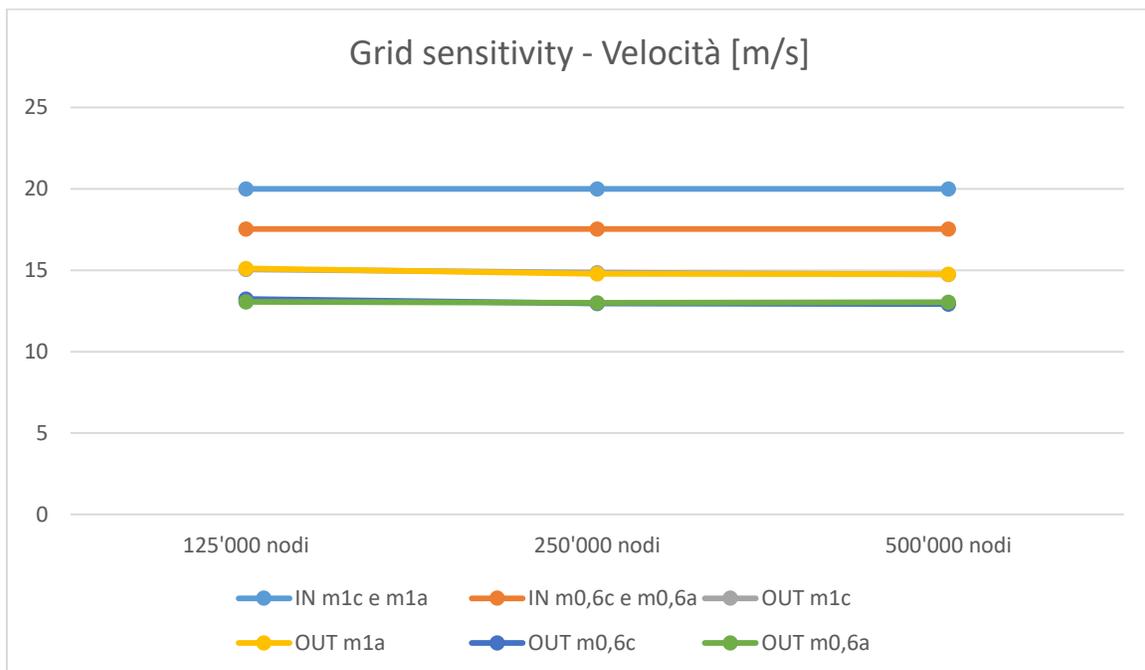
Come era stato rappresentato nel “layout approssimativo”, lo scarico dell’acqua veniva eseguito sul lato sinistro dell’airbox. Questa configurazione però non è più considerata ottimale, dato che in ogni simulazione eseguita, questa zona viene costantemente indicata con pressioni relative molto basse, se non leggermente negative. Questo può portare alla mancata apertura della valvola quando richiesto, data la non presenza di un differenziale di pressione consono all’operazione. Inoltre questo può portare alla crescita esponenziale del Kv del flusso attraverso la valvola, rendendo molto difficoltosa la ricerca di una valvola con Kv sufficientemente alto e di dimensioni ridotte,

Per questa ragione, la soluzione migliore sarà adottare un layout dell’airbox che preveda lo scarico dell’umidità sul lato destro del box.

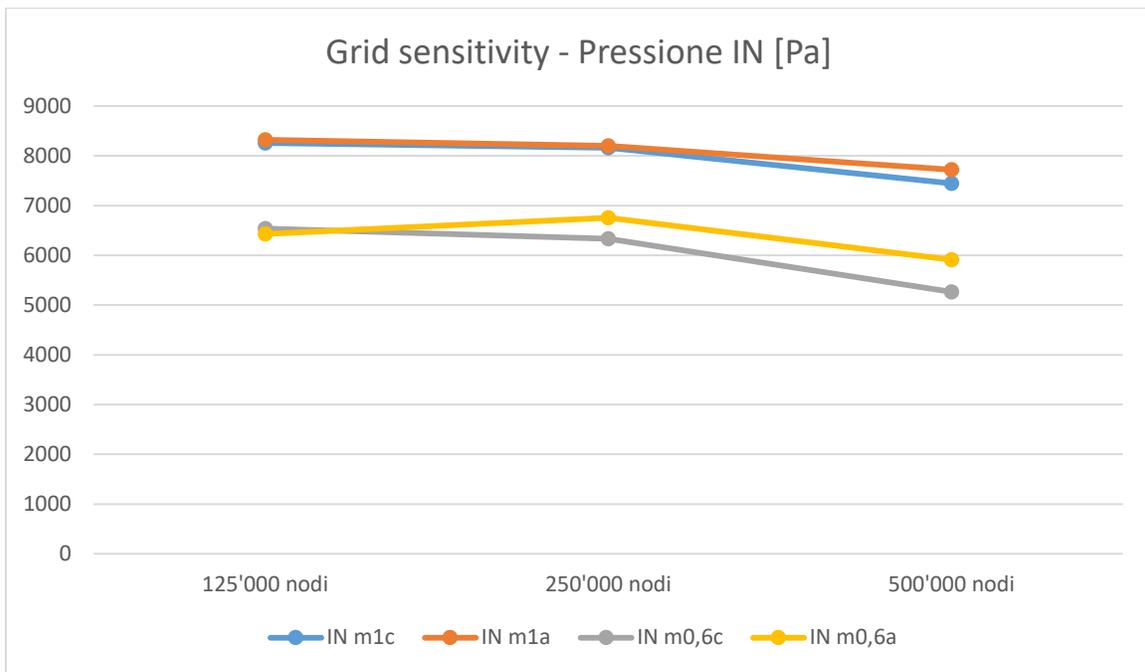
### 3.3.4 GRID SENSITIVITY

Come ultimo passaggio, va verificata la “grid sensitivity” delle simulazioni, ovvero, partendo dai risultati ottenuti con una mesh “grezza” bisogna accertarsi che aumentando la qualità della mesh, le soluzioni convergano tutte agli stessi valori, per poter accertare di aver eseguito delle simulazioni veritiere e trovare il valore minimo di dimensione della mesh che restituisca i risultati adatti, in modo da diminuire il peso computazionale e velocizzare il processo di simulazione.

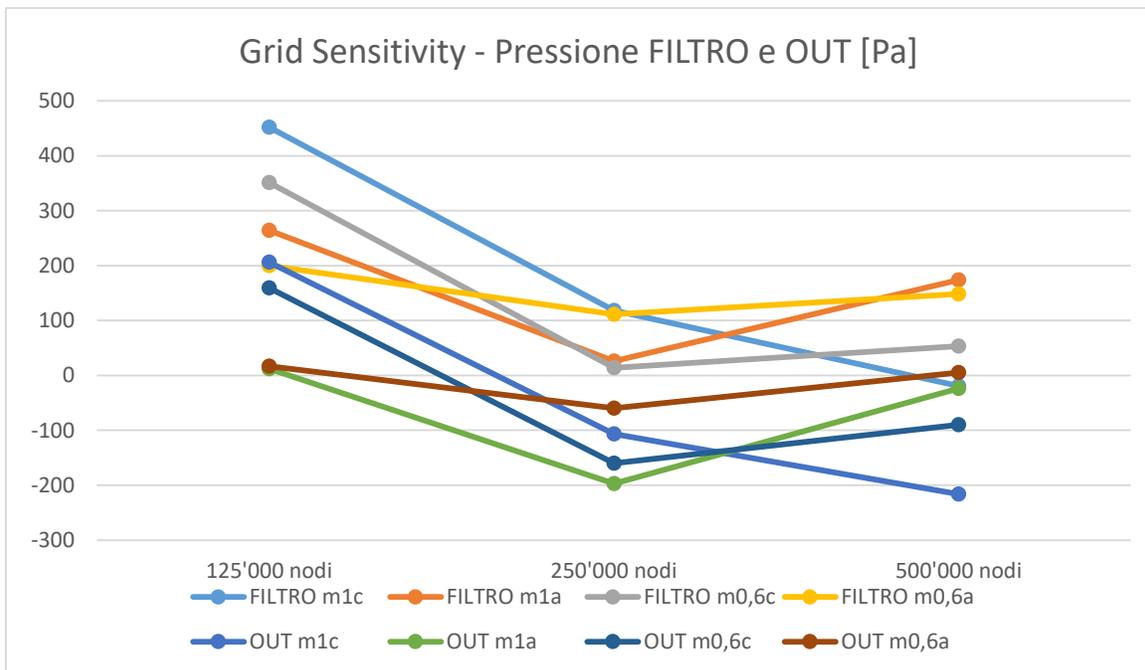
Perciò, verranno graficati di seguito nei prossimi diagrammi i valori di velocità e pressione per verificare che convergano agli stessi valori passando da 125'000, a 250'000 fino a 500'000 celle. Verrà eseguita solo per la configurazione senza tubo di flusso, sia per semplicità che perché così non verranno considerate le celle che il tubo di flusso “ruba” all’airbox, trovando così di quante celle (circa) esso necessita per avere convergenza dei risultati:



**Tabella 3.14:** Grid sensitivity dei valori medi di velocità



**Tabella 3.15:** Grid sensitivity dei valori medi di pressione in ingresso



**Tabella 3.15:** Grid sensitivity dei valori medi di pressione in uscita e sopra il filtro

Si può notare quindi come la velocità non abbia alcun problema di convergenza, dato l'andamento quasi lineare dei risultati a prescindere dal numero di celle impiegate per la mesh.

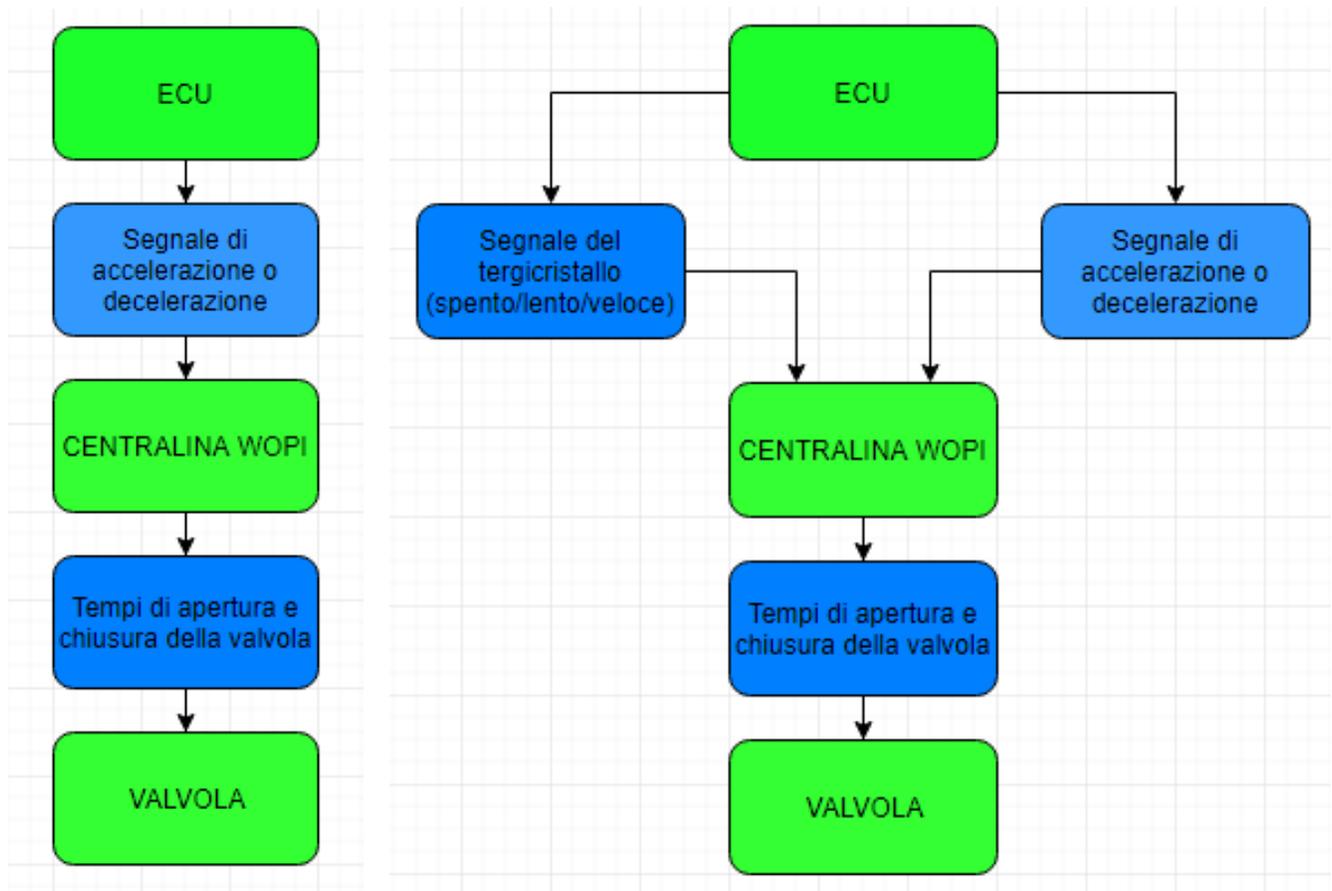
Tutt'altro discorso per quanto riguarda le pressioni. Quelle in ingresso sono ancora accettabili, ma già si nota che ancora non si ha una vera e propria convergenza dei risultati, invece in uscita e sopra il filtro ancora la convergenza è lontana. Sarà necessario quindi, in futuro, eseguire delle prove più "raffinate" a livello di mesh fino ad ottenere dei valori definitivi ed attendibili al 100%, questo sarà possibile, utilizzando un software diverso da quello utilizzato in questa circostanza, oppure che faccia riferimento ad una licenza professionale che non abbia il limite di circa 500'000 celle per la versione studentesca. In ogni caso le prove CFD condotte all'interno dell'azienda mostrano come anche questi risultati siano compatibili con quelli trovati in precedenza.

## 4. ECU e WOPI

In questo capitolo verranno illustrate entrambe le centraline simulate, sia la parte riguardante il codice che le comanda (software), che quella riguardante il loro assemblaggio (hardware).

### 4.1 Software

Il codice vero e proprio riguardante entrambe le centraline si troverà nell'allegato in fondo all'elaborato. In questo paragrafo, per semplicità, verrà riportato solamente un diagramma di flusso che faciliti la comprensione della logica di attuazione della valvola. Verranno infatti rappresentati i segnali digitali trasmessi dalla ECU, che verranno ricevuti dalla centralina del sistema WOPI che, in funzione di questi, comanderà l'apertura e la chiusura dell'elettrovalvola. Sono stati sviluppati due possibili tipi di logica:



**Diagramma 4.1:** Possibili logiche di controllo della valvola, semplice (a sinistra) e complessa (a destra)

Sono entrambe valide per lo scopo del WOPI ma, sia per poter assicurare alla futura valvola una vita molto maggiore che la possibilità di ricoprire entrambe le logiche col medesimo circuito, verrà sviluppata la seconda logica (quella più complessa). Sono presenti in tutto sei possibili configurazioni, che ora vengono divise in due macro categorie: accelerazione e decelerazione. Nel primo caso il veicolo avrà la massima richiesta di coppia, per questo tutti e tre i possibili segnali del tergicristalli (spento, lento o veloce) saranno totalmente bypassati e la valvola verrà lasciata chiusa, facendo in modo che tutta l'aria in ingresso andrà a finire nella camera di combustione. Nel secondo invece, sarà l'opposto, ovvero il motore non avrà richiesta di coppia. Ecco il momento in cui si andrà ad aprire la valvola di scarico dell'acqua in funzione del segnale del tergicristalli:

- Apertura poco frequente (in caso di tergicristalli spento, per esempio 1 secondo di apertura e 10 di chiusura);
- Apertura mediamente frequente (in caso di tergicristalli in posizione lenta, per esempio 1 secondo di apertura e 5 di chiusura);
- Apertura molto frequente (in caso di tergicristalli in posizione rapida, per esempio 1 secondo di apertura e 1 di chiusura).

Ovviamente si tratta di tempistiche settate per rendere evidenti i cambiamenti nell'utilizzo del prototipo a seconda della velocità del tergicristallo, per settarle in modo mirato sarà necessario eseguire dei test che stimino una media di riempimento d'acqua dell'airbox in funzione di queste velocità.

## 4.2 Hardware

Di seguito viene rappresentata una foto del modello fisico del sistema WOPI completo:

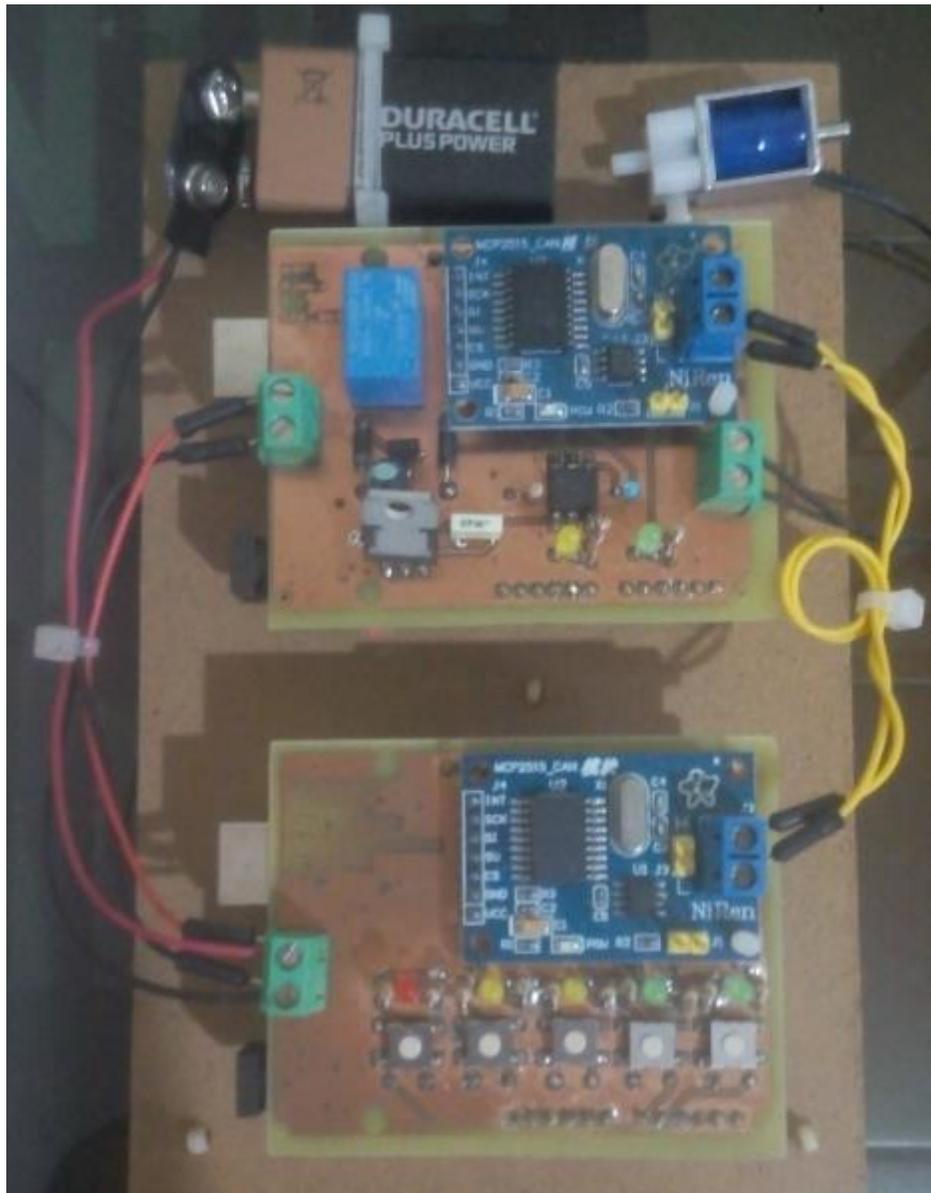


Foto 4.2: Prototipo del sistema WOPI

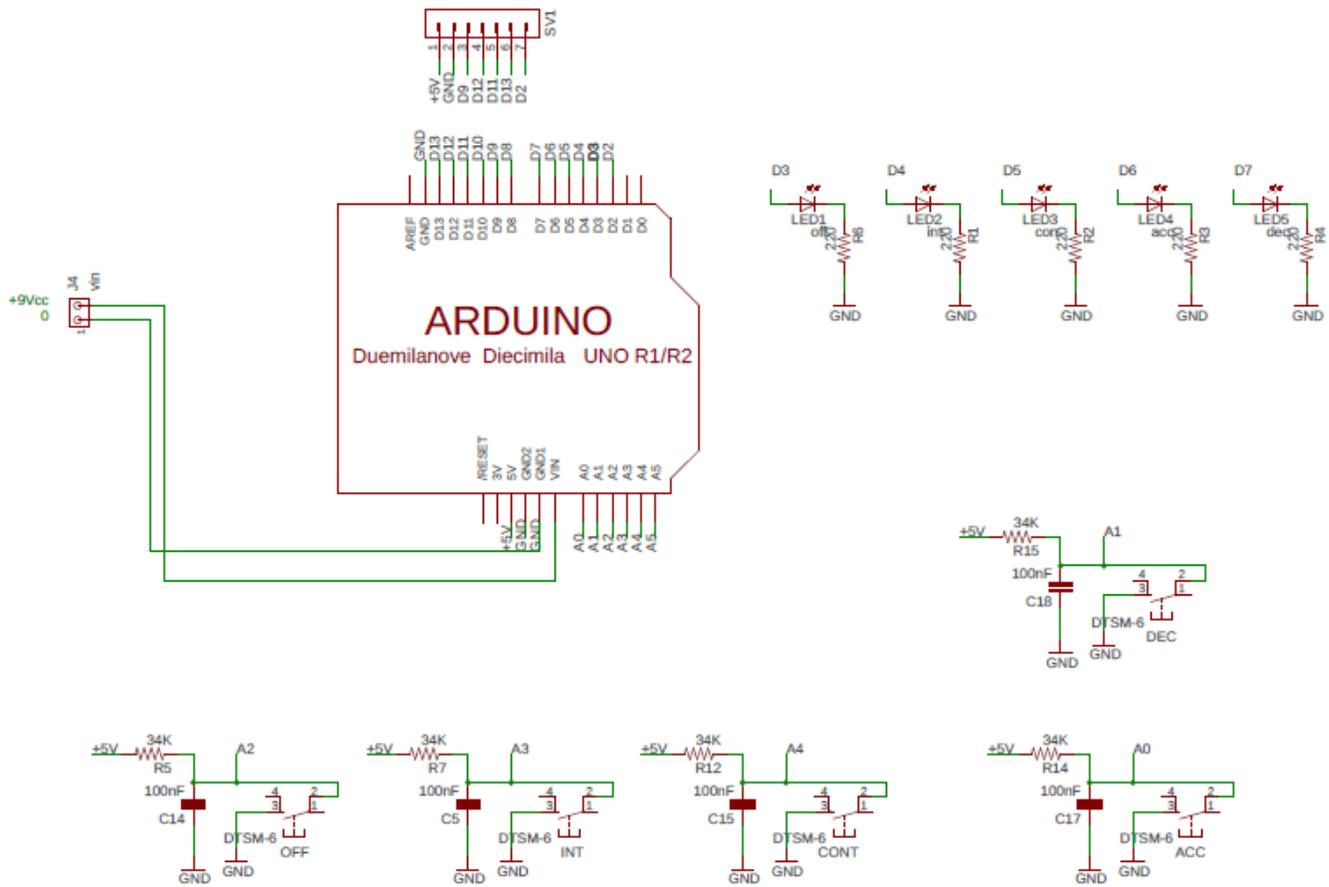
In basso si trova il simulatore del CAN (o Controller Area Network), che, in breve rappresenta lo standard seriale per connettere più ECU (Unità per il Controllo Elettronico). Il CAN simulato qui ha 5 tasti, inclusi i rispettivi led per poter monitorare il corretto funzionamento di tutto il circuito (le resistenze ai lati impediscono alla corrente troppo elevata di bruciarli). Da sinistra verso destra, i tasti simulano i segnali che riceverei dall'ECU (in questo caso attivati manualmente, per comodità dell'utilizzo del modello) rispettivamente di:

- Tergicristallo spento
- Tergicristallo a funzionamento intermittente
- Tergicristallo a funzionamento continuo
- Accelerazione
- Decelerazione

Il modello alimenta sia la ECU (il master o transmitter) che la scheda del WOPI (lo slave o receiver) attraverso i connettori verdi con una pila a 9 Volt, che poi verranno limitati a 5 per alimentare correttamente la bobina dell'elettrovalvola e non correre il rischio di surriscaldarla. I due moduli CAN BUS si interfacciano attraverso due segnali, detti low e high. Il cavo del "low" si riconosce essendo stato marchiato con un puntino nero agli estremi.

Nella parte alta invece si trova la centralina che controlla il sistema WOPI. I due led in questo caso rappresentano lo stato di chiusura e apertura dell'elettrovalvola (rispettivamente in giallo e verde). Se il WOPI riceve segnale di accelerazione entrambi i led saranno spenti, poiché così la valvola non viene "autorizzata" ad aprirsi.

### 4.2.1 SCHEMA DELLO SHIELD DEL MASTER (ECU)



Disegno 4.3: Shield del master

A sinistra dello shield (la grossa scheda al centro dello schema, che altro non è che un PCB, o “Printed Circuit Board” a doppia faccia, anche detto double layer) è visibile la batteria a 9 Volt che alimenta il master (e che sarà la stessa che alimenterà lo slave). Per una corretta alimentazione andrà collegata al pin della “massa” (ground) e al pin “Vin”, ovvero tensione in ingresso. Non utilizzo i pin “3V” o “5V” in quanto alimentandoli col triplo o il doppio della loro tensione nominale porterebbe brevemente la scheda al surriscaldamento.

Uno shield non è nient'altro che un circuito modulare che viene collegato alla scheda Arduino Uno che gli permette di eseguire altre funzioni che normalmente non potrebbe eseguire. Ogni shield di Arduino ha la stessa forma standard che le stesse schede Arduino posseggono. In questo caso però, non essendo attualmente disponibili degli shield per questo specifico utilizzo, sono stati progettati entrambi grazie al software "Eagle" e in seguito prodotti con piastre presensibilizzate in rame che sono state poi stampate da appositi macchinari per realizzare circuiti su PCB.

In basso sono stati rappresentati gli schemi elettrici che comandano i vari stati del tergitristallo: spento, intermittente e continuo (nota: i valori di resistenze e condensatori che sono riportati in tutti gli schemi sono riferiti alla costruzione di questo specifico modello, poi quando verranno implementati nelle vetture probabilmente subiranno delle lievi modifiche). I differenti casi in cui può trovarsi il tergitristallo vengono gestiti dallo shield ognuno da pin analogici diversi, rispettivamente A2, A3 ed A4.

In testa al circuito vengono posizionate delle resistenze dette di "pull up", resistenze utilizzate nei circuiti logici elettronici per avere la garanzia che gli ingressi di un sistema logico si trovino a livelli logici previsti, nel caso in cui esternamente i dispositivi siano scollegati oppure si trovino ad alta impedenza. Vengono anche utilizzate come interfaccia tra due diversi dispositivi logici, meglio se operanti a differenti tensioni di alimentazione.

Tra la resistenza e la massa (ground) viene interposto anche un condensatore (modello 1k63) per poter realizzare un circuito cosiddetto di debounce (o anti rimbalzo). Esso viene utilizzato per generare un solo impulso (e stabile) quando mi ritrovo con un ingresso elettricamente "rumoroso".

Esso viene utilizzato soprattutto in presenza di deviatori o tasti e serve ad eliminare i possibili impulsi falsati, cosa tipica in caso di contatti meccanici. L'origine di questi disturbi in presenza degli interruttori è data dal fatto che la lamina a contatto non può garantire, una volta commutata, che l'impulso sia stabile istantaneamente.

La chiusura del contatto elettromeccanico spesso produce, invece di un impulso continuo e preciso, un insieme di impulsi più ripidi e stretti, dovuti sia dal rimbalzo della lamina di contatto, che al degrado delle capacità conduttiva degli stessi contatti, effetti dovuti per esempio dall'ossido oppure lo sporco.

A lato dello shield invece sono stato rappresentati i circuiti rappresentanti i pulsanti di accelerazione e decelerazione, i cui segnali viaggeranno rispettivamente sui pin analogici A0 ed A1.

Sopra a questi due sono stati rappresentati i circuiti dei led, che si accenderanno quando ho attivato o meno uno di questi cinque stati appena descritti. Questi invece comunicheranno con lo shield attraverso i pin digitali D3, D4, D5, D6 e D7.

In alto infine ho lo schema del connettore del transceiver (ricetrasmittitore) del mio modulo CAN che, per essere alimentato ed eseguire il suo compito, va collegato ai pin di: alimentazione a 5V, massa e infine ai digitali D2, D9, D11, D12, D13.

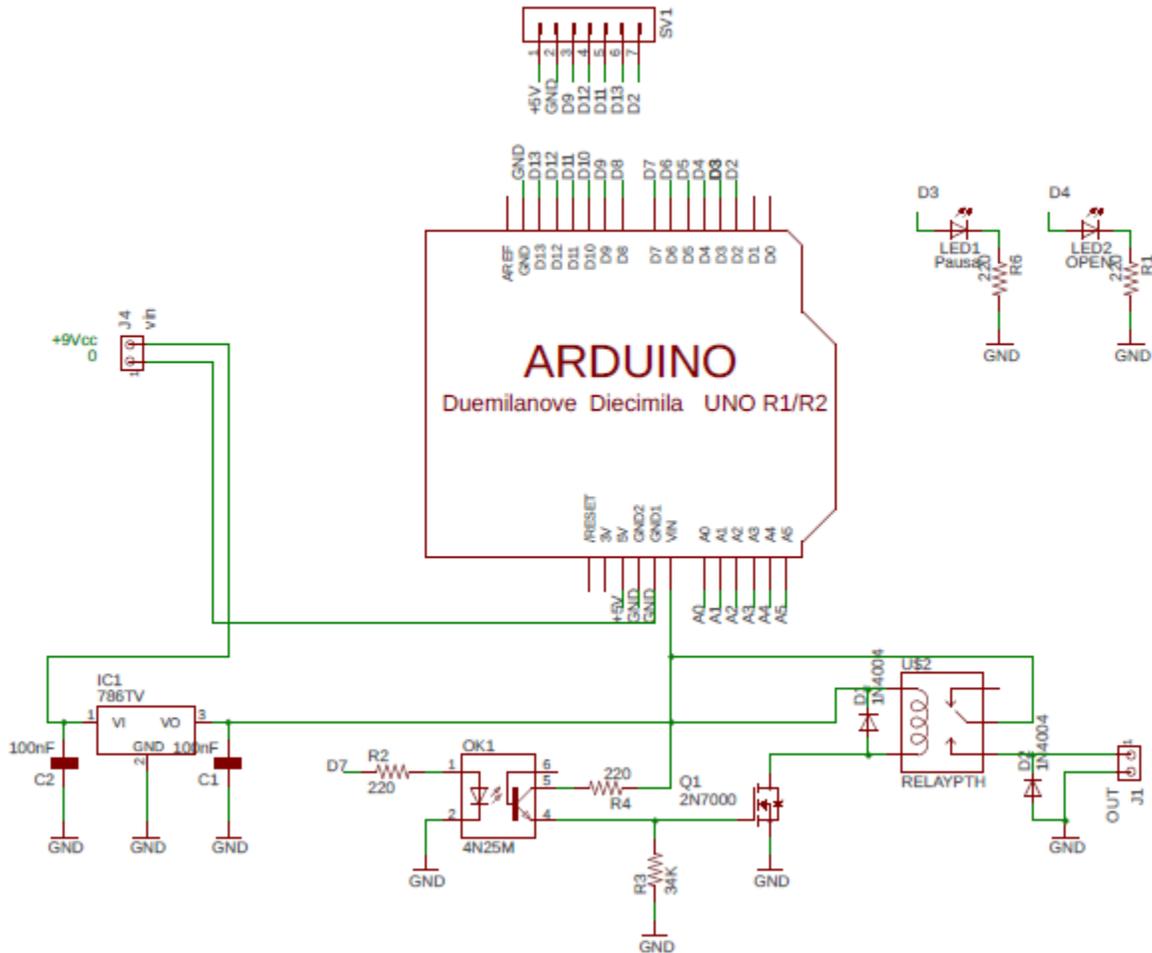
Possiamo ora vedere nel dettaglio la stampa di entrambi i layer del PCB prima quello superiore poi quello inferiore:



Le linee blu rappresentano la parte di circuito che finisce nel layer inferiore, cioè nel lato delle saldature, mentre le rosse rappresentano il lato della componentistica.

In sintesi, tutto ciò che non è un filo (una linea) fa parte della massa. Nel secondo layer infine sono rappresentati anche i collegamenti tra lo shield ed il modulo CAN-BUS compatibile con Arduino, che è ciò che simula il funzionamento dell'ECU.

## 4.2.2 SCHEMA DELLO SHIELD DELLO SLAVE (WOPI)



Disegno 4.5: Shield dello slave

In basso a sinistra possiamo vedere lo stabilizzatore di tensione. È colui che prende in ingresso i 9 Volt dell'alimentazione e li converte in 5. Questo è necessario in quanto l'elettrovalvola di scarico opera proprio a questa tensione e così evitiamo di sovraccaricarla.

Anche la board di Arduino nominalmente lavora a 5 Volt, ma portando la tensione in ingresso al pin "Vin" come abbiamo visto siamo in grado di gestirne tranquillamente anche 9 grazie al regolatore di tensione interno alla scheda (può arrivare anche a 12 ma è la soglia massima, meglio stare al di sotto di questo valore). Tra la massa, sia all'ingresso che all'uscita possiamo notare l'interposizione di due condensatori da 100 nF, che sono detti di stabilizzazione.

La loro presenza è giustificata dal fatto che, come visto nella sezione riguardante il master, potrei non avere una tensione “pulita” in ingresso allo shield e alla bobina dell’elettrovalvola; così facendo sono in grado di eliminare il problema.

La tensione, stabilizzata a 5 Volt, ora va in due direzioni: Una al pin “Vin” per alimentare lo shield e la scheda Arduino. L’altra invece va verso l’optoisolatore 4N25M, un componente che consente di trasferire il segnale di un circuito ad un altro, mantenendo però l’isolamento galvanico tra di essi, cioè è una misura adottata per la sicurezza circuitale. Di solito in un circuito elettronico vengono trattati segnali di tipo digitale, ma utilizzando questo modulo noi possiamo trasferire anche segnali analogici. Richiedendo noi un’elevata linearità, utilizzando questo componente siamo in grado di correggere le possibili non-linearità presenti a causa del passaggio analogico/digitale.

Ora il segnale, digitalizzato e linearizzato, passa attraverso un MOSFET 2N7000. Si tratta di un componente (il cui acronimo significa “transistor con metallo/ossido semiconduttore ad effetto di campo”) al cui interno si trova un condensatore formato da diversi strati di materiali (tre per la precisione) dove sono presenti presenti dei terminali: un gate, poi altri due, detti di sorgente e drenaggio (source e drain).

Applicando tensione al gate, col MOSFET io posso accumularla in uno strato sottostante al gate che poi procederò a “svuotare” controllandone il passaggio tra drain e source, regolando quindi quanta corrente passerà attraverso il mio dispositivo.

I MOSFET possono essere divisi in due tipologie: una è detta di tipo “p” quando il substrato del mio semiconduttore (di solito si tratta di silicio) viene drogato positivamente; in caso venga drogato negativamente invece, si parla di un MOSFET di tipo “n”. I primi faranno andare la corrente dal drain verso il source, viceversa per quanto riguarda i secondi, che sono la tipologia che andremo ad utilizzare noi, ma nulla ci vieterebbe di utilizzare i tipo “p”, cambierebbe semplicemente parte del circuito ma eseguirebbe ugualmente il suo compito. Questo componente, col mio segnale d’ingresso, accende il led al suo interno, poiché è un MOSFET “fotoaccoppiatore”, ovvero accoppia il circuito solo nel caso si sia acceso il led, in caso contrario non ho nessun accoppiamento con la parte successiva, ovvero col relè, che poi si chiude a massa.

Si può notare che tra l’optoisolatore e il MOSFET è stata applicata una resistenza di 34000 Ohm.

Essendo collocata tra il gate di questo secondo componente e la massa, quando viene tolta tensione dal gate la resistenza forza la sua apertura, dato che noi lo stiamo utilizzando a “interdizione e saturazione”. Infatti, la prima volta che viene alimentato il circuito dello slave, questa resistenza lo obbliga a restare aperto, ovvero “interdetto”.

Apriamo una piccola parentesi riguardante la struttura ed il funzionamento anche del relè per completezza: si tratta di un dispositivo che, tramite il campo magnetico che posso fornirgli alla bobina grazie al passaggio di corrente, è in grado di agire da deviatore senza l’ausilio di un operatore.

Tornando al nostro circuito, la bobina del relè quindi ora è eccitata e può muovere l’interruttore al suo interno verso il basso (nello schema rappresentato in posizione di riposo, quindi senza l’eccitazione dell’avvolgimento) che così andrà ad alimentare J1, che rappresenta la bobina dell’elettrovalvola.

L’utilizzo di un relè è giustificato dal fatto che lo shield e il circuito tutto sono alimentati continuamente (se così non fosse basterebbe alimentare il tutto solo quando serve, ma poi non sarebbe compatibile col funzionamento di un ECU d’automobile, perciò l’elettrovalvola per essere alimentata solo quando è necessario necessita il suo impiego).

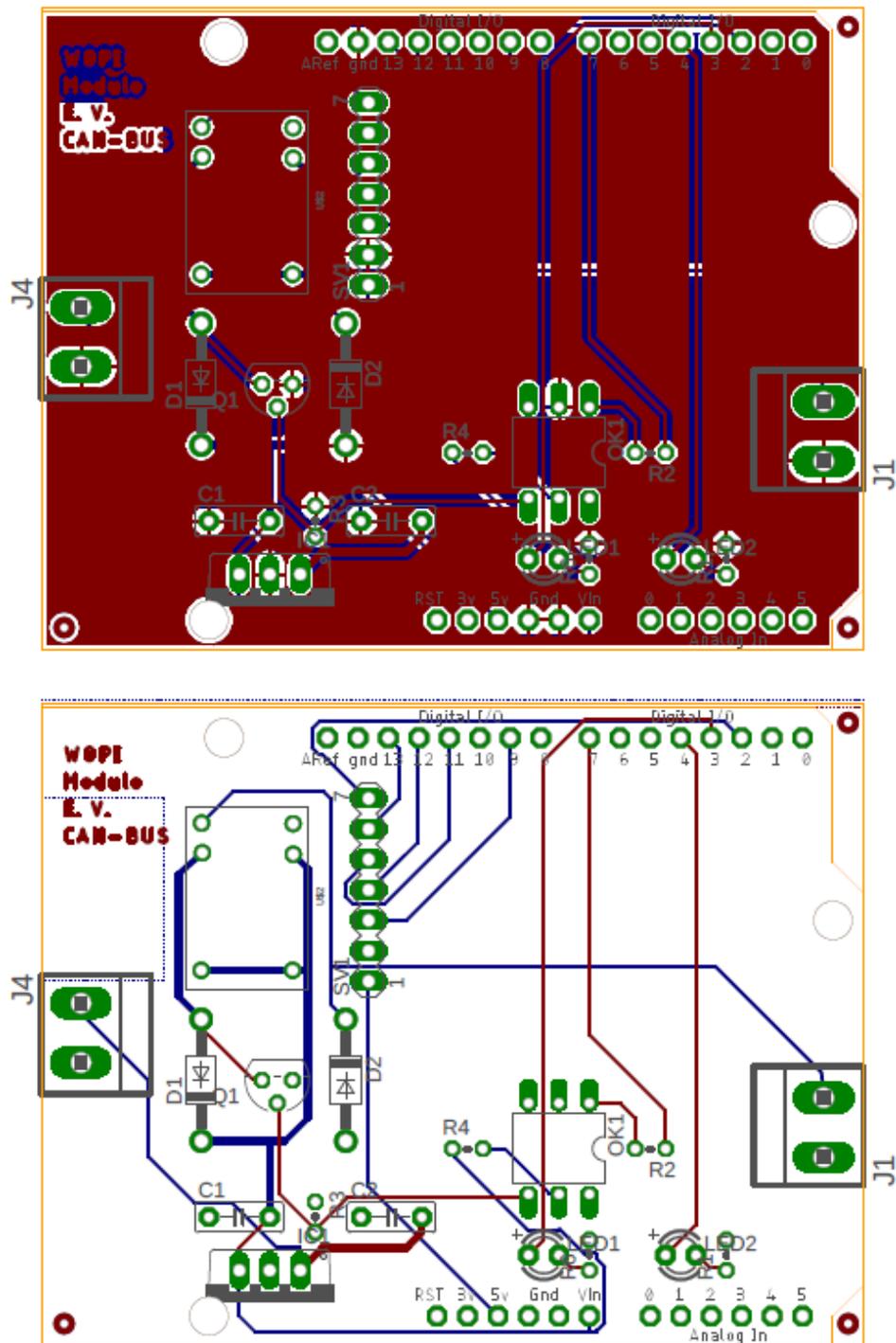
Si può notare sia in prossimità del relè che dell’elettrovalvola la presenza di due diodi; questi sono detti diodi “di flyback” e vengono utilizzati a causa delle bobine presenti.

La corrente, una volta che le ha attraversate ed hanno eseguito il loro scopo, esce verso la massa. A questo punto tolgo l’alimentazione (al gate del transistor e all’ingresso dell’elettrovalvola) e così la corrente smette di circolare. Solo che le bobine ne hanno immagazzinata una parte di quella precedentemente fornitagli e la restituirebbe al circuito, in più con verso opposto, ovvero dalla massa verso il polo positivo! Grazie a questo diodo però il tutto si chiude su sé stesso, impedendone il ricircolo involontario quando il circuito riapre, lasciando così che questa corrente venga “riassorbita” nella bobina. Potrebbe sembrare che così si incorra nel rischio di avere una riapertura involontaria della valvola, ma questa corrente residua non sarà mai sufficientemente intensa da consentirglielo e così verrà dissipata, ecco perché quella del diodo è un’ottima contromisura di sicurezza.

Spostandoci ora in alto a destra possiamo vedere i due che alimentano due led: quello di chiusura e di apertura della mia valvola, alimentati rispettivamente attraverso i pin digitali D3 e D4. In caso io mi trovi in stato di accelerazione l'elettrovalvola non verrà mai azionata ed entrambi i led saranno spenti, dato che voglio impedire la fuoriuscita di aria all'esterno quando ne ho bisogno alla massima pressione possibile. Saranno accesi invece durante lo stato di decelerazione con le tempistiche di scarico date dal funzionamento attuale del tergicristallo.

Infine in alto abbiamo lo schema del transceiver, analogo a quello che è già stato illustrato nella parte riguardante il master.

Vediamo adesso nel dettaglio i layer dello shield che comanda l'apertura della valvola:



Disegni 4.6: PCB dello slave

### 4.2.3 VALVOLA

Per questo tipo di applicazione si ha bisogno di una valvola due vie e due posizioni (2/2), normalmente chiusa (NC), in modo da garantire una tenuta il più possibile ermetica in condizioni di utilizzo ottimale dell'airbox, mentre si dovrà avere lo scarico delle eventuali gocce d'acqua aprendola per pochi istanti, anche se, con questa logica di controllo che la aprirà, nella maggioranza dei casi verrà elaborata anche dell'aria, in quanto la quantità di acqua dentro un airbox è pur sempre bassa. La dinamica di questo sistema varia molto in funzione di molti fattori (portata d'aria in ingresso, velocità del veicolo, umidità dell'aria ed eventuali gocce di pioggia). Ecco perché la scelta della valvola, che sarebbe eseguita ad hoc per un sistema statico, in questa situazione andrà perlomeno verificato con un prototipo completo ed in condizioni di utilizzo reali per certificarne l'effettivo funzionamento corretto.

Detto questo, si può procedere in due direzioni:

- Scegliere una valvola già presente in commercio che soddisfi le esigenze del progetto
- Progettare “ad hoc” una nuova valvola

Si procede da subito col primo metodo, lasciando il secondo come riserva, per diversi motivi:

- Non è necessario spendere mesi (o più) di progettazione, disegno, produzione e test dei componenti principali (corpo valvola, stelo, ecc.) e del complessivo finale
- Non serve mettersi alla ricerca e alla scelta uno ad uno dei componenti che normalmente non si produrrebbero in proprio (come molle e solenoidi)
- Una valvola adatta allo scopo prodotta da un'azienda che ha molti anni d'esperienza nel settore, anche se non perfettamente dimensionata, è comunque più affidabile rispetto ad un primissimo modello prodotto internamente. Una valvola è un dispositivo elettromeccanico molto delicato e che richiede alta precisione dei sottocomponenti, che se non lavora perfettamente potrebbe non garantire una tenuta stagna a valvola chiusa, oppure un solenoide non scelto col giusto criterio (o non progettato e sviluppato correttamente nel caso si volesse produrre “in casa” anche questo pezzo) potrebbe avere una vita utile di cicli di utilizzo molto minore di quanto sperato.

Seguiamo quindi il metodo di scelta di una valvola già presente in commercio. I vincoli da rispettare per il progetto WOPI sono i seguenti:

- Valvola a due vie e due posizioni, normalmente chiusa (NC)
- Dimensioni ridotte del corpo valvola, possibilmente nell'intorno di 50x30x30 mm (avere spazio libero sotto il cofano nelle vetture moderne spesso è un lusso; si potrebbe pensare di modificare altri organi per farle spazio ma non è affatto conveniente, specialmente perché il cliente avrà poca, se non per niente, voglia di cambiare un fornitore per una valvola di troppo, specie per un ipotetico modesto aumento di performance)
- Range di pressione di lavoro ridotto e mirato. La valvola lavorerà in un ambiente dove si avrà a che fare con differenziali di pressioni molto limitati spesso vicini a pressioni relative nulle, mentre in altri si andrà addirittura sotto la pressione atmosferica. Perciò si dovrà trovare una valvola progettata per lavorare in leggera sovrappressione o in condizioni di vuoto e dovrà essere installata nella zona dell'airbox più idonea
- Tipo di fluido che dovrà trattare, ovvero acqua, anche se normalmente non verrà elaborata da sola
- Un Kv (o Cv) il più simile alle condizioni di lavoro (possibilmente maggiore)

Procederemo a breve alla scelta della valvola adatta, ma per farlo bisogna introdurre il coefficiente di flusso della valvola Kv (o Cv, l'analogo statunitense, dove  $Kv = 0,86 * Cv$ ) e da catalogo scegliere una valvola che avrà circa questo coefficiente, possibilmente maggiore. In questo ambiente di lavoro si opererà in un ambiente dinamico, dove la velocità e il differenziale di pressione tra monte e valle non sarà mai costante e varierà notevolmente durante la guida, ma la scelta migliore sarà eseguita seguendo comunque il metodo classico di scelta di una valvola per fluidi incomprimibili.

Supposto che la maggior parte del fluido elaborato dalla valvola sarà acqua, il KV elaborato viene determinato dalla formula seguente:

$$KV = \frac{Q * \sqrt{\rho}}{\sqrt{\Delta p}}$$

Dove  $Q$  rappresenta la portata volumetrica che attraversa l'orifizio della valvola (misurata in  $m^3/h$ ),  $\rho$  la densità dell'acqua (in  $kg/dm^3$ , che per l'acqua vale 1) e  $\Delta p$  il differenziale di pressione tra ingresso ed uscita della valvola (in bar).

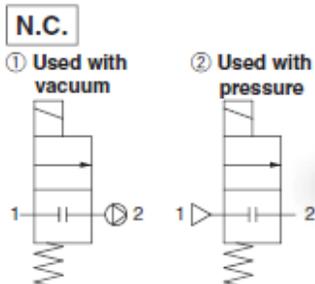
La ricerca di una valvola di dimensioni così ridotte è stata molto lunga, spesso dovuta al fatto che chi le produce nelle schede tecniche è “avidio” nel fornire tutte le specifiche tecniche e sarebbe necessario contattarli uno per uno. Ma alla fine è stato trovato un modello che sembra adatto allo scopo, si tratta del VX214 (o dell'analogo VX212) con porta da 1/8" e orifizio da 2 mm, il cui corpo valvola supera leggermente gli ingombri ipotizzati. Qui di seguito vengono riportate la loro schede tecniche:



## For Medium Vacuum Single Unit

### Model/Valve Specifications

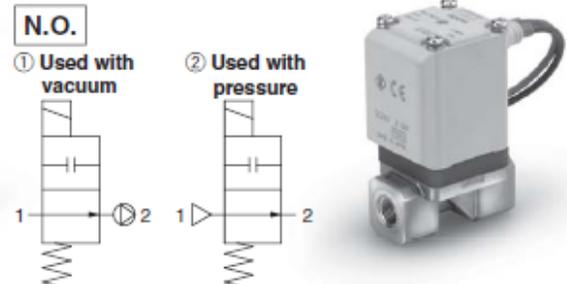
Passage symbol (Application example)



Refer to "Glossary of Terms" on page 29 for passage symbols.



Passage symbol (Application example)



Refer to "Glossary of Terms" on page 29 for passage symbols.



### Normally Closed (N.C.)

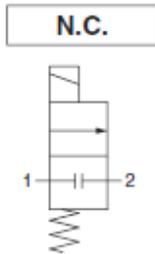
Size	Port size	Orifice diameter [mm]	Model	Flow-rate characteristics			Operating pressure range		Weight [g]
				C [dm <sup>3</sup> /(s·bar)]	b	Cv	① Used with vacuum (Pa-abs)	② Used with pressure (MPa-G)	
1	1/8, 1/4	2	VX214	0.63	0.63	0.23	0.1 to atmospheric pressure	0 to 1.0	300
		3		1.05	0.68	0.41		0 to 0.6	300
		5		2.20	0.39	0.62		0 to 0.2	300
2	1/4, 3/8	4	VX224	1.90	0.52	0.62		0 to 1.0	460
		7		3.99	0.44	1.08		0 to 0.15	460
3	1/4, 3/8	5	VX234	1.96	0.55	0.75		0 to 1.0	580
		8		5.67	0.33	1.58		0 to 0.3	580
		10		5.74	0.64	2.21		0 to 0.1	580
	1/2	10		8.42	0.39	2.21		0 to 0.1	630



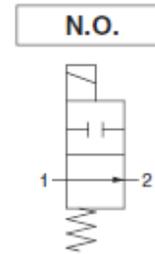
# For Water Single Unit

## Model/Valve Specifications

### Passage symbol



### Passage symbol



Refer to "Glossary of Terms" on page 29 for passage symbols.

Refer to "Glossary of Terms" on page 29 for passage symbols.

### Normally Closed (N.C.)

Size	Port size	Orifice diameter [mmø]	Model	Flow-rate characteristics		Maximum operating pressure differential [MPa]	Weight [g] <sup>Note)</sup>
				AV (x 10 <sup>-6</sup> m <sup>2</sup> )	Conversion Cv		
1	1/8, 1/4	2	VX212	5.5	0.23	1	300
		3		10.0	0.42	0.6	300
		5		15.0	0.63	0.2	300
2	1/4, 3/8	4	VX222	15.0	0.63	1	460
		7		26.0	1.08	0.15	460
3	1/4, 3/8	5	VX232	18.0	0.75	1	580
		8		38.0	1.58	0.3	580
		10		53.0	2.21	0.1	580
	10	53.0		2.21	0.1	630	

### Fluid and Ambient Temperature

Fluid temperature [°C]	Ambient temperature [°C]
1 to 60 <sup>Note)</sup>	-20 to 60

Note) With no freezing

### Valve Leakage

#### Internal Leakage

Seal material <sup>Note 2)</sup>	Leakage rate (Water) <sup>Note 1)</sup>
NBR (FKM)	0.1 cm <sup>3</sup> /min or less

#### External Leakage

Seal material <sup>Note 2)</sup>	Leakage rate (Water) <sup>Note 1)</sup>
NBR (FKM)	0.1 cm <sup>3</sup> /min or less

Note 1) Leakage is the value at ambient temperature 20°C.

Note 2) For seal material/FKM, refer to "Other options".

**How to Order (Single Unit)**



VX2 1 2 A A A

**Fluid**

2	Water
---	-------

**Interchangeable with existing bracket**

—	No
<b>XB</b>	Yes

(Note) For more information, please see table below.

**Specifications**

**For Air**

**For Medium Vacuum**

**For Water**

**For Oil**

**For Steam**

**Construction**

**Dimensions**

**Coil size/Valve type**

Size	Symbol	Valve type
Size 1	1	N.C.
	4	N.O.
Size 2	2	N.C.
	5	N.O.
Size 3	3	N.C.
	6	N.O.

**Body material/Port size/Orifice diameter**

Symbol	Body material	Port size	Orifice diameter
A	C37 (Brass)	1/8	2
B			3
C		1/4	2
D			3
E	Stainless steel	1/8	5
F			2
H		1/4	3
J			5
K	1/2	3/8	2
L			4
M		1	3
N	5		

**Other option**

Symbol	Seal material *1	Oil-free	Port thread
—	NBR	—	Standard (Rc)
A	NBR	—	G
B	NBR	—	NPT
C	FKM	—	Standard (Rc)
D	NBR	—	G
E	NBR	○	NPT
F	FKM	—	G
G	FKM	—	NPT
H	FKM	○	Standard (Rc)
K	FKM	○	G
L	FKM	○	NPT
Z	NBR	○	Standard (Rc)

\*1 For low concentration ozone resistant and deionised water select seal material FKM.

**Voltage/Electrical entry (coil insulation type: Class B)**

Symbol	Voltage	Electrical entry	Symbol	Voltage	Electrical entry
A	24 VDC	Grommet	Z1P	48 VAC	Conduit (With surge voltage suppressor)
B	100 VAC	Grommet (With surge voltage suppressor)	Z1Q	220 VAC	
C	110 VAC		Z1R	240 VAC	
D	200 VAC		Z1Y	24 VAC	
E	230 VAC	DIN terminal (With surge voltage suppressor)	Z1S	12 VDC	Faston terminal
F	24 VDC		Z1T	12 VDC	
G	24 VDC		Z2A	24 VDC	
H	100 VAC	Conduit terminal (With surge voltage suppressor)	Z2B	100 VAC	DIN terminal (With surge voltage suppressor, with light)
J	110 VAC		Z2C	110 VAC	
K	200 VAC		Z2D	200 VAC	
L	230 VAC	Conduit terminal (With surge voltage suppressor)	Z2E	230 VAC	Conduit terminal
M	24 VDC		Z2F	48 VAC	
N	100 VAC		Z2G	220 VAC	
P	110 VAC	Conduit (With surge voltage)	Z2H	240 VAC	Conduit terminal
O	200 VAC		Z2V	24 VAC	
R	230 VAC		Z2J	12 VDC	
S	24 VDC	Conduit (With surge voltage)	Z2K	24 VDC	Conduit terminal
T	100 VAC		Z2L	100 VAC	
U	110 VAC		Z2M	110 VAC	
V	200 VAC		Z2N	200 VAC	

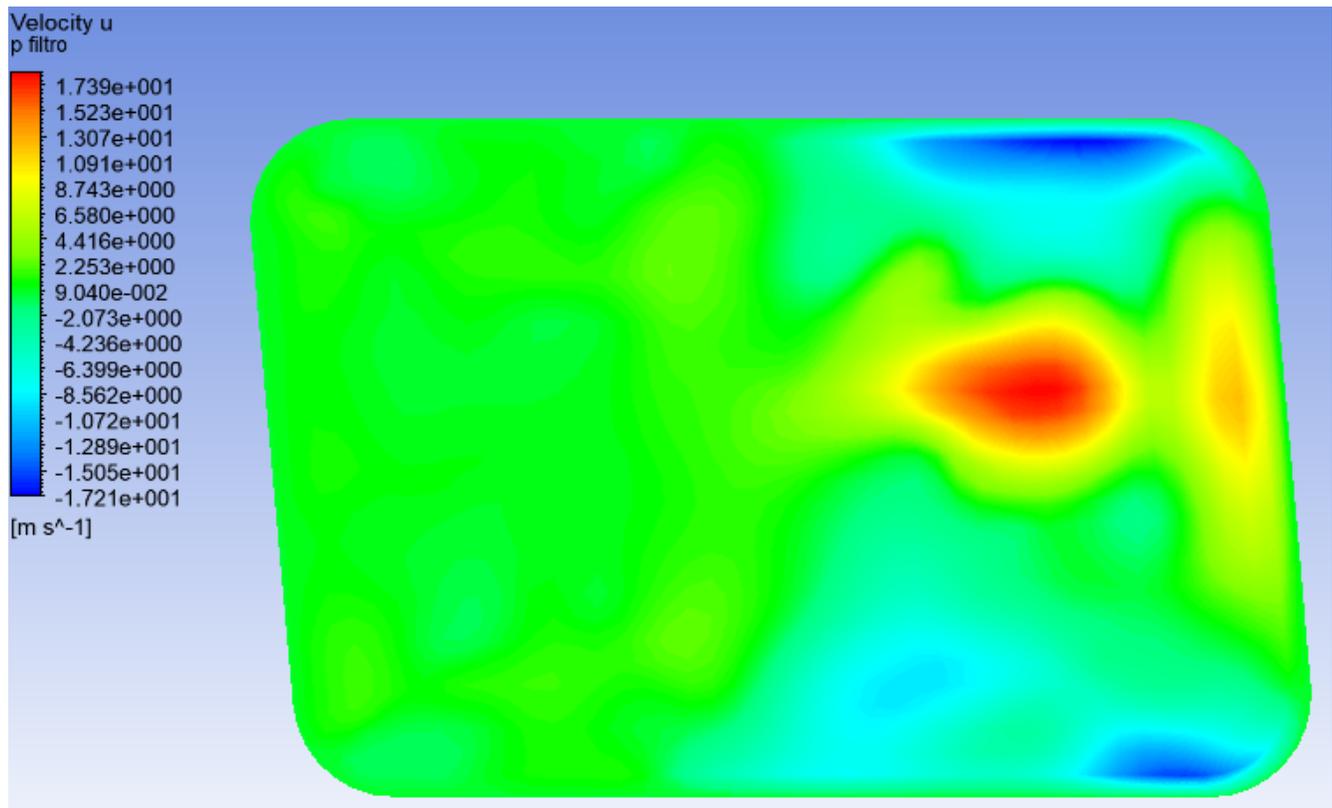
**Tabella 4.7:** Scheda tecnica della valvola

Dato che le dimensioni, il CV e il range di pressione sono gli stessi per entrambe le valvole e che supportano l'utilizzo con aria e acqua, la scelta di una o dell'altra per l'utilizzo "in pressione" (con pressioni relative maggiori di zero) è del tutto indifferente. Se si volesse procedere invece ad installare una valvola in condizioni di vuoto bisognerebbe ricorrere forzatamente alla VX214, che lavora in un campo che varia dagli 0,1 Pascal assoluti fino a pressione atmosferica.

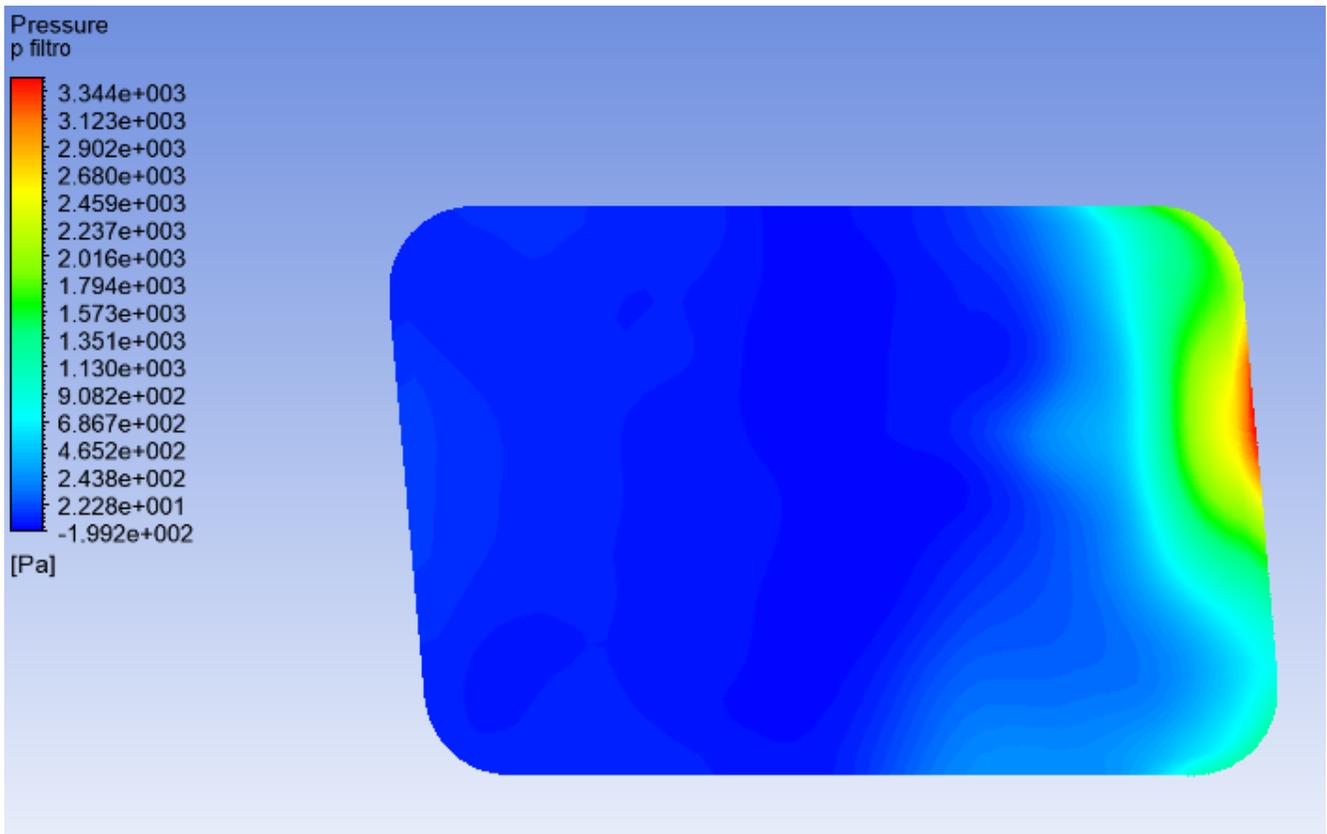
La portata d'aria non sarà mai la stessa durante il moto del veicolo in quanto, come accennato qualche riga fa, dipenderà dalla sua velocità, dalla dinamica del flusso in quel particolare istante e dalla velocità in direzione parallela alla bocca d'ingresso, in quanto l'ingresso della valvola sarà posizionato in quella medesima direzione.

Per questo viene eseguita un'ipotesi semplificativa ai fini della qualità del risultato, utilizzando i dati delle simulazioni ottenute in condizioni statiche e a valvola chiusa, data la sua apertura quasi istantanea.

La portata (valore che ricordiamo essere solo ideale in questo utilizzo) è definita dalla velocità perpendicolare all'orifizio del corpo valvola che lo attraversa e appunto la sua sezione di passaggio. Vedremo di seguito tutti valori di CV che troveremo nelle 6 configurazioni di simulazione (senza tubo di flusso) nel caso di valvola "in pressione" posizionata nel lato destro dell'airbox. Utilizziamo per esempio i dati ottenuti a Mach 1 con 250'000 celle:



**Immagine 4.8:** Campo di velocità lungo l'asse x. "Images used courtesy of ANSYS, Inc."



**Immagine 4.9:** Campo di pressione. "Images used courtesy of ANSYS, Inc."

Con un diametro di 2 mm dell'orifizio, una velocità lungo x di circa 3 m/s ed un salto di pressione di 0,033 bar:

$$Q = v \frac{\pi D^2}{4} = 3 * \frac{\pi * 0,002^2}{4} = 0,00000942 \frac{m^3}{s} = 0,033912 m^3/ora$$

$$KV = \frac{Q * \sqrt{\rho}}{\sqrt{\Delta p}} = 0,18$$

$$CV = \frac{KV}{0,86} = 0,21$$

Valore che è di poco inferiore al CV della nostra valvola, condizione ideale di lavoro per una valvola.

In altre occasioni coi dati raccolti dalle simulazioni si ottengono altre volte valori di poco inferiori a 0,23 della valvola, mentre in altri di poco superiori.

acqua		125'000 celle m1	acqua		125'000 celle m0,6
diametro orifizio [m]		0,002	diametro orifizio [m]		0,002
area orifizio [m2]		0,00000314	area orifizio [m2]		0,00000314
velocità [m/s]		4	velocità [m/s]		2,5
portata [m3/s]		0,00001256	portata [m3/s]		0,00000785
portata [m3/h]		0,045216	portata [m3/h]		0,02826
densità [kg/m3]		1000	densità [kg/m3]		1000
DP (P1-P2) [bar]		0,05	DP (P1-P2) [bar]		0,025
KV		0,20	KV		0,18
CV		0,23	CV		0,21

acqua		250'000 celle m1	acqua		250'000 celle m0,6
diametro orifizio [m]		0,002	diametro orifizio [m]		0,002
area orifizio [m2]		0,00000314	area orifizio [m2]		0,00000314
velocità [m/s]		3	velocità [m/s]		3
portata [m3/s]		0,00000942	portata [m3/s]		0,00000942
portata [m3/h]		0,033912	portata [m3/h]		0,033912
densità [kg/m3]		1000	densità [kg/m3]		1000
DP (P1-P2) [bar]		0,033	DP (P1-P2) [bar]		0,022
KV		0,19	KV		0,23
CV		0,21	CV		0,26

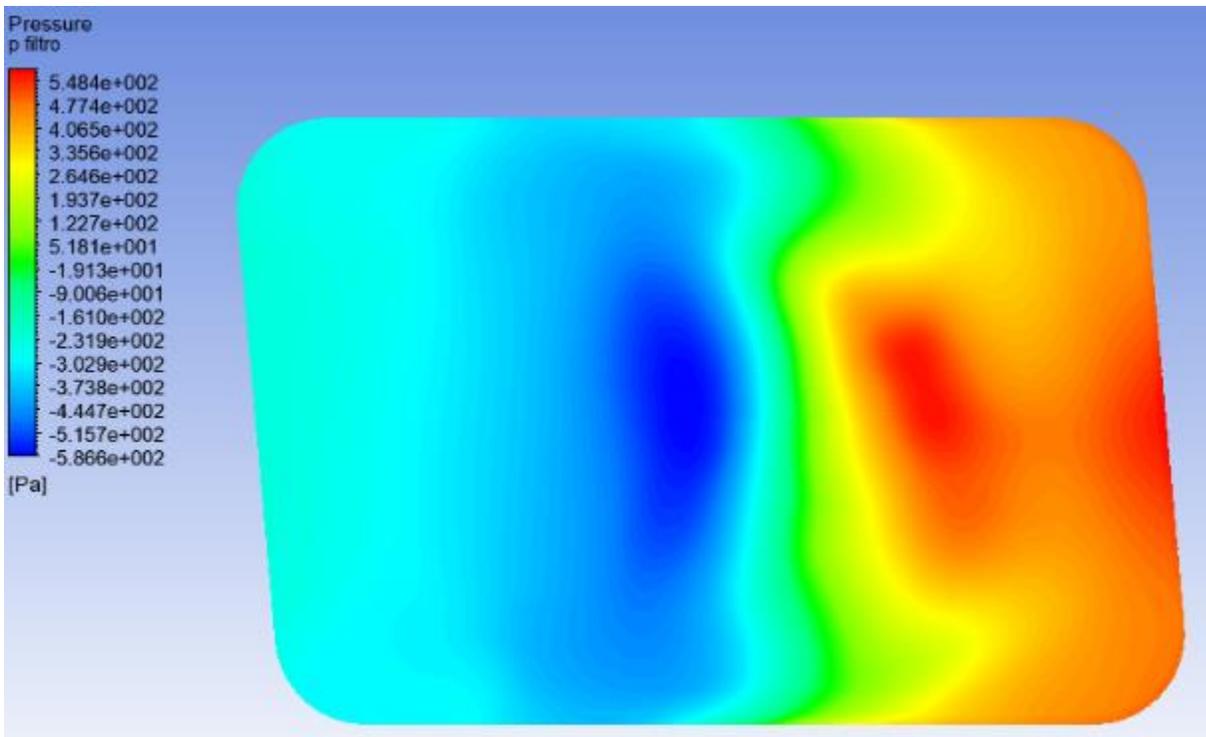
  

acqua		500'000 celle m1	acqua		500'000 celle m0,6
diametro orifizio [m]		0,002	diametro orifizio [m]		0,002
area orifizio [m2]		0,00000314	area orifizio [m2]		0,00000314
velocità [m/s]		1,5	velocità [m/s]		2
portata [m3/s]		0,00000471	portata [m3/s]		0,00000628
portata [m3/h]		0,016956	portata [m3/h]		0,022608
densità [kg/m3]		1000	densità [kg/m3]		1000
DP (P1-P2) [bar]		0,006	DP (P1-P2) [bar]		0,009
KV		0,22	KV		0,24
CV		0,25	CV		0,27

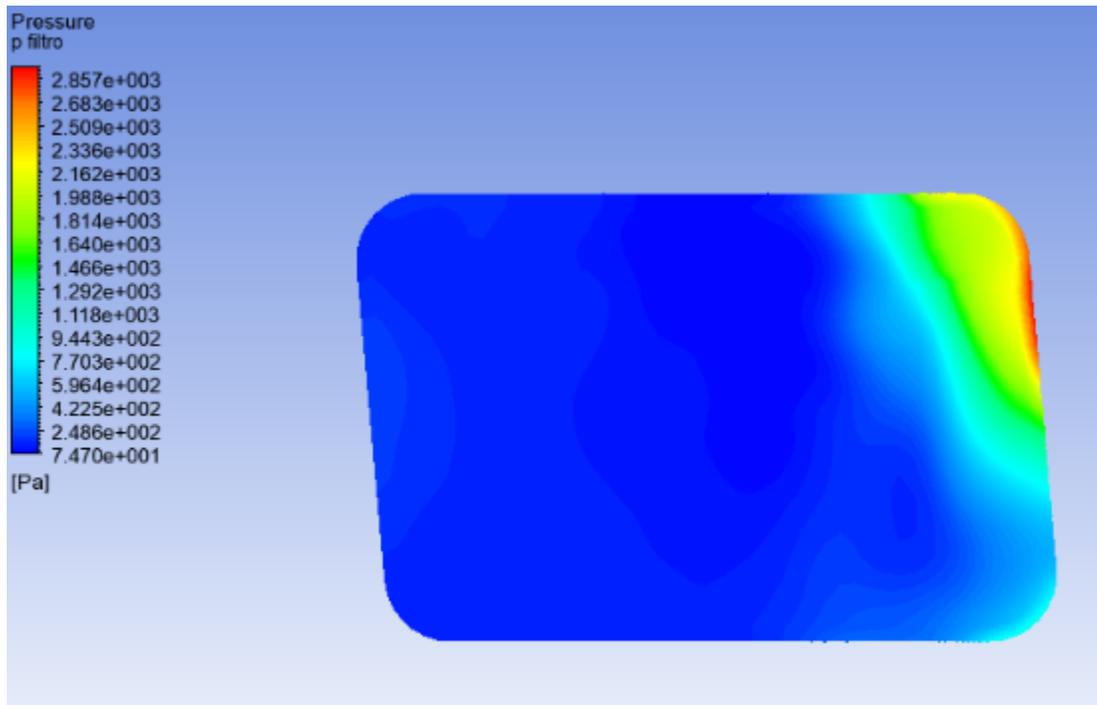
Ma, come ampiamente accennato poche pagine fa, questo dimensionamento va preso molto alla larga, essendo che i valori di velocità e pressione possono variare ampiamente e che spesso all'apertura della valvola verrà elaborata verso l'esterno anche aria e non solo acqua.

Inoltre, se lo si ritenesse necessario, si può vedere dai cataloghi che esiste la medesima valvola (a pari dimensioni) con un Cv che arriva fino a 0,63 e che funziona ad un range di pressione ancora compatibile col caso del WOPI. Si potrebbero eseguire ulteriori simulazioni, ma l'aggiunta di un condotto di scarico di 2 millimetri di diametro è troppo minuto per poter essere simulato decentemente con certe dimensioni della mesh, infatti facendo dei tentativi anche con 500'000 celle i residui che inizialmente si abbassano fino a valori di  $10^{-2}$ , dopo un centinaio di iterazioni hanno un repentino aumento, che porta inevitabilmente a risultati irreali in termini fisici. Per questo sono stati utilizzati i valori sopracitati per i calcoli non possedendone altri al momento.

Esiste anche una seconda possibilità di configurazione, ma conviene studiarla a fondo con simulazioni più precise (maggior numero di celle e di iterazioni) per avere pressioni e velocità il più precise possibile: installare una valvola per vuoto nella parete sinistra dell'airbox. Da risultati delle simulazioni mostrati qui sotto si può vedere come le pressioni in quella zona attraversino un range critico, poiché si sono ottenute pressioni relative da circa -250 Pascal relativi fino a pochi Pascal sopra la pressione atmosferica, per questo motivo l'utilizzo di una valvola o l'altra risulterebbe incorretto:



**Immagine 4.10:** Campo di pressione. "Images used courtesy of ANSYS, Inc."



**Immagine 4.11:** Campo di pressione. "Images used courtesy of ANSYS, Inc."

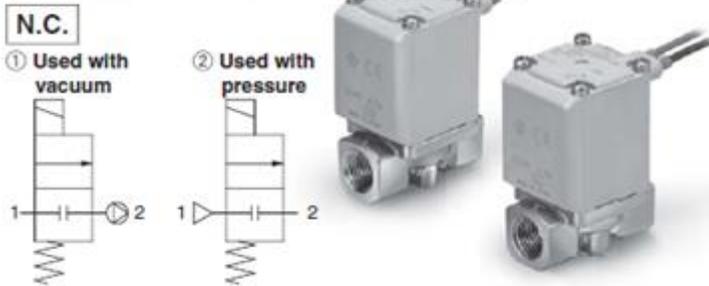
Nel caso le future simulazioni evidenzino come nella zona interessata ci si ritroverà sempre di fronte pressioni relative negative, seppur piccole, si potrà allora ricorrere alla valvola “per vuoto” VX214, che viene mostrata di nuovo da un estratto del suo catalogo:



## For Medium Vacuum Single Unit

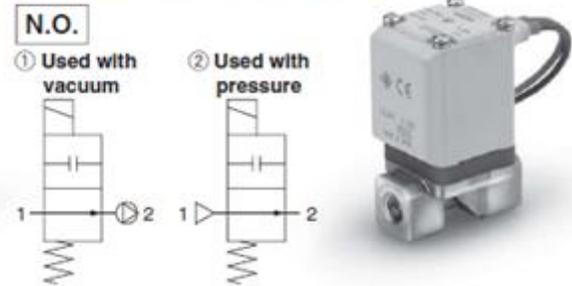
### Model/Valve Specifications

Passage symbol (Application example)



Refer to "Glossary of Terms" on page 29 for passage symbols.

Passage symbol (Application example)



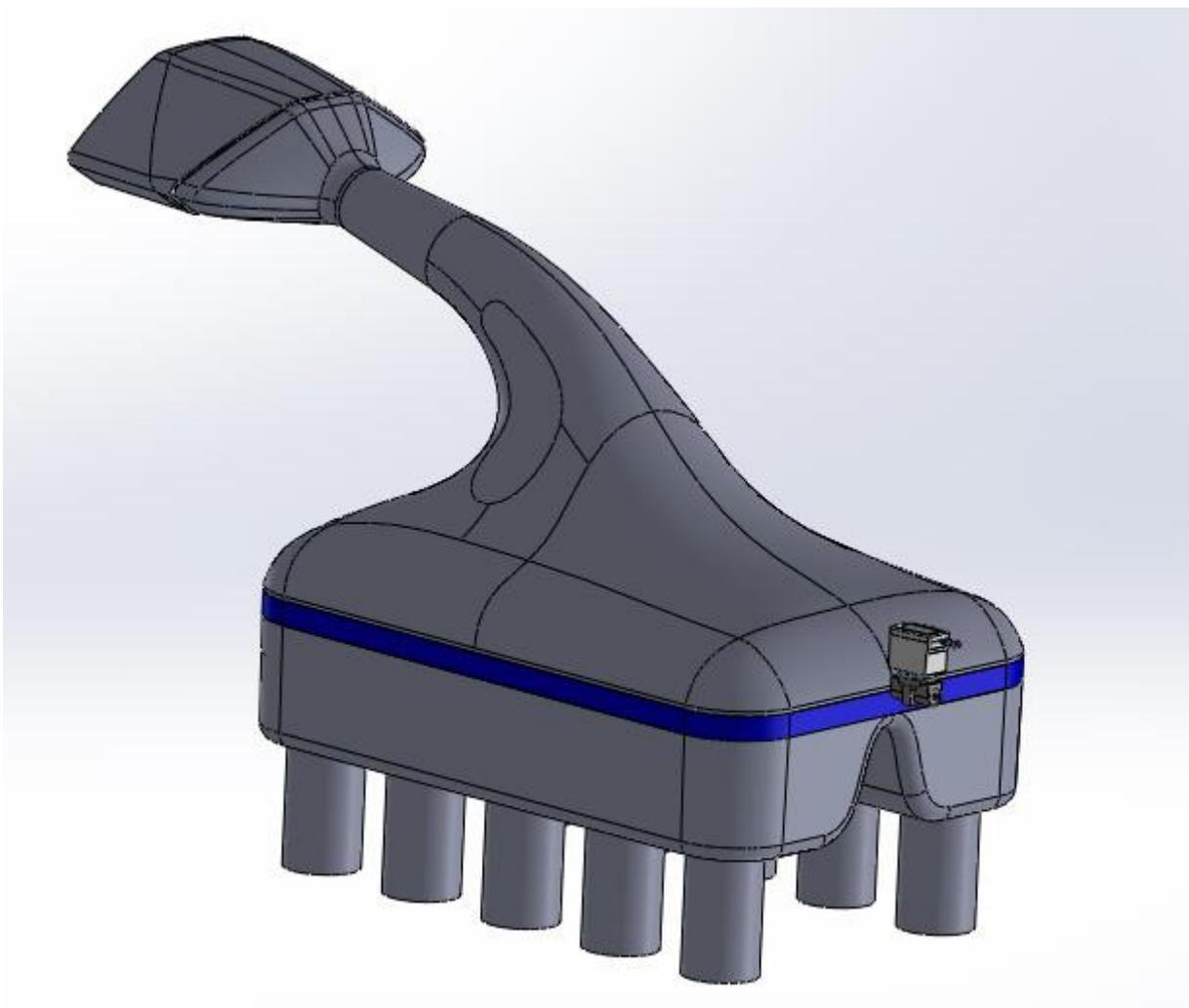
Refer to "Glossary of Terms" on page 29 for passage symbols.

### Normally Closed (N.C.)

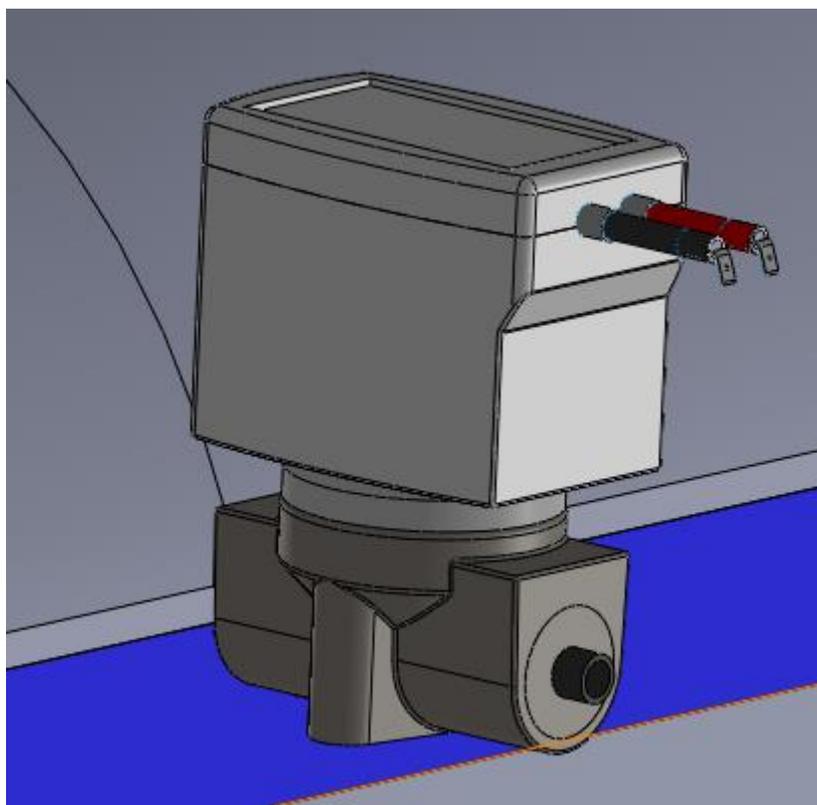
Size	Port size	Orifice diameter [mm]	Model	Flow-rate characteristics			Operating pressure range		Weight [g]
				C [dm <sup>3</sup> /(s·bar)]	b	Cv	① Used with vacuum (Pa abs)	② Used with pressure (MPa G)	
1	1/8, 1/4	2	VX214	0.63	0.63	0.23	0.1 to atmospheric pressure	0 to 1.0	300
		3		1.05	0.68	0.41		0 to 0.6	300
		5		2.20	0.39	0.62		0 to 0.2	300
2	1/4, 3/8	4	VX224	1.90	0.52	0.62		0 to 1.0	460
		7		3.99	0.44	1.08		0 to 0.15	460
3	1/4, 3/8	5	VX234	1.96	0.55	0.75		0 to 1.0	580
		8		5.67	0.33	1.58	0 to 1.0	580	
		10		5.74	0.64	2.21	0 to 0.1	580	

**Tabella 4.12:** Scheda tecnica della valvola

Di seguito appare quindi il modello finale dell'airbox con l'elettrovalvola posizionata nella parete destra, con tutte le supposizioni e le scelte applicate dall'inizio della pianificazione fino ad ora:



**Immagine 4.13:** Disegno del complessivo finale



**Immagine 4.14:** Particolare della valvola



## 5. CONCLUSIONI

Questa tesi ha riguardato il metodo di sviluppo, la simulazione e la realizzazione di un prototipo di sistema WOPI (Water Out Pressure In) che andrà implementato in un airbox al fine di eliminare la presenza di umidità e pioggia al suo interno migliorando la prestazione del motore a combustione interna del veicolo, raggiungendo una minore caduta di pressione tra il suo ingresso e la sua uscita.

Gli obiettivi che erano stati prefissati in fase preliminare erano i seguenti:

- Realizzare un metodo di sviluppo che portasse alla scelta dell'impostazione più performante per il sistema WOPI
- La simulazione fluidodinamica (CFD) dell'airbox in condizione di aria a velocità di numero di Mach 0,6 e 1 sia in caso di WOPI aperto e chiuso
- L'analisi dei risultati ottenuti
- La scelta di una elettrovalvola adatta allo scopo
- La produzione di un prototipo che simuli il funzionamento del WOPI (tramite codice Arduino e componenti elettromeccanici compatibili con esso)

Il metodo di sviluppo è proseguito molto linearmente nelle scelte effettuate, soprattutto perché molte delle seguenti sono risultate lampanti per produrre il prototipo maggiormente funzionale allo scopo.

Le simulazioni CFD, a causa del fatto che per ottenere risultati attendibili sono necessari giorni interi per ogni singola prova, hanno richiesto molto tempo per essere affinate e eseguite con criterio. Alla fine comunque sono stati ottenuti risultati compatibili con le previsioni, anche se positivi in maniera meno sensibile di quanto si potesse sperare.

La valvola, a causa delle ridotte dimensioni desiderate, è stata di difficile reperimento sul mercato. Comunque la scelta è ricaduta su un modello che è in grado di soddisfare sia gli ingombri che le limitate differenze di pressione tra i suoi monte e valle.

## Conclusioni

---

Infine il prototipo, che personalmente è la parte del progetto che più mi ha dato soddisfazione, con relativa semplicità elettronica è in grado di aprire e chiudere la valvola in totale autonomia, in funzione dei soli segnali elettrici del potenziometro dell'acceleratore e del tergitristalli frontale dell'auto.

## **BIBLIOGRAFIA**

- G. Ferrari, *Motori a combustione interna*, 2008
- ANSYS FLUENT 12.0, *User's Guide*, 2009
- S. Fitzgerald e M. Shiloh, *Arduino – Il libro dei progetti*, 2013
- P. J. Roache, *Verification and Validation in Computational Science and Engineering*, 1998
- Z. Zeng e R. Grigg, *A Criterion for Non-Darcy Flow in Porous Media*
- C. Saccani, A. Bianchini, M. Pellegrini, *Richiami di componentistica*, A.A. 2015-2016
- F. De Crescenzo, *Progettazione concettuale e raccolta dei bisogni*, A.A. 2016-2017
- F. De Crescenzo, *Architettura di prodotto*, A.A. 2016-2017
- <https://www.arduino.cc/>
- <http://forum.arduino.cc/index.php>
- <https://www.cfd-online.com/>
- [https://www.cfd-online.com/Wiki/Turbulence\\_free-stream\\_boundary\\_conditions](https://www.cfd-online.com/Wiki/Turbulence_free-stream_boundary_conditions)
- <https://www.researchgate.net/>
- <https://www.grc.nasa.gov/www/k-12/rocket/mflchk.html>



## **RINGRAZIAMENTI**

*I miei primi ringraziamenti vanno alla prof.ssa Francesca De Crescenzo per tutti i consigli e i suggerimenti che mi ha dato, sia per quanto riguarda la sua materia che per il mondo del lavoro e per tutto l'aiuto che non ha mai esitato a fornirmi.*

*Grazie alla prof.ssa Beatrice Pulvirenti per aver accettato di farmi da correlatrice e avermi introdotto ad un mondo, quello delle simulazioni CFD, che all'epoca ancora era una totale scoperta e incognita per me.*

*Ringrazio il dott. Gaetano Bergami e tutti i dipendenti della BMC per avermi dato l'opportunità di fare quest'esperienza che mi ha fatto imparare tante cose, sia didattiche che sul mondo del lavoro.*

*Ringrazio tutto il FabLab di Cesena per avermi iniziato al mondo di Arduino, ancora pura fantasia per me quando ho cominciato a venire alle vostre serate. Specialmente ringrazio Roberto Raggini, senza di te probabilmente il mio modellino sarebbe ancora da finire.*

*Grazie a tutti i miei amici, troppi per potervi nominare uno per uno, per esserci sempre stati e avermi aiutato a superare tutti i momenti difficili che ho incontrato in questi ultimi anni di università, specialmente alla fine quando la mia voglia di studiare stava vacillando pericolosamente.*

*Grazie anche a te Isa, mi hai sempre spronato a diventare una persona ed un ingegnere migliore di quello che ero quando mi hai conosciuto e, modestia a parte, penso tu abbia fatto un buon lavoro.*

*Grazie ai miei genitori Gilberto e Renata, per aver creduto ancora una volta in me nonostante le premesse non fossero delle migliori, visto il risultato e il tempo impiegato per terminare la mia laurea triennale, e non di meno per l'ennesimo sostegno economico che mi hanno dato, d'ora in poi sarò io a provvedere a me stesso e se necessario anche a voi.*

## Bibliografia e ringraziamenti

---

*Infine, come per la mia laurea triennale, ringrazio e saluto i miei nonni: Germano, Aldo, Virginia ed Anna. Ovunque voi siate spero vi arrivi il mio pensiero e vorrei foste presenti alla fine di questo difficile percorso.*

*A tutti voi, grazie di cuore.*

# Codice Arduino

## Software

Tramite il compilatore di Arduino sono stati elaborati due file di codice, uno per l'ECU e uno per la centralina del WOPI, in modo da simulare il più fedelmente possibile il funzionamento del sistema di apertura e chiusura della valvola se fosse installato in un veicolo. Vediamo nello specifico questi file di codice (dopo // ci sono i commenti che spiegano il significato della relativa stringa):

### ECU:

```
<<
#define ledPin 7           //Rimpiazza ogni menzione del ledPin col valore 7 durante la compilazione
#include <mcp_can.h>       //Libreria CAN
#include <SPI.h>           //Libreria per dialogare col Transceiver CAN

const int SPI_CS_PIN = 9; //Pin di controllo periferica SPI
int ledOff = 3;           //Pin di segnalazione tergi OFF
int ledInt = 4;           //Pin di segnalazione tergi Intermittente
int ledCont = 5;          //Pin di segnalazione tergi ON continuo
int ledAcc = 6;           //Pin di segnalazione tergi Accelerazione
int ledDec = 7;           //Pin di segnalazione tergi Decelerazione
int tergi0 = 16;          //Pulsante simulazione tergi OFF
int tergi1 = 17;          //Pulsante simulazione tergi Intermittente
int tergi2 = 18;          //Pulsante simulazione tergi ON Continuo
int acc_ne = 14;          //Pulsante simulazione Accelerazione
int dec_ne = 15;          //Pulsante simulazione Decelerazione
int CAN_pl1 = 0;          //Payload del Frame CAN1
int CAN_pl2 = 0;          //Payload del Frame CAN2
int premuta;              //Flag rilevamento premuta pulsante
```

```
unsigned char stmp[8];
long int attesa = 0;

MCP_CAN CAN(SPI_CS_PIN);

void setup() {
    //qui definisco quali sono i miei OUTPUT e i miei INPUT (i pulsanti)
    pinMode(ledOff, OUTPUT);           //LED monitor segnale
    pinMode(ledInt, OUTPUT);           //LED monitor segnale
    pinMode(ledCont, OUTPUT);          //LED monitor segnale
    pinMode(ledAcc, OUTPUT);           //LED monitor segnale
    pinMode(ledDec, OUTPUT);           //LED monitor segnale
    pinMode(16, INPUT);                 //Accelerazione
    pinMode(17, INPUT);                 //Decelerazione
    pinMode(18, INPUT);                 //Tergicristallo fermo
    pinMode(14, INPUT);                 //Tergicristallo intermittente
    pinMode(15, INPUT);                 //Tergicristallo continuo

    Serial.begin(115200);               //Inizializzazione Seriale per dialogare col PC

    while (CAN_OK != CAN.begin(CAN_500KBPS)) // inizializziamo il messaggio del can bus e la
    velocità del segnale : baudrate = 500 Kbps
    {
        Serial.println("CAN BUS Shield init fail"); //avvisi se la centralina è partita o no
        Serial.println(" Init CAN BUS Shield again");
        delay(100); //Attesa prima di reinizializzare il transceiver CAN in caso di fallimento
    }

    Serial.println("CAN BUS Shield init ok!");
    Serial.println("Inizializzazione Led");
    //un ciclo che verifichi che tutti i led siano funzionanti e li accendo con segnale 1
}
```

```
digitalWrite(ledOff,1);
digitalWrite(ledInt,1);
digitalWrite(ledCont,1);
digitalWrite(ledAcc,1);
digitalWrite(ledDec,1);
delay(2000);
digitalWrite(ledOff,1);           //ora lascio acceso solo il led che mi dà segnale di tergicristallo off
digitalWrite(ledInt,0);
digitalWrite(ledCont,0);
digitalWrite(ledAcc,1);
digitalWrite(ledDec,0);
for (int i = 0; i <=7; i++) {
```

*//inizializzazione del frame CAN a 0 (sequenza di bit con la prima posizione che non ci interessa, ma solo la seconda e la terza dove metterò i valori). Inizialmente metterò 0 in tutte le posizioni perché va vuotato per 8 volte. Poi gli dico che il tergicristallo è spento, come abbiamo definito prima col led rosso acceso. La prima posizione non ci interessa, nella seconda metto la posizione del tergicristallo, nella terza se stiamo accelerando o decelerando*

```
    stmp[i] = 0;
}
stmp[1] = 1 ;           //inizializzazione frame CAN con tergi spento (allineamento alla condizione sopra)
stmp[2] = 1 ;           //inizializzazione del frame CAN in accelerazione

}
```

*//qui inizia il ciclo dove definisco cinque condizioni tutte uguali dove gestisco i segnali che trasmetterò al WOPI coi pulsanti.*

```
void loop() {

  if (digitalRead (tergi0) == 0) {
    stmp[1] = 1;           //avrò qui 1 2 o 3 a seconda della posizione del tergicristallo
    digitalWrite(ledOff,1);
    digitalWrite(ledInt,0);
    digitalWrite(ledCont,0);
    premuta = 1;         //flag che ho definito io per rappresentare la premuta che mette la variabile
a 1. Se il pulsante è stato premuto stampa a monitor cosa sta inviando, poi compone il messaggio in
codice esadecimale 0x70 (preso in rete dalla libreria dei CAN ID della Mazda)
  }

  if (digitalRead (tergi1) == 0) {
    stmp[1] = 2;
    digitalWrite(ledOff,0);
    digitalWrite(ledInt,1);
    digitalWrite(ledCont,0);
    premuta = 1;
  }

  if (digitalRead (tergi2) == 0) {
    stmp[1] = 3;

    digitalWrite(ledOff,0);
    digitalWrite(ledInt,0);
    digitalWrite(ledCont,1);
    premuta = 1;

  }

//adesso faccio lo stesso nella posizione 2, quella riguardante l'accelerazione
}
```

```
if (digitalRead (acc_ne) == 0) {
  stmp[2] = 1;
  digitalWrite(ledAcc,1);
  digitalWrite(ledDec,0);
  premuta = 1;

}
if (digitalRead (dec_ne) == 0) {
  stmp[2] = 0;
  digitalWrite(ledAcc,0);
  digitalWrite(ledDec,1);
  premuta = 1;
}

if (premuta or ((millis() - attesa) >= 50)) { //Ogni 50 milliSec se non ci sono altri invii reinvia il
Frame CAN
  Serial.print("Invio frame tergicristallo = ");
  Serial.println(stmp[1]);
  Serial.print("Invio frame Acceleratore = ");
  Serial.println(stmp[2]);
  CAN.sendMsgBuf(0x70,0, 8, stmp); //Invio frame stmp con 8 posizioni
  premuta = 0; //resetto il flag premuta
  delay(100); //pausa di 0,1 secondi tra un invio e l'altro
}
}
>>
```

Aperto il monitor di Arduino possiamo vedere l'inizializzazione della scheda, il loop del bus che imposta la velocità d'informazione ed inizia a trasmetterle. Inizializza lo shield CAN e verifica che funzioni.

Vediamo ora l'inizializzazione dei led sia sulla scheda che sul monitor, poi come visto anche nel codice verranno tutti spenti (portati a 0) tranne quello del tergicristallo spento e dello stato di accelerazione. Ora se schiaccio il pulsante di altri stati del tergicristallo o dell'acceleratore cambieranno ovviamente sia l'illuminazione che gli stati 0 e 1 del mio frame. Tutto questo viene scritto dal codice che vedremo poi come verrà inviato allo slave. Se l'ECU non riceve nessuna premuta (variazione di segnale) comunque ogni 50 millisecondi reinvia lo stesso frame per tenere occupato il bus.

## WOPI:

```
<<
#include <SPI.h>                //Libreria per dialogare col Transceiver CAN
#include "mcp_can.h"           //Libreria CAN

const int SPI_CS_PIN = 9;     //Pin di controllo periferica SPI
const int led_pausa = 3;     //Dichiarazione degli altri pin e delle variabili che servono
const int led_ev = 4;
const int ev = 7;

bool acc_ne = 1;
unsigned int last_cycle;
unsigned int last_command;
int stato_tergi = 0;
int stato_acc = 0;
int canId;
int ciclo = 0;
unsigned char len = 0;

unsigned char buf[8];
int seq1[7] = {1,0,0,0,0,0,0};
```

*//1 indica l'apertura della valvola (un secondo), 0 la chiusura (un secondo), e cinque secondi di pausa (data dai cinque 0). Se voglio aumentare la pausa devo aggiungere degli 0 e aumentare di conseguenza la dimensione [] dell'array (idem per aumentare l'apertura ma aggiungerei degli 1)*

```
int seq2[5] = {1,0,0,0,0};
```

```
int seq3[2] = {1,0};
```

```
unsigned long cont = 0;
```

```
int flag = 0;
```

```
MCP_CAN CAN(SPI_CS_PIN); //Inizializzazione libreria CAN
```

```
void setup()
```

```
{
```

```
  Serial.begin(115200); //inizializzo la seriale che serve per dialogare col PC per il debug
```

```
  while (CAN_OK != CAN.begin(CAN_500KBPS)) //inizializzo il can bus con v = 500kbps
```

```
  {
```

```
    Serial.println("CAN BUS Shield init fail"); //avvisi se la centralina è partita o no
```

```
    Serial.println("Init CAN BUS Shield again"); //se ci sono problemi reinizializzo
```

```
    delay(100);
```

```
  }
```

```
  pinMode(led_pausa, OUTPUT); //definisco i miei pin di output (ev = elettrovalvola)
```

```
  pinMode(led_ev, OUTPUT); //con questo pilota il led verde, con quello sopra il giallo
```

```
  pinMode(ev, OUTPUT); //qui pilota il relè
```

```
//inizializzazione dei led facendoli lampeggiare uno alla volta
```

```
  Serial.println("CAN BUS Shield init ok!");
```

```
  digitalWrite(led_pausa , 1);
```

```
  digitalWrite(led_ev , 0);
```

```
digitalWrite(ev , 0);
delay(1000);           //durata della lampeggiata
digitalWrite(led_pausa , 0);
digitalWrite(led_ev , 0);
digitalWrite(ev , 0);
delay(1000);          //durata pausa della lampeggiata
digitalWrite(led_pausa , 0);
digitalWrite(led_ev , 1);
digitalWrite(ev , 0);
delay(1000);
digitalWrite(led_pausa , 0);
digitalWrite(led_ev , 0);
digitalWrite(ev , 0);
delay(1000);

digitalWrite(led_pausa , 0);
digitalWrite(led_ev , 0);
digitalWrite(ev , 1);
delay(1000);          //durata apertura dell'elettrovalvola
digitalWrite(led_pausa , 0);
digitalWrite(led_ev , 0);
digitalWrite(ev , 0);  //ora richiudo l'elettrovalvola
Serial.println("Init Complete!"); //inizializzazione board completata

//il mio frame trasmesso dalla ECU arriva all'interno di un buffer (memoria intermedia) che
//viene ricevuto ora dalla scheda del WOPI e viene "svuotato" per sicurezza nel caso arrivino file
//corrotti/spazzatura, può succedere

for (int i = 0; i<=7; i++) {           //pulizia del buffer che contiene il frame CAN
  buf[i]=0;
}
```

```
}
```

```
void loop() {
```

```
leggi_CAN();           //va a leggere sul BUS, funzione che ho scritto in fondo
```

```
condizioni_CAN();     //altra funzione sotto che mi verifica le condizioni
```

```
if (ciclo==1) {
```

```
    int j = 0;
```

```
    int val;
```

```
    cont=millis();
```

```
    //crea una variabile temporanea da usare in val per creare il lampeggio del led per un secondo, che ogni secondo dà la scansione periodica dell'array, dovuto dopo a val >=1000 millisecondi
```

```
    while (j<7) {
```

```
        val = millis() - cont;
```

```
        digitalWrite(ev,seq1[j]);
```

```
        digitalWrite(led_ev,seq1[j]);
```

```
    //seq1 è il mio array che vado a scansionare, definito all'inizio del codice
```

```
    if (val >= 1000) {
```

```
        digitalWrite(led_pausa,1);
```

```
        j++;           //io continuo a scansionare e campionare finchè non finisce l'array
```

```
        cont=millis();
```

```
        leggi_CAN();   //ripeto i comandi qui altrimenti il frame rimarrebbe "non letto"
```

```
        condizioni_CAN(); //se mentre ciclo il ciclo 1 cambiano le condizioni del frame io devo essere pronto a leggerlo o lo perdo
```

```
    if (ciclo!=1) j=7;
```

```
    }  
  }  
  Serial.println("Passo di qui....");  
  digitalWrite(ev,0);  
  digitalWrite(led_pausa,0);  
}
```

```
if (ciclo==2) {  
  int j = 0;  
  
  int val;  
  cont=millis();  
  
  while (j<5) {  
    val = millis() - cont;  
    digitalWrite(ev,seq2[j]);  
    digitalWrite(led_ev,seq2[j]);  
  
    if (val >= 1000) {  
      digitalWrite(led_pausa,1);  
      j++;  
      cont=millis();  
      leggi_CAN();  
  
      condizioni_CAN();  
      if (ciclo!=2) j=5;  
    }  
  }  
}
```

```
Serial.println("Passo di qui...");  
digitalWrite(ev,0);  
digitalWrite(led_pausa,0);  
}
```

```
leggi_CAN();  
condizioni_CAN();
```

```
if (ciclo==3) {  
  int j = 0;  
  int val;  
  cont=millis();
```

```
  while (j<2) {  
    val = millis() - cont;  
    digitalWrite(ev,seq3[j]);  
    digitalWrite(led_ev,seq3[j]);
```

```
    if (val >= 1000) {  
      digitalWrite(led_pausa,1);  
      j++;  
      cont=millis();  
      leggi_CAN();  
      condizioni_CAN();  
      if (ciclo!=3) j=2;
```

```
    }  
  }  
  Serial.println("Passo di qui....");  
  digitalWrite(ev,0);
```

```
digitalWrite(led_pausa,0);

}

}

void leggi_CAN () {
  if(CAN_MSGAVAIL == CAN.checkReceive()) { // controllo se c'è data coming
    CAN.readMsgBuf(&len, buf); // read data, len: data length, buf: data buf
    unsigned char canId = CAN.getCanId();
    Serial.println("-----"); //riga vuota per spaziare i risultati

    Serial.print("get data from ID: ");
    Serial.println(canId); //stampa a video il mio frame e il mio canId

    //0x70 in esadecimale corrisponde a 112 in decimale. Questo serve per fare in modo che solo
    //questo tipo di frame venga accettato per aprire/chiedere l'elettrovalvola, qui non ho problemi perché
    //ho solo un'informazione ma se ne arrivassero altre farebbe confusione

    if (canId == 0x70) {
      for (int i = 0; i<=7; i++) {
        Serial.print(buf[i]);

        //qui separo valore per valore l'array che verrà stampato a monitor da sinistra a destra. La posizione 0
        //dell'array non mi interessa, la posizione 1 e 2 invece rappresentano rispettivamente la posizione del
        //tergicristallo e dell'accelerazione, se spingo un pulsante diverso vedrò che varierà l'array

      }
    }
    Serial.println();

  }
}
```

*//leggo il frame e l'ho messo nel buffer, la condizione 0 non mi interessa, prendo solo la 1 e la 2*

```
void condizioni_CAN() {
  if (( buf[1] == 1) and buf[2]==0 and !(ciclo ==1) ){
    ciclo = 1;
    //se in buf[1] ho 1 il ciclo lo pongo = 1 e farà una cosa, se = 2 lo pongo = 2 e farà un'altra
    cosa e via così. Per ciclo = 1 io apro la valvola per un secondo e la tengo chiusa per 5 nel caso
    io stia decelerando (dovuto al buf[2]==0) e il tergicristallo sia spento (misura di sicurezza nel
    caso si accumulaci acqua o umidità, poi ovviamente posso modificare le tempistiche di tutti i casi.
  }
  if (( buf[1] == 2) and buf[2]==0 ){
    ciclo = 2; //in questo caso scarico per 1 secondo e chiudo per 3 secondi
  }
  if (( buf[1] == 3) and buf[2]==0 ){
    ciclo = 3; //in questo caso scarico per 1 secondo e chiudo per 1 secondo
  }
  if (buf[2]==1){
    ciclo = 4;           //ciclo vuoto nel caso io stia accelerando
    digitalWrite(ev,0); //scrivo in questo caso di porre l'ev = 0
    digitalWrite(led_ev,0); //scrivo anche di spegnere il suo led
  }
}
>>
```

