# PROMETHEUS ECO RACING NTUA

Prototype electric vehicles paving the way to a more sustainable electrified future



Από που προερχόμαστε;





# Διαγωνισμός Shell Eco Marathon





Παγκόσμιος διαγωνισμός εξοικονόμησης ενέργειας προσανατολισμένος στην αυτοκίνηση. Οι φοιτητές κατασκευάσουν βιώσιμα οχήματα υπερυψηλής απόδοσης και χαμηλής κατανάλωσης.



## ΙΔΡΥΣΗ ΟΜΑΔΑΣ





Ο διαγωνισμός Shell Eco Marathon



# Οχήματα Ομάδας





ΠΥΡΦΟΡΟΣ "Ι"

ΠΥΡΦΟΡΟΣ "ΙΙ"



# ΦΕΤΙΝΟΙ ΣΤΟΧΟΙ-Urban Concept





# ΦΕΤΙΝΟΙ ΣΤΟΧΟΙ-Urban Concept





Η στροφή αυτή αποτελεί την μεγαλύτερη πρόκληση που έχει αντιμετωπίσει ποτέ η ομάδα μας.





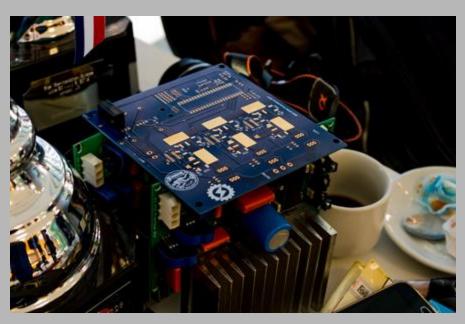


#### Τι πάει να πει κατασκευή ενός οχήματος από το μηδέν;



Εκεί που άλλες ομάδες αγοράζουν ένα μεγάλο μέρος των υποσυστημάτων τους, εμείς είμαστε η μοναδική ομάδα στην Ελλάδα και από τις ελάχιστες στην Ευρώπη, η οποία σχεδιάζει και κατασκευάζει εξ'ολοκλήρου όλα τα συστήματα του οχήματός της, από τον κινητήρα και τον αντιστροφέα μέχρι το σασί και το κέλυφος.

#### ΥΠΟΣΥΣΤΗΜΑΤΑ ΟΧΗΜΑΤΟΣ

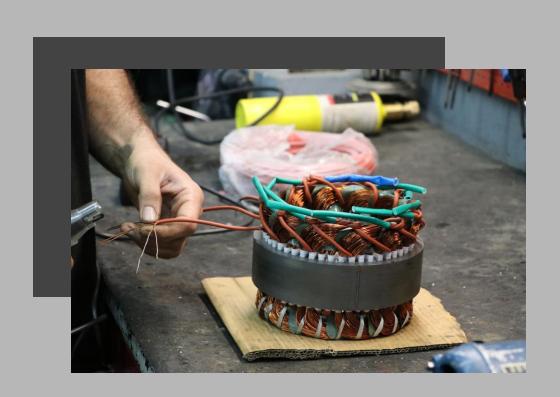




Ο αντιστροφέας και άποψη της θέσης όλων των ηλεκτρολογικών στοιχείων στο όχημα

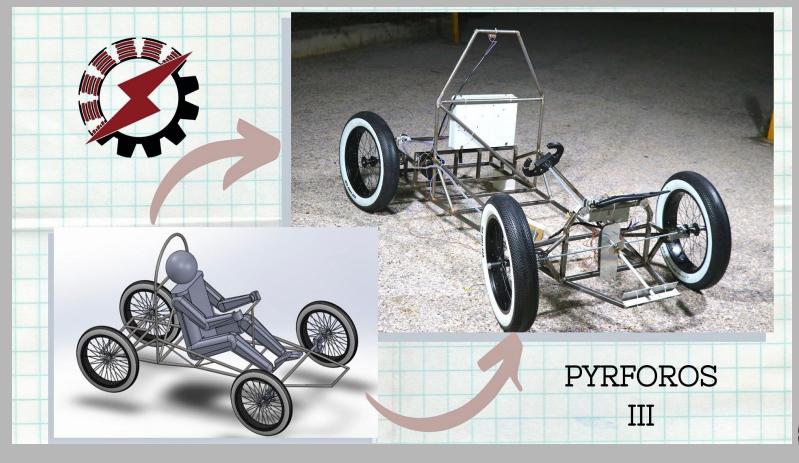


#### ΚΙΝΗΤΗΡΑΣ URBAN ΟΧΗΜΑΤΟΣ





#### ΣΑΣΙ URBAN ΟΧΗΜΑΤΟΣ









#### ΟΙ ΣΥΝΕΡΓΑΤΕΣ ΜΑΣ





Raycap











#### ΙΔΡΥΜΑ ΑΝΤΩΝΙΟΣ ΚΑΙ ΙΩΑΝΝΗΣ ΑΓΓΕΛΙΚΟΥΣΗΣ





















#### PROMETHEUS ECO RACING

#### **EMBEDDED AI:**

real-time electric motor predictive maintenance interactive workshop σχετικά με:

- Προβλεπτική συντήρηση ηλεκτρικού κινητήρα και ανίχνευση πιθανών σφαλμάτων με χρήση μηχανικής μάθησης
- Υλοποίηση συστήματος πραγματικού χρόνου σε επεξεργαστή Cortex-M series με Tensorflow Lite





powered by





#### Σκοπός του Workshop

- Εισαγωγή στον προγραμματισμό ενός μικροελεγκτή
- Εισαγωγή στα βήματα εκπαίδευσης ενός νευρωνικού δικτύου
- Παρουσίαση εφαρμογών της μηχανικής μάθησης στην προβλετική συντήρηση κινητήρων
- Βασικός Στόχος: Να αναπτύξετε περαιτέρω το πρότζεκτ και να υλοποιήσετε τα δικά σας πρότζεκτ μηχανικής μάθησης σε μικροελεγκτές





#### Που διαφοροποιείται το Embedded AI;

- Real-time
- Συλλογή δεδομένων μέσω αισθητήρων
- Περιβάλλοντα μικρής υπολογιστικής ισχύος
- Περιβάλλοντα περιορισμένης μνήμης
- Εκπαίδευση σε διαφορετικό περιβάλλον και η μεταφορά της γνώσης στον μικροελεγκτή





#### Predictive Maintenance Use Case

Προβλεπτική συντήρηση κινητήρα: η ανάλυση του φάσματος των κραδασμών με σκοπό την διάγνωση προβλημάτων (ζυγοστάθμισης κλπ.)

Industry standard: ανάλυση κραδασμών Η διάγνωση αυτή μπορεί να γίνει και με νευρωνικά δίκτυα.

- Καταγραφή δεδομένων (επιταχυνσιόμετρο)
- Εκπαίδευση νευρωνικού δικτύου
- Χρήση του μικροελεγκτή για προβλέψεις





# **Hands-on:** υλοποίηση σε Teensy 4.0 (Cortex-M) με Tensorflow Lite

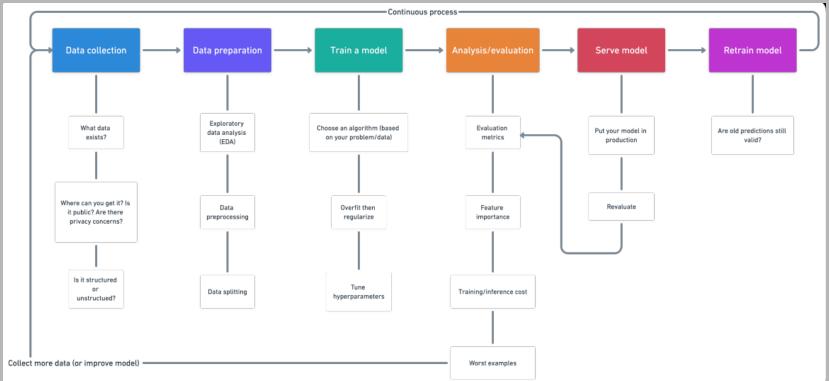
"Embedded AI; real-time electric motor predictive maintenance"

Kindly sponsored by





#### Τυπικά βήματα για σχεδιασμό συστημάτων Machine Learning





#### Workshop Setup

- Laptops
- Clone το github repo:
   <a href="https://github.com/prometheus-eco-racing/embedded-ai-sfhmmy22-att">https://github.com/prometheus-eco-racing/embedded-ai-sfhmmy22-att</a>
- Θα πρέπει να έχεις εγκαταστημένα στον Η/Υ σου
- Arduino IDE
- Teensyduino add-on
- Arduino MPU6050 library
- Arduino Tensorflow Lite library





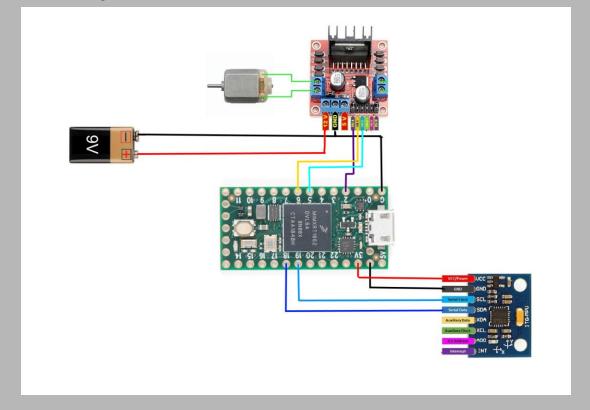
### Τα υλικά που θα χρησιμοποιηθούν

- Teensy 4.0
- Jumpers
- 9V Battery Clip
- USB CABLE 2.0 MICRO
- BREADBOARD 170 TIE MINI
- DC MOTOR
- L289N motor driver





## Step 0: Κύκλωμα

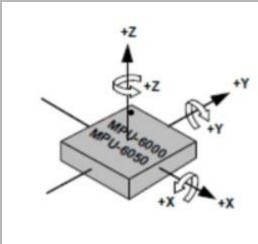






#### Step 1: Data Collection

- Αισθητήρας MPU6050 (αξελερόμετρο + γυροσκόπιο)
- Συλλογή επιταχύνσεων 6 DOG (3 Μεταφορικές, 3 Στροφικές)



#### Aρχείο create\_dataset:

- Το πρόγραμμα αυτό παίρνει μετρήσεις από γυροσκόπιο και αξελερόμετρο
- Εκτύπωνει τις μετρήσεις στο Serial
   Μοπίτοι σε κατάλληλη μορφή (ώστε να αποθηκευτούν σε csv)





#### Step 2: Train the Model

- Ανοιγμα google colab (link στο github repo)
- Ανέβασμα των αρχείων balanced.csv και unbalanced.csv στο google colab data folder.
- Parse μετρήσεων (60000 μετρήσεις από το create\_dataset.ino).
- Χωρισμός μετρήσεων σε 1000 samples των 60 features.
- Φιλτράρισμα σε παράθυρα των 40ms (10 μετρήσεις κάθε 4 ms) για 6
   παραμέτρους του αισθητήρα (aX, aY, aZ, gX, gY, gZ)
- Δημιουργία sample, επανάληψη 1000 φορές για την δημιουργία του dataset.



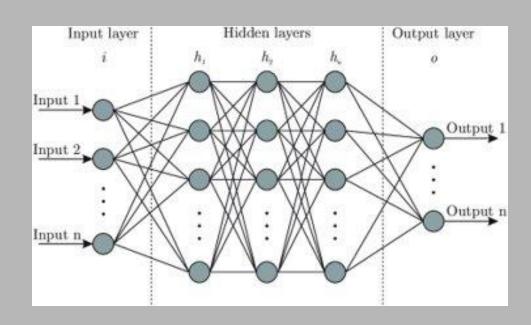


#### Άσκηση για το σπίτι...

- Πόσα layers εχει το δικό μας νευρωνικό δίκτυο;
- Πόσους νευρώνες έχει το κάθε layer;

Υπόδειξη: Δείτε τον κώδικα train.ipynb / Build and Train the Model (Google Collab)

 Add a dropout layer to prevent overfitting







### Step 3: real time classification in Teensy

Με το script classify.inο γίνεται η κατηγοριοποίηση του κινητήρα σε ζυγοσταθμισμένο και αζυγοστάθμιστο. Ο τρόπος με τον οποίο ενημερώνεται ο χρήστης για το αποτέλεσμα της κατηγοριοποίησης είναι μέσω του Serial output του Teensy, στο οποίο τυπώνεται η πρόβλεψη του μοντέλου. Για τιμές > 0.5 ο κινητήρας θεωρείται ζυγοσταθμισμένος.





#### Exercise: complete the code in classify.ino

#### Fill the input tensor:

```
// Allocate memory for the model's input and output tensors
tflInterpreter->AllocateTensors();
// Get pointers for the model's input and output tensors
tflInputTensor = tflInterpreter->input(0);
// Finally fill the input tensor
tflInputTensor->data.f[????] = ???;
tflInputTensor->data.f[????] = ???;
tflInputTensor->data.f[????] = ???;
```





#### Exercise: complete the code in classify.ino

#### Fill the output tensor:

```
// Get pointers for the model's output tensors
tflOutputTensor = tflInterpreter->output(0);
```

```
// Fetch the output (prediction)
float prediction = tflOutputTensor->data.f[0];
```

Με βάση αυτό εκτυπώστε στο serial monitor αν είναι balanced ή unbalanced





### Ευχαριστούμε για την προσοχή σας

Περιμένουμε να μας στείλετε τα δικά σας project και ερωτήσεις





# Ευχαριστούμε τον χορηγό των υλικών που χρησιμοποιήθηκαν σήμερα





#### Η ομάδα μας σήμερα...



