SELEN

Smart ELectrolysis based on GrEen ENergy

noesisHUB Team (K-12):

Βουτυράκης Μάνος, Δακορώνιας Λάμπρος, Κατσούρης Μπάμπης, Κίτσιος Μάξιμος, Παργινού Δέσποινα, Τάρλας Γιάννης, Φιοράκης Κωνσταντίνος, Χωματάς Απόστολος

Υπεύθυνοι Συντονισμού της ομάδας:

Μιχαέλα Βλάχου (εκπαιδεύτρια της εταιρείας noesisHUB)

Ιωάννης Μανωλόπουλος

Τι είναι το SELEN ;

Το SELEN είναι ένα εργοστάσιο παραγωγής υδρογόνου. Η διαδικασία παραγωγής γίνεται με φιλικό τρόπο προς το περιβάλλον αξιοποιώντας τις ανανεώσιμες πηγές ενέργειας του ήλιου και του αέρα. Το υδρογόνο που παράγεται χρησιμοποιείται για καύσιμο των αυτοκινήτων αντί των συμβατικών καυσίμων (π.χ., πετρέλαιο, βενζίνη). Στόχος του SELEN είναι να καταπολεμηθεί η παγκόσμια κλιματική αλλαγή.

Εικόνα 1: Συνολική μακέτα του προτεινόμενου συστήματος.

Πώς λειτουργεί;

- Παράγεται ηλεκτρική ενέργεια με τη χρήση φωτοβολταϊκού πάνελ και ανεμογεννήτριας.
- Η ενέργεια αυτή αποθηκεύεται σε μία μπαταρία (δηλαδή η μπαταρία φορτίζεται από την «πράσινη» ενέργεια) και στη συνέχεια από την μπαταρία στέλνεται στην συσκευή ηλεκτρόλυσης.
- Στη συσκευή ηλεκτρόλυσης γίνεται η διάσπαση του μορίου του νερού στα άτομα από τα οποία αποτελείται, δηλαδή σε δύο άτομα υδρογόνου

και σε ένα άτομο οξυγόνου. Η διάσπαση του μορίου του νερού πραγματοποιείται με την παροχή συνεχούς ηλεκτρικού ρεύματος από την μπαταρία.

- Η παραγωγή υδρογόνου ελέγχεται με έναν αισθητήρα υδρογόνου.
- Μόλις το ποσοστό υδρογόνου είναι αρκετό, στέλνεται σήμα μέσω ενός ασύρματου πομπού στον ασύρματο δέκτη του αυτοκινήτου.
- Όταν το αυτοκίνητο λάβει το σήμα, τότε είναι προγραμματισμένο να κινηθεί προς τα εμπρός μέχρι να συναντήσει κάποιο εμπόδιο και να σταματήσει.

Τα πλεονεκτήματα του SELEN

- Χρησιμοποιεί ανανεώσιμες πηγές ενέργειας για παραγωγή ηλεκτρικής ενέργειας.
- Δεν μολύνει το περιβάλλον.
- Το υδρογόνο ως καύσιμο είναι φιλικό προς το περιβάλλον.
- Συμβάλλει στην επίλυση του προβλήματος της κλιματικής αλλαγής.

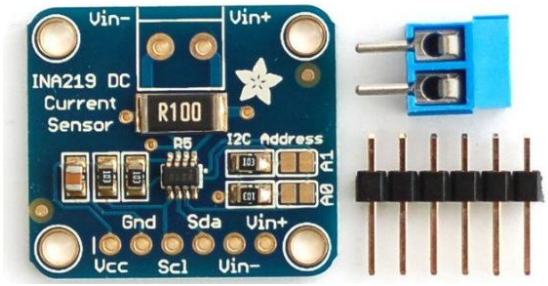
Τα μέρη του εργοστασίου μας

Η ανεμογεννήτρια θα συμβάλει στην παραγωγή ηλεκτρικής ενέργειας από την αιολική ενέργεια. Συγκεκριμένα, η κινητική ενέργεια του αέρα περιστρέφει τον DC κινητήρα της ανεμογεννήτριας ο οποίος με τη σειρά του παράγει συνεχές ηλεκτρικό ρεύμα (δηλαδή η διαδικασία είναι αντίστροφη αυτής που παρατηρούμε

συνήθως από έναν κινητήρα ο οποίος περιστρέφεται όταν τροφοδοτείται από ηλεκτρικό ρεύμα που προέρχεται από μια μπαταρία).

Το φωτοβολταϊκό πάνελ απορροφά την ηλικιακή ακτινοβολία και παράγει ηλεκτρική ενέργεια. Μαζί με την ανεμογεννήτρια μας βοηθάνε να έχουμε ηλεκτρικό ρεύμα χρησιμοποιώντας ανανεώσιμες πηγές ενέργειας.

Το συνεχές ηλεκτρικό ρεύμα που παράγεται από τις Ανανεώσιμες Πηγές Ενέργειας χρησιμοποιείται για τη φόρτιση της μπαταρίας (η μπαταρία που χρησιμοποιήθηκε είναι τύπου Lithium Polymer Battery, 3.7V, 1850mAh) η οποία παρέχει στη συσκευή ηλεκτρόλυσης την απαραίτητη ηλεκτρική ενέργεια.



Το ποσοστό ενέργειας που συνεισφέρει η ανεμογεννήτρια και το φωτοβολταϊκό πάνελ στην φόρτιση της μπαταρίας, ελέγχεται με τη χρήση δύο ολοκληρωμένων κυκλωμάτων INA219 [1]. Η πληροφορία ρεύματος και τάσης που παράγει κάθε μια διάταξη (δηλαδή η ανεμογεννήτρια και το φωτοβολταϊκό πάνελ) εισάγονται στον μικροελεγκτή Arduino Uno, που διαθέτει το εργοστάσιο παραγωγής Υδρογόνου, μέσω του πρωτοκόλλου επικοινωνίας I2C. Συγκεκριμένα, το κάθε ολοκληρωμένο INA219 επικοινωνεί με τον μικροελεγκτή μέσω δύο καλωδίων μεταφοράς σήματος τα οποία συνδέονται στους ακροδέκτες SCL και SDA του μικροελεγκτή Arduino Uno. Τέλος, προκειμένου να ελέγχουμε και την ενέργεια (δηλαδή το ρεύμα και την τάση) που εισέρχεται από την μπαταρία προς τη συσκευή ηλεκτρόλυσης, συνδέουμε με αντίστοιχο τρόπο ένα τρίτο ολοκληρωμένο κύκλωμα INA219. Επομένως, τα τρία INA219 αποτελούν στοιχεία εισόδου του μικροελεγκτή Arduino Uno του εργοστασίου παραγωγής Υδρογόνου. Η ανάγνωση των τιμών που εισάγει

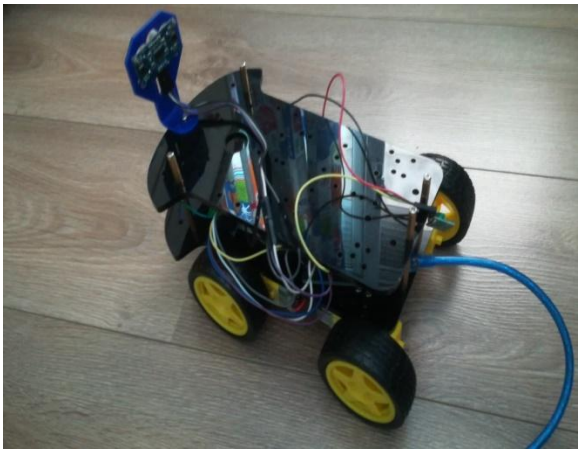
το κάθε INA219 στον μικροελεγκτή γίνεται μέσα από το περιβάλλον προγραμματισμού Arduino IDE, με τη διαδικασία AnalogReadSerial.

Η συσκευή ηλεκτρόλυσης Hoffman θα μας βοηθήσει να πραγματοποιήσουμε τον τελικό στόχο του εργοστασίου μας, δηλαδή να πραγματοποιηθεί η παραγωγή του υδρογόνου το οποίο θα χρησιμοποιηθεί ως καύσιμο των αυτοκινήτων. Με τη χρήση νερού (ή διαλύματος αλατόνερου για γρηγορότερο αποτέλεσμα) και του ηλεκτρικού ρεύματος που προέρχεται από την μπαταρία και τις ανανεώσιμες πηγές ενέργειας, θα πραγματοποιηθεί η ηλεκτρόλυση. Συγκεκριμένα, θα διασπαστεί το μόριο του νερού στα άτομα από τα οποία αποτελείται, δηλαδή σε δύο άτομα Υδρογόνου και σε ένα άτομο Οξυγόνου. Τα άτομα του Υδρογόνου και του Οξυγόνου συλλέγονται με τη μορφή φυσαλίδων στους σωλήνες που βρίσκονται πάνω από τα ηλεκτρόδια της καθόδου (αρνητικό ηλεκτρόδιο) και της ανόδου (θετικό ηλεκτρόδιο) αντίστοιχα, σε αναλογία 2:1 (όπως αυτή καθορίζεται από το μόριο του νερού).

Για να συλλέξουμε το αέριο Υδρογόνο, μόλις ανοίξουμε την στρόφιγγα του σωλήνα που βρίσκεται πάνω από το ηλεκτρόδιο της καθόδου, χρησιμοποιούμε ένα μπαλόνι (το κίτρινο μπαλόνι στην προηγούμενη εικόνα) μέσα στο οποίο έχουμε τοποθετήσει τον αισθητήρα Υδρογόνου τύπου MQ-8 (απεικονίζεται αριστερά). Το στόμιο από το μπαλόνι τοποθετείται στην έξοδο της στήλης που συλλέγεται το υδρογόνο. Με αυτόν τον τρόπο ενημερωνόμαστε για το ποσοστό Υδρογόνου που έχουμε έτοιμο προς διάθεση στα ενδιαφερόμενα οχήματα. Οι τιμές του αισθητήρα Υδρογόνου στέλνονται σε έναν από τους αναλογικούς ακροδέκτες του (έχει επιλεγεί ο A0) και η ανάγνωση των τιμών αυτών γίνεται μέσα από τη διαδικασία AnalogReadSerial. Επομένως, ο αισθητήρας υδρογόνου παρέχει ένα ακόμα στοιχείο εισόδου προς τον μικροελεγκτή Arduino Uno.

Όταν οι τιμές Υδρογόνου ξεπεράσουν ένα καθορισμένο όριο (την τιμή του ορίου την προσδιορίσαμε στις 150 μονάδες και προέκυψε μετά από πολλαπλές μετρήσεις που πραγματοποιήσαμε όταν η στρόφιγγα της συσκευής ηλεκτρόλυσης ήταν κλειστή και όταν αυτή ήταν ανοικτή) τότε ο μικροελεγκτής του εργοστασίου παραγωγής υδρογόνου θα δώσει εντολή στον ασύρματο πομπό (transmitter module) για να εκπέμψει σήμα ειδοποίησης προς τα γειτονικά οχήματα που είναι εφοδιασμένα με έναν ασύρματο δέκτη. Ο ασύρματος πομπός έχει συνδεθεί στον ψηφιακό ακροδέκτη 10 του Arduino Uno και αποτελεί το στοιχείο εξόδου του μικροελεγκτή. Σημειώνουμε, ότι στοιχείο εξόδου θεωρείται και η οθόνη του υπολογιστή που τυπώνει τα μηνύματα (μέσα από την εντολή Serial.println όπως θα δούμε στη συνέχεια) και η οποία θα μπορούσε να αντικατασταθεί με μια απλή οθόνη Liquid-Crystal Display (LCD).

Τα μέρη του οχήματός μας



Το Arduino Car κατασκευάζεται από ορισμένα βασικά εξαρτήματα όπως βάσεις στήριξης και τροχοί). Για τον προγραμματισμό της επιθυμητής λειτουργίας του οχήματος χρησιμοποιούμε έναν μικροελεγκτή Arduino Uno. Στον μικροελεγκτή αυτό συνδέεται ένας αισθητήρας απόστασης υπερήχων (το trigger και το echo είναι συνδεδεμένα με τον μικροελεγκτή στους ακροδέκτες 12 και 13 αντίστοιχα) και ένας ασύρματο δέκτης (RF που είναι συνδεδεμένος στον ψηφιακό ακροδέκτη 10) που έχουν τον ρόλο των στοιχείων εισόδου προς τον μικροελεγκτή.

Επίσης, τέσσερεις κινητήρες συνεχούς ρεύματος (DC motors) είναι συνδεδεμένα στον μικροελεγκτή, μέσω

δύο συσκευών οδήγησης κινητήρα (Dual DC motor driver controller board που είναι στην ουσία ένας ενισχυτής ρεύματος), αποτελώντας τα στοιχεία εξόδου του μικροελεγκτή.

Η λειτουργία του οχήματος θα ενεργοποιείται με τη λήψη από τον ασύρματο δέκτη του σήματος (receiver module) που θα εκπέμψει ο ασύρματος πομπός του εργοστάσιο παραγωγής υδρογόνου. Ο αισθητήρας απόστασης θα σταματάει την κίνηση του οχήματος σε περίπτωση εμποδίου, δηλαδή μόλις φτάσει κοντά στο εργοστάσιο παραγωγής Υδρογόνου.

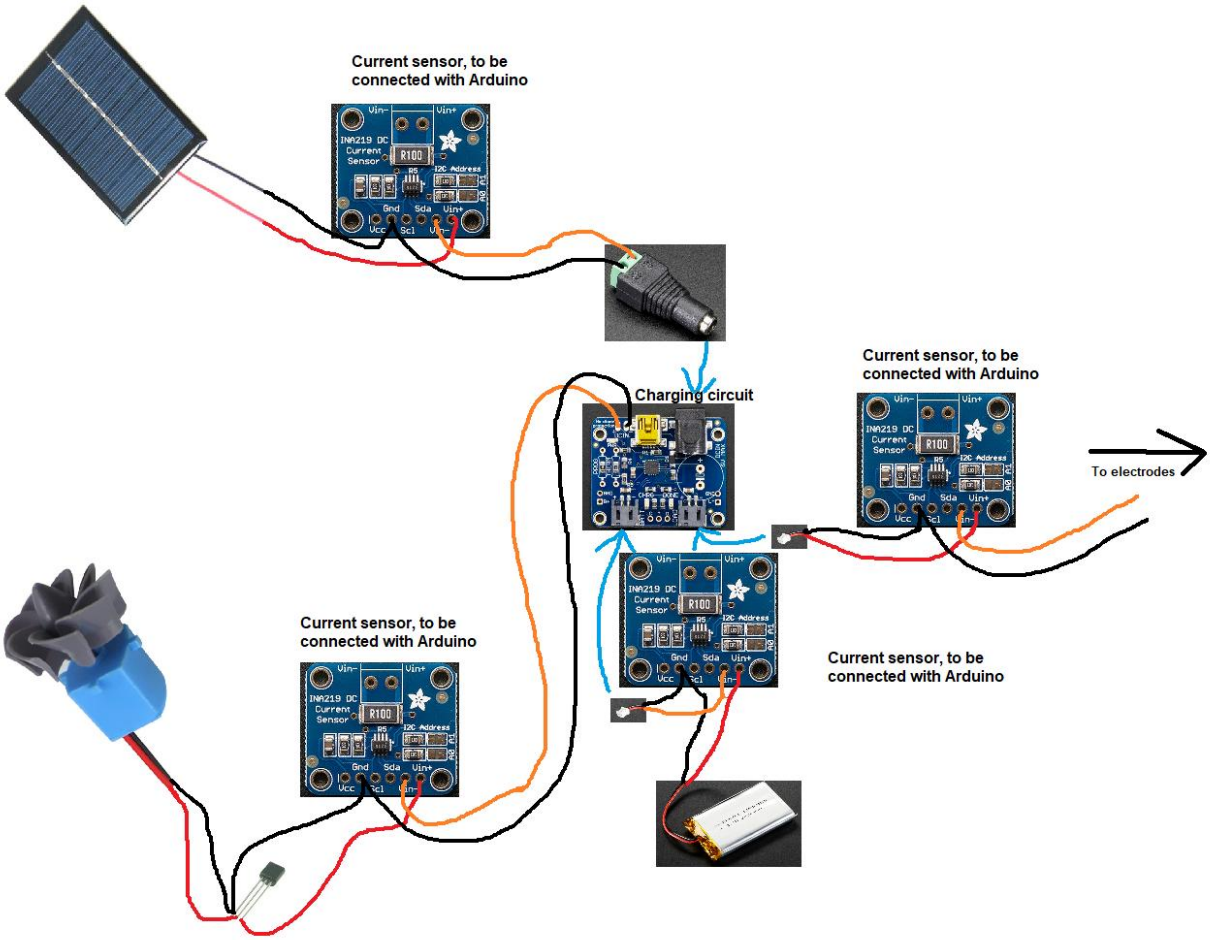
3-D Εκτύπωση για την κατασκευή της ανεμογεννήτριας και της βάσης του φωτοβολταϊκού πάνελ

Η ανεμογεννήτριά μας (βάση, προσαγωγέας για τον ρότορα του κινητήρα, πτερύγια) σχεδιάστηκε και μοντελοποιήθηκε με τη βοήθεια του ανοικτού προγράμματος 3-D εκτύπωσης FreeCAD. Τα τμήματα αυτά εκτυπώθηκαν στο εργαστήριο της noesisHUB σε 3-D εκτυπωτή.

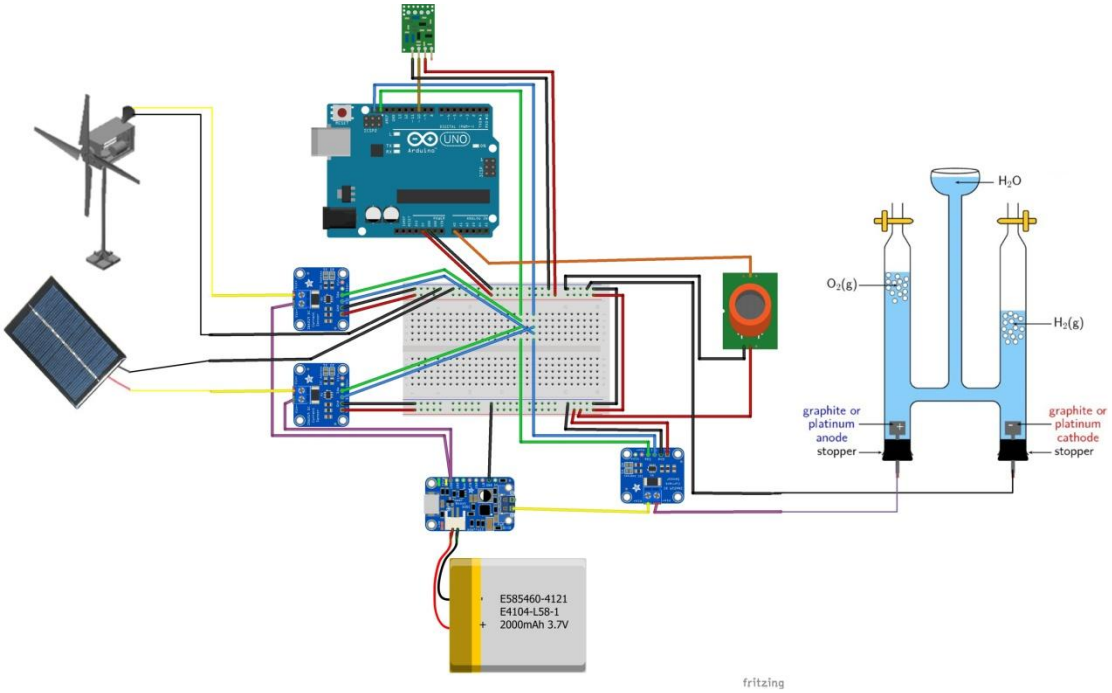
Συνδέσεις και Κυκλώματα (Hardware)

Για να κατασκευαστούν σωστά όλα τα κυκλώματα χρειάστηκε πρώτα να σχεδιαστούν χρησιμοποιώντας ανοικτά σχεδιαστικά προγράμματα (όπως για παράδειγμα το σχεδιαστικό πρόγραμμα προσομοίωσης κυκλωμάτων Fritzing). Στη συνέχεια προχωρήσαμε στις συνδέσεις οι οποίες γίνονταν με κύριο άξονα τους δύο μικροελεγκτές Arduino Uno (για το εργοστάσιο παραγωγής Υδρογόνου και για το όχημα) όπως εξηγήθηκε παραπάνω.

Στην Εικόνα 2 υπάρχει φωτογραφία της σχεδίασης του κυκλώματος για το εργοστάσιο παραγωγής Υδρογόνου. Έμφαση δίνεται στην συνδεσμολογία των κυκλωμάτων παρακολούθησης INA219 (για λόγους απλοποίησης του κυκλώματος δεν παρουσιάζεται ο μικροελεγκτής Aduino Uno). Στη συνέχεια, στην Εικόνα 3 δίνεται η πλήρης διάταξη του εργοστασίου παραγωγής Υδρογόνου με τη βοήθεια του προγράμματος Fritzing.

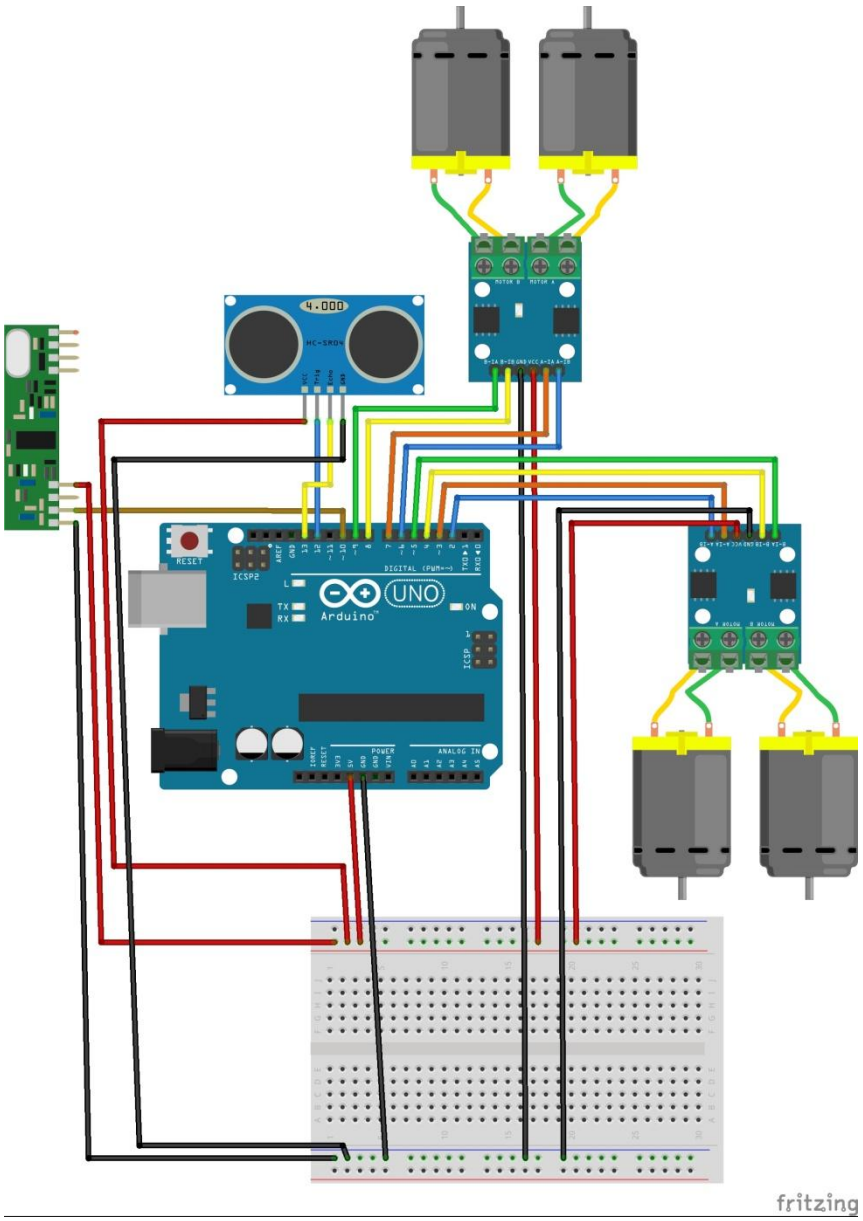


Εικόνα 2: Μέρος της συνδεσμολογίας που ακολουθήσαμε για το εργοστάσιο παραγωγής υδρογόνου. Κύριος σκοπός του διαγράμματος είναι ο τρόπος σύνδεσης των κυκλωμάτων INA219 (σημείωση: Η μετρική διάταξη INA219 που συνδέεται στην μπαταρία, δεν χρησιμοποιήθηκε στην τελική κατασκευή).



Εικόνα 3: Πλήρης συνδεσμολογία του εργοστασίου παραγωγής Υδρογόνου σχεδιασμένο στο πρόγραμμα ανοικτού κώδικα Fritzing.

Επίσης, στην Εικόνα 4 βρίσκεται η συνδεσμολογία του κυκλώματος του οχήματος, η οποία πραγματοποιήθηκε και αυτή στο σχεδιαστικό πρόγραμμα Fritzing.



Εικόνα 4: Πλήρης συνδεσμολογία του Οχήματος σχεδιασμένο στο πρόγραμμα ανοικτού κώδικα Fritzing.

Προγραμματιστικός κώδικας (Software)

Προγραμματίσαμε τους μικροελεγκτές του εργοστασίου παραγωγής υδρογόνου (Εικόνα 5) και του οχήματος (Εικόνα 7) σε Arduino IDE. Συγκεκριμένα, με τη βοήθεια του περιβάλλοντος οπτικού προγραμματισμού Ardublock (βλέπε Εικόνα 6) παρουσιάστηκαν στην ομάδα απλά τμήματα προγραμματιστικού κώδικα και έγινε ο παραλληλισμός με τον παραγόμενο κώδικα στο Arduino IDE (Εικόνα 7). Με αυτόν τον τρόπο ήταν εύκολη η

υλοποίηση των περισσότερο απαιτητικών κομματιών λογισμικού στο προγραμματιστικό περιβάλλον ανοικτού κώδικα για μικροελεγκτές Wiring.

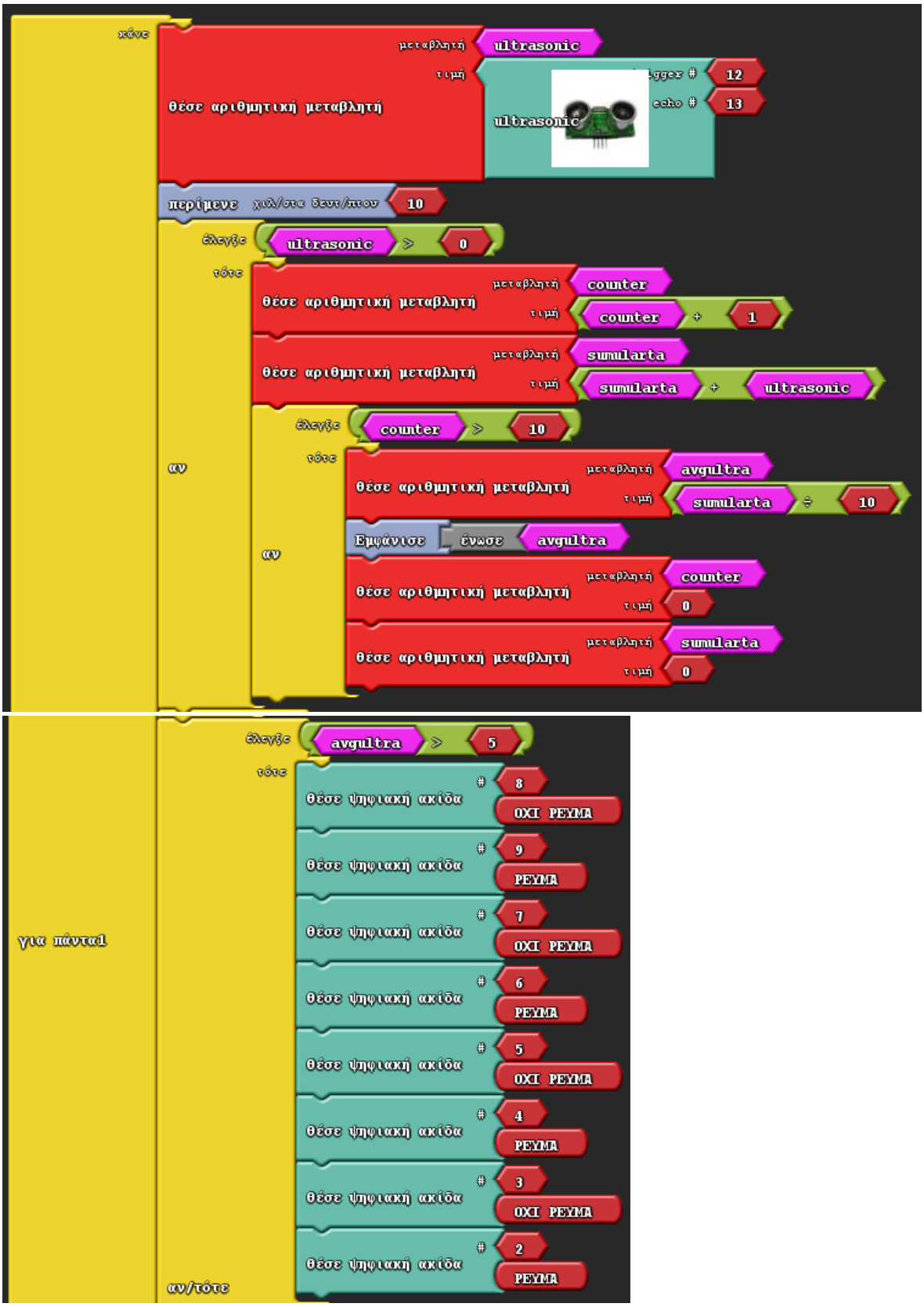
FACTORY | Arduino 1.6.0

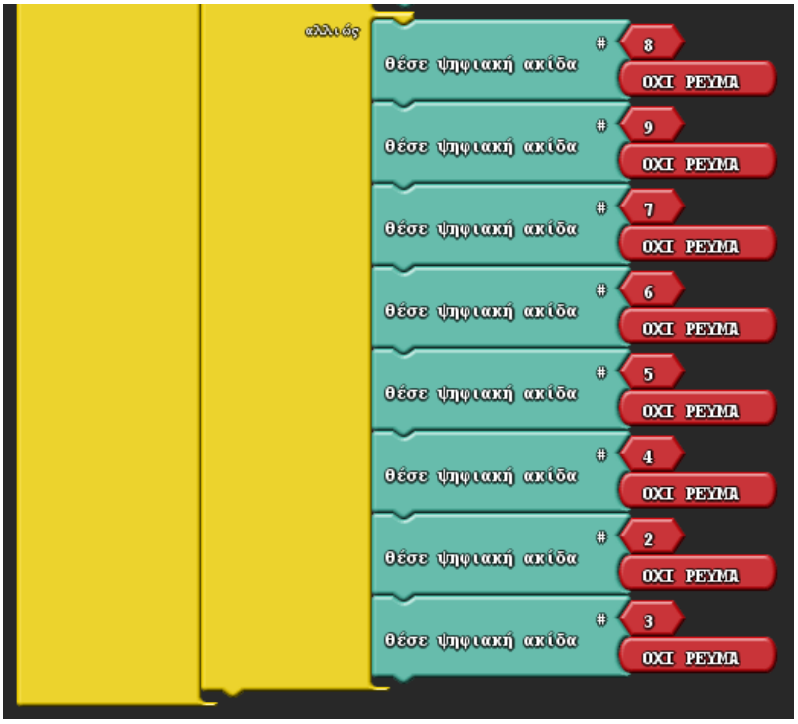
File Edit Sketch Tools Help

FACTORY

```
#include <Wire.h>
#include <Adafruit_INA219.h>
#include <RH_ASK.h>
#include <SPI.h> // Not actually used but needed to compile

RH_ASK driver(2000, 11, 10, 8, false); // Default speed is 2000bps; R
Adafruit_INA219 ina219_A;
Adafruit_INA219 ina219_B(0x41);
Adafruit_INA219 ina219_D(0x45);

float shunt_voltage;
float bus_voltage;
float current_mA;
float load_voltage;

void setup(void) {
  Serial.begin(9600);
  Serial.println("Hello!");
  ina219_A.begin();  // Initialize first board (default address 0x40)
  ina219_B.begin();  // Initialize second board with the address 0x41
  ina219_D.begin();  // Initialize second board with the address 0x45
  if (!driver.init())
     Serial.println("init failed");

}
 void loop() {
   read_print(ina219_A, "Sun");
   read_print(ina219_B, "Wind");
   read_print(ina219_D, "electrolysis");
   int sensorValue = analogRead(A0);
   Serial.print("Hydrogen concentration: ");
   Serial.println(sensorValue);
   if (sensorValue > 150)
   {
     Serial.println("Hydrogen detected, sending signal to car!");
     const char *msg = "Move_car";
     driver.send((uint8_t *)msg, strlen(msg));
     driver.waitPacketSent();
     delay(1000);
   }
   Serial.println("");
   Serial.println("");
   Serial.println("");
   delay(1000);
 }

 void read_print(Adafruit_INA219 ina_219, const String& ina_219_assigned_letter) {
   shunt_voltage = 0;
   bus_voltage = 0;
   current_mA = 0;
   load_voltage = 0;

  shunt_voltage = ina_219.getShuntVoltage_mV();
  bus_voltage = ina_219.getBusVoltage_V();
  current_mA = ina_219.getCurrent_mA();
  load_voltage = bus_voltage + (shunt_voltage / 1000);

//  Serial.print("INA_219_"); Serial.print(ina_219_assigned_letter); Serial.print(" Bus Voltage:   "); Serial.print(bus_voltage); Serial.println(" V");
//  Serial.print("INA_219_"); Serial.print(ina_219_assigned_letter); Serial.print(" Shunt Voltage: "); Serial.print(shunt_voltage); Serial.println(" mV");
  Serial.print("INA_219_"); Serial.print(ina_219_assigned_letter); Serial.print(" Load Voltage:  "); Serial.print(load_voltage); Serial.println(" V");
  Serial.print("INA_219_"); Serial.print(ina_219_assigned_letter); Serial.print(" Current:       "); Serial.print(current_mA); Serial.println(" mA");
}
```

Εικόνα 5: Προγραμματιστικός κώδικας στο περιβάλλον Arduino IDE για τον μικροελεγκτή Arduino Uno του εργοστασίου παραγωγής Υδρογόνου.





Εικόνα 6: Προγραμματιστικός κώδικας του οχήματος στο περιβάλλον οπτικού προγραμματισμού Ardublock για τον μικροελεγκτή Arduino Uno.

CAR | Arduino 1.6.0

File Edit Sketch Tools Help

CAR

```
#include <RH_ASK.h>
#include <SPI.h> // Not actualy used but needed to compile

RH_ASK driver(2000, 10, 12, 8, false); // Default speed is 2000bps; RX pin = 11;

int _ABVAR_4_avgultra;
int _ABVAR_2_counter;
bool move_bool = false;
int ardublockUltrasonicSensorCodeAutoGeneratedReturnCM(int trigPin, int echoPin)
{
  int duration;
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(20);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  duration = duration / 59;
  return duration;
}

int _ABVAR_1_ultrasonic;
int _ABVAR_3_sumularta;
```

```
void setup()
{
_ABVAR_4_avgultra = 0;
pinMode( 7 , OUTPUT);
Serial.begin(9600);
_ABVAR_2_counter = 0;
pinMode( 8 , OUTPUT);
pinMode( 2 , OUTPUT);
pinMode( 6 , OUTPUT);
pinMode( 4 , OUTPUT);
_ABVAR_3_sumularta = 0;
_ABVAR_1_ultrasonic = 0;
digitalWrite( 12 , LOW );

pinMode( 9 , OUTPUT);
pinMode( 5 , OUTPUT);
pinMode( 3 , OUTPUT);
if (!driver.init())
     Serial.println("init failed");

}

void loop()
{
uint8_t buf[8];
uint8_t buflen = sizeof(buf);
if (driver.recv(buf, &buflen) || move_bool) // Non-blocking
{
  if (!move_bool) {
    // Message with a good checksum received, dump it.
    Serial.print("Hydrogen full, moving car!");
  }
  move_bool = true;
//  Serial.println((char*)buf);
  _ABVAR_1_ultrasonic = ardublockUltrasonicSensorCodeAutoGeneratedReturnCM( 12 , 13 ) ;
  delay( 10 );
  if (( ( _ABVAR_1_ultrasonic ) > ( 0 ) ))
  {
  _ABVAR_2_counter = ( _ABVAR_2_counter + 1 ) ;
  _ABVAR_3_sumularta = ( _ABVAR_3_sumularta + _ABVAR_1_ultrasonic ) ;
  if (( ( _ABVAR_2_counter ) > ( 10 ) ))
  {
  _ABVAR_4_avgultra = ( _ABVAR_3_sumularta / 10 ) ;
  Serial.print( _ABVAR_4_avgultra );
  Serial.println("");
  _ABVAR_2_counter = 0 ;
  _ABVAR_3_sumularta = 0 ;
  }
  }
```

```
  if (( ( _ABVAR_4_avgultra ) > ( 5 ) ))
  {
  digitalWrite( 8 , LOW );
  digitalWrite( 9 , HIGH );
  digitalWrite( 7 , LOW );
  digitalWrite( 6 , HIGH );
  digitalWrite( 5 , LOW );
  digitalWrite( 4 , HIGH );
  digitalWrite( 3 , LOW );
  digitalWrite( 2 , HIGH );
  }
  else
  {
  digitalWrite( 8 , LOW );
  digitalWrite( 9 , LOW );
  digitalWrite( 7 , LOW );
  digitalWrite( 6 , LOW );
  digitalWrite( 5 , LOW );
  digitalWrite( 4 , LOW );
  digitalWrite( 2 , LOW );
  digitalWrite( 3 , LOW );
  }
}

}
```

Εικόνα 7: Προγραμματιστικός κώδικας στο περιβάλλον Arduino IDE για τον μικροελεγκτή Arduino Uno του οχήματος.

Αναφορές:

[1] https://learn.adafruit.com/adafruit-ina219-current-sensor-breakout