



Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών
Σχολή Θετικών Επιστημών
Τμήμα Πληροφορικής και Τηλεπικοινωνιών

M146 – Σύγχρονες Επικοινωνίες Μικρής και Μεγάλης Εμβέλειας

Διδάσκων: Καθ. Δημοσθένης Βουγιούκας

Κατασκευή Συσκευής Αλκοτέστ

<AM>

Ευάγγελος Σιατήρας

ΠΜΣ

Μηχανική Υπολογιστών

Αθήνα, 13 Ιουλίου, 2020



Περιεχόμενα

1.ΕΙΣΑΓΩΓΗ.....	5
1.1 ΣΚΟΠΟΣ ΤΗΣ ΕΡΓΑΣΙΑΣ	5
1.2 ΡΟΛΟΣ ΤΗΣ ΣΥΣΚΕΥΗΣ ΚΑΙ Η ΣΗΜΑΝΤΙΚΟΤΗΤΑ ΤΗΣ.....	5
2 ΣΧΕΔΙΑΣΜΟΣ ΣΥΣΤΗΜΑΤΟΣ.....	6
2.1 ΕΠΙΛΟΓΗ ΑΙΣΘΗΤΗΡΑ.....	6
2.2 ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΤΟΥ ARDUINO BOARD.....	7
2.2.1 Είσοδοι και έξοδοι	7
2.2.2 Είσοδοι τροφοδότησης	8
2.3 ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΤΟΥ ΑΙΣΘΗΤΗΡΑ MQ-3	8
2.4 ΔΟΜΗ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ	9
3 ΥΛΟΠΟΙΗΣΗ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ.....	11
3.1 ΒΑΘΜΟΝΟΜΗΣΗ ΤΟΥ ΑΙΣΘΗΤΗΡΑ	11
3.1.1 Παραμετροποίηση σε επίπεδο Υλικού (Hardware)	11
3.1.2 Παραμετροποίηση σε επίπεδο Λογισμικού (Software)	11
3.2 ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ	14
3.3 ΔΙΑΔΙΚΑΣΙΑ ΜΕΤΡΗΣΗΣ	16
3.4 ΥΛΟΠΟΙΗΣΗ ΤΟΥ ΙΟΤ ΜΕΡΟΥΣ ΜΕΣΩ ΤΗΣ NODE-RED ΠΛΑΤΦΟΡΜΑΣ.....	16
3.4.1 ΑΠΑΙΤΗΣΕΙΣ NODE-RED	17
3.4.2 NODE-RED ARDUINO BOARD FLOW	17
3.4.3 NODE-RED CENTRAL SERVER FLOW	20
3.4.3.1 Βάση Δεδομένων	22
3.5 ΠΑΡΑΤΗΡΗΣΕΙΣ.....	24
4 ΣΥΣΧΕΤΙΣΗ ΤΡΟΠΟΥ ΛΕΙΤΟΥΡΓΕΙΑΣ ΜΕ ΤΗΝ ΠΡΑΓΜΑΤΙΚΗ ΧΡΗΣΗ	25
5 ΕΞΕΛΙΞΗ.....	26
6 ΒΙΒΛΙΟΓΡΑΦΙΑ	27



Κατάλογος Σχημάτων

Εικόνα 1: Μέρη του Συστήματος.....	7
Εικόνα 2: Ευαισθησία του αισθητήρα σε διαφορά αέρια	9
Εικόνα 3: Σχεδιασμός Συστήματος.....	10
Εικόνα 5: Ελάχιστα Τετράγωνα.....	14
Εικόνα 6: Arduino Board Flow	17
Εικόνα 7: Serial In Node.....	18
Εικόνα 8: MQTT Out Node	18
Εικόνα 9: Get Measurement Node	18
Εικόνα 10: Alcohol in BAC Node	19
Εικόνα 11: Form Driving Ability Node	19
Εικόνα 12: Define Result Node.....	19
Εικόνα 13: Dashboard Result with Passed Alcohol Test	19
Εικόνα 14: Dashboard Result with Failed Alcohol Test	20
Εικόνα 15: Central Server Flow	20
Εικόνα 16: MQTT In Node.....	21
Εικόνα 17: Switch Node.....	21
Εικόνα 18: Non Zero Measurement Node.....	21
Εικόνα 19: Resulting Mail	24



M146 – Σύγχρονες Επικοινωνίες Μικρής και Μεγάλης Εμβέλειας

Τίτλος Μελέτης: Κατασκευή Συσκευής Αλκοτέστ

Ευάγγελος Σιατήρας

Κατάλογος Πινάκων

<u>Πίνακας 1: Επιδράσεις του Αλκοόλ στον Άνθρωπο</u>	6
--	----------



1.Εισαγωγή

1.1 Σκοπός της Εργασίας

Σκοπός της εργασίας είναι η κατασκευή ενός πρωτοτύπου (Proof of Concept) αναλυτή της αναπνοής οπου ανιχνεύει τα επίπεδα αλκοόλ στο αίμα συμβάλλοντας στην ευαισθητοποίηση των οδηγών για τους κινδύνους που σχετίζονται με την κατανάλωση αλκοολούχων ποτών και την οδική ασφάλεια. Ανάλογα με την μέτρηση θα εκτυπώνει τα καταλληλά μηνύματα. Συγχρόνως τα δεδομένα θα συγκεντρώνονται κεντριοποιημένα για την περεταίρω ανάλυση τους. Θα είναι ένα πρωτότυπο οπου θα μπορούσε να αποτελέσει μια συσκευή η οποία τοποθετείται μέσα στο αυτοκίνητο και θα λαμβάνει ένα δείγμα και με βάση το αποτέλεσμα θα ενημερώνει τον οδηγό για την ικανότητα οδήγησής του και σε περίπτωση που είναι ανεπαρκής θα ειδοποιεί τον ορισμένο από τον ίδιο τον οδηγό οπου θα οδηγήσει αντί αυτού με ασφάλεια μέχρι τον προορισμό όπως ορίζει η νομοθεσία της Ελλάδος. Συγχρόνως τα δεδομένα από πολλές συσκευές θα συγκεντρώνονται σε μια IoT πλατφόρμα για την περεταίρω αξιοποίησή τους.

1.2 Ρόλος της Συσκευής και η Σημαντικότητά της

Είναι μια συσκευή που θα παίζει καθοριστικό ρόλο στην αποφυγή ατυχημάτων και θα καλλιεργήσει μια καλύτερη οδηγική παιδεία. Η τοποθέτηση και άρα η αναγκαστική χρήση του αισθητήρα νηφαλιότητας οδηγού στο αυτοκίνητο προλαμβάνει την επιπολαιότητα και την αδυναμία υπό την επήρεια αλκοόλ του οδηγού. Κάθε άνθρωπος και ιδιαίτερα τα άτομα νεαρής ηλικίας υπερεκτιμούν τις δυνατότητες τους μετά την κατανάλωση αλκοόλ αφού χρήζουν νηφαλιότητας και καθαρής σκέψης. Πολλές είναι οι απώλειες σε ανθρώπινες ζωές εξαιτίας αυτού του φαινομένου και ίσως στο μέλλον μία τέτοια κατασκευή βάλει τέλος στο ανεπιθύμητο αυτό φαινόμενο.

Αναλυτικότερα τα πρόσφατα στατιστικά στοιχεία δείχνουν ότι:

- 3,3 εκατομμύρια θάνατοι προκαλούνται από την κατάχρηση αλκοόλ παγκοσμίως.
- το εύρος των νέων ηλικίας 11 έως 24 ετών αποτελούν το μεγαλύτερο κίνδυνο λόγω μεγαλύτερης κατανάλωσης αλκοόλ.
- περίπου το 30% των τροχαίων ατυχημάτων στην Ελλάδα σχετίζονται με την κατάχρηση αλκοόλ.
- Ο αριθμός των ατόμων που επιλέγουν υπεύθυνα να περιορίσουν την κατανάλωση αλκοόλ ως συνάρτηση της ασφάλειας οδήγησης διαρκώς μειώνεται.



2 Σχεδιασμός Συστήματος

2.1 Επιλογή Αισθητήρα

Δεδομένου ότι οι ανθρώπινες ζωές εξαρτώνται από την ακρίβεια της μέτρησης του επιπέδου αλκοόλης στο αίμα, η αξιοπιστία του μέσου είναι ένα βασικό χαρακτηριστικό. Γι' αυτόν τον λόγο επέλεξα να χρησιμοποιήσω έναν συγκεκριμένο αισθητήρα που έχει υψηλή ευαισθησία και γρήγορο χρόνο απόκρισης, τον MQ-3.

Οι μετρήσεις που θα λαμβάνουμε μέσω του MQ-3 αφορούν το [BAC](#) δηλαδή είναι το μέτρο της ποσότητας του αλκοόλ στο αίμα ενός ατόμου επί της %.

Συγκεκριμένα στον παρακάτω πίνακα παρουσιάζεται η επίδραση του αλκοόλ με μονάδα μέτρησης το BAC στον άνθρωπο.

Αλκοόλ (g / l)	Επιδράσεις (Ενδεικτικά)
0.2 – 0.4	Ήπια ευφορία, ομιλία, ελαφρά κινητική διαταραχή και τελικά μέτρια μείωση της ικανότητας της κρίσης και της προσοχής.
0.5 – 0.8	Μείωση αντίληψης , περαιτέρω μείωση της κρίσης, τάση για επικίνδυνη οδήγηση.
0.8 – 1.0	Υπερεκτίμηση της ικανότητας οδήγησης, της τάσης για οδήγηση στη μέση του δρόμου, εμφανής καθυστέρηση στο χρόνο αντίδρασης σε ένα ερέθισμα.
1.0 – 2.0	Συναισθηματική αστάθεια, εξασθένηση της μνήμης, απώλεια κρίσης, αταξία και δυσαρθρία (δυσκολία στην άρθρωση του λόγου).
2.0-4.0	Σημάδια δηλητηρίασης, διανοητική σύγχυση, αποπροσανατολισμός, απάθεια, έντονες αντιληπτικές αλλαγές, έμετος, ακράτεια, αδυναμία να διατηρηθεί σε όρθια θέση.
4.0-5.0	Κρίσιμη κατάσταση της υγείας, κώμα, υποθερμία, ΥΠΟ αερισμός, αρτηριακή υπόταση, αναισθησία.

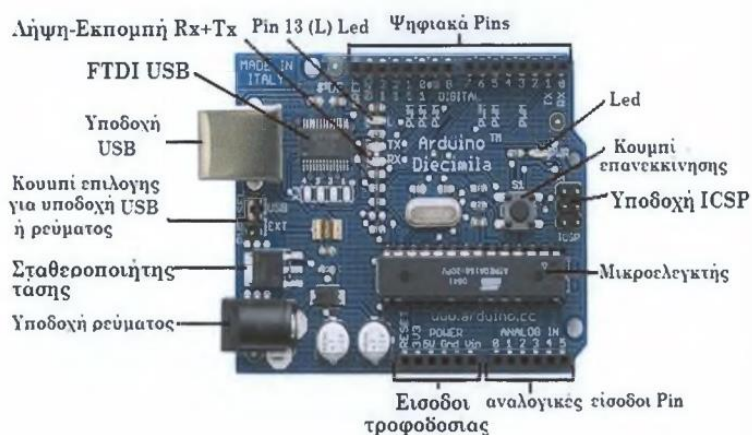
Πίνακας 1: Επιδράσεις του Αλκοόλ στον Άνθρωπο

Σε περίπτωση ανιχνευμένης περιεκτικότητας σε αλκοόλ στο αίμα είναι υψηλότερη από 0.24 g / l, σε αντίθεση με την συμβατική συσκευή μέτρησης στέλνει mail μέσω της πλατφόρμας node-red σε μια ορισμένη διεύθυνση ενημερώνοντας για την δυνητικά επικίνδυνη κατάσταση του οδηγού.



2.2 Χαρακτηριστικά του Arduino Board

Η έκδοση του Arduino που χρησιμοποιείται είναι η Duemilanove. Το arduino Duemilanove (2009) είναι μία πλατφόρμα βασισμένη στο μικροεπεξεργαστή ATmega168. Έχει 14 ψηφιακές εισόδους και εξόδους (εκ των οποίων οι έξι μπορούν να χρησιμοποιηθούν ως αποτελέσματα PWM), 6 αναλογικές εισόδους, ένα ταλαντωτή κρυστάλλου 16 MHZ, μια σύνδεση USB, μία είσοδο παροχής τροφοδοσίας, ένα αγωγό ICSP, και ένα κουμπί επαναφοράς (reset). Περιέχει όλα όσα απαιτούνται για να υποστηρίξουν τη λειτουργία του μικροεπεξεργαστή. Μπορεί να συνδεθεί με έναν υπολογιστή μέσω ενός καλωδίου USB ή να τροφοδοτηθεί με έναν προσαρμογέα ή με μία μπαταρία. Μια σχηματική αναπαράσταση με τα μέρη του συστήματος παρουσιάζεται παρακάτω.



Εικόνα 1: Μέρη του Συστήματος

2.2.1 Είσοδοι και εξόδοι

Το Arduino Duemilanove έχει 14 ψηφιακούς ακροδέκτες που μπορούν να χρησιμοποιηθούν ως εισόδοι ή εξόδοι, χρησιμοποιώντας τις pinMode (), digitalWrite (), και digitalRead () λειτουργίες. Λειτουργούν σε 5 Volt. Τα ψηφιακά pin βρίσκονται στην κορυφή της πλακέτας. Μερικά από τα ψηφιακά pin επιτρέπονται για την διαμόρφωση του πλάτους (PWM) που σημαίνει ότι μπορούν να χρησιμοποιηθούν για να στέλνουν αναλογικές πληροφορίες σε ένα εξάρτημα.

PWM: 3, 5, 6, 9, 10, και 11. Παρέχουν παραγωγή PWM μαζί με την analogWrite () λειτουργία. Οι θύρες αυτές μπορούν να ρυθμιστούν για να κάνουν διαφορετικά πράγματα από μια ψηφιακή επικοινωνία. Οι πιθανές τιμές των pin είναι in που σημαίνει ότι η πληροφορία έρχεται από τον επεξεργαστή και out ότι πηγαίνει στο εξάρτημα από τον επεξεργαστή. Επιπλέον έχει 6 αναλογικούς ακροδέκτες εισόδου. Αυτοί μπορούν να διαβάσουν αναλογικές τιμές και να τις μετατρέψουν σε αριθμό από 0-1023. Η μέτρηση της τάσης γίνεται προκαθορισμένα από 0-5Volts

Σειριακές: 0 (RX) και 1 (TX). Χρησιμοποιούνται για την λήψη (RX) και την εκπομπή (TX) των σειριακών στοιχείων TTL. Αυτές οι εισοδοί συνδέονται με τις αντίστοιχες εισόδους του τμηματικού τσιπ FTDI USB- to-TTL Serial.

SPI: 10,11, 12,13. Αυτές οι εισοδοί υποστηρίζουν την επικοινωνία SPI, η οποία, αν και παρέχεται από το hardware υλικό, δεν είναι διαθέσιμη ακόμα στη γλώσσα arduino. Το Πρωτόκολλο SPI ή



Serial Peripheral Interface Bus επιτρέπει την σειριακή σύγχρονη επικοινωνία μεταξύ ολοκληρωμένων σε πλήρης αμφίδρομη επικοινωνία.

LED: 13. Υπάρχει ενσωματωμένο led που συνδέεται με την ψηφιακή είσοδο 13. Όταν η τιμή της εισόδου είναι HIGH, το led είναι αναμμένο, ενώ όταν η τιμή της εισόδου είναι LOW, το led δεν είναι αναμμένο

Reset είναι το κουμπί επαναφοράς. Αυτό μας επιτρέπει να ξεκινήσουμε το πρόγραμμα μας πάλι. Είναι σημαντικό γιατί εκεί αποθηκεύεται οποιοδήποτε πρόγραμμα έχει φορτωθεί. Για να αλλάξουμε το πρόγραμμα που είναι αποθηκευμένο στο arduino πρέπει να στείλουμε ένα νέο πρόγραμμα στην μνήμη.

2.2.2 Είσοδοι τροφοδότησης

Οι είσοδοι τροφοδότησης είναι οι ακόλουθες: VIN. Η είσοδος τάσης στην πλατφόρμα Arduino όταν χρησιμοποιείται εξωτερική πηγή ενέργειας (διαφορετική από τα 5V της USB σύνδεσης ή άλλης προσαρμοζόμενης πηγής ενέργειας). Η παροχή ενέργειας μπορεί να γίνει διαμέσω αυτού του pin, ή, παρέχοντας τάση από το βύσμα παροχής ενέργειας, μπορούμε να έχουμε πρόσβαση στην τάση αυτή από το pin αυτό.

5V. Η προσαρμοσμένη παροχή ηλεκτρικού ρεύματος που χρησιμοποιείται για να τροφοδοτήσει το μικροεπεξεργαστή και άλλα εξαρτήματα στην πλακέτα. Αυτό μπορεί να προέλθει είτε από το VIN διαμέσω του ενσωματωμένου στην πλατφόρμα προσαρμογέα, είτε να παρασχεθεί από τη USB σύνδεση ή άλλη προσαρμοζόμενη 5V πηγή. Χρησιμοποιείται για να τροφοδοτήσει άλλες συσκευές.

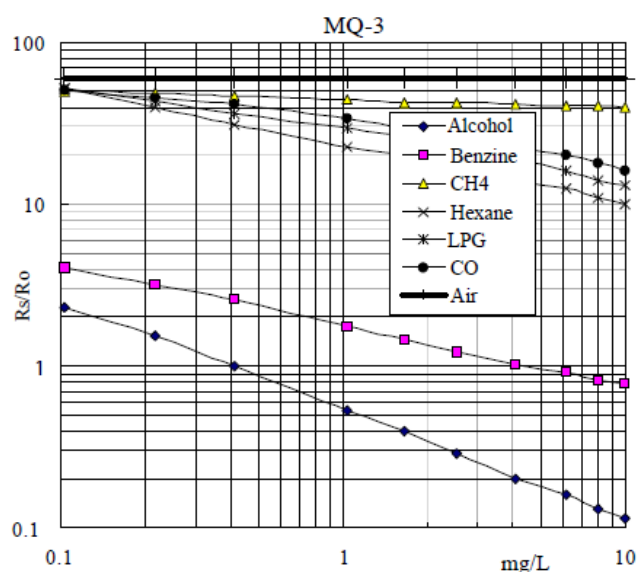
3V3. Το ολοκληρωμένο FTDI που βρίσκεται στην πλακέτα του Arduino παράγει τάση των 3.3V με μέγιστο ρεύμα 50mA. Το FTDI FT232RL παρέχει σειριακή επικοινωνία με τον ηλεκτρονικό υπολογιστή για προγραμματισμό πάνω από την θύρα USB με την βοήθεια των ανάλογων FTDI drivers.

GND. Ακροδέκτες γείωσης

2.3 Χαρακτηριστικά του αισθητήρα MQ-3

Ο αισθητήρας όπως περιγράφεται στο [datasheet](#) έχει 4 pins – VCC, GND, AO(Analog Output) and DO (Digital Output). Είναι κατασκευασμένος από Al_2O_3 (Ceramic tube) και από ένα ευαίσθητο στρώμα διοξειδίου του κασσιτέρου. Έχει μια θερμική αντίσταση η οποία αντλεί ένα σημαντικό ποσοστό ρεύματος και έτσι δεν συνίσταται η usb θύρα του Arduino ως πηγή ρεύματος χωρίς εξωτερική τροφοδοσία. Η αντίσταση του αισθητήρα εξαρτάται από την ποσότητα του αλκοόλ στον αέρα. Υψηλή περιεκτικότητα στον αέρα σημαίνει χαμηλότερη αντίσταση του αισθητήρα. Η αντίσταση του αισθητήρα μαζί με μια αντίσταση φορτίου σχηματίζει ένα διαχωριστικό τάσης. Ο αισθητήρας MQ3 δίνει την τάση σε όλη αυτή την αντίσταση φορτίου ως έξοδο. Αυτή η τιμή αυξάνεται με τη συγκέντρωση αλκοόλης στο δείγμα.

Ο αισθητήρας αντιδρά στην παρουσία μορίων αλκοόλης μεταβάλλοντας την αντίσταση R_s . Συγκεκριμένα, όπως φαίνεται από το γράφημα στο παρακάτω Σχήμα από το φύλλο δεδομένων του αισθητήρα, για αύξηση της συγκέντρωσης αλκοόλης στον αέρα υπάρχει μείωση της αντίστασης R_s .



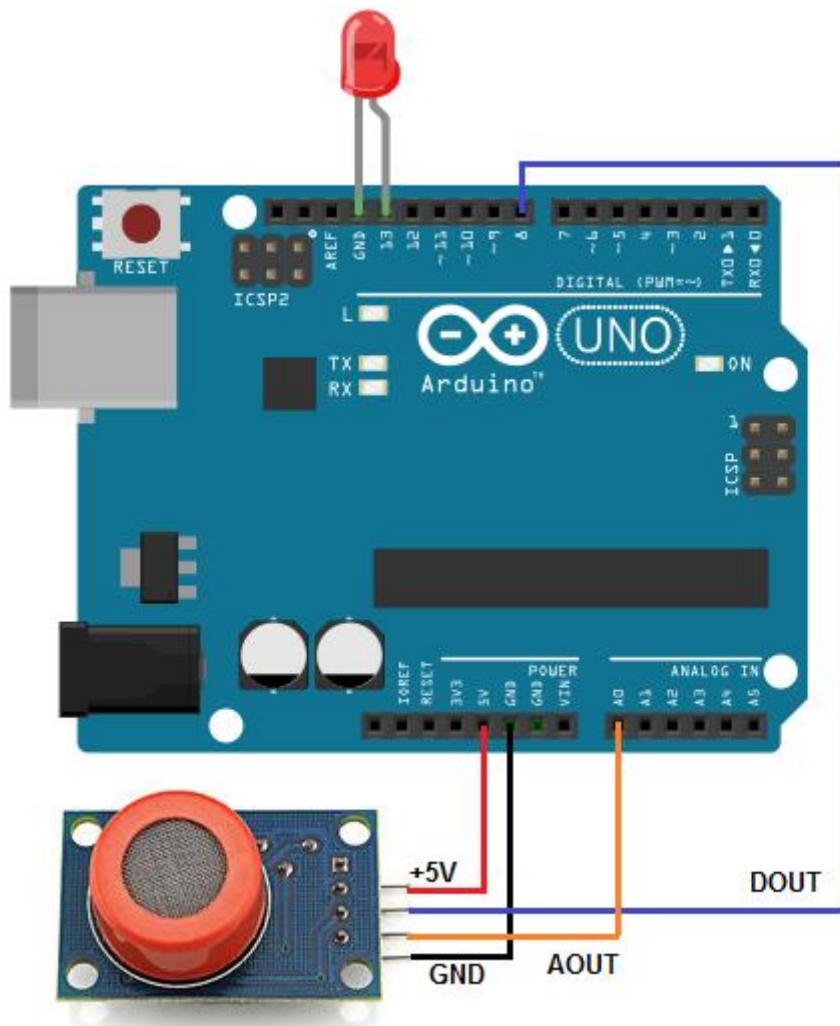
Εικόνα 2: Ευαισθησία του αισθητήρα σε διάφορα αέρια

Ο MQ-3 έχει υψηλή ευαισθησία στο αλκοόλ και έχει καλή αντίσταση στις παρεμβολές όπως από την βενζίνη τον καπνό και τους υδρατμούς. Επίσης ο συγκεκριμένος αισθητήρας μπορεί να χρησιμοποιηθεί για την ανίχνευση αλκοόλ διαφορετικών συγκεντρώσεων και είναι χαμηλού κόστους και χρησιμοποιείται σε πολλές εφαρμογές.

Το κύκλωμα του ουσιαστικά είναι ένας απλοποιημένος διαχωριστής τάσης, που αλλάζει αναλογικά την τάση V_{out} σε R_s . Το σήμα τάσης αποστέλλεται στο Arduino. Η αντίσταση R_L είναι ρυθμιζόμενη μέσω ενός ποτενσιόμετρου στην πλακέτα στην οποία συγκολλάται ο αισθητήρας, ρυθμίζει την ταχύτητα με το οποίο το σύστημα φτάνει σε σταθερές συνθήκες.

2.4 Δομή του Συστήματος

Σύμφωνα με τα παραπάνω οι θύρες που επιλέχθηκαν προκειμένου να συνδεθεί ο αισθητήρας MQ-3 στο Arduino είναι η θύρα 5V για την τροφοδοσία του αισθητήρα η θύρα GND(Ground) για την γείωση του αισθητήρα και η αναλογική θύρα AOUT για την λήψη των μετρήσεων από τον αισθητήρα. Χρησιμοποιείται και η ψηφιακή θύρα D8 από την οποία δεν χρησιμοποιούμε το output της αλλά σε μια γενικότερη περίπτωση μπορεί να χρησιμοποιηθεί για debugging.



Εικόνα 3: Σχεδιασμός Συστήματος



3 Υλοποίηση του Συστήματος

3.1 Βαθμονόμηση του αισθητήρα

3.1.1 Παραμετροποίηση σε επίπεδο Υλικού (Hardware)

Σύμφωνα με το [datasheet](#) για την σωστή παραμετροποίηση του λαμβάνουμε υπόψιν το led που ανάβει στον αισθητήρα κατά την διεξαγωγή του πειράματος. Αναλυτικότερα ρυθμίζουμε το ποτενσιόμετρο ώστε κατά την διεξαγωγή του πειράματος το led να ανάβει και αφού πλησιάσουμε το οινόπνευμα και πάρει μια μέτρηση να σβήσει έπειτα από 60 δευτερόλεπτα. Επιπλέον η διάρκεια που ενδείκνυται ως Preheat Time είναι περισσότερο από 24 ώρες.

Ωστόσο στα πλαίσια της εργασίας παρατηρείται ότι οι τιμές εξόδου του αισθητήρα σταθεροποιούνται στα 50 δευτερόλεπτα οπού είναι και ο χρόνος που ενδείκνυται ως χρόνος αναμονής πριν την διεξαγωγή του πειράματος με τον συγκεκριμένο αισθητήρα.

3.1.2 Παραμετροποίηση σε επίπεδο Λογισμικού (Software)

Αφού πάρουμε μια μέτρηση από τον αισθητήρα πρέπει να την μετατρέψουμε σε κατάλληλη κλίμακα που αναπαριστά το BAC.

Η μεταβλητή αντίσταση μέσα στον αισθητήρα είναι η R_s , και η τιμή της R_s κατά την έκθεση στην ατμόσφαιρα (κανένα δείγμα) ονομάζεται R_o . Όσο περισσότερο αλκοόλ, τόσο χαμηλότερη είναι η τιμή της R_s . Αντί να μετράμε την αντίσταση, μετράμε την τάση στο σημείο μεταξύ του αισθητήρα και μιας αντίστασης φορτίου. Ο αισθητήρας και η αντίσταση φορτίου σχηματίζουν ένα διαχωριστικό τάσης. Όσο χαμηλότερη είναι αντίσταση του αισθητήρα, τόσο υψηλότερη είναι η τιμή της τάση που διαβάζουμε στην αντίσταση φορτίου. Αυτή είναι η τιμή που διαβάζεται από τη συνάρτηση `analogRead` κυμαίνεται μεταξύ 0 και 1024 και αντιστοιχεί σε εύρος τάσης 0 έως 5 volt. Οι υψηλότερες τιμές τάσης σημαίνουν υψηλότερη συγκέντρωση αλκοόλης στο δείγμα. Η μέθοδος βαθμονόμησης όπως προτείνεται στο `datasheet` του αισθητήρα είναι η λήψη ορισμένων δειγμάτων με διαφορετικές, γνωστές συγκεντρώσεις και έπειτα να σχεδιάσουμε την $\frac{\log(R_s / R_o)}{\log(C)}$.

Αυτή είναι μια εξαιρετικά δύσκολη προσέγγιση για τους ακόλουθους λόγους:

- Ο αισθητήρας είναι ευαίσθητος στην υγρασία και τη θερμοκρασία του περιβάλλοντος και η τιμή του R_o δεν είναι σταθερή, ειδικά κατά τη διάρκεια επαναλαμβανόμενων μετρήσεων με δείγματα διαφορετικών συγκεντρώσεων.
- Είναι πρακτικά αδύνατο να υπάρχει συγκεκριμένη συγκέντρωση αλκοόλ στον αέρα ώστε να προσδιορίσουμε την αναλογία R_s / R_o .
- Το μεγαλύτερο πρόβλημα με την προσέγγιση R_s / R_o , είναι ότι δεν υπάρχουν επαρκή δεδομένα στο φύλλο δεδομένων (εκτός από ένα γράφημα το οποίο δεν δίνει αρκετή πληροφορία).

Επιλέχθηκε λοιπόν να χρησιμοποιηθεί μια διαφορετική προσέγγιση για την βαθμονόμηση του αισθητήρα όπως παρακάτω.



Θα χρησιμοποιήσουμε δείγματα τα οποία έχουν νερό και διαφορετικές ποσότητες κρασιού οπότε οι ποσότητες του υγρού στα ποτήρια είναι ίδιες. Αρχικά έχοντας τοποθετημένο τον αισθητήρα πάνω από ένα δείγμα παρατηρείται ότι η τιμή από την `analogRead()` αρχίζει να σταθεροποιείται έπειτα από 50 δευτερόλεπτα όπως αναφέρθηκε προηγουμένως. Αρά θα εισάγουμε αυτήν την καθυστέρηση στην αρχή αλλά και πριν από κάθε μέτρηση προκειμένου να είναι αξιόπιστη.

```
int mq3_analogPin = A0; // connected to the output pin of MQ3
int val;
int tempPin = 1;
unsigned long time;
unsigned long time_in_second;
void setup() {
    Serial.begin(9600); // open serial at 9600 bps
}

void loop()
{
    // give ample warmup time for readings to stabilize
    float arduino_voltage = 5 ;
    int mq3_value = analogRead(mq3_analogPin);
    float voltage = mq3_value * (arduino_voltage / 1023.0);

    //time
    time = millis(); //Since arduino has booted
    time_in_second = time/1000 ;

    //Serial.print("Voltage=");
    Serial.print(voltage);
    Serial.print(" ");

    //Serial.print(" Time(S): ");
    Serial.print(time_in_second);
    Serial.print("\n");

    delay(1000); //slow down output
}
```

Έπειτα θα χρησιμοποιήσουμε ένα κανονικό αλκοτέστ για να μετρήσουμε την συγκέντρωση του αλκοόλ σε κάθε δείγμα. Για κάθε δείγμα παίρνουμε μια μέτρηση και με το MQ-3 και καταλήγουμε στα παρακάτω αποτελέσματα.

Διαφορά Δυναμικού (V)	Συγκέντρωση Αλκοόλ (g/l)
2.92	0.50
3.15	0.75
3.29	1.10



3.50	1.50
3.81	1.90

Τώρα αυτό που θέλουμε είναι μια σχέση που θα συνδέει τις 2 παραπάνω μεταβλητές. Άρα σκοπός μας είναι να βρούμε τα a, b που συνδέουν με τον βέλτιστο τρόπο την σχέση $V(n) = aX(n) + b$ όπου ως V αναπαρίσταται η διαφορά δυναμικού και ως X η συγκέντρωση σε αλκοόλ και n είναι η κάθε μέτρηση. Ένας εύκολος τρόπος να εκτιμήσουμε τα a και b είναι μέσω της εφαρμογής της μεθόδου των ελάχιστων τετραγώνων (Least Squares).

Εφαρμόζοντας least squares όπως φαίνεται παρακάτω :

```
import matplotlib.pyplot as plt
import numpy as np
import math
from scipy.optimize import curve_fit

xdata = np.array([0.5, 0.75, 1.1, 1.5, 1.9])
ydata = np.array([2.92, 3.15, 3.29, 3.5, 3.81])

def func(x, a, b):
    return a*x + b

popt, pcov = curve_fit(func, xdata, ydata)

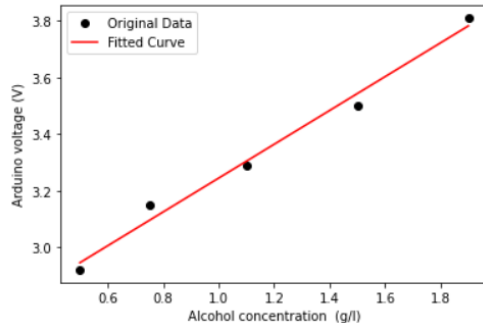
print ("The optimal parameters of the line are:" + str(popt))

plt.figure()
plt.plot(xdata, ydata, 'ko', label="Original Data")
plt.plot(xdata, func(xdata, *popt), 'r-', label="Fitted Curve")
plt.ylabel('Arduino voltage (V)')
plt.xlabel('Alcohol concentration (g/l)')
plt.legend(loc='best')
plt.show()
```

Καταλήγουμε ότι οι παράμετροι της γραμμής $y = a * x + b$ η οποία αναπαριστά με τον βέλτιστο τρόπο την σχέση μεταξύ των τιμών του αισθητήρα με τις πραγματικές τιμές της περιεκτικότητας του αλκοόλ στα δείγματα όπως φαίνεται στην επόμενη εικόνα είναι :



The optimal parameters of the line are:[0.5984252 2.64581102]



$$V(n) = 0.59842 * x(n) + 2.64581$$

Εικόνα 5: Ελάχιστα Τετράγωνα

3.2 Προγραμματισμός του Συστήματος

Το κανονικό αλκοτέστ μετράει πρακτικά το [BAC](#) και η παραπάνω σχέση αναπαριστά την σχέση της εξόδου του αισθητήρα με το BAC. Σύμφωνα με το BAC οι τιμές αναφοράς είναι οι >0.2 και το calibration του αισθητήρα έχει γίνει με βάση αυτά τα δεδομένα οπότε αγνοούμε τις τιμές εκτός του πεδίου ορισμού.

Επίσης παρατηρείται ότι κατά την διεξαγωγή του πειράματος παρατηρήθηκε μια συνέχεια στα αποτελέσματα αφού φέρναμε κοντά στον αισθητήρα αλκοόλη η οποία συνεχιζόταν και αφού την απομακρύνουμε. Γι' αυτόν τον λόγο λαμβάνονται μετρήσεις που το πλήθος τους ορίζεται στην μεταβλητή READ_SAMPLE_TIMES ανά διάστημα που ορίζεται στην μεταβλητή READ_SAMPLE_INTERVAL και υπολογίζεται η μέση τιμή όπου αυτή λαμβάνεται υπόψιν σαν τιμή εξόδου του αισθητήρα.

Έπειτα η επικοινωνία του Arduino με την node-red πλατφόρμα γίνεται μέσω σειριακής θύρας. Μέσω αυτής αποστέλλεται ένα json ως string το οποίο έχει 3 key/value pairs. Τα keys είναι τα εξής:

- 1) Serial_no μέσω του serial number μπορούμε να ξεχωρίσουμε τις συσκευές που είναι τοποθετημένες στα αυτοκίνητα της αυτοκινητοβιομηχανίας.
- 2) Driving ability Η ικανότητα της οδήγησης είναι μια παράμετρος που εξάγεται στην στιγμή της μέτρησης προκειμένου να μπορεί να αξιοποιηθεί και τοπικά στο αλκοτέστ όπως συμβαίνει με όλα τα αλκοτέστ της αγοράς.
- 3) Measurement που είναι το αποτέλεσμα της μέτρησης όπου η ακρίβεια εξαρτάται από την ευαισθησία του αισθητήρα και από περιβαλλοντικούς παράγοντες όπως παραπάνω.
- 4) Η παράμετρος που ορίζει την ικανότητα οδήγησης έχει οριστεί στα 2.78 με βάση την παραπάνω σχέση καθώς η νομοθεσία στην χώρα μας ορίζει τα 0.24 χιλιοστά του γραμμαρίου ανά λίτρο εμπνευσμένου αέρα ως τιμή αναφοράς η οποία ισοδυναμεί με 2,78 volt.

Με βάση τις παραπάνω παρατηρήσεις ο κώδικας του Arduino είναι ο παρακάτω:

```
#define Serial_No "1025af"
```



M146 – Σύγχρονες Επικοινωνίες Μικρής και Μεγάλης Εμβέλειας

Τίτλος Μελέτης: Κατασκευή Συσκευής Αλκοτέστ

Ευάγγελος Σιατήρας

```
#define          READ_SAMPLE_INTERVAL          (100)    //define how many samples you are going to
take

#define          READ_SAMPLE_TIMES            (10)      //define the time interval(in milisecond)
between each samples

int mq3_analogPin = A0; // connected to the output pin of MQ3

float get_BAC_from_Voltage(float v){
    float a = 0.59842 ;
    float b = 2.64581 ;
    float BAC = (v-a)/b ;
    return (BAC > 0 ? BAC : 0);
}

void create_json(float BAC, int driving_ability){
    Serial.print("{\"serial_no\":\"");
    Serial.print(Serial_No);
    Serial.print("\",");
    Serial.print("\"driving_ability\":");
    Serial.print(driving_ability);
    Serial.print(",");
    Serial.print("\"measurement\":");
    Serial.print(BAC);
    Serial.println("}");
}

void setup() {
    pinMode(8, INPUT);
    pinMode(12, OUTPUT);
    Serial.begin(9600); // open serial at 9600 bps
}

void loop() {
    // Serial.println("Breath in");
    digitalWrite(12, HIGH); // turn the LED on (HIGH is the voltage level)
    delay(2000);           // wait for a second
    digitalWrite(12, LOW);  // turn the LED off by making the voltage LOW

    float mq3_value=0;

    for (int i=0;i<READ_SAMPLE_TIMES;i++) {
        mq3_value += analogRead(mq3_analogPin);
        delay(READ_SAMPLE_INTERVAL);
    }
    mq3_value = mq3_value/READ_SAMPLE_TIMES;
```



```
float voltage = mq3_value * (5.0 / 1023.0); // get mq3 voltage
float response = get_BAC_from_Voltage(voltage);

if(voltage<2.78){
    create_json (response,1);
    delay(5000);
}else if(voltage>=2.78){
    float response = get_BAC_from_Voltage(voltage);
    create_json (response,0);
    delay(5000);
}
}
```

Επιπλέον 2 παραδείγματα του output είναι τα εξής:

1) {"serial_no":"1025af","driving_ability":1,"measurement":0.00}

2) {"serial_no":"1025af","driving_ability":0,"measurement":0.97}

Στην 2^ο παράδειγμα παρατηρούμε ότι το value για το key driving_ability είναι μηδέν διότι η τιμή 0.97 είναι μεγαλύτερη από το μέγιστο όριο που είναι το 0.24 όπως αναφέρθηκε παραπάνω.

3.3 Διαδικασία Μέτρησης

Αφού συνδέσουμε με τροφοδοσία την συσκευή δηλαδή τον μικροϋπολογιστή arduino ανάβει το μπλε led που μας προειδοποιεί ότι πρέπει να φυσήξουμε στα επόμενα 2 δευτερόλεπτα. Αφού φυσήξουμε τότε ο αισθητήρας αλκοόλης μετράει το οινόπνευμα στην αναπνοή μας κατασκευάζει το json θέτοντας το αποτέλεσμα της μέτρησης και με βάση αυτό την ικανότητα οδήγησης του οδηγού το οποίο και θα αποσταλεί στα ανωτέρα layers.

3.4 Υλοποίηση του IOT Μέρους Μέσω της Node-Red Πλατφόρμας

Το IOT μέρος της εργασίας υλοποιείται μέσω της πλατφόρμας node-red και περιλαμβάνει 2 flows. Ένα από τα μεγάλα πλεονεκτήματα που προσφέρει το Node-RED στον developer είναι η ευκολία με την οποία μπορεί να χτίσει μία απλή διεπαφή χρήστη. Στις περισσότερες εφαρμογές IoT αυτό που μας ενδιαφέρει είναι η αποτελεσματικότητα και το χαμηλό ενεργειακό κόστος. Έτσι οι απλές εύκολα αναγνώσιμες διεπαφές είναι προτιμότερες από τις βαριές και πιο σύνθετες. Πριν παρουσιάσουμε τον τρόπο χτισίματος της διεπαφής θα πρέπει να μιλήσουμε για τα ιεραρχικά επίπεδα της. Το υψηλότερο επίπεδο της διεπαφής είναι το Tab. Με το που δημιουργήσουμε ένα Tab ουσιαστικά κατοχυρώνουμε ένα παράθυρο, όπου μπορούμε να χτίσουμε την διεπαφή. Τα Tabs που έχουμε δημιουργήσει είναι τα: *Arduino_Board* και *Central Server*.



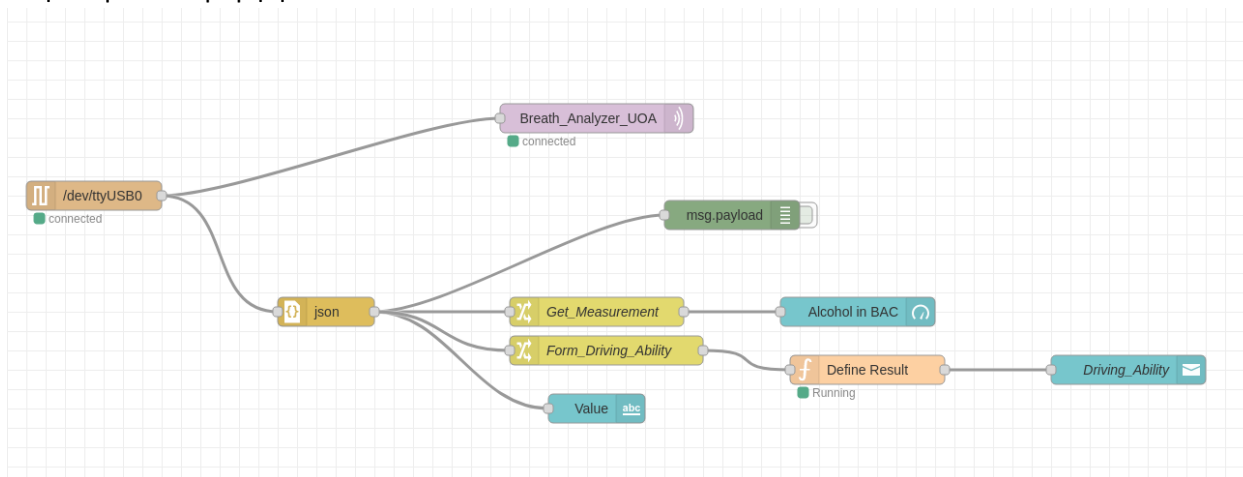
3.4.1 Απαιτήσεις Node-Red

Τα nodes που έχουν εγκατασταθεί και χρησιμοποιούνται στα πλαίσια της εργασίας είναι:

- node-red-contrib-email-out
- node-red-contrib-python3-function
- node-red-dashboard
- node-red-node-mysql
- node-red-node-serialport

3.4.2 Node-Red Arduino Board Flow

Το πρώτο flow επεκτείνει τις λειτουργίες του Arduino board. Το συγκεκριμένο σενάριο είναι εφικτό καθώς το node-red είναι μια πλατφόρμα με χαμηλές απαιτήσεις όπου μπορεί να εγκατασταθεί και να αξιοποιηθεί αποδοτικά σε αρκετά boards που τρέχουν linux όπως το raspberry. Το OS του Arduino δεν το υποστηρίζει οπότε στα πλαίσια της εργασίας χρησιμοποιείται ένα Linux VM με ubuntu 18.04 όπου είναι συνδεδεμένο το Arduino μέσω της σειριακής θύρας (πρακτικά USB). Το Arduino μέσω αυτής της αρχιτεκτονικής μπορεί να αποστείλει τις μετρήσεις και την ικανότητα του οδηγού να οδηγήσει σε οπουδήποτε κεντρικό server συλλογής και αξιοποίησης δεδομένων. Επιπλέον μπορεί να δείξει στον χρήστη με ένα γράφημα κατά πόσο έχει ξεπεράσει το όριο και πόσο κοντά είναι στο ανώτερο επίπεδο κατανάλωσης αλκοόλ (το 5) όπως περιγράφεται παραπάνω. Αναλυτικότερα το Arduino board flow έχει την παρακάτω μορφή:



Εικόνα 6: Arduino Board Flow

Τα δεδομένα τροφοδοτούνται μέσω του serial input που έχει παραμετροποιηθεί να κοιτάει στην συγκεκριμένη σειριακή θύρα, με το συγκεκριμένο baud rate (9600) που στέλνει το Arduino όπως φαίνεται παρακάτω:



Serial Port: /dev/ttyUSB0

Settings: Baud Rate (9600), Data Bits (8), Parity (None), Stop Bits (1), DTR (auto), RTS (auto), CTS (auto), DSR (auto)

Εικόνα 7: Serial In Node

Έπειτα για την αποστολή των δεδομένων στον central server χρησιμοποιείται ένας public mqtt broker ο [hivemq](https://hivemq.com). Τα δεδομένα αποστέλλονται κάνοντας publish στο topic «Breath_Analyzer_Uoa» με QoS = 2 για μέγιστο reliability όπου ο αποστολέας και ο παραλήπτης συμμετέχουν σε ένα handshake δύο επιπέδων ώστε να εξασφαλίσουν ότι ένα και μόνο ένα αντίγραφο λήφθηκε (assured delivery).

Server: broker.hivemq.com:1883

Topic: Breath_Analyzer_UOA

QoS: 2, Retain: [checked]

Name:

Εικόνα 8: MQTT Out Node

Έπειτα τα δεδομένα που λαμβάνονται είναι σε μορφή string. Για την αξιοποίησή τους, μετατρέπονται σε JSON. Όπως αναφέρθηκε προηγουμένως το JSON 'κουβαλάει' την μέτρηση και την ικανότητα του οδηγού να οδηγήσει. Η μέτρηση αφού εξαχθεί από το JSON μέσω ενός change node

Set: msg. payload

to: msg. payload.measurement

Εικόνα 9: Get Measurement Node

Τροφοδοτείται σε ένα ui_gauge node το οποίο έχει παραμετροποιηθεί καταλληλά ώστε να απεικονίζει την μέτρηση και χρωματικά με κριτήριο το όριο.



Εικόνα 10: Alcohol in BAC Node

Όμοια με το measurement key χρησιμοποιούμε ένα change node προκειμένου να εξάγουμε το driving_ability key.



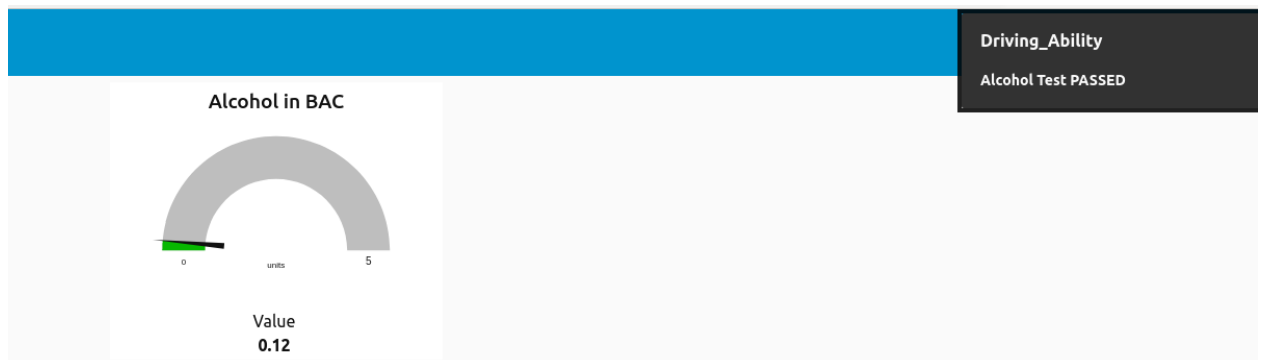
Εικόνα 11: Form Driving Ability Node

Έπειτα χρησιμοποιούμε ένα rpython3-function node προκειμένου να οπτικοποιήσουμε το driving ability value όπως φαίνεται παρακάτω.

```
1  
2 if msg['payload'] == 0:  
3     msg['payload'] = "Alcohol Test FAILED"  
4 else :  
5     msg['payload'] = "Alcohol Test PASSED"  
6  
7 return msg
```

Εικόνα 12: Define Result Node

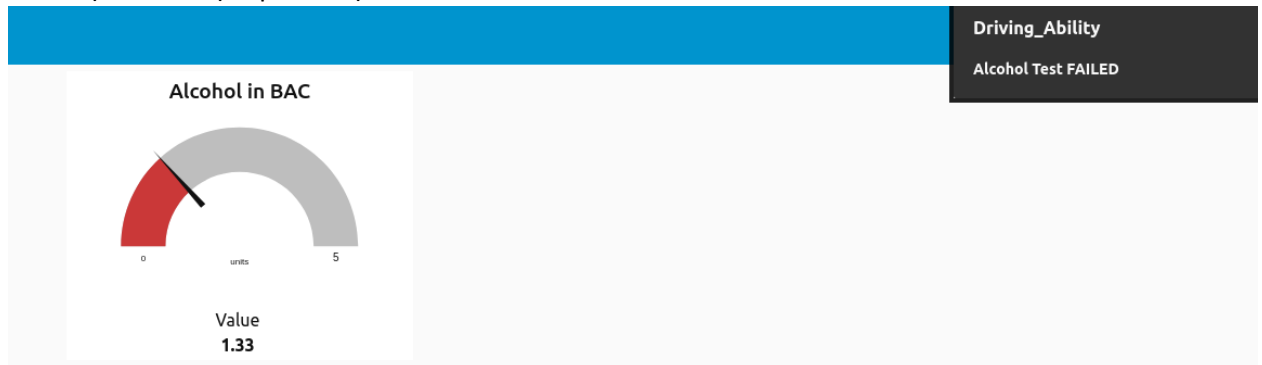
Και έπειτα χρησιμοποιούμε ένα ui_toast node προκειμένου παράλληλα με το gauge να γίνεται trigger και ένα popup notification panel όπου θα τυπώνεται το κατάλληλο μήνυμα. Στην περίπτωση μιας μέτρησης κάτω από το όριο το dashboard θα είναι όπως στην παρακάτω εικόνα



Εικόνα 13: Dashboard Result with Passed Alcohol Test



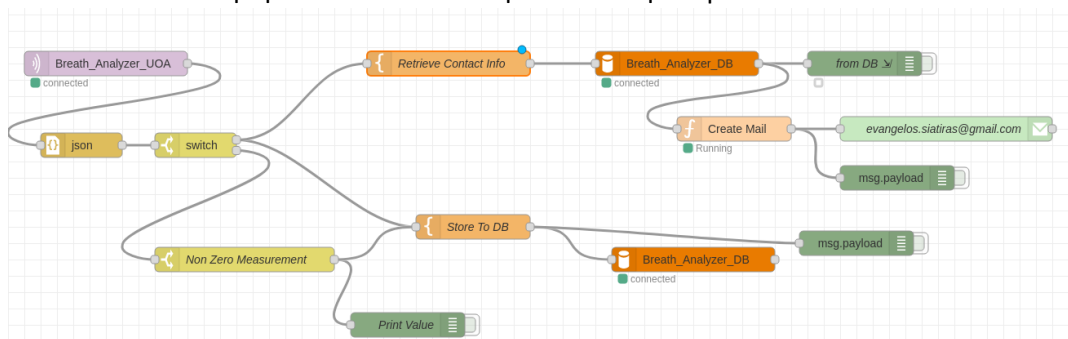
Και στην αντίθετη περίπτωση



Εικόνα 14: Dashboard Result with Failed Alcohol Test

3.4.3 Node-Red Central Server Flow

Το central server flow στα πλαίσια της εργασίας είναι υλοποιημένο στο ίδιο VM και στο ίδιο node-red instance αλλά είναι τελείως ανεξάρτητο με το Arduino board flow. Πρακτικά μπορεί να εγκατασταθεί οπουδήποτε και να μπορεί να συλλεγεί δεδομένα από μεγάλο αριθμό από “clients”. Η αναφορά σε clients γίνεται γιατί ένας central server πρέπει να είναι scalable και efficient. Η υλοποίηση του central server φαίνεται στην παρακάτω εικόνα.



Εικόνα 15: Central Server Flow

Αρχικά κάνει subscribe στο «Breath_Analyzer_UOA» topic και δέχεται δεδομένα στο συγκεκριμένο JSON format όπως παραπάνω.



The screenshot shows the configuration for an MQTT In Node. It includes fields for: Server (broker.hivemq.com:1883), Topic (Breath_Analyzer_UOA), QoS (2), Output (auto-detect (string or buffer)), and Name (Name).

Εικόνα 16: MQTT In Node

Έπειτα αφού μετατραπεί το input string σε input JSON (μέσω ενός json Node) έπειτα διαχωρίζεται ανάλογα με το value του driving ability.

The screenshot shows the configuration for a Switch Node. The Name field is 'Name'. The Property is 'msg. payload.driving_ability'. There are two switch conditions: the first is '== a_z 0' leading to output 1, and the second is '== a_z 1' leading to output 2.

Εικόνα 17: Switch Node

Στην δεύτερη έξοδο του switch node δρομολογούνται τα μηνύματα που παράγονται στο σενάριο όπου το αλκοτέστ είναι εντός ορίου. Έπειτα μέσω ενός άλλου switch node κρατάμε τα μηνύματα στα οποία η μέτρηση είναι θετική.

The screenshot shows the configuration for a Non Zero Measurement Node. The Name field is 'Non Zero Measurement'. The Property is 'msg. payload.measurement'. There is one condition: '> a_z 0' leading to output 1.

Εικόνα 18: Non Zero Measurement Node



Τα μηνύματα αυτά αποθηκεύονται στην βάση δεδομένων.

Από την πρώτη έξοδο του παραπάνω switch node δρομολογούνται τα μηνύματα τα οποία αφορούν μετρήσεις εκτός ορίων. Τα μηνύματα αυτά αποθηκεύονται στην βάση δεδομένων και έπειτα στέλνεται ένα ενημερωτικό mail στον προκαθορισμένο υπεύθυνο να οδηγήσει το όχημα αντί του μεθυσμένου οδηγού.

3.4.3.1 Βάση Δεδομένων

Αναλυτικότερα η βάση δεδομένων που χρησιμοποιείται στα πλαίσια της εργασίας είναι η

10.1.44-MariaDB-Ubuntu0.18.04.1

Και μπορεί να εγκατασταθεί μέσω της εντολής

```
sudo apt install mariadb-server
```

Στην βάση δεδομένων αποθηκεύεται όλη η πληροφορία που φέρνει το json (serial_no,driving_ability,measurements) όταν η μέτρηση είναι θετική. Σε αυτήν προστίθεται και το current timestamp προκειμένου να σώζεται η ώρα και η ημερομηνία που έγινε το alcotest. Το sql query που χρησιμοποιείται το οποίο τροφοδοτείται στην βάση μέσω του template node Store to DB

Είναι:

```
insert into alcohol_stats (serial_no,state,measurement,timestamp)
values
('{{payload.serial_no}}','{{payload.driving_ability}}','{{payload.measurements}}',CURRENT_TIMESTAMP);
```

Τα παραπάνω δεδομένα αποθηκεύονται στον πίνακα “alcohol_stats” όπως παρακάτω.

```
MariaDB [Breath_Analyzer_DB]> select * from alcohol_stats limit 1;
```

```
+-----+-----+-----+-----+-----+
| id | serial_no | state | measurement | timestamp |
+-----+-----+-----+-----+-----+
| 2 | 1025af | 0 | 0.75 | 2020-07-11 03:15:10 |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

Να σημειώσουμε ότι η στήλη «id» αυξάνεται αυτόματα ενώ εισάγονται γραμμές στην βάση (Auto Increment).

Έπειτα χρησιμοποιείται και άλλο ένα table με όνομα “account_details”. Εδώ κατά την αγορά του αυτοκινήτου η κατά την εγκατάσταση του αλκοτέστ στο όχημα πρέπει η



αυτοκινητοβιομηχανία να κρατάει σε ένα αρχείο τα στοιχεία επικοινωνίας του οδηγού για τον οποίο το αλκοτέστ με το συγκεκριμένο serial number θα λαμβάνει μετρήσεις καθώς και τα στοιχεία επικοινωνίας του αντικαταστατή (που θα δηλώνει ο ίδιος ο οδηγός) στην περίπτωση που το τεστ βγει εκτός ορίων.

Μια εγγραφή στο table φαίνεται όπως παρακάτω:

```
MariaDB [Breath_Analyzer_DB]> select * from account_details;
```

Serial_No	Driver_Name	Driver_Contact_Info	Secondary_Driver_Name	Secondary_Driver_Contact_Info
1025af	Nick Pappas	6983555874	George Pappas	vangelis_12@hotmail.com

1 row in set (0.00 sec)

Με την πληροφορία που αποθηκεύουμε στην βάση δίνουμε την δυνατότητα της εξαγωγής στατιστικών σε βάθος χρόνου είτε συγκεντρωτικών είτε ατομικών για κάθε οδηγό καθώς μπορούμε να αναλύσουμε τις μετρήσεις από κάθε φορά που μπαίνει στο όχημα του και να «χτίσουμε» το προφίλ του με σκοπό τις κατάλληλες επεμβάσεις για την ασφάλεια του, έπειτα από την αξιολόγηση των αποτελεσμάτων.

λεπτομέρειες για τα creation_scripts και τα data βρίσκονται στο sqldump αρχείο Breath_Analyzer_DB.sql

Επιστρέφοντας στο Central Server flow , σε περίπτωση που μια μέτρηση είναι εκτός ορίων με βάση το serial number της συσκευής από την οποία έγινε η μέτρηση κατασκευάζουμε το query στο Retrieve Contact Info template node με τα οποία ζητάμε από την βάση τα contact details του οδηγού αντικαταστατή για τον οποίο η μέτρηση είναι εκτός ορίων. Το query είναι:

```
select * from account_details where Serial_No = '{{payload.serial_no}}';
```

Έτσι για το συγκεκριμένο serial number επιστρέφονται όλες οι λεπτομέρειες όπως απεικονίζονται παραπάνω.

Η πληροφορία αυτή τροφοδοτείται σε ένα python3-function Node όπου γίνεται ένα data manipulation και κατασκευάζεται το θέμα του mail και το κείμενο στο body όπως παρακάτω.

```
import json
```

```
payload = msg['payload'][0]
```



```
mail = "Dear "+str(payload.get("Secondary_Driver_Name"))+",\nKindly get  
in touch with "+str (payload.get("Driver_Name"))+" through "+ str  
(payload.get("Driver_Contact_Info"))+" as he/she is not able to drive!  
\n"
```

```
msg['topic'] = "Critical Alert for "+ str(payload.get('Driver_Name'))+  
"!!!"
```

```
msg['to'] = str(payload.get('Secondary_Driver_Contact_Info'))
```

```
msg['body'] = str(mail)
```

```
return msg
```

Το αποτέλεσμα φαίνεται στην παρακάτω εικόνα:

Critical Alert for Nick Pappas!!!



12 July 2020 23:45

evangelos.siatiras@gmail.com

Details

Dear George Pappas,
Kindly get in touch with Nick Pappas through
[6983555874](#) as he/she is not
able to drive!

Εικόνα 19: Resulting Mail

3.5 Παρατηρήσεις

Στην συγκεκριμένη υλοποίηση η ανάκτηση και η αποθήκευση των δεδομένων καθώς και η αποστολή των μηνυμάτων γίνεται πλήρως δυναμικά μέσω των δεδομένων από τη βάση. Αυτό καθιστά την υλοποίηση scalable αφού μπορεί να λαμβάνει μετρήσεις από πολλούς αισθητήρες ταυτόχρονα, να τις αποθηκεύει στην βάση και ταυτόχρονα να έχει όλες τις πληροφορίες που χρειάζεται προκειμένου να ειδοποιήσει μέσω του ενημερωτικού email ταχύτατα και αποδοτικά καθώς δεν χρειάζεται να κρατάει δεδομένα τοπικά.



4 Συσχέτιση Τρόπου Λειτουργίας με την Πραγματική Χρήση

Η συσχέτιση αυτής της κατασκευής με την εφαρμογή σε μηχανοκίνητο όχημα γίνεται ως εξής. Με την είσοδο του κλειδιού και έπειτα την ενεργοποίηση του διακόπτη όπου περιμένουμε κάποια δευτερόλεπτα (θέση προθέρμανσης στους diesel) αυτόματα τροφοδοτείται με ρεύμα η συσκευή. Το μπλε λαμπάκι led υποδηλώνει τους ενδεικτικούς φωτισμούς στο καντράν του οχήματος. Το dashboard όπου περιλαμβάνει το gauge plot και το popup notification θα εμφανίζεται σε μια από τις οθόνες ιδανικά σε αυτήν που είναι στο καντράν του οχήματος. Από εκεί θα ενημερώνεται για το αποτέλεσμα του αλκοτέστ και εφόσον βγει αρνητικό θα γνωρίζει ότι έχει ειδοποιηθεί ένας δεύτερος προκαθορισμένος από τον ίδιο, οδηγός για να έρθει να παραλάβει το όχημα του.



5 Εξέλιξη

Η εξέλιξη που μπορεί να πάρει ο συγκεκριμένος αισθητήρας (κατασκευή) ώστε με την χρήση του σε όχημα να εξυπηρετεί στο ακέραιο τον σκοπό δημιουργίας του είναι αρκετά πολύπλοκη. Ο αισθητήρας αλκοόλης να είναι μεγαλύτερης ακριβείας και χρόνου απόκρισης, όπως δηλαδή των διαπιστευμένων εταιριών κυκλοφορίας του και τον ελεγκτικών μηχανισμών. Έπειτα δεν συνίσταται η χρήση του Arduino για μια τέτοια συσκευή καθώς έχει πολύ μειωμένη επεξεργαστική ισχύ και πολύ λίγη RAM σε συνδυασμό με το ότι δεν υποστηρίζει linux. Αυτό έχει σαν αποτέλεσμα οι δυνατότητες του όσον αφορά το “scalability” είναι πολύ περιορισμένες. Μια αντίστοιχη συσκευή που ενδείκνυται είναι το raspberry το οποίο είναι μια πολύ πιο δυνατή από άποψη επιδόσεων συσκευή ικανή να τρέχει το node-red τοπικά όπως και μια βάση δεδομένων προκειμένου να κρατάμε τις μετρήσεις και σε επίπεδο εφαρμογής. Τέλος επιτρέπει τον προγραμματισμό του σε οποιαδήποτε γλώσσα προγραμματισμού και μέσω του linux μπορούμε να έχουμε διαφορετικά processes να τρέχουν παράλληλα που είναι το πλέον αποδοτικό για την κατασκευή οποιασδήποτε εφαρμογής.

Τέλος είναι σημαντικό να αναφέρουμε ότι η προσφερόμενη βοήθεια των αισθητήρων στον οδηγό και στον άνθρωπο γενικότερα είναι επιθυμητή μέχρι το σημείο όπου δεν υπονομεύει ή επηρεάζει την ελευθερία του αλλά και την γνώση για σωστή χρήση του αυτοκινήτου.



6 Βιβλιογραφία

- [1] L. HANWEI ELETRONICS CO., «Technical Data MQ-3 Gas Sensor,» 2 8 2010. [Ηλεκτρονικό]. Available: https://components101.com/sites/default/files/component_datasheet/MQ-3%20Sensor%20Datasheet%20In%20Detail.pdf?fbclid=IwAR0ac_uEOiimY8Qnu7ivtDuEKTE90Ep2HzBOUDkHK3YQUzpL7qe4Z48H7mg.
- [2] «Blood Alcohol Content,» [Ηλεκτρονικό]. Available: https://en.wikipedia.org/wiki/Blood_alcohol_content.
- [3] Y. Ibrahim, «Alcohol Detection Sensor,» [Ηλεκτρονικό]. Available: https://create.arduino.cc/projecthub/yashikibrahim/alcohol-detection-sensor-dc32cd?fbclid=IwAR3kTsZIO_wV89h7qw1FSQDkFuiYELmPaRe0QCvgexnRIAlicoPM58-WM1Y.
- [4] Σ. Σ. Δημητρίου, «Κατασκευή Συστήματος Καλλιέργειας Ακριβείας,» 08 2019. [Ηλεκτρονικό]. Available: https://nemertes.lis.upatras.gr/jspui/bitstream/10889/13439/1/%ce%94%ce%b9%cf%80%ce%bb%cf%89%ce%bc%cf%83%cf%84%ce%b9%ce%ba%ce%b7_%cf%84%ce%b5%ce%bb.pdf.
- [5] «ΠΕΡΙΓΡΑΦΗ ΛΕΙΤΟΥΡΓΙΑΣ ΣΥΓΧΡΟΝΩΝ ΑΙΣΘΗΤΗΡΩΝ ΑΥΤΟΚΙΝΗΤΩΝ ΚΑΙ ΣΧΕΔΙΑΣΗ ΚΑΙ ΚΑΤΑΣΚΕΥΗ ΑΙΣΘΗΤΗΡΑ ΕΛΕΓΧΟΥ ΝΗΦΑΛΙΟΤΗΤΑΣ ΟΔΗΓΟΥ,» 2015. [Ηλεκτρονικό]. Available: <http://repository.library.teiwest.gr/xmlui/bitstream/handle/123456789/3915/%ce%a0%ce%95%ce%a1%ce%99%ce%93%ce%a1%ce%91%ce%a6%ce%97%20%ce%9b%ce%95%ce%99%ce%a4%ce%9f%ce%a5%ce%a1%ce%93%ce%99%ce%91%ce%a3%20%ce%a3%ce%a5%ce%93%ce%a7%ce%a1%ce%9f%ce%9d%ce%a9%ce%9d>.
- [6] D. GOMBA, «Arduino Breathalyzer: Calibrating the MQ-3 Alcohol Sensor,» 23 09 2010. [Ηλεκτρονικό]. Available: https://blog.arduino.cc/2010/09/23/arduino-breathalyzer-calibrating-the-mq-3-alcohol-sensor/?fbclid=IwAR3kTsZIO_wV89h7qw1FSQDkFuiYELmPaRe0QCvgexnRIAlicoPM58-WM1Y.