

Σχεδιασμός και ανάπτυξη συστήματος αυτοματισμού για μελισσοκομικές εφαρμογές



ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Πρόγραμμα Μεταπτυχιακών Σπουδών

Επιστήμη και Τεχνολογία της Πληροφορικής και των Υπολογιστών

Ειδίκευση Υλικού και Υπολογιστικών Συστημάτων

Διπλωματική Εργασία

**Σχεδιασμός και ανάπτυξη συστήματος αυτοματισμού για
μελισσοκομικές εφαρμογές**



Αντώνιος Μπέττας

A.M. 19055

Επιβλέπων:

Διονύσης Κανδρής,

Καθηγητής

Αιγάλεω, Ιούνιος 2021



UNIVERSITY OF WEST ATTICA
FACULTY OF ENGINEERING
DEPARTMENT OF INFORMATICS & COMPUTER ENGINEERING
POSTGRADUATE COURSE: SCIENCE & TECHNOLOGY OF
INFORMATICS AND COMPUTERS

Diploma Thesis

**Design and Development of an Automation System for Beekeeping
Applications**



Antonios Bettas
Registration Number: 19055

Supervisor:
Dionisis Kandris
Professor

Athens-Egaleo, June 2021

Σχεδιασμός και ανάπτυξη συστήματος αυτοματισμού για μελισσοκομικές εφαρμογές

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Σχεδιασμός και ανάπτυξη συστήματος αυτοματισμού για
μελισσοκομικές εφαρμογές

Αντώνιος Μπέττας

A.M. 19055


Εισηγητής:

Διονύσης Κανδρής

Εξεταστική Επιτροπή:

Ημερομηνία εξέτασης: 29 Ιουνίου 2021

Η Διπλωματική Εργασία έγινε αποδεκτή και βαθμολογήθηκε από την εξής τριμελή επιτροπή:

Διονύσης Κανδρής Καθηγητής	Αντώνης Μπόγρης Καθηγητής	Σταύρος Φατούρος Αναπληρωτής Καθηγητής
		

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΜΕΤΑΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

Ο κάτωθι υπογεγραμμένος Αντώνιος Μπέττας του Βασιλείου, με αριθμό μητρώου 19055 φοιτητής του Προγράμματος Μεταπτυχιακών Σπουδών Επιστήμη και Τεχνολογία της Πληροφορικής και των Υπολογιστών του Τμήματος Μηχανικών Πληροφορικής και Υπολογιστών της Σχολής Μηχανικών του Πανεπιστημίου Δυτικής Αττικής, δηλώνω ότι:

«Είμαι συγγραφέας αυτής της μεταπτυχιακής εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της, είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, οι όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε ακριβώς είτε παραφρασμένες, αναφέρονται στο σύνολό τους, με πλήρη αναφορά στους συγγραφείς, τον εκδοτικό οίκο ή το περιοδικό, συμπεριλαμβανομένων και των πηγών που ενδεχομένως χρησιμοποιήθηκαν από το διαδίκτυο. Επίσης, βεβαιώνω ότι αυτή η εργασία έχει συγγραφεί από μένα αποκλειστικά και αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο δικής μου, όσο και του Ιδρύματος.

Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του πτυχίου μου».

Ο Δηλών



Αντώνιος Μπέττας

Ευχαριστίες

Η παρούσα διπλωματική εργασία πραγματοποιήθηκε με την συνεχή υποστήριξη ενός μεγάλου συνόλου σημαντικών ανθρώπων στους οποίους θα ήθελα να εκφράσω τις ευχαριστίες μου.

Αρχικά θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου, Διονύσιο Κανδρή, για την συνεχή, άμεση και πολύτιμη βοήθεια του στην υλοποίηση της διπλωματικής εργασίας.

Στην συνέχεια θα ήθελα να ευχαριστήσω τους φίλους μου Στέλιο Αθανασόπουλο και Παντελεήμων Γιαννουδάκο για την κατασκευή ενός αντικειμένου μέσω του 3D εκτυπωτή τους, καθώς και τον Ανδρέα Ιορδανίδη για την συγκόλληση ακροδεκτών στις πλακέτες που χρησιμοποιήθηκαν στην εργασία.

Εν κατακλείδι θα ήθελα να ευχαριστήσω την οικογένειά μου που είναι συνοδοιπόροι και συμπαραστάτες σε όλες τις επιλογές της ζωής μου και την κοπέλα μου που ήταν στο πλευρό μου καθ' όλη τη διάρκεια των μεταπτυχιακών σπουδών μου.

Περίληψη

Η παρούσα διπλωματική εργασία έχει ως αντικείμενο τη σχεδίαση και ανάπτυξη ενός συστήματος αυτοματισμού για την εποπτεία και τον έλεγχο των συνθηκών που επικρατούν σε μελισσοκομικές μονάδες. Το συγκεκριμένο σύστημα καταγράφει τις συνθήκες που επικρατούν σε ένα μελίσσι μέσω κατάλληλων αισθητήρων, και μέσω κεντρικού ελεγκτή επεξεργάζεται τα αντίστοιχα δεδομένα και παράγει κατάλληλα σήματα ελέγχου. Επιπρόσθετα μέσω κατάλληλης εφαρμογής που αναπτύχθηκε καθίσταται δυνατός ο απομακρυσμένος έλεγχος λειτουργίας από συσκευή ευφυούς κινητού τηλεφώνου.

Abstract

The subject of this dissertation is the design and development of an automation system for the supervision and control of the conditions prevailing in beekeeping units. The specific system records the conditions prevailing in a beehive through appropriate sensors, and through a central controller processes the corresponding data and produces appropriate control signals. In addition, through a suitable application developed, it is possible to remotely control the operation of an intelligent mobile phone device.

Πίνακας Περιεχομένων

Ευχαριστίες	iii
Περίληψη	iv
Abstract	iv
Πίνακας Περιεχομένων	v
Κεφάλαιο 1 : Εισαγωγή στη μελισσοκομία	1
1.1 Εισαγωγή	1
1.1 Σπουδαιότητα των μελισσών από την αρχαιότητα.....	1
1.2 Γενικά στοιχεία μελισσών	3
1.3 Η κοινωνική ζωή της αποικίας.....	8
1.4 Η σημασία των μελισσών και των προϊόντων τους για τον άνθρωπο	13
1.5 Κίνδυνοι μελισσοκομίας	19
1.6 Η μελισσοκομία στην Ελλάδα και την Ευρωπαϊκή Ένωση	23
Κεφάλαιο 2 : Ο μικροελεγκτής Arduino	27
2.1 Εισαγωγή	27
2.2 Ιστορική αναδρομή Arduino	27
2.3 Η πλακέτα Arduino	28
2.4 Γιατί να επιλεγεί το Arduino	28
2.5 Μοντέλα Arduino	29
2.6 Arduino Shields.....	33
2.7 Arduino Uno γενικά χαρακτηριστικά.....	37
2.7.1 Ακροδέκτες.....	38
2.7.2 Τροφοδοσία	41
2.7.3 Μικροελεγκτής (microcontroller).....	42
2.7.4 Μνήμη	43
2.8 Λογισμικό Arduino.....	43
2.8.1 Arduino IDE.....	43
2.8.2 Arduino Web Editor.....	45
2.8.3 MiniBlox	49
Κεφάλαιο 3: Σύστημα αυτοματοποιημένου ελέγχου μελισσιών	53
3.1 Εισαγωγή	53
3.2 Σύνοψη συστήματος	53
3.3 Ανάλυση της δομής και της λειτουργίας του συστήματος	55
3.3.1 Arduino Uno.....	56
3.3.2 Arduino Nano.....	56
3.3.2.1 Συνδεσμολογία Arduino Uno & Nano	56

3.3.3 Lcd 1602 & I2C Serial Interface	57
3.3.3.1 Συνδεσμολογία Arduino Uno & LCD (with I2C Serial Interface)	58
3.3.4 Αισθητήρες βάρους & HX711	59
3.3.4.1 Συνδεσμολογία Αισθητήρων βάρους και HX711	62
3.3.4.2 Κώδικας της διαδικασίας ζύγισης	65
3.3.4.3 Επεξήγηση κώδικα ζύγισης	69
3.3.5 SIM900 GPRS/GSM	70
3.3.5.1 Συνδεσμολογία SIM900 & FTDI FT232RL USB	71
3.3.5.2 Συνδεσμολογία SIM900 & Arduino Uno	74
3.3.6 GPS Module	75
3.3.6.2 Συνδεσμολογία GPS Module & Arduino Uno	76
3.3.7 IR Wireless Remote Control	77
3.3.7.1 Συνδεσμολογία IR Wireless Remote Control	78
3.3.7.2 Κώδικας IR Wireless Remote Control	78
3.3.8 Αισθητήρας δόνησης (vibrator sensor)	80
3.3.8.1 Συνδεσμολογία αισθητήρα δόνησης	81
3.3.9 Μαγνητικός αισθητήρας συναγερμού	81
3.3.9.1 Συνδεσμολογία μαγνητικού αισθητήρα συναγερμού	82
3.3.10 Αισθητήρας θερμοκρασίας και υγρασίας (DHT22 module)	83
3.3.10.1 Συνδεσμολογία DHT22 module	84
3.3.11 RGB Led	85
3.3.11.1 Συνδεσμολογία RGB Led	85
3.4. Κώδικας προγραμματισμού συστήματος	86
3.4.1 Κώδικας Slave (Nano)	86
3.4.1.1 Διευκρινήσεις κώδικα Slave (Nano)	90
3.4.2 Κώδικας Master (UNO)	91
3.4.2.2 Διευκρινήσεις κώδικα Master (Uno)	108
3.5.1 Android Εφαρμογή	109
3.5.2 MIT App Inventor & Κώδικας	110
Κεφάλαιο 4: Επίλογος	115
4.1 Εισαγωγή	115
4.2 Σύνοψη εργασίας	115
4.3 Προβλήματα και λύσεις	115
4.4 Συμπεράσματα	116
4.5 Προτάσεις για μελλοντική εξέλιξη	117
BIBΛΙΟΓΡΑΦΙΑ	119

Κεφάλαιο 1 : Εισαγωγή στη μελισσοκομία

1.1 Εισαγωγή

Το παρόν κεφάλαιο παρέχει εισαγωγικές έννοιες στον τομέα της μελισσοκομίας. Ειδικότερα, γίνεται αναφορά στην σπουδαιότητα της εκτροφής μελισσών από την αρχαιότητα, περιγράφονται στοιχεία των συγκεκριμένων εντόμων και της οργάνωσής τους σε κοινοβιακούς σχηματισμούς, και επισημαίνονται τα οφέλη που παρέχουν στον άνθρωπο. Επίσης, αναλύονται οι κίνδυνοι που απειλούν τις αποικίες μελισσών και τέλος αποτιμάται η απήχηση της μελισσοκομίας στην Ελλάδα και την Ευρωπαϊκή Ένωση.

1.1 Σπουδαιότητα των μελισσών από την αρχαιότητα

Χρονική αφετηρία της σημερινής μελισσοκομικής πανίδας τοποθετείται πριν από 70 εκατομμύρια έτη[1] και μάλιστα αυτό τεκμηριώνεται και από την ύπαρξη άριστα διατηρημένου απολιθώματος που χρονολογείται 50 εκατομμύρια χρόνια πριν. Η σπουδαιότητα της μέλισσας και των παραγόμενων προϊόντων της ανάγεται ήδη στους αρχαίους χρόνους[2].

Αρχικά, τόσο στην Παλαιά και την Καινή Διαθήκη, όσο και στο Κοράνι, στα Ιερά κείμενα της Κίνας, στα χειρόγραφα της Αιγύπτου κτλ. γίνεται εμφανής ο σεβασμός στις μέλισσες. Για παράδειγμα στην Καινή Διαθήκη υπάρχει πλήθος χωρίων που καταδεικνύουν τη σημασία της μέλισσας. Στο κατά Ματθαίου Ευαγγέλιο, το μέλι αναφέρεται ως τροφή του Αγίου Ιωάννη Προδρόμου, ενώ στο κατά Λουκά Ευαγγέλιο η κηρήθρα παρουσιάζεται ως προϊόν κατανάλωσης από τον νεοαναστηθέντα Χριστό. Αναφορές για τις μέλισσες έχουμε και στους Βίους των Αγίων. Ο Άγιος Φιλάρετος ο Ελεήμων, δώρισε στους φτωχούς ολόκληρη την περιουσία του μαζί με 250 κυψέλες, γεγονός που τον έκανε να τιμάται ιδιαίτερα από τους μελισσοκόμους και να θεωρείται προστάτης τους. Αξίζει ιδιαίτερης αναφοράς ότι στο κοράνι πολλά κεφάλαια έχουν ονόματα ζώων ως τίτλο, ένα από αυτά είναι

και μέλισσα. Ο Ιουδαϊσμός αναγνωρίζει επίσης την σπουδαιότητα των μελισσών όχι μόνο για τα παραγόμενα προϊόντα τους, αλλά κυρίως για την κοινωνική τους δομή, καθώς όπως οι μέλισσες ακολουθούν την βασίλισσα, έτσι και οι εβραίοι ακολουθούν τους προφήτες. Ακόμα και στα Ινδουιστικά κείμενα το γάλα με το μέλι θεωρείται ένα ιατρικό ποτό[3].

Στην αρχαία ελληνική μυθολογία, η μέλισσα διαδραματίζει σημαντικό ρόλο, άλλωστε έχουν δημιουργηθεί διάφοροι μύθοι για αυτή. Από τις πιο χαρακτηριστικές αναφορές στην αρχαία ελληνική μυθολογία, αυτή κατά την οποία η Μέλισσα, κόρη του Μελισσέα –βασιλιά της Κρήτης– με τη βοήθεια της αδελφής της Αμάθειας, ανάθρεψε με μέλι και γάλα το νεογέννητο Δία ώστε να μεγαλώσει γρηγορότερα και να διεκδικήσει τη θέση του ανάμεσα στους θεούς. Ο Δίας ονομάστηκε και Μελιπτεύς, προς τιμήν της τροφού του.

Το αρχαιότερο πρόσωπο που εμφανίζεται στο χώρο της μελισσοκομίας και θεωρείται πατέρας της κατά την μυθολογία είναι ο Αρισταίος. Ο Αρισταίος ήταν γιός του Απόλλωνα και της νύμφης Κυρήνης. Ανατράφηκε από την Γαία και Ώρες, οι οποίες τον έκαναν αθάνατο στάζοντας στα χείλη του νέκταρ και αμβροσία. Την εκπαίδευση του ανέλαβαν οι Μούσες και οι Νύμφες, οι οποίες του έμαθαν την καλλιέργεια του αμπελιού, της ελιάς και την τέχνη της μελισσοκομίας, την οποία δίδαξε στους κατοίκους της Κέας για την οποία λατρευόταν.

Μέλισσα ονομαζόταν και η Ιέρεια της θεάς Δήμητρας, η οποία μυήθηκε από την θεά στα κρυφά της μυστήρια. Σύμφωνα με τη μυθολογία, η Μέλισσα δεν αποκάλυψε ποτέ τα μυστικά της θεάς και για αυτό οι άλλες γυναίκες την σκότωσαν. Από τα κομμάτια του σώματός της, γεννήθηκαν οι μέλισσες.

Σύμφωνα με έναν άλλο μύθο, μία μέλισσα πήγε κάποτε στον Δία και του προσέφερε μέλι. Εκείνος ευχαριστήθηκε πολύ και της είπε ότι μπορεί να της εκπληρώσει οποιαδήποτε χάρη θέλει. Η μέλισσα ζήτησε να μπορεί να χρησιμοποιεί το κεντρί της για να προστατεύει την κυψέλη και το μέλι της. Ο Δίας της έκανε τη χάρη αλλά της είπε ότι « θα γίνει αυτό που θέλεις, αλλά όταν θα χρησιμοποιείς το κεντρί σου θα χάνεις τη ζωή σου για το καλό της κυψέλης»[4].

Εκτός από τους μύθους, δημιουργήθηκαν και αρκετοί θρύλοι που αναδεικνύουν το σεβασμό και τη σημασία που πάντα είχαν οι μέλισσες. Ένας θρύλος από τη Βιζύη της Θράκης αναφέρεται στη σχέση της μέλισσας με την Αγιά

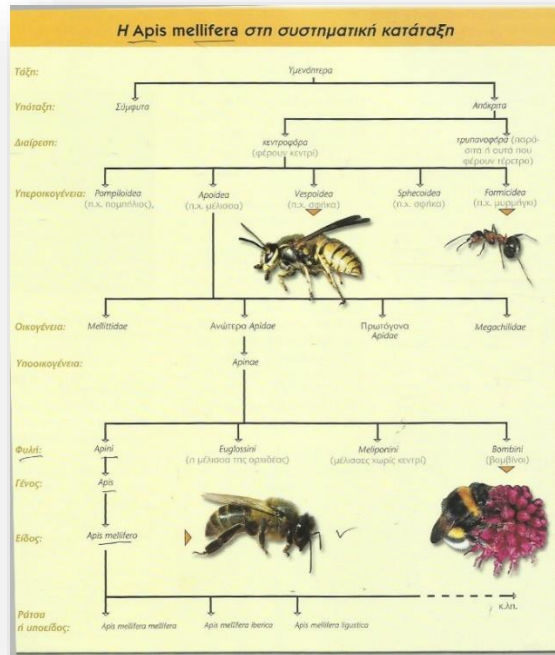
Σοφία της Κωνσταντινούπολης. Ο αυτοκράτορας Ιουστινιανός ήθελε να χτίσει ένα μεγαλοπρεπή ναό και το ανέθεσε στον πρωτομάστορά του. Ο καιρός περνούσε και κανένα σχέδιο του πρωτομάστορα δεν τον ικανοποιούσε. Μία Κυριακή κατά την Θεία Λειτουργία, έπεσε το αντίδωρο από τα χέρια του Ιουστινιανού και μία μέλισσα το πήρε έξω από το ναό. Ο αυτοκράτορας διέταξε, όσοι εκτρέφανε μέλισσα να ψάξουν τις κυψέλες τους για να το βρουν. Τελικά, το αντίδωρο βρέθηκε στη κυψέλη του πρωτομάστορα, όπου οι μέλισσες μέσα στην κυψέλη είχαν χτίσει με κερύ μία μεγαλόπρεπη εκκλησία σε μικρογραφία[5].

1.2 Γενικά στοιχεία μελισσών

Η μέλισσα είναι ένα εξαιρετικά περίπλοκο έντομο, το οποίο γοητεύει τον άνθρωπο ήδη από τους αρχαίους χρόνους μέχρι και σήμερα. Για παράδειγμα οι ανακαλύψεις του Karl von Frisch, αναφορικά με τους χορούς των μελισσών, του χάρισαν το 1973 το βραβείο Νόμπελ[6]. Είναι, λοιπόν, ιδιαίτερα σημαντικό να μελετηθεί η «οικογένεια των μελισσών».

Όλες οι μέλισσες ανήκουν στην τάξη των Υμενοπτέρων, υπόταξη απόκριτα, είναι κεντροφόρες και ανήκουν στην υπεροικογένεια Apoidea. Η υπεροικογένεια αυτή τρέφεται κυρίως με νέκταρ και γύρη και χαρακτηρίζεται από μακριά προβοσκίδα και τριχωτή επιδερμίδα. Βασικό χαρακτηριστικό των ανώτερων Apoidea είναι ότι διαθέτουν κοινωνικότητα, ενώ αντίθετα τα κατώτερα αυτής της υπεροικογένειας είναι μοναχικά.

Η *Apis Mellifera* (κοινή μέλισσα) μαζί με τους Βομβίνους (αγριομέλισσα) και τις μέλισσες χωρίς κεντρί (μέλισσες μικρού μεγέθους που ζουν κυρίως στην Αμερική) ανήκουν στην οικογένεια ανώτερα Apidae. Η *Apis Mellifera* ανήκει στη φυλή *Apini latreille*, η οποία περιλαμβάνει μόνο ένα γένος την *Apis*. Πρόκειται για την μελιτοφόρο μέλισσα που χαρακτηρίζεται από έντονα κοινωνική συμπεριφορά.



Εικόνα 1.1: Η Apis mellifera στη συστηματική κατάταξη

Πηγή: H. Clement, (2007), Σύγχρονη μελισσοκομία (σελ. 13)

Οι μέλισσες του παρελθόντος, έμοιαζαν με τις σημερινές, χωρίς όμως να διαθέτουν κεντρί. Με άλλα λόγια, οι πρώτες μέλισσες μορφολογικά έμοιαζαν με σφήκες (Vespoidea), υπόθεση που ενισχύεται από την ύπαρξη των σφηκών της οικογένειας των Masaginae, που τρέφονται αποκλειστικά με νέκταρ και γύρη. Γενικά οι φυλές των μελισσών, σε γενικά πλαίσια, πλέον διαμορφώνονται με βάση την γεωγραφική τους περιοχή, το μέγεθός τους, την τριχοφυΐα τους και τα άλλα ανατομικά χαρακτηριστικά τους. Γίνεται λοιπόν κατανοητό, ότι για να αποκτήσει κανείς μία ολοκληρωμένη εικόνα για τις μέλισσες, θα πρέπει να μελετήσει την ανατομία τους[6].

Μορφολογικά, το σώμα της μέλισσας αποτελείται από το κεφάλι, το θώρακα και την κοιλιά. Το κεφάλι έχει ωοειδές σχήμα και σε αυτό υπάρχουν τα διάφορα όργανα αίσθησης της μέλισσας (αφή, όραση, γεύση, όσφρηση)[7], ο εγκέφαλος και κάποιοι αδένες (υποφαρυγγικοί, χειλικοί και σιαγονικοί). Η όραση επιτυγχάνεται μέσω των απλών και των σύνθετων οφθαλμών, ενώ οι υπόλοιπες αισθήσεις από τις κεραίες και τα στοματικά μόρια. Οι απλοί οφθαλμοί αντιλαμβάνονται την ένταση του φωτός

Σχεδιασμός και ανάπτυξη συστήματος αυτοματισμού για μελισσοκομικές εφαρμογές

και βρίσκονται στο πάνω μέρος του κεφαλιού, ενώ οι σύνθετοι που βρίσκονται στα πλάγια του κεφαλιού αντιλαμβάνονται τις εικόνες[8].

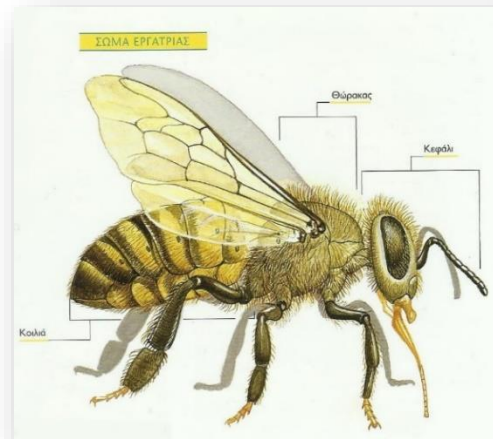


Εικόνα 1.2: Κεφάλι εργάτριας

Πηγή: H. Clement, (2007), Σύγχρονη μελισσοκομία (σελ. 18)

Ο θώρακας βρίσκεται ανάμεσα στην κοιλιά και το κεφάλι της μέλισσας και αποτελείται από τρία τμήματα. Το καθένα από αυτά φέρει ένα ζευγάρι πόδια, ενώ το δεύτερο και το τρίτο τμήμα φέρουν και από ένα ζεύγος μεμβρανωδών φτερών[8].

Η κοιλιά των μελισσών αποτελείται από δέκα κοιλιακούς δακτυλίους, (εφτά εμφανείς και τρεις μη εμφανείς). Στην κοιλιά βρίσκεται το σύνολο των εσωτερικών οργάνων καθώς και το κεντρί. Το κεντρί είναι ένα όργανο άμυνας, το οποίο όταν αποκολλάται, αποκολλάται μαζί με το σάκο δηλητηρίου. Για αυτό μία μέλισσα μπορεί να κεντρίσει μία μόνο φορά και μετά χάνει την ζωή της[6].



Εικόνα 1.3: Σώμα εργάτριας

Πηγή: H. Clement, (2007), Σύγχρονη μελισσοκομία (σελ. 18)

Πέρα από την μορφολογία της μέλισσας ιδιαίτερο ενδιαφέρον έχει να μελετηθεί και η ανατομία της, καθώς διαθέτει κυκλοφορικό, πεπτικό και απεκκριτικό, αναπνευστικό, νευρικό, αναπαραγωγικό και αδενικό σύστημα.

Όπως σε όλα τα έντομα έτσι και στην μέλισσα, το κυκλοφορικό σύστημα αποτελείται από την καρδιά και την αορτή, η οποία δεν μεταφέρει αίμα αλλά αιμολέμφο. Η αιμολέμφος είναι ένα άχρωμο υγρό το οποίο δεν μεταφέρει οξυγόνο αλλά θρεπτικά στοιχεία. Οι συσπάσεις της καρδιάς της μέλισσας, λοιπόν, μεταφέρουν αιμολέμφο στον εγκέφαλο μέσω της αορτής[8].

Το πεπτικό σύστημα της μέλισσας διακρίνεται από τρία μέρη. Αρχικά το πρόσθιο έντερο που περιλαμβάνει, το φάρυγγα, τον οισοφάγο, τον πρόλοβο, και το προστόμαχο. Το μέσο έντερο που αποτελεί ουσιαστικά το στομάχι της και τέλος το οπίσθιο έντερο που περιλαμβάνει το λεπτό και το παχύ έντερο. Με το πεπτικό σύστημα συνδέονται και ορισμένοι αδένες, όπως οι σιελογόνοι (παράγουν σάλιο) και οι υποφαριγγικοί αδένες (οι οποίοι παράγουν τον βασιλικό πολτό)[8].

Οι μέλισσες δεν διαθέτουν πνεύμονες αλλά τραχείες και για αυτό το αναπνευστικό της σύστημα χαρακτηρίζεται ως τραχειακό. Το σύστημα αυτό διοχετεύει το οξυγόνο σε όλα τα όργανα του σώματος και ταυτόχρονα αποβάλλει το διοξείδιο του άνθρακα. Αξίζει να αναφερθεί ότι κάποιες τραχείες διευρύνονται, σχηματίζοντας κάτι σαν αερόσακους, ώστε να υποβοηθούν την μέλισσα κατά την πτήση της[6].

Το νευρικό σύστημα της μέλισσας αποτελείται από τον εγκέφαλο και την κεντρική νευρική αλυσίδα (γάγγλια). Ο εγκέφαλος περιλαμβάνει τρία μέρη: τον πρωτεγκέφαλο, το δεύτερο εγκέφαλο και τον τρίτο εγκέφαλο. Ο πρωτεγκέφαλος είναι ο πιο ανεπτυγμένος και συνδέεται άμεσα με την όραση. Ο δεύτερος εγκέφαλος λειτουργεί ως ενδιάμεσος σταθμός μεταξύ κάθε κεραίας και του πρωτεγκέφαλου. Τέλος ο τρίτος εγκέφαλος κατά κύριο λόγο ελέγχει τους αδένες[6].

Πλήρως ανεπτυγμένο αναπαραγωγικό σύστημα διαθέτουν μόνο οι βασίλισσες και οι κηφήνες. Ορισμένες εργάτριες κάτω από ορισμένες συνθήκες γεννούν αγονιμοποίητα αυγά («ωοτόκες εργάτριες» ή «ψευδομητέρες»)[7]. Η βασίλισσα διαθέτει δύο ωοθήκες, κάθε μία εκ των οποίων διαθέτει ωοφόρους σωλήνες (οβαριόλες), που παράγουν ωά, τα οποία περνάνε από την σπερματοθήκη και εξέρχονται από τον κόλπο. Στην σπερματοθήκη βρίσκονται συγκεντρωμένα τα σπερματοζωάρια των κηφήνων, με τους οποίους η βασίλισσα ζευγάρωσε στην αρχή της ζωής της. Ο κηφήνας είναι σεξουαλικά ώριμος, περίπου δώδεκα μέρες μετά την έξοδό του από το κελί. Κατά την σύζευξη τα γεννητικά όργανα του κηφήνα, μέσω της σύσπασης των μυών της κοιλίας, βγαίνουν από το σώμα του, εκσπερματώνει, και τα γενετικά του όργανα αποκόπτονται και στην συνέχεια πεθαίνει[8].

Οι μέλισσες διαθέτουν δύο ήδη αδένων, τους ενδοκρινείς και τους εξωκρινείς. Οι ενδοκρινείς εκκρίνουν ορμόνες και σχετίζονται με την ανάπτυξη και την συμπεριφορά του εντόμου. Οι εξωκρινείς αδένες είναι πιο γνωστοί καθώς σχετίζονται με τα παραγόμενα προϊόντα της μέλισσας την άμυνα της, την επικοινωνία της και την επεξεργασία της τροφής της.

Είναι άξιο αναφοράς, ότι οι μέλισσες διαθέτουν και τις πέντε ανθρώπινες αισθήσεις. Όπως προαναφέρθηκε, μέσω των οφθαλμών, έχουν την αίσθηση της όρασης, την οποία χρησιμοποιούν τόσο για να συλλέξουν τροφή όσο και για να γυρίσουν στην κυψέλη τους. Η αίσθηση της αφής της μέλισσας, δεν περιορίζεται στις κεραίες της, όπως περιορίζεται η αίσθηση της αφής στο χέρι του ανθρώπου, αλλά ένα σύνολο αισθητήριων οργάνων καλύπτει ολόκληρο το σώμα της ώστε να αντιλαμβάνεται πλήρως το φυσικό της περιβάλλον. Μέσω των κεραίων, των ταρσών (στα μπροστινά πόδια της μέλισσας) και της προβοσκίδας της, η μέλισσα αποκτά επίσης την αίσθηση της γεύσης. Αντιλαμβάνεται και τις τέσσερις γεύσεις

που αντιλαμβάνεται και ο άνθρωπος (ξινό, γλυκό, αλμυρό, πικρό) με την διαφορά ότι η πικρή γεύση δεν έχει καμία σημασία για αυτές. Στις κεραίες επίσης βρίσκονται τα αισθητήρια όργανα της όσφρησης, η οποία είναι δύο φορές πιο έντονη από την όσφρηση του ανθρώπου και τ χρησιμοποιούν τόσο για τα λουλούδια όσο και για να αναγνωρίζουν τα άτομα της ίδιας κυψέλης. Αναφορικά με την ακοή η μέλισσες αντιλαμβάνονται και ήχους και δονήσεις. Εκτός των βασικών αυτών αισθήσεων οι μέλισσες αντιλαμβάνονται το διοξείδιο του άνθρακα καθώς και διάφορες αλλαγές του καιρού, όπως αλλαγή θερμοκρασίας, υγρασίας, ατμοσφαιρικής πίεσης και άλλα[6].

1.3 Η κοινωνική ζωή της αποικίας

Επειδή η μέλισσα, όπως και ο άνθρωπος είναι κοινωνικό ον, δεν μπορεί να ζήσει απομονωμένη, γι' αυτό το λόγο δημιουργεί αποικίες[9]. Με άλλα λόγια, ζει σε οργανωμένο σύνολο (σμήνος), το οποίο αποτελείται από τρεις κάστες με διαφορετικές μορφολογίες και ρόλους. Υπάρχουν πάρα πολλές εργασίες, οι οποίες πρέπει να επιτευχθούν και για αυτό το λόγο είναι απαραίτητη μία αυστηρή οργάνωση.

Οι μέλισσες ζουν σε ένα ιδιαίτερο χώρο (κυψέλη), η οποία αποτελεί το ζωτικό κέντρο της αποικίας. Η κυψέλη απαρτίζεται από ομοιόμορφες θήκες από κερί (κηρήθρες), τοποθετημένες παράλληλα. Από τους Αρχαίους χρόνους μέχρι και σήμερα, έχουν χρησιμοποιηθεί διάφορα συστήματα κυψελών, όπως για παράδειγμα η κυψέλη Langstroth, η κυψέλη Jumbo, η κυψέλη Dadant-Modifiee, η Standart (πρότυπη) κυψέλη και άλλες[7]. Οι πλευρές της κηρήθρας κατασκευάζονται από τις εργάτριες μέλισσες και αποτελούνται από κελιά, που χρησιμεύουν για να τοποθετήσει η βασίλισσα τα αυγά της, να γίνει η ανατροφή του γόνου και για να αποθηκευτούν τα πολύτιμα αποθέματα που είναι απαραίτητα για την ομαλή και σίγουρη διαβίωση του πληθυσμού. Ο γόνος τοποθετείται στο κέντρο της κυψέλης και το μέγεθος που καταλαμβάνει διαφέρει ανάλογα την εποχή, καθώς την άνοιξη έχει πολύ μεγαλύτερο όγκο, ο οποίος ο χειμώνα συρρικνώνεται. Αξίζει να αναφερθεί ότι τα κελιά είναι εξάγωνα επειδή είναι το σχήμα με την μικρότερη περίμετρο και συνεπώς δαπανάτε μικρότερη ποσότητα κεριού χωρίς να δημιουργούνται κενά, αφετέρου διότι όπως αποδεικνύεται από ανώτερα

μαθηματικά (λογισμό μεταβολών), τα κανονικά εξάγωνα καταστούν δυνατό τον διαμερισμό ενός δοχείου, με το μέγιστο δυνατό όγκο στα διαμερισμένα κελιά[10].



Εικόνα 1.4: Γόνος (καφετί χρώμα)

Σε κάθε κυψέλη, ή αλλιώς ένα μελίσσι, διακρίνουμε τριών ειδών κατοίκους (ακμαία άτομα) την βασίλισσα, τις εργάτριες και τους κηφίνες. Η διαφορετική τους μορφολογία, φυσιολογία και συμπεριφορά καθιστά δυνατή την καλύτερη δυνατή περάτωση των εργασιών που πρέπει να συντελεστούν στην κυψέλη[6]. Η βασίλισσα είναι το μόνο θηλυκό, το οποίο έχει την ευθύνη της αναπαραγωγής, δηλαδή εξασφαλίζει τη συνέχεια της αποικίας και ηγείται του μελισσιού. Στη συνέχεια, διακρίνουμε τους κηφίνες, που είναι τα αρσενικά άτομα της κυψέλης, είναι μερικές εκατοντάδες σε ορισμένη εποχή και είναι επιφορτισμένοι με τη διαδικασία αναπαραγωγής της βασίλισσας. Τέλος, υπάρχουν οι εργάτριες των οποίων βασική ευθύνη είναι η δημιουργία της κυψέλης, η διαδικασία παραγωγής όλων των προϊόντων και την εκκόλαψη των αυγών. Με άλλα λόγια, οι εργάτριες πραγματοποιούν όλες τις λειτουργίες που χρειάζονται για την ανάπτυξη και την επιβίωση της αποικίας. Με το όρο μέλισσες αναφερόμαστε σε όλα αυτά τα ακμαία έντομα.

Ανάλογα με το πόσο μεγάλη και το πόσο αναπτυγμένη είναι η κάθε αποικία το σύνολο του πληθυσμού της διαφέρει. Συνήθως κυμαίνεται μεταξύ 20.000-80.000, εκ των οποίων υπάρχει μία και μοναδική βασίλισσα (προσωρινά μπορεί να υπάρχουν και περισσότερες) και 1.000-1.400 κηφίνες. Ο υπόλοιπος πληθυσμός αποτελείται από τις εργάτριες[6].

Η ανάπτυξη της αποικίας (συγκομιδή και ωτοκία της βασίλισσας) συντελείται σε στάδια ενός έτους, τα οποία καλούνται και «βιολογικός κύκλος» και επηρεάζονται κυρίως από τις εποχές και το περιβάλλον. Αναφορικά με τα γενετικά τους χαρακτηριστικά, η κάθε ράτσα μελισσών διαθέτει διαφορετικό βιολογικό κύκλο και ενσωματώνεται στις διαφορετικές περιβαλλοντολογικές και κλιματολογικές συνθήκες κάθε τόπου, ώστε να αποτελέσουν τμήμα του βιότοπου αυτού.

Η βασίλισσα αποτελεί τον «εγκέφαλο» της κυψέλης, αφού με τις φερομόνες της ρυθμίζει τις δραστηριότητες μέσα στην αποικία. Οι βασιλικές φερομόνες παράγονται από τους σιαγονικούς αδένες, για αυτό και στις βασίλισσες οι αδένες αυτοί είναι υπερτροφικοί. Η μορφολογία της διαφέρει από αυτή των εργατριών και των κηφήνων καθώς είναι προσαρμοσμένη στην αναπαραγωγή. Χαρακτηρίζεται από μια ιδιαίτερα ανεπτυγμένη κοιλιά και έναν πιο ογκώδη θώρακα. Επιπλέον, το κεντρί της διαθέτει έναν ιδιαίτερα ανεπτυγμένο σάκο δηλητηρίου, με αποτέλεσμα ακόμα και αν κεντρίσει να μην πεθαίνει, σε αντίθεση με τις εργάτριες. Όπως έχει προαναφερθεί, μόνο η βασίλισσα διαθέτει ένα ολοκληρωμένο αναπαραγωγικό σύστημα. Ανάλογα με την ηλικία της βασίλισσας, μεταβάλλεται και η μορφολογία της. Μία νεαρή βασίλισσα, τόσο μορφολογικά όσο και κινησιολογικά μοιάζει με τις εργάτριες για αυτό και είναι δύσκολο να εντοπιστεί. Μετά την γονιμοποίησή της αρχίζει να κινείται πιο αργά, αυξάνεται αρκετά το μέγεθος της κοιλιάς της και οι εργάτριες σχηματίζουν γύρω της έναν κύκλο, τη λεγόμενη «αυλή της βασίλισσας». Καθώς γερνάει, χάνει το τρίχωμα της και τα φτερά της σταδιακά καταστρέφονται. Τελευταίο διαφοροποιητικό στοιχείο της βασίλισσας είναι ότι δεν διαθέτει μηχανισμούς για την συλλογή γύρης[6]. Ορισμένοι συγγραφείς υποστηρίζουν ότι το όνομα «μάννα του μελισσιού» της ταιριάζει καλύτερα, καθώς όπως προαναφέρθηκε ο προορισμός της είναι να γεννά το σμήνος. Οι υπόλοιπες μέλισσες, επειδή τη θεωρούν μάννα τους την περιβάλλουν με πολύ μεγάλη αγάπη και αφοσίωση. Χαρακτηριστικό παράδειγμα αποτελεί το γεγονός ότι σε περίπτωση εξάντλησης της τροφής του μελισσιού το χειμώνα, η τελευταία σταγόνα μέλι προορίζεται για την βασίλισσα[7].



Εικόνα 1.5: Βασίλισσα

Ο κηφήνας, ονομάζεται αλλιώς και ψευδο-βομβίνος, υπάρχει σε μία αποικία, μόνο όταν αυτή διαθέτει καλά αποθέματα. Ο κηφήνας μορφολογικά διαθέτει πιο χοντρό και πιο μαύρο σώμα από τις εργάτριες, πιο σύνθετα μάτια και δεν διαθέτει κεντρί. Η γνωστή λειτουργία του, είναι η γονιμοποίηση της βασίλισσας. Μόνο ένας κηφήνας θα γονιμοποιήσει την βασίλισσα, γεγονός που συντελείται στον αέρα και όχι μέσα στην κυψέλη. Τις πρώτες μέρες γέννησης τους, οι κηφήνες τρέφονται από τις τροφούς, ενώ αργότερα τρέφονται και μόνοι τους με μέλι. Μετά το καλοκαίρι οι κηφήνες δεν έχουν πλέον χρησιμότητα, ωστόσο, αν υπάρχει αφθονία φαγητού οι μέλισσες τους ανέχονται, ενώ αν η ποσότητα του φαγητού μειωθεί σημαντικά, οι εργάτριες συχνά κεντρίζουν και σκοτώνουν τον κηφήνα[7]. Ο αριθμός των κηφήνων σε μία κυψέλη αυξάνεται τη άνοιξη, που γίνεται η σμηνουργία (μία διαδικασία διαχωρισμού της κυψέλης σε δύο πληθυσμούς)[6].



Εικόνα 1.6: Κηφήνας

Τα γρανάζια της εύρυθμης λειτουργίας της κυψέλης αποτελούν οι εργάτριες, οι οποίες είναι θηλυκού γένους και δεν διαθέτουν ολοκληρωμένο αναπαραγωγικό σύστημα. Ωστόσο, διαθέτουν άλλα εξειδικευμένα όργανα, τα οποία τους δίνουν την ικανότητα να επιτελούν διάφορες λειτουργίες μέσα στην κυψέλη. Με άλλα λόγια, η εργάτρια είναι μια ατελής μέλισσα, η οποία δε γονιμοποιείται, προκειμένου να επιτελεί αποκλειστικά όλες τις άλλες εργασίες, οι οποίες την επιβαρύνουν[7]. Μορφολογικά, χαρακτηρίζονται από πολύ μακριά προβοσκίδα, ώστε να συλλέγει το νέκταρ. Επιπλέον με τα πίσω πόδια τους, συλλέγουν την γύρη και την πρόπολη. Όπως η βασίλισσα, έτσι και οι εργάτριες διαθέτουν κεντρί το οποίο χρησιμοποιούν ως αμυντικό όπλο. Ανάλογα με την ηλικία τους οι εργάτριες επιτελούν διαφόρων ειδών εργασίες. Οι νεαρές, είναι υπεύθυνες για την εξασφάλιση της υγιεινής της κυψέλης. Επιτελούν, δηλαδή, χρέη «καθαρίστριας», καθώς ετοιμάζουν τα κελιά για την υποδοχή των αυγών ή την αποθήκευση τροφής. Οι λίγο μεγαλύτερες εργάτριες απομακρύνουν και τα υπολείμματα πτωμάτων από την αποικία. Άλλες εργάτριες «τροφοί», φροντίζουν για την ανάπτυξη των γόνων. Οι λεγόμενες «αρχιτέκτονες και μαραγκοί» είναι επιφορτισμένες να κατασκευάσουν και να σφραγίσουν τα κελιά της κηρήθρας. Σε αναλογία με αυτή την γραμμή σκέψης, διακρίνουμε και την «συλλέκτρια», που συλλέγει την τροφή, την «αποθηκάριο», που διαχειρίζεται την τροφή με την οποία η συλλέκτρια επιστρέφει στην κυψέλη, την «αερίστρια», που

ρυθμίζει το μικροκλίμα και την θερμοκρασία της αποικίας (αυτό συμβαίνει κυρίως κατά τη σμηνουργία) και την «φρουρό», που υπερασπίζονται την κυψέλη[6].



Εικόνα 1.7: Εργάτριες

1.4 Η σημασία των μελισσών και των προϊόντων τους για τον άνθρωπο

Η μελισσοκομία αποτελεί κλάδο της εντομολογίας, ο οποίος ασχολείται με την εκτροφή της μέλισσας, καταλήγοντας μέχρι την εκμετάλλευση της από τον άνθρωπο. Οι μέλισσες είναι κοινωνικά έντομα που ζουν κατά χιλιάδες στο «σπίτι» τους, την κυψέλη. Επειδή, λοιπόν, οι μέλισσες έχουν την τάση να δημιουργούν φωλιές, ο άνθρωπος οδηγήθηκε στη σκέψη ότι είναι δυνατό να τις συλλάβει και να τις βάλει να ζήσουν μέσα σε κάποιο κουτί, που να μοιάζει με κουφάλα δέντρου ή με τρύπα σε βράχο, προκειμένου να παράγουν μέλι γι' αυτόν[11]. Με αυτό τον τρόπο άρχισε η ενασχόληση του ανθρώπου με τη μελισσοκομία. «Εκτρέφονται για το μέλι και τα άλλα προϊόντα που παράγουν, για την επικοινωνιακή τους δράση στα καλλιεργούμενα φυτά ή ως ευχάριστη ερασιτεχνική απασχόληση. Η σχέση του ανθρώπου με τη μέλισσα χάνεται στα βάθη των αιώνων. Η συνεισφορά όμως, της μέλισσας στην αύξηση της φυτικής παραγωγής μέσω της επικοινωνίας είναι ανυπολόγιστη. Αν συνυπολογίσει κανείς το γενικότερο όφελος της φύσης από την

επικοινωνία, καταλαβαίνει το πόσο σημαντικός είναι ο κλάδος της μελισσοκομίας παγκοσμίως. Μας ενδιαφέρουν τόσο τα προϊόντα που παίρνουμε από την μελισσοκομική ενασχόληση όσο η γονιμοποιός προσφορά της μέλισσας.

Είναι γεγονός πως η μέλισσα αποτελεί τον σημαντικότερο οργανισμό πάνω στον πλανήτη καθώς, όπως αναφέρει και ο Albert Einstein «αν οι μέλισσες εξαφανιστούν, απομένουν στον άνθρωπο τέσσερα χρόνια ζωής». Είναι υπεύθυνη για την γονιμοποίηση των φυτών μέσω της επικοινωνίας. Με το όρο επικοινωνία εννοούμε τη μεταφορά γύρης από τους ανθήρες ενός άνθους, στο στίγμα ενός άλλου άνθους. Η διαδικασία αυτή είναι ζωτικής σημασίας για την αναπαραγωγή, την επιβίωση και την εξέλιξη, όχι μόνο των ανθών και των φυτών γενικότερα, αλλά και για των ζωικών ειδών, τα οποία τρέφονται από τα πρώτα, οδηγώντας στην εύρυθμη λειτουργία και ανάπτυξη όλου του οικοσυστήματος. Έρευνες έδειξαν πως η μέλισσα ευθύνεται για το 60-70% της γονιμοποίησης των φυτών και πως η σημασία της ύπαρξής της δεν εστιάζεται στο μέλι και τα υπόλοιπα προϊόντα που παράγονται στην κυψέλη και καταναλώνονται κυρίως από τον άνθρωπο, αλλά σε όλες τις σημαντικές λειτουργίες που εκτελεί στη φύση, προς όφελος όλων των ζωντανών πλασμάτων επί γης.

Η κύρια προσφορά της μέλισσας, είναι η διατήρηση του φυσικού περιβάλλοντος. Ωστόσο, εξαιτίας της κοινωνικής της συμπεριφοράς της, παράγει στην κυψέλη της διάφορα προϊόντα, τα οποία αποφέρουν μεγάλη χρησιμότητα στην διατροφή και την υγεία του ατόμου.

Το μέλι αποτελεί το σπουδαιότερο προϊόν της μελισσοκομίας, ενώ γνωστή είναι η ιστορία του, η οποία ξεκινά από τους αρχαίους χρόνους. Άλλα παράγωγα της μελισσοκομικής εκμετάλλευσης είναι η γύρη, το κερί, η πρόπολη, ο βασιλικός πολτός και το δηλητήριο της μέλισσας.

Με τον όρο μέλι (σύμφωνα με τον ορισμό του Διεθνούς Οργανισμού Γεωργίας και Τροφίμων (FAO) αναφερόμαστε στο «γλυκό προϊόν που παράγουν οι μέλισσες του είδους *Apis mellifera*, καθώς συλλέγουν, μετατρέπουν και αποθηκεύουν στις κηρήθρες το νέκταρ ή άλλους φυτικούς χυμούς από διάφορα ζωντανά μέρη του φυτού»[12]. Από την αρχαιότητα μέχρι το 18ο αιώνα αποτελούσε αποκλειστικό ζαχαρώδες τρόφιμο για τον άνθρωπο[8]. Εξαιτίας της άμεσα αφομοιώσιμης γλυκόζης που περιέχει καταναλώνεται ευρύτατα ως δυναμωτικό.

Αξίζει να αναφερθεί ότι ανάλογα με την εποχή παραγωγής του μελιού, παρατηρείται διαφορετική γεύση, οσμή και διαφορετικό χρώμα. Για παράδειγμα, τα πρώτα μέλια του χρόνου χαρακτηρίζονται από ανοιχτό χρώμα και «φρουτώδη» μυρωδιά, ενώ τα μέλια που παράγονται αργότερα έχουν πιο έντονο χρώμα και πιο «δασώδη» μυρωδιά[6].



Εικόνα 1.1: Μέλι στην κηρύθρα

Η γύρη είναι ένα προϊόν που συγκεντρώνουν οι μέλισσες από διάφορα άνθη. Αν παρατηρήσει κανείς ένα λουλούδι, θα δει τους στημόνες, οι οποίοι καταλήγουν στους ανθήρες (κάτι σαν μικροί σάκοι) οι οποίοι περιέχουν μία χρωματιστή σκόνη, την γύρη. Με τη «ληηλασία» των ανθών η μέλισσα εξασφαλίζει τόσο την δική της διατροφή, όσο και την επικοινωνία των λουλουδιών. Αποτελεί μία φυσική τροφή, που διαθέτει ένα εντυπωσιακό θρεπτικό προφίλ καθώς είναι πλούσια σε πρωτεΐνες, αμινοξέα, βιταμίνες κτλ. Γι' αυτό άλλωστε αποτελεί και την βασική Πηγή τροφής των μελισσών. Μια αποικία, προκειμένου να εκτρέψει τον γόνο, καταναλώνει 35-40 κιλά γύρης το χρόνο[6].



Εικόνα 1.9: Γύρη (κίτρινες κυψέλες)

Ο βασιλικός πολτός, συχνά χρησιμοποιείται και ο όρος «γάλα των μελισσών», είναι μία κρεμώδης ουσία που εκκρίνεται από τους υποφαρυγγικούς αδένες, που βρίσκονται στο κεφάλι των εργατριών. Με τον βασιλικό πολτό τρέφονται όλες οι προνύμφες τις πρώτες τους μέρες και έπειτα μόνο οι προνύμφες, οι οποίες ως τέλεια έντομα θα γίνουν βασίλισσες. Οι βασίλισσες μέλισσες τρέφονται αποκλειστικά από τον βασιλικό πολτό και σε αυτόν οφείλουν το μέγεθος, την γονιμότητα και την μακροβιότητα τους. Τα βασιλικά κελιά περιέχουν το πιο πλούσιο πολτό και στην μεγαλύτερη ποσότητα. Γίνεται σαφές λοιπόν, ότι ο βασιλικός πολτός είναι πλούσια Πηγή βιταμινών και θεωρείται ότι έχει ευεργετικές ουσίες για τον ανθρώπινο οργανισμό. Εκτός όμως από την θρεπτική του αξία, ο βασιλικός πολτός έγινε αντικείμενο πολλών ερευνών αναφορικά με τις φαρμακευτικές του ιδιότητες στις νευρασθένειες, στην γηριατρική, στην εγκυμοσύνη, στην υπερέκπτωση και σε άλλες περιπτώσεις διαταραχών του οργανισμού[8].



Εικόνα 1.10: Βασιλικός πολτός

Πηγή:

https://el.wikipedia.org/wiki/%CE%92%CE%B1%CF%83%CE%B9%CE%BB%CE%B9%CE%BA%CF%8C%CF%82_%CF%80%CE%BF%CE%BB%CF%84%CF%8C%CF%82

Η πρόπολη είναι ένα υλικό με ποικίλες ιδιότητες που χρησιμοποιείται τόσο στην φαρμακευτική παραγωγική διαδικασία, όσο και στην παραγωγή καλλυντικών προϊόντων. Η πρόπολη είναι ρητινώδης κολλητική ουσία που συλλέγουν οι μέλισσες από διάφορα φυτά, όπως για παράδειγμα φτελιές, ιτιές, έλατα και πεύκα, την εμπλουτίζουν με κερί, γύρη, ένζυμα και άλλες ουσίες και τη χρησιμοποιούν για να στεγανοποιήσουν και να απολυμάνουν το εσωτερικό της κυψέλης, όπου θα φιλοξενηθούν οι γόννοι των μελισσών και θα αποθηκευτούν το μέλι και η γύρη. Η ονομασία (προ + πόλις) οφείλεται στο ότι οι μέλισσες την τοποθετούν στην είσοδο της κυψέλης ώστε να εμποδίζουν την είσοδο διαφόρων εχθρών στη φωλιά τους καθώς και για την προστασία τους από τις διάφορες δυσμενής καιρικές συνθήκες[13].



Εικόνα 1.11: Πρόπολη

Στην παραγωγή καλλυντικών χρησιμοποιείται και το κερι των μελισσών. Το κερι εκκρίνεται από τους κηρογόνους αδένες των εργατριών μελισσών ηλικίας 12-18 ημερών και το χρησιμοποιούν για να φτιάχνουν την κυψέλη τους και να σφραγίζουν τα κελία της κηρήθρας τους. Όταν βγει από τους αδένες μοιάζει σαν λέπτι και έχει λευκό χρώμα, όμως κατά την επεξεργασία του από τις μέλισσες, επειδή αναμειγνύεται με την γύρη αποκτά κίτρινο χρώμα. Η χρησιμότητα κεριού από τον άνθρωπο δεν είναι κάτι καινούριο, αλλά ανάγεται πολύ βαθιά στο παρελθόν. Για παράδειγμα οι άνθρωποι χρησιμοποιούσαν το κερι για γραφή, για επισκευές πλοίων, για φωτισμό, για γλυπτική και άλλα[6].

Τέλος, πολλές είναι οι φαρμακολογικές ιδιότητες που έχει το δηλητήριο της μέλισσας, το οποίο αποτελεί ένα πολύπλοκο μείγμα χημικών ουσιών και επηρεάζει την φυσιολογία ενός οργανισμού, για αυτό και αποτελεί το «όπλο» της μέλισσας εναντίον τον εχθρών της. Η έκκριση του δηλητηρίου συντίθεται στους αδένες των θηλυκών μελισσών και με το τσίμπημα εκτοξεύεται μέσω το κεντριού. Μία ώριμη εργάτρια έχει 100-150μm δηλητήριο ενώ η βασίλισσα περίπου 700μm. Στις μέρες

μας γίνονται έρευνες αναφορικά με τις φαρμακευτικές ιδιότητες του δηλητηρίου των μελισσών για παράδειγμα στην ρευματοειδή αρθρίτιδα, στις μυαλγίες, στις νευραλγίες, στον καρκίνο αλλά και σε άλλες ασθένειες[6].

1.5 Κίνδυνοι μελισσοκομίας

Παρά το γεγονός ότι ένας από τους βασικούς μηχανισμούς άμυνας των μελισσών είναι η διαρκής ανανέωση του πληθυσμού τους, παρατηρείται ότι τα τελευταία χρόνια ο πληθυσμός τους διαρκώς μειώνεται. Η συνεχής αυτή μείωση αποτελεί αντικείμενο ενδιαφέροντος, καθώς η εξάλειψή τους είναι απειλή για την ύπαρξη ζωής πάνω στον πλανήτη. Από τα τέλη της δεκαετίας του '90 είχε επιβεβαιωθεί από μελισσοκόμους σε Ευρώπη και Β. Αμερική πως οι πληθυσμοί των μελιφόρων μελισσών μειώνονταν με ασυνήθιστα ταχύτατους ρυθμούς. Αυτό συμβαίνει τόσο εξαιτίας των «εχθρών» τους, που θέλουν να εκμεταλλευτούν τα παραγόμενα προϊόντα τους, όσο και εξαιτίας διαφόρων ασθενειών και δηλητηριάσεων που πλήττουν το μέλισσι. Επομένως καθίσταται κατανοητό ότι πρέπει να αναλυθούν ορισμένες από τις βασικές αιτίες θανάτου των μελισσών[8].

Υπάρχουν πάρα πολλοί εχθροί των μελισσών, όπως διάφορα έντομα ή άλλα ζώα. Κάποια από αυτά είναι πιο επιβλαβή από κάποια άλλα. Στο σημείο αυτό, θα αναλυθούν οι εχθροί εκείνοι που προκαλούν τις μεγαλύτερες ζημιές σε μία αποικία.

Αρχικά, ένας βασικός εχθρός του μελισσιού είναι η σφήκα. Η σφήκα μοιάζει αρκετά με την μέλισσα, καθώς ανήκει επίσης στην τάξη των Υμενόπτερων, στην υπόταξη των απόκριτων, και στην διαίρεση των κεντροφόρων. Οι ενήλικες σφήκες, παρόλο που μπορούν να τραφούν και με χυμούς φρούτων, για τις προνύμφες τους προτιμούν λευκωματούχες ενώσεις (πχ κρέας, ψάρι). Για αυτό το λόγο τον Ιούνιο που ο πληθυσμός των σφηκών αυξάνεται, ο θώρακας των μελισσών θεωρείται ως εύκολο θήραμα[8].



Εικόνα 1.12: Παγίδα αντιμετώπισης σφηκών

Μία δεύτερη απειλή για τα μελίσσια είναι ο κηρόσκορος. Ο κηρόσκορος είναι ένα Λεπιδόπτερο (νυκτόβια γκριζα πεταλούδα) με όνομα είτε *Galleria melonella* (αν είναι μεγάλος), είτε *Achroea grisella* (αν είναι μικρός). Τα θηλυκά συχνά εναποθέτουν τα αυγά τους μέσα στις κηρήθρες οι οποίες καταστρέφονται από τις κάμπιες που εκκολάπτονται από τα αυγά, καθώς οι κάμπιες στηρίζουν τη διατροφή τους, στην αποθηκευμένη γύρη στο κουκούλι του γόνου και στα κόπρανα των προνυμφών[6].

Την άνοιξη, ο μελισσοφάγος που μεταναστεύει στην Ελλάδα για να αναπαραχθεί, δημιουργεί «πρόβλημα» στα μελίσσια, καθώς τρέφεται με έντομα και έχει ιδιαίτερη προτίμηση σε υμενόπτερα όπως μέλισσες και σφήκες. Την ίδια περίπου εποχή, και τα χελιδόνια, συνιστούν απειλή για τις μέλισσες, στην προσπάθειά τους να ταΐσουν τα μικρά τους[8].

Οι μέλισσες όπως και άλλοι οργανισμοί είναι δυνατόν να προσβληθούν από διάφορες οργανισμούς, όπως ιούς και βακτήρια, με αποτέλεσμα να αρρωστήσουν. Αυτό μπορεί να συμβεί τόσο στους γόνους των μελισσών όσο και στις ενήλικες μέλισσες.

Ένα από τα μεγαλύτερα προβλήματα της μελισσοκομίας σχεδόν παγκοσμίως είναι η βαρόαση, η οποία προκαλείται από το παρασιτικό άκαρι *Varroa jacobsoni*. Η βαρόαση είναι εμφανής με το μάτι και στους γόνους και στις μέλισσες, καθώς μπορεί να διακρίνει κανείς μεμονωμένα μικρά καφεκόκκινα παράσιτα. Με την προσβολή από βαρρόα οι μέλισσες κινδυνεύουν από οξεία παράλυση και σε μεμονωμένες περιπτώσεις και σηψαιμία. Επιπλέον, οι βαρρόες καταναλώνουν την τροφή των προνυμφών, με αποτέλεσμα να μειώνεται η διάρκεια ζωής των χειμερινών μελισσών[14].



Εικόνα 1.13: Βαρρόας

Πηγή: <https://www.melissokomika-souani.gr/blog/item/115-varoa-kai-therapeia>

Συχνά, οι γόνους ταλανίζονται από την Αμερικανική και την Ευρωπαϊκή σηψιγονία. Η Αμερικανική σηψιγονία (κακοήθης) αποτελεί βακτηριακή λοίμωξη, που προκαλείται από το βακτήριο *Bacillus larvae* και συνήθως μεταφέρεται στους γόνους μέσω της τροφής. Ιδιαίτερα οι νεαρές προνύμφες μολύνονται από το βακτήριο αυτό. Τα σφραγίσματα των κελιών συνήθως αποκτούν τρύπες και ο γόνος αποσυντίθεται. Η Ευρωπαϊκή σηψιγονία (κύριο αίτιό της το βακτήριο *Melissococcus pluton*), μοιάζει αρκετά με την Αμερικανική σηψιγονία, ωστόσο θεωρείται καλοήθης καθώς η αυτοϊάση της ευνοείται από την τάση των μελισσών για καθαρισμό[14].

Ωστόσο, είναι δυνατόν να ασθενήσουν και οι ενήλικες μέλισσες. Για παράδειγμα κάποιες από τις ασθένειες των μελισσών είναι: σηψαιμία, δυσεντερία, μελισσόψειρα, Νόσος του Μάη και άλλες.

Η σηψαιμία έχει ως συχνότερο αίτιο το βακτήριο *Peudomonas apisertica*. Μέσω της αιμολέμφου, μεταφέρονται διάφορα βακτήρια στις μέλισσες και μάλιστα ορισμένα από αυτά προσβάλλουν αποκλειστικά τις μέλισσες. Όταν ένα μελίσι υποφέρει από σηψαιμία, τότε παρατηρείται μπροστά στην κυψέλη, ένα πλήθος μελισσών που δεν μπορούν να πετάξουν και σέρνονται, έχοντας χάσει το μεγαλύτερο ποσοστό ή και ολόκληρο το τρίχωμά τους[14].

Στις χειμερινές μέλισσες, παρατηρείται ορισμένες φορές δυσεντερία, εξαιτίας αφενός της ανικανότητάς τους να κάνουν πτήσεις καθαρισμού και αφετέρου του είδους της τροφής που καταναλώνουν. Αυτό έχει ως αποτέλεσμα, το εσωτερικό της κυψέλης να έχει καφέ λεκέδες εξαιτίας των περιπτώματων των μελισσών[14].

Όπως προαναφέρθηκε, ο βαρρός αποτελεί ένα από τα σημαντικότερα προβλήματα στην μελισσοκομία. Συχνά όμως μπερδεύουμε το βαρρόα με την μελισσόψειρα (*Braula coeca*), η οποία εντοπίζεται συνήθως στο θώρακα των μελισσών, μέχρι την στιγμή που θα μετακινηθούν στο κεφάλι τους για να τραφούν. Επί της ουσίας εισχωρεί στους υποφαρυγγικούς αδένες, για να καταναλώσει βασιλικό πολτό για αυτό και το καλύτερο γέυμα της αποτελεί η βασίλισσα[14].

Η Νόσος του Μάη εμφανίζεται κυρίως κατά το Μάιο (εξού το όνομά της) στις νεαρές μέλισσες, και συνηθέστερα στις μέλισσες τροφούς. Οι μέλισσες έχουν διογκωμένες κοιλιές και συνήθως σέρνονται έξω από την κυψέλη για να πεθάνουν. Βασική αιτία της είναι η αφυδάτωση, ο χρόνος εγκλεισμός λόγω καιρικών συνθήκων και λόγω τοξικών ουσιών για παράδειγμα φυτοφάρμακα[14].

Τέλος, ο μεγαλύτερος κίνδυνος που καλούνται οι μέλισσες να αντιμετωπίσουν αναφορικά με τη βιωσιμότητα τους, είναι η εκτεταμένη και αλόγιστη χρήση φυτοφαρμάκων. Τα εντομοκτόνα που χρησιμοποιούνται ιδιαίτερα κατά την περίοδο της ανθοφορίας από τους αγρότες για την παραγωγή των προϊόντων τους, επιδρούν αρνητικά στα άνθη και κατ'επέκταση τη γύρη τους, δηλητηριάζοντάς τα κατά κάποιον τρόπο, καθώς τα τοξικά φάρμακα αυτά μεταφέρονται και μέσω του αέρα. Ωστόσο, όταν το δηλητήριο που βρίσκεται στη γύρη έχει βραδεία δράση, παρατηρείται ότι η τροφή αυτή έχει ήδη μεταφερθεί στην κυψέλη πριν από το

θάνατο της συλλέκτριας, αποτελώντας τροφή και για τις υπόλοιπες, αλλά και για τους γόνους. Συνεπώς, η χρήση φυτοφαρμάκων έχει άμεση και έμμεση επίπτωση το θάνατο μιας ολόκληρης αποικίας[6][14].

1.6 Η μελισσοκομία στην Ελλάδα και την Ευρωπαϊκή Ένωση

Η Ελλάδα ανέκαθεν ασκούσε την μελισσοκομία. Εξαιτίας της ανάπτυξης του διεθνούς εμπορίου, το μέλι αποτελεί βασικό εξαγωγικό προϊόν με σπουδαία οικονομικά κέρδη. Η Μελισσοκομία αποτελεί έναν παραγωγικό κλάδο, με σημαντική επίδραση στην οικονομία της χώρας. Είναι, λοιπόν, εξαιρετικής σημασίας η ανάλυση στατιστικών στοιχείων, αναφορικά με την ελληνική και την παγκόσμια αγορά.

Στην Ελλάδα, η μελισσοκομία αποτελεί έναν κλάδο της γεωργίας με ιδιαίτερη δυναμική, που καλύπτει σχεδόν το 1% της ακαθάριστης αξίας της γεωργικής παραγωγής. Γι' αυτό και θεωρείται ένας από τους πιο σημαντικούς κλάδους του ελληνικού πρωτογενούς τομέα. Περίπου 25.000 άτομα ασχολούνται με την μελισσοκομία, με το μεγαλύτερο ποσοστό (90-95%) να ασχολούνται ερασιτεχνικά[15].

Σύμφωνα με στοιχεία του Υπουργείου Αγροτικής Ανάπτυξης και Τροφίμων, περίπου 23.000 μελισσοκόμοι με 1.380.000 κυψέλες ασχολούνται με την μελισσοκομία στη χώρα μας εκ των οποίων οι 5.000 κατέχουν άνω των 150 κυψελών και χαρακτηρίζονται ως είναι δεύτερη στην ευρωπαϊκή ένωση (μετά την Ισπανία) από απόψεως κατοχής μελισσοσμηνών και παράγει κατά μέσο όρο 14.000 τόνους μέλι ετησίως. Η εγχώρια παραγωγή καλύπτει περίπου το 90% της κατανάλωσης.

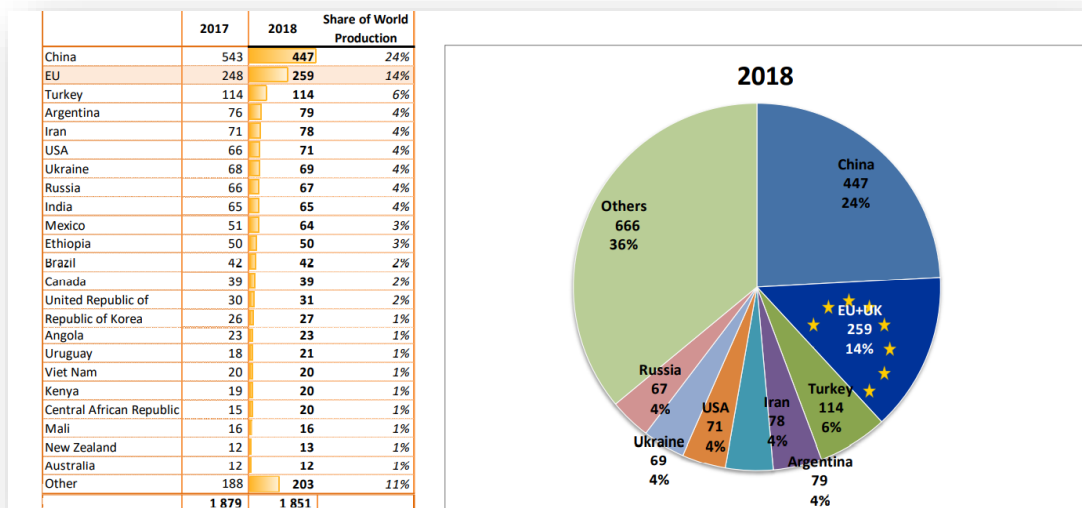
Από πλευράς γεωργικής κατανομής, η μελισσοκομία είναι διαδεδομένη σε όλη την Χώρα. Υπάρχουν όμως περιοχές που έχουν αυξημένο μελισσοκομικό ενδιαφέρον, όπως εκείνες των Νομών Χαλκιδικής Καβάλας, Φθιώτιδας, Εύβοιας, Αττικής, Αρκαδίας, Ηρακλείου, Χανίων και άλλες[15].

	Κράτος – Μέλος	Μελισσοκομικό Κεφάλαιο (αριθμός κυψελών)
1.	Βέλγιο	110.750
2.	Δανία	160.000
3.	Γερμανία	893.000
4.	Ελλάδα	1.388.000
5.	Ισπανία	2.464.601
6.	Γαλλία	1.150.000
7.	Ιρλανδία	20.000
8.	Ιταλία	1.100.000
9.	Λουξεμβούργο	11.077
10.	Κάτω Χώρες	80.000
11.	Αυστρία	327.000
12.	Πορτογαλία	590.000
13.	Φινλανδία	42.000
14.	Σουηδία	145.000
15.	Ηνωμένο Βασίλειο	274.000
16.	Τσεχία	477.743
17.	Κύπρος	45.714
18.	Λιθουανία	83.800
19.	Λετονία	54.173
20.	Ουγγαρία	872.650
21.	Μάλτα	1.938
22.	Πολωνία	949.200
23.	Σλοβενία	143.152
24.	Σλοβακία	192.002
25.	Εσθονία	50.500
	Σύνολο:	11.626.300

Εικόνα 1.2: Αριθμός κυψελών στην Ευρωπαϊκή Ένωση

Πηγή: Υπουργείο Αγροτικής Ανάπτυξης και Τροφίμων

Χρήσιμη είναι και μία μελέτη αναφορικά με το επίπεδο παραγωγής μελιού στην Ευρωπαϊκή Ένωση συγκριτικά με τον υπόλοιπο κόσμο, καθώς η Ευρωπαϊκή Ένωση έρχεται δεύτερη παγκοσμίως (παράγει 280.000 τόνους μελιού) μετά την Κίνα, όπως φαίνεται και στον παρακάτω πίνακα[16].



Εικόνα 1.3: Επίπεδο παραγωγής μελιού στην Ευρωπαϊκή Ένωση συγκριτικά με τον υπόλοιπο κόσμο

Πηγή: Honey Market Presentation of European Commission

Γίνεται, λοιπόν, κατανοητό ότι η μελισσοκομία αν και αποτελεί ένα μικρό κλάδο της γεωργικής παραγωγής έχει μεγάλη σημασία όχι μόνο για την Ελλάδα, αλλά και για όλο τον κόσμο. Αυτό αποδεικνύεται και αν αναλογιστεί κανείς το γεγονός ότι στην Ευρωπαϊκή Ένωση υπάρχουν περίπου 18.2 εκατομμύρια κυψέλες το 2019, παρουσιάζοντας περίπου 5.1% αύξηση συγκριτικά με το 2018[16].

Ωστόσο, είναι αξιοσημείωτο ότι η παραγωγή μελιού στην Ευρωπαϊκή Ένωση, δεν επαρκεί για αυτοκατανάλωση παρά μόνο για ένα ποσοστό της τάξεως του 60%. Το 2018 οι εισαγωγές της Ευρωπαϊκής Ένωσης ήταν περίπου 207.000 τόνοι (με αξία 451.000.000€) και οι εξαγωγές τη 21.000 τόνοι (με αξία 119.000.000€). Αν δει κανείς αυτά τα νούμερα συγκριτικά με το 2013, παρουσιάζεται αύξηση στις εισαγωγές κατά 25% και αύξηση στις εξαγωγές κατά 40%[16].

	2017		2018		Share of World Imports	
	Tons	1 000 €	Tons	1 000 €	Tons	1 000 €
European Union	208 317	457 218	207 376	451 085	38.5%	33.8%
North America	204 416	532 155	202 309	451 710	37.6%	33.8%
<i>of which USA</i>	202 565	503 411	197 867	421 395	36.7%	31.6%
Asia	72 134	319 544	62 333	211 508	11.6%	15.8%
<i>of which Japan</i>	42 821	126 573	44 521	123 068	8.3%	9.2%
<i>China</i>	5 682	80 896	3 824	59 362	0.7%	4.4%
Near and Middle Eastern Countries	33 582	112 904	31 827	116 741	5.9%	8.7%
Oceania and Polar Regions	8 713	41 777	15 588	45 500	2.9%	3.4%
Other European Countries	11 006	43 165	11 545	46 234	2.1%	3.5%
Africa	10 311	17 841	6 669	9 826	1.2%	0.7%
South America	840	2 205	610	1 562	0.1%	0.1%
Central America and Carribeans	912	2 374	272	701	0.1%	0.1%
Total	550 231	1 529 185	538 529	1 334 867		

Εικόνα 1.4: Παγκόσμιες εισαγωγές μελιού

Πηγή: Honey Market Presentation of European Commission

Σχεδιασμός και ανάπτυξη συστήματος αυτοματισμού για μελισσοκομικές εφαρμογές

	2017		2018		Share of World Imports	
	Tons	1 000 €	Tons	1 000 €	Tons	1 000 €
Asia	219 909	441 525	214 336	398 202	40.9%	28.7%
<i>of which China</i>	129 260	239 570	123 477	211 027	23.6%	15.2%
European Union	27 801	160 047	27 283	150 078	5.2%	10.8%
South America	111 774	306 635	111 411	261 019	21.3%	18.8%
Other European Countries	78 342	152 542	59 049	113 847	11.3%	8.2%
Central America and Carribeans	31 523	102 681	55 170	100 105	10.5%	7.2%
North America	27 190	76 945	26 691	73 315	5.1%	5.3%
<i>of which Canada</i>	19 462	53 632	18 837	51 807	3.6%	3.7%
<i>USA</i>	7 728	23 313	7 853	21 508	1.5%	1.5%
Oceania and Polar Regions	13 433	264 420	14 782	235 405	2.8%	17.0%
Near and Middle Eastern Countries	15 221	61 479	12 337	46 267	2.4%	3.3%
Africa	2 695	8 765	2 843	9 640	0.5%	0.7%
Total	527 888	1 575 040	523 901	1 387 878		

Εικόνα 1.5: Παγκόσμιες εξαγωγές μελιού

Πηγή: Honey Market Presentation of European Commission

Σε αυτό το σημείο, είναι ιδιαίτερα σημαντικό να αναφερθούν τόσο οι χώρες της Ευρωπαϊκής Ένωσης που αποτελούν τους «πρωταθλητές» εισαγωγών, όσο και οι κύριες χώρες από τις οποίες η Ευρωπαϊκή Ένωση εισάγει μέλι. Αρχικά, σύμφωνα με τα στοιχεία του 2018, η κυριότερη χώρα που εισάγει μέλι στην Ευρωπαϊκή Ένωση είναι η Γερμανία (60.000 τόνοι). Ακολουθούν, το Ηνωμένο Βασίλειο (45.000 τόνοι), το Βέλγιο (22.000 τόνοι), η Πολωνία (21.000 τόνοι) και τέλος η Ισπανία (17.000 τόνοι). Οι εισαγωγές αυτές γίνονται κατά κύριο λόγο από την Κίνα (80.000 τόνοι), την Ουκρανία (41.000 τόνοι), την Αργεντινή (25.000 τόνοι), το Μεξικό (21.000 τόνοι) και τέλος την Χιλή (8.000 τόνοι)[16].

	2017		2018		2019		Jan-Aug 2020		Compared to Jan-Aug
		% Extra EU		% Extra EU		% Extra EU		% Extra EU	
Ukraine	47 096	22.6%	40 964	19.7%	44 988	21.9%	31 343	28.5%	↑ +14.5%
China	78 600	37.7%	80 243	38.6%	79 201	38.6%	25 895	23.6%	↓ -49.7%
Argentina	24 826	11.9%	25 405	12.2%	21 828	10.6%	14 825	13.5%	↑ +6.7%
Mexico	20 857	10.0%	20 860	10.0%	21 543	10.5%	11 791	10.7%	↓ -11.4%
Uruguay	4 357	2.1%	4 550	2.2%	5 424	2.6%	6 554	6.0%	↑ +97.2%
Cuba	7 890	3.8%	4 974	2.4%	6 477	3.2%	5 074	4.6%	↑ +11.0%
Brazil	1 506	0.7%	4 587	2.2%	4 117	2.0%	4 248	3.9%	↑ +74.6%
Turkey	2 802	1.3%	3 482	1.7%	2 957	1.4%	2 206	2.0%	↑ +28.2%
Others	20 755	9.9%	22 734	10.9%	18 538	9.0%	7 874	7.2%	↓ -32.0%
Extra EU	208 687		207 799		205 074		109 808		↓ -15.3%
% Change	↑ +7.2%		↓ -0.4%		↓ -1.3%				

Εικόνα 1.6: Χώρα προέλευσης εισαγωγών της Ε.Ε

Πηγή: Honey Market Presentation of European Commission

Κεφάλαιο 2 : Ο μικροελεγκτής Arduino

2.1 Εισαγωγή

Το κεφάλαιο αυτό εισάγει τον αναγνώστη στον μικροελεγκτή Arduino ο οποίος και επιλέχθηκε να είναι ο ελεγκτής του συστήματος που αναπτύχθηκε. Ειδικότερα, γίνεται μία γενική περιγραφή του συγκεκριμένου ελεγκτή, παρουσιάζονται διάφορα μοντέλα και Shields του Arduino επισημαίνοντας τις δυνατότητές τους, αναλύεται το μοντέλο Arduino Uno, που έχει επιλεγεί για την δημιουργία του έργου, και γίνεται αναφορά στο λογισμικό καθώς και στα ολοκληρωμένα περιβάλλοντα ανάπτυξης που μπορούν να χρησιμοποιηθούν από την εκάστοτε πλακέτα Arduino.

2.2 Ιστορική αναδρομή Arduino

Το 2003 ο Hernando Barragan στα πλαίσια της μεταπτυχιακής διπλωματικής του εργασίας στο Interaction Design Institute Ivrea, υπό την επίβλεψη του Massimo Banzi και του Casey Reas, ανέπτυξε το σύστημα Wiring, ένα συνδυασμό μικροϋπολογιστικού συστήματος και κατάλληλου προγραμματιστικού περιβάλλοντος βασισμένου στη γλώσσα προγραμματισμού Processing που λίγα χρόνια νωρίτερα (το 2001) είχαν παρουσιάσει ο Ben Fry και ο Casey Reas[17]. Αξίζει να αναφερθεί ότι οι Ben Fry και ο Casey Reas, ήταν μαθητές του John Maeda και βάσισαν τη γλώσσα Processing στην γλώσσα προγραμματισμού Design By Numbers του καθηγητή τους[18].

Το 2005, εξαιτίας της ανάγκης των μαθητών για ένα λειτουργικό, φθηνότερο και πιο φιλικό στο χρήστη αναπτυξιακό σύστημα[19] άρχισε να δημιουργείται στην Ivrea της Ιταλίας, η πλατφόρμα Arduino, από τους Massimo Banzi, David Cuartielles, η οποία βασίστηκε τόσο στο σύστημα Wiring, όσο και στην γλώσσα προγραμματισμού Processing και κατά λογική συνέχεια από την γλώσσα προγραμματισμού Design By Numbers του John Maeda. Το όνομα Arduino λέγεται ότι προέκυψε από έναν βασιλιά της Ιταλίας του ενάτου αιώνα όπου κατοικούσε στην ίδια πόλη και σήμαινε " ισχυρός φίλος". Μία άλλη εκδοχή, αναφορικά με την

ονομασία του, είναι ότι προέρχεται από την ονομασία μιας γνωστής ρυθ που αποτέλεσε το στέκι της ομάδας του Banzi εκείνη την περίοδο[20].

2.3 Η πλακέτα Arduino

Το Arduino ορίζεται ως «μια ανοιχτού κώδικα ηλεκτρονική πλατφόρμα βασισμένη σε ένα ευκολόχρηστο Hardware και Software»[21]. Η πλακέτα Arduino είναι μία πλακέτα τυπωμένου κυκλώματος (PCB), με ενσωματωμένο μικροελεγκτή, και ένα αριθμό πυλών που χρησιμεύουν ως είσοδοι ή έξοδοι. Με άλλα λόγια, η πλακέτα Arduino είναι ένα μικρό κύκλωμα που περιέχει ένα ολοκληρωμένο σύστημα υπολογιστή χρησιμοποιώντας ένα μικρό chip (ολοκληρωμένο κύκλωμα) που είναι ο μικροελεγκτής[20]. Ο μικροελεγκτής (microcontroller) αποτελεί ένα είδος μικροεπεξεργαστή, ο οποίος εξαιτίας των διαφόρων ενσωματωμένων συστημάτων του, έχει τη δυνατότητα να λειτουργήσει με ελάχιστα εξωτερικά εξαρτήματα[22]. Με τον μικροελεγκτή παρέχεται η δυνατότητα στο χρήστη να προγραμματίσει το Arduino στο προγραμματιστικό περιβάλλον Arduino IDE (ο χρήστης έχει την δυνατότητα να το κατεβάσει δωρεάν από την επίσημη σελίδα του Arduino <https://www.arduino.cc/en/software>) κάνοντας χρήση της γλώσσας Wiring (η οποία μοιάζει με την C++).

2.4 Γιατί να επιλεγεί το Arduino

Το Arduino χρησιμοποιείται για εκατοντάδες διαφορετικές εφαρμογές, εξαιτίας της απλής και εύκολα προσβάσιμης εμπειρίας που προσφέρει, τόσο στους νέους χρήστες, όσο και στους πιο έμπειρους. Αρχικά, οι πλακέτες Arduino χαρακτηρίζονται από χαμηλό κόστος. Αυτό καθίσταται σαφές, ιδίως αν τις συγκρίνουμε με άλλες πλακέτες μικροελεγκτών. Το μέγιστο κόστος μίας έτοιμης (μονταρισμένης) πλακέτας Arduino είναι τα 50\$ (περίπου 41 ευρώ), ωστόσο το κόστος αυτό μπορεί να μειωθεί σημαντικά, καθώς παρέχεται η δυνατότητα στο χρήστη να αγοράσει μη συναρμολογημένο το Arduino. Ένα δεύτερο πλεονέκτημα της πλακέτας Arduino, είναι ότι τρέχει σε διάφορα λειτουργικά συστήματα, όπως για παράδειγμα σε Windows, Linux και Macintosh OSX, σε αντίθεση με

ανταγωνιστικούς μικροελεγκτές, που συνήθως περιορίζονται στα Windows. Επιπλέον, το Arduino χαρακτηρίζεται από ένα απλό και ξεκάθαρο προγραμματιστικό περιβάλλον, το οποίο ενδείκνυται για αρχάριους, παραμένοντας ταυτόχρονα και ευέλικτο για την εξυπηρέτηση των πιο έμπειρων χρηστών. Επιπρόσθετα, αξίζει να αναφερθεί, ότι το software του Arduino είναι ανοιχτού λογισμικού και διατίθεται προς επέκταση με τη χρήση των γλωσσών C++ και AVR C. Τέλος, το Arduino επιτρέπει στους χρήστες τόσο να αντιγράψουν την κατασκευή της πλακέτας προκειμένου να την κατανοήσουν, όσο και να την εξελίξουν ή να την αναπτύξουν, χωρίς να κινδυνεύουν από ενδεχόμενες νομικές κυρώσεις[23]. Αυτό έχει ως αποτέλεσμα πολλές εταιρίες να εκμεταλλευτούν αυτή την κατάσταση και να δημιουργούν απομιμήσεις των πλακετών αυτών προσφέροντας ακόμα πιο ανταγωνιστικές τιμές στους χρήστες του Arduino.

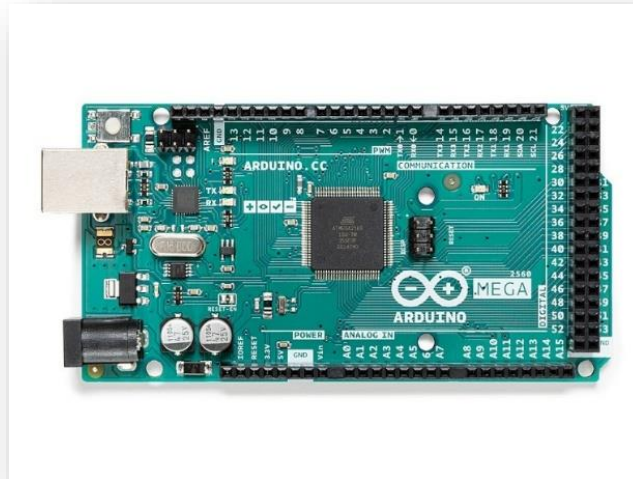
2.5 Μοντέλα Arduino

Όπως προαναφέρθηκε το Arduino είναι μία πλακέτα ανοιχτού τύπου. Είναι αξιοσημείωτο το γεγονός ότι υπάρχουν περισσότερα του ενός μοντέλα Arduino με διαφορετικά τεχνικά χαρακτηριστικά, δυνατότητες και μεγέθη, δίνοντας έτσι στον χρήστη ποικίλες δυνατότητες στην υλοποίηση των διαφόρων έργων του. Στην συνέχεια θα αναλυθούν ορισμένα από αυτά τα μοντέλα προκειμένου να καταστούν κατανοητές οι διαφοροποιήσεις τους.

Αρχικά, μοντέλο αναφοράς αποτελεί το Arduino Uno, το οποίο θα αναλυθεί εκτενέστερα στο επόμενο υποκεφάλαιο, καθώς αυτό χρησιμοποιείται στην εκπόνηση της παρούσας εργασίας. Στην συνέχεια, γίνεται αναφορά μερικών μοντέλων Arduino που κατατάσσονται ως τα πιο δημοφιλέστερα σύμφωνα με την επίσημη σελίδα Arduino Store[24].

- Arduino MEGA 2560 Rev3 : Ο μικροελεγκτής που βρίσκεται στον Arduino Mega είναι ο ATmega2560. Προσφέρει 54 ψηφιακές εισόδους /εξόδους από τις οποίες οι 15 παρέχονται και για PWM(Pulse Width Modulation) έξοδο, 16 αναλογικές εισόδους, 4 σειριακές συνδέσεις με ηλεκτρονικό υπολογιστή ή άλλα Arduino (UARTs) και ρολόι στα 16MHz. Ο λόγος χρήσης του είναι ότι παρέχει μεγαλύτερη μνήμη και περισσότερες ψηφιακές και αναλογικές ακίδες

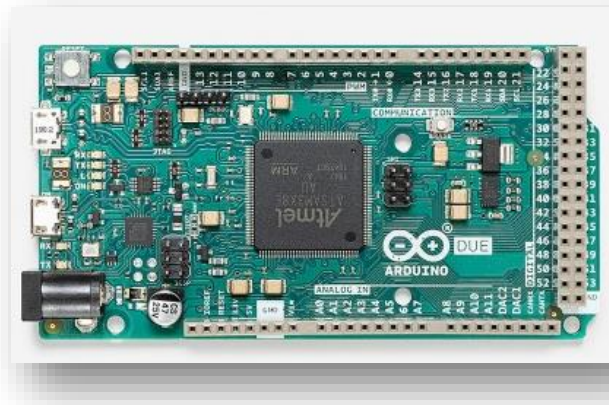
σύνδεσης από τα περισσότερα μοντέλα Arduino. Εν κατακλείδι, παρέχει την δυνατότητα στον χρήστη να συνδέσει μεγάλο αριθμό αισθητήρων καθώς και φόρτωση μεγάλων βιβλιοθηκών και προγραμμάτων[25].



Εικόνα 2.1: ARDUINO MEGA 2560 REV3´

Πηγή : <https://store.arduino.cc/arduino-mega-2560-rev3>

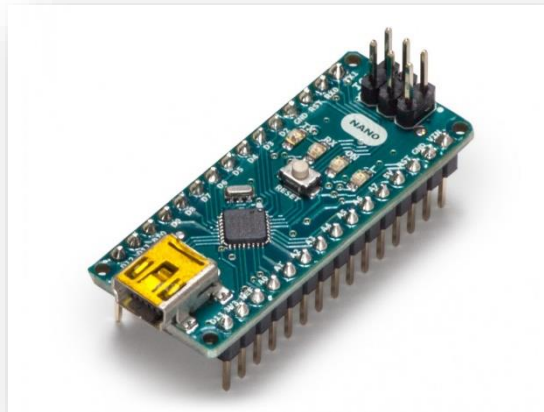
- Arduino Due : Ο μικροελεγκτής που διαθέτει η παρούσα συσκευή είναι ο AT91SAM3X8E και είναι η πρώτη πλακέτα Arduino που έχει 32-bit ARM μικροελεγκτή, εκτοξεύοντας την απόδοση, όσο αναφορά την επεξεργασία των δεδομένων, στα ύψη. Προσφέρει 54 ψηφιακές εισόδους και εξόδους, από τις οποίες οι 12 μπορούν να μετατραπούν σε PWM έξοδοι, 12 αναλογικές εισόδους, 4 σειριακές διασυνδέσεις, ρολόι στα 84 MHz καθώς και 2 αναλογικούς εξόδους (DAC). Το αγκάθι του Arduino Due είναι ότι η τάση λειτουργίας του βρίσκεται στα 3.3V, δηλαδή η μέγιστη τάση που μπορεί να διαχειριστεί από τους ακροδέκτες, δίχως να προκαλεστούν προβλήματα στην συσκευή, είναι τα 3.3V. Το παρόν μοντέλο λοιπόν, ενδείκνυται για την δημιουργία ενός μεγάλου έργου, δίνοντας στον χρήστη μέγιστη απόδοση σε αυτό, αλλά και περιορισμό στην σύνδεση των διαφόρων περιφερειακών συσκευών, λόγω της τάσης λειτουργίας της συσκευής[26].



Εικόνα 2.7: ARDUINO DUE

Πηγή : <https://store.arduino.cc/arduino-due>

- Arduino Nano : Αποτελεί ίσως, την δημοφιλέστερη επιλογή για την δημιουργία έργων, που έχουν ως γνώμονα την ελαχιστοποίηση του μεγέθους του project, δίχως όμως να ελαχιστοποιηθούν οι δυνατότητες της συσκευής καθώς και η φαντασία του χρήστη. Διαθέτει τον μικροελεγκτή ATmega328, 22 ψηφιακούς ακροδέκτες εισόδου/εξόδου (6 από τις οποίες είναι PWM) , 8 αναλογικοί είσοδοι και ρολόι των 16 MHz[27].



Εικόνα 2.3: ARDUINO NANO

Πηγή: <https://store.arduino.cc/arduino-nano>

Arduino Leonardo : Το μέγεθος του συμπίπτει με αυτό της πλακέτας, Arduino Uno, δείχνοντας με πρώτη ματιά ότι δεν διαφοροποιείται τόσο από την συγκεκριμένη πλακέτα. Παρόλα αυτά το Arduino

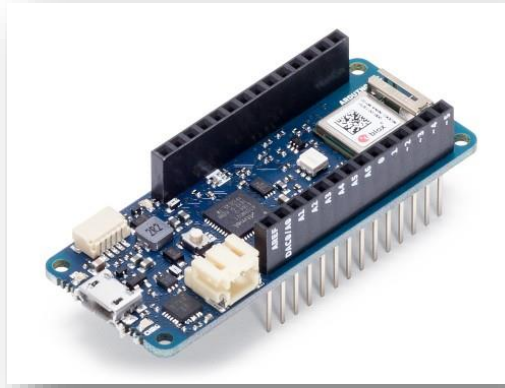
Leonardo διαθέτει τον μικροελεγκτή ATmega32u4 ο οποίος διαθέτει μια ενσωματωμένη USB διεπαφή, με αποτέλεσμα, για παράδειγμα, να μπορεί να αναγνωριστεί ως πληκτρολόγιο ή ποντίκι στον υπολογιστή. Επιπρόσθετα έχει 20 ψηφιακούς ακροδέκτες εισόδου/εξόδου από τους οποίους τα 7 μπορούν να χρησιμοποιηθούν ως PWM έξοδοι, 12 αναλογικά πινάκια και ρολόι ταχύτητας των 16MHz[28].



Εικόνα 2.4: ARDUINO LEONARDO WITH HEADERS

Πηγή: <https://store.arduino.cc/arduino-leonardo-with-headers>

- Arduino MKR Wifi 1010 : Το συγκεκριμένο μοντέλο αποτελεί μία λύση σταθμό για τους χρήστες που επιθυμούν να σχεδιάσουν εφαρμογές με διεπαφή στο διαδίκτυο, αφού παρέχει δυνατότητες σύνδεσης Wifi και Bluetooth. Ο κύριος επεξεργαστής της συσκευής είναι ο χαμηλής κατανάλωσης Arm Cortex-M0 32-bit SAMD21, ενώ υπεύθυνος για τις συνδέσεις Bluetooth και Wifi είναι το chip NINA-W102 που βρίσκεται στο κάτω μέρος της συσκευής, το οποίο λειτουργεί στα 2.4GHz. Δεν θα μπορούσε να παραληφθεί η ανάγκη για ασφάλεια της επικοινωνίας, η οποία διασφαλίζεται από το Microchip ECC508. Επιπρόσθετα, παρέχονται 8 ψηφιακοί ακροδέκτες εισόδου/εξόδου, 13 ακροδέκτες PWM εξόδων, 7 αναλογικούς ακροδέκτες εισόδου των 8/10/12 bit αντίστοιχα και ένα αναλογικό ακροδέκτη εξόδου[29].



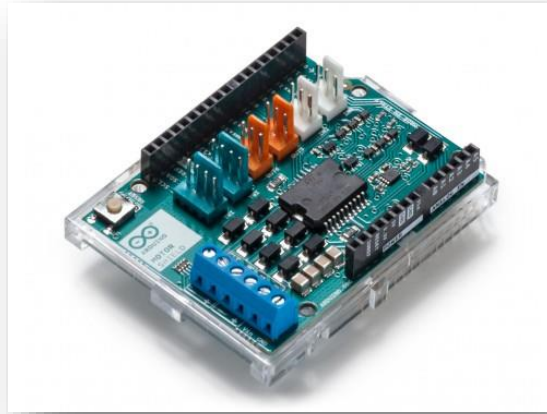
Εικόνα 2.8: ARDUINO MKR WIFI 1010

Πηγή: <https://store.arduino.cc/arduino-mkr-wifi-1010>

2.6 Arduino Shields

Έχοντας γίνει μία μικρή αναφορά στα διαφορετικών ειδών μοντέλα του Arduino, είναι οφθαλμοφανές πως το Arduino επιθυμεί να καλύψει ακόμα και τους πιο απαιτητικούς χρήστες, βάζοντας όριο μόνο στην φαντασία αυτών και όχι στις δυνατότητες του υλικού. Έχοντας αυτό ως γνώμονα και οδηγό, το Arduino δημιουργεί τα Shields. Τα Shields είναι πλακέτες που κουμπώνουν επάνω από τις πλακέτες του Arduino. Αυτό έχει ως αποτέλεσμα να επεκτείνονται οι δυνατότητες του εκάστοτε μοντέλου δίνοντας νέα μορφή στο έργο του χρήστη. Στην συνέχεια γίνεται μια μικρή δειγματοληπτική αναφορά σε διαφορετικών ειδών Shields.

- **Arduino Motor Shield Rev3:** Το συγκεκριμένο Shield είναι πλήρως συμβατό με το Arduino Uno. Κύριο χαρακτηριστικό του είναι το L298, το οποίο έχει σχεδιαστεί για να οδηγεί επαγωγικά φορτία, όπως για παράδειγμα ρελέ, κινητήρες συνεχούς ρεύματος (Direct Current motor) κ.α. Επιπρόσθετα, επιτρέπει στον χρήστη να ελέγχει εύκολα την ταχύτητα και την κατεύθυνση του κινητήρα, ενώ προσφέρεται και η δυνατότητα χρήσης δύο κινητήρων. Ο χρήστης μπορεί με ασφάλεια, που του παρέχεται από την πλακέτα, μέσω εξωτερικής τροφοδοσίας, να τροφοδοτεί έως 12V και 2A στον εκάστοτε κινητήρα[30].



Εικόνα 2.6: ARDUINO MOTOR SHIELD REV3

Πηγή:<https://store.arduino.cc/arduino-motor-shield-rev3?queryID=06d5df445f8f153ba2611e6c49aee229>

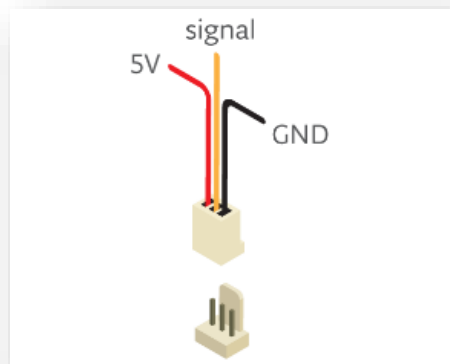
- Arduino Ethernet Shield 2: Το εικονιζόμενο Shield της εικόνας παρέχει, στην πλακέτα που θα συνδεθεί, την δυνατότητα πλοήγησή της στο διαδίκτυο μέσω ενός καλωδίου δικτύου RJ45. Επίσης, ο χρήστης μπορεί να διαβάσει και να αποθηκεύσει δεδομένα στην SD κάρτα. Ο Ethernet controller, ένα ολοκληρωμένο κύκλωμα τσιπ που ελέγχει τις επικοινωνίες Ethernet, προσφέρει 32K buffer, ενώ η ταχύτητα δικτύου μπορεί να φτάσει τα 100Mb[31].



Εικόνα 2.7: Arduino Ethernet Shield 2

Πηγή: <https://www.arduino.cc/en/Guide/ArduinoEthernetShield#connecting-the-shield>

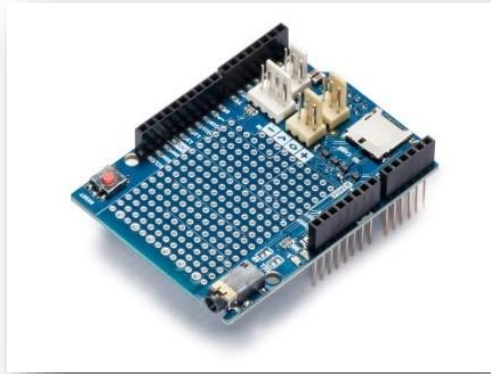
- **Arduino Education Shield:** Το συγκεκριμένο Shield, όπως αναφέρεται και από την ονομασία του, ενδείκνυται για εκπαιδευτικούς σκοπούς, επιτρέποντας την γρήγορη, ευχάριστη και ευκολότερη εκμάθηση κατά την κατασκευή έργων. Το Arduino Education Shield είναι συμβατό με το Arduino Uno με τάση στα 5V, όπως και με το Arduino 101 με τάση στα 3.3V. Μερικές από τις παροχές της πλακέτας είναι η δυνατότητα στον χρήστη να ενσωματώσει ένα breadboard και να γίνουν οι συνδέσεις πάνω σε αυτό, καθώς και μία υποδοχή ηχείου. Επιπρόσθετα, διαθέτει θύρα 3 ακροδεκτών(3-pin header port) που επιτρέπει την άμεση και εύκολη σύνδεση, όπως για παράδειγμα ενός servo.



Εικόνα 2.9: 3-pin header port

Πηγή: <https://store.arduino.cc/arduino-education-shield?queryID=09182a65688ec4496eae53e7fbb4ad3a>

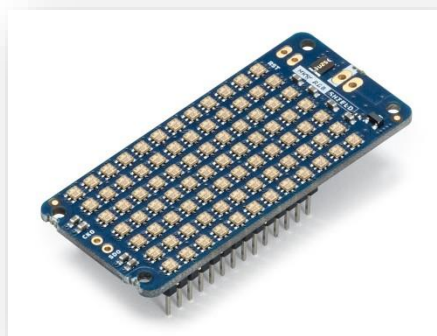
Τέλος, διαθέτει υποδοχή για τοποθέτηση κάρτας SD, που ο χρήστης μπορεί να αποθηκεύσει ή να διαβάσει δεδομένα, εκτοξεύοντας τις δυνατότητες του έργου του[32].



Εικόνα 2.8: Arduino Education Shield

Πηγή:<https://store.arduino.cc/arduino-education-shield?queryID=09182a65688ec4496eae53e7fbb4ad3a>

- Arduino MKR RGB Shield: Σε περίπτωση που ο χρήστης θέλει να δώσει έμφαση στην εμφάνιση φωτεινών μηνυμάτων, τότε το Arduino MKR RGB Shield είναι η λύση. Ο χρήστης έχει 84 RGB Led στην διάθεσή του, όπου μπορεί να παράγει όποιο μήνυμα επιθυμεί και σε όποιο χρώμα επιθυμεί, αφού τα RGB διαθέτουν τα 3 βασικά χρώματα (μπλε, κόκκινο, πράσινο). Το μόνο που χρειάζεται να κάνει ο χρήστης είναι να τοποθετήσει το Shield με ένα μοντέλο της σειράς MKR, δίχως να χρειάζεται κάποιον αντάπτορα ή συγκόλληση[33].



Εικόνα 2.10: ARDUINO MKR RGB SHIELD

Πηγή:<https://store.arduino.cc/arduino-mkr-rgb-shield?queryID=2985bccb93c329d2d169fac35369bafb>

2.7 Arduino Uno γενικά χαρακτηριστικά

Η συγκεκριμένη πλακέτα αποτελεί ίσως το πιο δημοφιλές προϊόν της σειράς Arduino. Η λέξη «Uno» στα ιταλικά σημαίνει ένα. Η ονομασία αυτή δόθηκε στην συγκεκριμένη πλακέτα, καθώς αποτέλεσε την πρώτη και βασική έκδοση Arduino. Το Arduino Uno αποτελεί σημείο αναφοράς, καθώς όλα τα υπόλοιπα μοντέλα βασίζονται στην δική του φιλοσοφία. Η συγκεκριμένη πλακέτα ενδείκνυται κυρίως για αρχάριους χρήστες καθώς είναι ιδιαίτερα εύχρηστη, με δυνατότητες ωστόσο που καλύπτουν ένα ευρύ φάσμα απαιτήσεων.



Εικόνα 2.11: Arduino Uno

Πηγή:<https://www.controlvoltage.net/arduino-arduino-uno-r3-development-board.html>

Οι δυνατότητες αυτές αποκαλύπτονται αν εξετάσει κανείς τα τεχνικά χαρακτηριστικά της πλακέτας.

Όπως φαίνεται και στον παρακάτω πίνακα, το Arduino Uno εμπεριέχει ένα μικροελεγκτή Atmel ATmega328P με τάση λειτουργίας τα 5V, προσφέροντας 14 ψηφιακούς ακροδέκτες, εκ των οποίων οι 6 μπορούν να προσφέρουν έξοδο PWM (Pulse Width Modulation). Επιπρόσθετα, διαθέτει 6 αναλογικούς ακροδέκτες και ταχύτητα χρονισμού στα 16MHz δίνοντας μία αρκετά ικανοποιητική ταχύτητα στην εκτέλεση των προγραμμάτων που είναι αποθηκευμένα στην 32KB flash μνήμη[21].

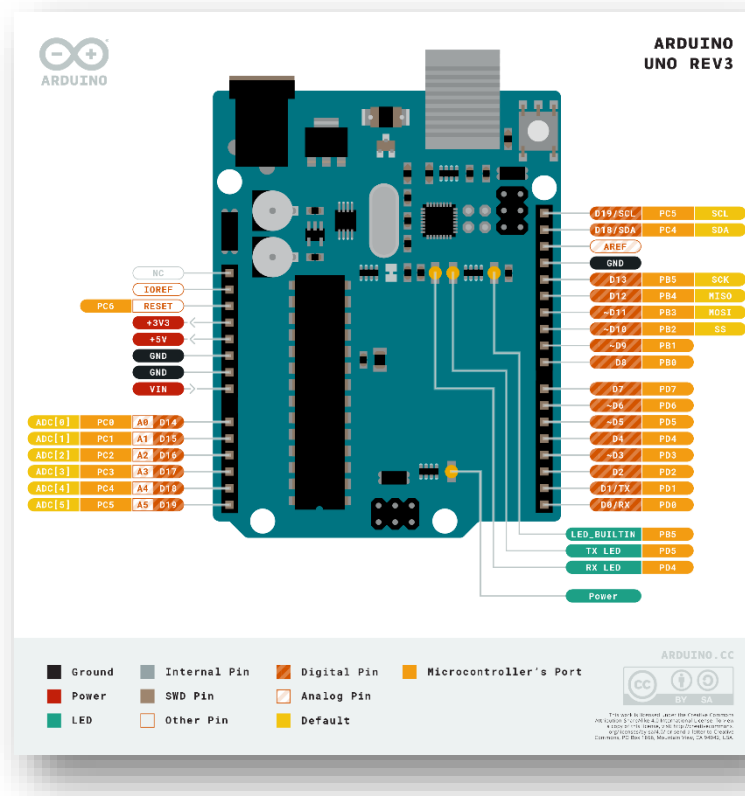
ΜΟΝΤΕΛΑ	ARDUINO UNO
ΜΙΚΡΟΕΛΕΓΚΤΗΣ	ATmega328P
ΤΑΣΗ ΛΕΙΤΟΥΡΓΙΑΣ	5V
ΤΑΣΗ ΕΙΣΟΔΟΥ	7-12V
ΟΡΙΑ ΤΑΣΗΣ	6-20V
ΨΗΦΙΑΚΟΙ ΑΚΡΟΔΕΚΤΕΣ ΕΙΣΟΔΟΥ/ ΕΞΟΔΟΥ	14(από τα οποία 6 παρέχουν έξοδο PWM)
ΚΑΝΑΛΙΑ PWM	6
ΑΝΑΛΟΓΙΚΟΙ ΑΚΡΟΔΕΚΤΕΣ ΕΙΣΟΔΟΥ	6
DC ΡΕΥΜΑ ΑΝΑ ΕΙΣΟΔΟ/ ΕΞΟΔΟ ΑΚΡΟΔΕΚΤΗ	40mA
DC ΡΕΥΜΑ ΓΙΑ 3.3 V ΑΚΡΟΔΕΚΤΗ	50mA
ΜΝΗΜΗ FLASH	32KB
SRAM	2KB
EEPROM	1KB
ΤΑΧΥΤΗΤΑ ΡΟΛΟΓΙΟΥ	16 MHz

Πίνακας 2.1: Χαρακτηριστικά Arduino Uno

Πηγή: <http://www.arduino.cc/>

2.7.1 Ακροδέκτες

Το Arduino Uno διαθέτει 20 ακροδέκτες (pins), εκ των οποίων οι 6 είναι αναλογικοί και οι 14 ψηφιακοί. Οι αναλογικοί ακροδέκτες εντοπίζονται στην κάτω πλευρά της πλακέτας με σήμανση ANALOG IN και είναι αριθμημένοι από το 0 έως το 5. Οι ακροδέκτες αυτοί χρησιμεύουν αποκλειστικά ως είσοδοι, όπου καθένας από αυτούς εμπεριέχει μετατροπέα A/D (Analog to Digital) με ανάλυση 10bit. Αυτό έχει ως αποτέλεσμα να δημιουργείται ένα ευρύ φάσμα διαφορετικών τιμών, της τάξεως του 2^{10} . Αυτές οι τιμές διαφοροποιούνται ανάλογα με τις αυξομειώσεις της διεύρυνσης του ρεύματος, από 0V μέχρι 5V, που παρέχεται στην εκάστοτε ακίδα[34].



Εικόνα 2.12: ARDUINO UNO REV 3

Πηγή <https://store.arduino.cc/arduino-uno-rev3-smd>

Επιπρόσθετα δίνεται η δυνατότητα στον χρήστη να διαφοροποιήσει το εύρος του ρεύματος (2V-5V) τροφοδοτώντας το pin με την επιθυμητή για αυτόν τάση χρησιμοποιώντας τον ακροδέκτη AREF (Analogue REFerence) και την συνάρτηση `analogReference()`[35]. Πάνω από τον ακροδέκτη που μόλις αναφέρθηκε, αναγράφονται άλλα δύο πινάκια το SCL και SDA που είναι υπεύθυνα για την χρήση του σειριακού δίαυλου επικοινωνίας I^2C . Με την χρήση λοιπόν, του πρωτοκόλλου επικοινωνίας αυτού, απαιτείται η σύνδεση δύο καλωδίων (εκτός από τους αγωγούς τροφοδοσίας των συσκευών) μεταξύ της κύριας συσκευής (master) και της υποτελής συσκευής (slave). Το ένα καλώδιο είναι υπεύθυνο για τον χρονισμό (SCL), ενώ το δεύτερο αποτελεί την γραμμή δεδομένων (SDA). Για την υλοποίηση του συγκεκριμένου πρωτόκολλου επικοινωνίας είναι απαραίτητη η χρήση της βιβλιοθήκης `Wire` που γίνεται μέσω του λογισμικού[36]. Άξιο προσοχής είναι πως οι ακροδέκτες SDA και SCL, μπορούν να αντικατασταθούν από τους αναλογικούς ακροδέκτες A4 και A5 αντίστοιχα, λόγω του ότι είναι εσωτερικά συνδεδεμένοι[37].

Οι ψηφιακές υποδοχές της πλακέτας είναι 14 και χρησιμεύουν τόσο ως είσοδοι αλλά και ως έξοδοι, μερικές από τις οποίες διαφοροποιούνται ανάλογα με την λειτουργία που διαπράττουν. Υπεύθυνοι να την σειριακή επικοινωνία της πλακέτας είναι οι ακροδέκτες 0 (RX) και 1 (TX) που είναι επιφορτισμένοι με την λήψη και την εκπομπή σειριακών δεδομένων αντίστοιχα. Συνεχίζοντας, οι ακροδέκτες 2 και 3, έχουν την δυνατότητα να λειτουργήσουν ως εξωτερικές διακοπές (interrupt 0 και 1 αντίστοιχα), δηλαδή προκειμένου να επιτευχθεί ακριβής συγχρονισμός, διακόπτεται η ροή του προγράμματος εκτελώντας ένα συγκεκριμένο μέρος του προγράμματος ή μια συνάρτηση. Οι ακροδέκτες 3,5,6,9,10 και 11 παρουσιάζονται ως ψευδοαναλογικές έξοδοι, δηλαδή έχουν την δυνατότητα να επιλεχθούν για την λειτουργία Pulse Width Modulation (PWM), δίνοντας πλήθος τιμών στην έξοδο της τάξεως 2^8 . Με άλλα λόγια παρέχονται 256 καταστάσεις, όπου στη κατάσταση 0, η τάση εξόδου είναι μηδενική, ενώ αντίθετα στην κατάσταση 255 η τάση εξόδου είναι +5V. Στο υπόλοιπο φάσμα τιμών εξάγονται τετραγωνικοί παλμοί που εναλλάσσονται με μεγάλη συχνότητα και για ίσους χρόνους μεταξύ των τιμών 0 και 5V. Επιπρόσθετα αξίζει να αναφερθεί πως το Arduino Uno διαθέτει τρεις Timers. Τον Timer 0 των 8 bit, που ελέγχει τους ακροδέκτες 5,6 και χρησιμοποιείται με το κάλεσμα των συναρτήσεων `millis()`, `delay()` και `micros()`, τον Timer 1 των 16 bit, που ελέγχει τους ακροδέκτες 9,10 και γίνεται χρήση αυτού με την βιβλιοθήκη `Servo`, και τέλος τον Timer 2 των 8 bit που ελέγχει τους 3,11 ακροδέκτες και την συνάρτηση `tone()`. Άξιο προσοχής είναι πως όταν χρησιμοποιούνται οι προαναφερθέντες συναρτήσεις και η `Servo library`, των timer 1 και 2 αντίστοιχα, δεν είναι δυνατή η χρήση των συγκεκριμένων ακροδεκτών σε λειτουργία PWM[38].

Παρόλα αυτά οι ακροδέκτες 10,11,12 και 13 χρησιμοποιούνται και για το πρωτόκολλο συγχρονισμένης σειριακής μεταφοράς δεδομένων SPI (Serial Peripheral Interface) της συσκευής, που επικοινωνεί με μία ή και περισσότερες περιφερειακές συσκευές άμεσα. Με την χρήση του συγκεκριμένου πρωτοκόλλου δίνεται η δυνατότητα να επικοινωνούν ακόμα και δύο μικροελεγκτές. Στο συγκεκριμένο πρωτόκολλο επικοινωνίας η μία, κύρια συσκευή καλείται master (που συνήθως είναι μικροελεγκτής) και όλες οι υπόλοιπες συσκευές χαρακτηρίζονται ως slave. Ακόμα και στο πρωτόκολλο SPI, ο κάθε ακροδέκτης επιτελεί μία συγκεκριμένη λειτουργία. Ο ακροδέκτης 11 (αποκαλείται και ως Master

Out Slave In (MOSI), χρησιμοποιείται από την κύρια συσκευή (master), ώστε να αποσταλούν δεδομένα στις περιφερειακές συσκευές (slaves). Αντίστοιχα, το pin 12 (Master In Slave Out (MISO)), είναι υπεύθυνο ώστε να αποστέλλονται δεδομένα από τις περιφερειακές συσκευές στην κύρια. Ο συγχρονισμός των συσκευών επιτυγχάνεται από το ρολόι της κύριας συσκευής, στο πινάκι 13 (Serial Clock (SCK)). Τέλος, με τον ακροδέκτη 10 (Slave Select (SS)) η κύρια συσκευή επιλέγει ποια περιφερειακή συσκευή θα τεθεί σε λειτουργία[39].

2.7.2 Τροφοδοσία

Από την αρχή της δημιουργίας της συσκευής, οι κατασκευαστές είχαν ως γνώμονα την παροχή εύκολων λύσεων, άσχετα από την εκάστοτε εμπειρία των χρηστών. Αυτό ακολουθείτε και στον τομέα της τροφοδοσίας. Αναφορικά, λοιπόν, με την τροφοδοσία του Arduino Uno, υπάρχει η δυνατότητα παροχής ρεύματος μέσω της θύρας USB, καθώς και μέσω εξωτερικής Πηγής ενέργειας, η οποία παρέχεται αφενός με την χρήση μπαταριών και αφετέρου με τη χρήση ενός μετασχηματιστή (DC: προσαρμογέας εναλλασσόμενου ρεύματος). Η σύνδεση του μετασχηματιστή στην πλακέτα συντελείται στην κάτω γωνία, δίπλα στην υποδοχή του USB μέσω μιας υποδοχής φισ των 2.1mm (θετικός πόλος στο κέντρο). Σε περίπτωση που η εξωτερική τροφοδοσία συντελεστεί με μπαταρία, η μπαταρία θα συνδεθεί στην υποδοχή GND και VIN. Η εξωτερική τροφοδοσία που παρέχεται πρέπει να κυμαίνεται μεταξύ των 7-12 V έτσι ώστε να μην εμφανίζονται προβλήματα μη επαρκούς τροφοδότησης στους ακροδέκτες εξόδου ή υπερθέρμανσης του σταθεροποιητή τάσης της πλακέτας αντίστοιχα[20].

Στο Arduino Uno οι ακροδέκτες που σχετίζονται με την τροφοδοσία βρίσκονται πάνω από τους αναλογικούς ακροδέκτες και δίπλα από την επιγραφή «POWER». Ο ακροδέκτης με την ένδειξη «VIN» είναι υπεύθυνος για την τροφοδοσία του συστήματος από εξωτερική Πηγή. Στην συνέχεια,, αναγράφεται το πινάκι «5V» όπου χρησιμοποιείται για την τροφοδοσία των περιφερειακών συσκευών που θα βρίσκονται σε σύνδεση με την συσκευή και θα τροφοδοτούνται με σταθερή τάση ρεύματος στα 5V. Κατά λογική συνέχεια η ακίδα «3,3V» συμπίπτει με την προαναφερθείσα, μόνο που η τάση εξόδου είναι στα 3,3V και ενδείκνυται για συσκευές χαμηλής τάσης τροφοδοσίας. Ο ακροδέκτης «GND» ,δηλαδή

«Ground», έχει το ρόλο της γείωσης, ενώ ο ακροδέκτης «Reset» ευθύνεται για την επανεκκίνηση της συσκευής, όταν γειωθεί. Τέλος, εμφανίζεται ο ακροδέκτης «IOREF» όπου προδίδει την αναφορά τάσης με την οποία λειτουργεί ο μικροελεγκτής εκτός των 5V[36].

2.7.3 Μικροελεγκτής (microcontroller)

Ο μικροελεγκτής, που θα μπορούσε να χαρακτηριστεί ως ο εγκέφαλος στο Arduino Uno είναι ο ATmega328P της Atmel, όπως έχει ήδη προαναφερθεί. Ο μικροελεγκτής, λοιπόν, είναι το μακρόστενο μαύρο στοιχείο που είναι τοποθετημένο στο κέντρο της πλακέτας και ο ρόλος του είναι να κατέχει τον μεταγλωττισμένο κώδικα, καθώς και να εκτελεί τις εντολές που έχει δημιουργήσει ο χρήστης μέσω αυτού. Με την χρήση της γλώσσας προγραμματισμού Wiring δίνεται η δυνατότητα στον χρήστη να έχει πρόσβαση στα περιφερειακά του μικροελεγκτή, όπως τους ακροδέκτες εισόδου εξόδου, τους μετατροπείς αναλογικού σε ψηφιακού (A/C), πρωτόκολλα επικοινωνίας (I2C και SPI) και σειριακές διεπαφές. Κάθε μικρή ακίδα του μικροελεγκτή συνδέεται περιμετρικά στην πλακέτα του Arduino και προσφέρει τους ακροδέκτες της συσκευής, που έχουν ήδη αναφερθεί, με αποτέλεσμα ο χρήστης να μπορεί να συνδέσει ή να αφαιρέσει από την αυτή τους αισθητήρες ή τα περιφερειακά της συσκευής εύκολα και γρήγορα[40].



Εικόνα 2.13: ATmega328P

Πηγή: J. Blum, (2013)

2.7.4 Μνήμη

Ο μικροελεγκτής της πλακέτας Arduino Uno είναι οπλισμένος με ενσωματωμένη μνήμη τριών τύπων. Αρχικά, διαθέτει μνήμη Flash των 32KB όπου τα 0.5KB κατέχουν το bootloader της συσκευής, που απαιτείται για την εγκατάσταση των προγραμμάτων των χρηστών μέσω της θύρας USB. Τα 31,5KB είναι υπεύθυνα για την αποθήκευση των προγραμμάτων αυτών που θα εκτελεστούν, τα οποία διατηρούνται ακόμα και με την επανεκκίνηση της συσκευής, καθώς και με την απώλεια τροφοδοσίας της. Στην συνέχεια, έχει σειρά η μνήμη SRAM (Static random-access memory), η οποία χρησιμοποιείται για την προσωρινή αποθήκευση των μεταβλητών που δημιουργούνται ή διαχειρίζονται από την εκτέλεση ενός προγράμματος. Αναφέρθηκε ο όρος «προσωρινή αποθήκευση» καθώς αν σταματήσει η τροφοδοσία της συσκευής ή πραγματοποιηθεί reset, τότε η μνήμη χάνει τα δεδομένα της. Τέλος, ο τύπος μνήμης EEPROM (Electrically Erasable Programmable Read-Only Memory) έχει μέγεθος 1KB και υπάρχει για την διατήρηση των τιμών που δημιουργούνται από την εκτέλεση των προγραμμάτων του χρήστη[36].

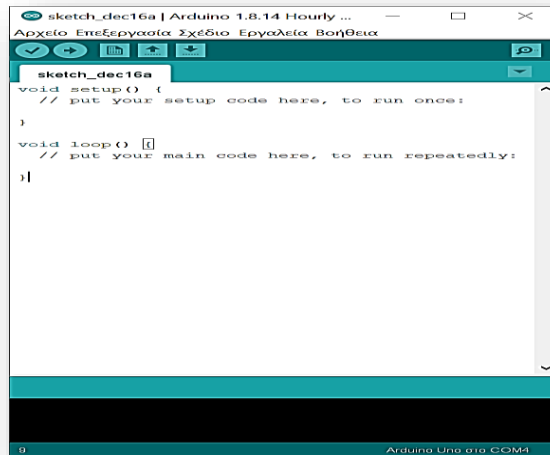
2.8 Λογισμικό Arduino

Το λογισμικό του Arduino είναι το πρόγραμμα που ευθύνεται για την κωδικοποίηση και την μεταφορά του κώδικα στο υλικό. Μετατρέπει τον κώδικα σε μία γλώσσα κατανοητή για τον μικροελεγκτή, παρέχοντας έμμεσα στον χρήστη μία κατανοητή γλώσσα σε αυτόν, την Wiring. Στην συνέχεια, γίνεται αναφορά σε τρία διαφορετικά ολοκληρωμένα περιβάλλοντα ανάπτυξης[41].

2.8.1 Arduino IDE

Το ολοκληρωμένο περιβάλλον ανάπτυξης ή αλλιώς IDE(Integrated Development Environment) του Arduino αποτελεί και την επίσημη εφαρμογή της εταιρίας, που αποτελεί την γέφυρα μεταξύ της εκάστοτε πλακέτας με τον υπολογιστή του χρήστη. Το Arduino IDE έχει βασιστεί στην γλώσσα προγραμματισμού Processing[21], προσφέροντας έτσι μεγάλη συμβατότητα στον μεγαλύτερο αριθμό των λειτουργικών συστημάτων, όπως Windows, Linux και Mac OS X. Το κάθε πρόγραμμα συγγραφής, ή αλλιώς sketch, που καλείται να συντάξει ο χρήστης γράφεται στη γλώσσα προγραμματισμού Wiring. Όταν γίνει ανέβασμα

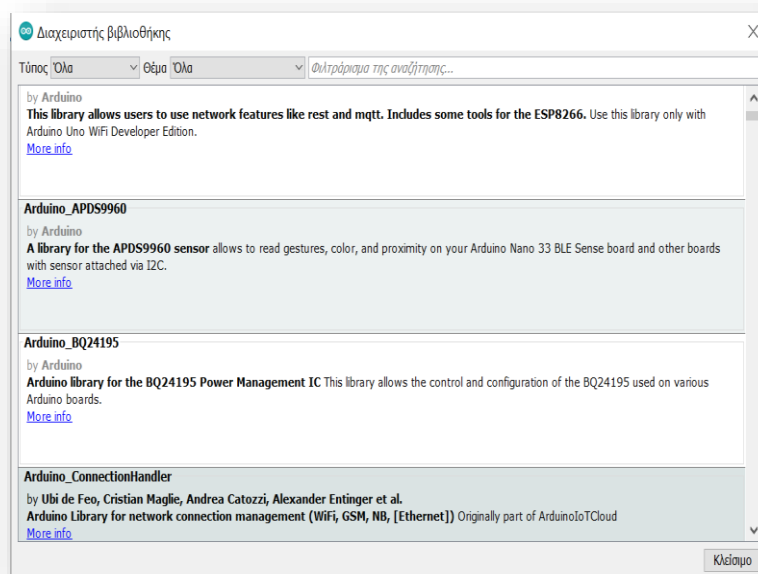
του sketch στην εκάστοτε πλακέτα Arduino, ο κώδικας μετατρέπεται σε γλώσσα προγραμματισμού C, όπου στην συνέχεια ο μετατροπέας Avr-gcc δίνει την τελική μορφή του κώδικα, η οποία είναι κατανοητή από τον μικροελεγκτή[42]. Αξίζει να αναφερθεί πως το συγκεκριμένο ολοκληρωμένο περιβάλλον ανάπτυξης παρέχεται δωρεάν στους χρήστες, οι οποίοι μπορούν να το κατεβάσουν από την επίσημη σελίδα του Arduino.



Εικόνα 2.9: Σύνταξη sketch στο Arduino IDE

Το Arduino IDE, λοιπόν, προσφέρει ένα πολύ φιλικό περιβάλλον στον χρήστη καθώς και έτοιμα παραδείγματα όπου μπορούν να χρησιμοποιηθούν στην πλακέτα που είναι συνδεδεμένη. Μέσω του serial monitor παρακολουθείται η σειριακή επικοινωνία μεταξύ της συσκευής με τον υπολογιστή, όπου δίνεται η δυνατότητα αποστολής και λήψης μηνυμάτων, τον εντοπισμό σφαλμάτων καθώς και την εμφάνιση δεδομένων με την κατάλληλη εντολή στον κώδικα που έχει συνταχθεί. Επιπρόσθετα, προσφέρονται έτοιμες βιβλιοθήκες, δίνοντας την δυνατότητα για την εύρυθμη και ολοκληρωμένη λειτουργία των εκάστοτε εξαρτημάτων που θα επιλέξει ο χρήστης να συνδέσει στην πλακέτα του. Οι βιβλιοθήκες που αλληλεπιδρούν με το Arduino είναι αρχεία κώδικα γραμμένα σε γλώσσα προγραμματισμού C ή C++ και εισέρχονται στην αρχή του sketch. Αυτό έχει ως αποτέλεσμα, αφενός να προσφέρεται στον χρήστη μία πολύ θετική ώθηση στο να διαχειριστεί εύκολα και γρήγορα τα εξαρτήματα που συνδέονται με το Arduino, και αφετέρου να προσφέρεται μία επεκτασιμότητα στην γλώσσα προγραμματισμού. Στο Arduino IDE υπάρχει η επιλογή «διαχείριση πλακετών» και

«διαχείριση βιβλιοθηκών» όπου δίνεται στον χρήστη η δυνατότητα να εγκαταστήσει νέα πακέτα λογισμικού ή βιβλιοθηκών του Arduino αντίστοιχα, αναβαθμίζοντας και εν συνεχεία βελτιώνοντας τις ήδη υπάρχουσες δυνατότητες, ή ακόμα παρατηρείται και η δημιουργία νέων λειτουργιών και δυνατοτήτων[36]. Παρόλα αυτά ο χρήστης έχει την δυνατότητα να συντάξει και να προσθέσει την δική του βιβλιοθήκη, μέσω του Arduino IDE, μεγαλώνοντας το φάσμα των δυνατοτήτων του έργου του. Μάλιστα, η επίσημη σελίδα του Arduino, ωθεί τον χρήστη στην δημιουργία της δικιάς του βιβλιοθήκης προσφέροντας μία δοκιμαστική που μπορεί να κατεβάσει δωρεάν, καθώς και οδηγίες για την εγκατάσταση και την σωστή συγγραφή της[43].

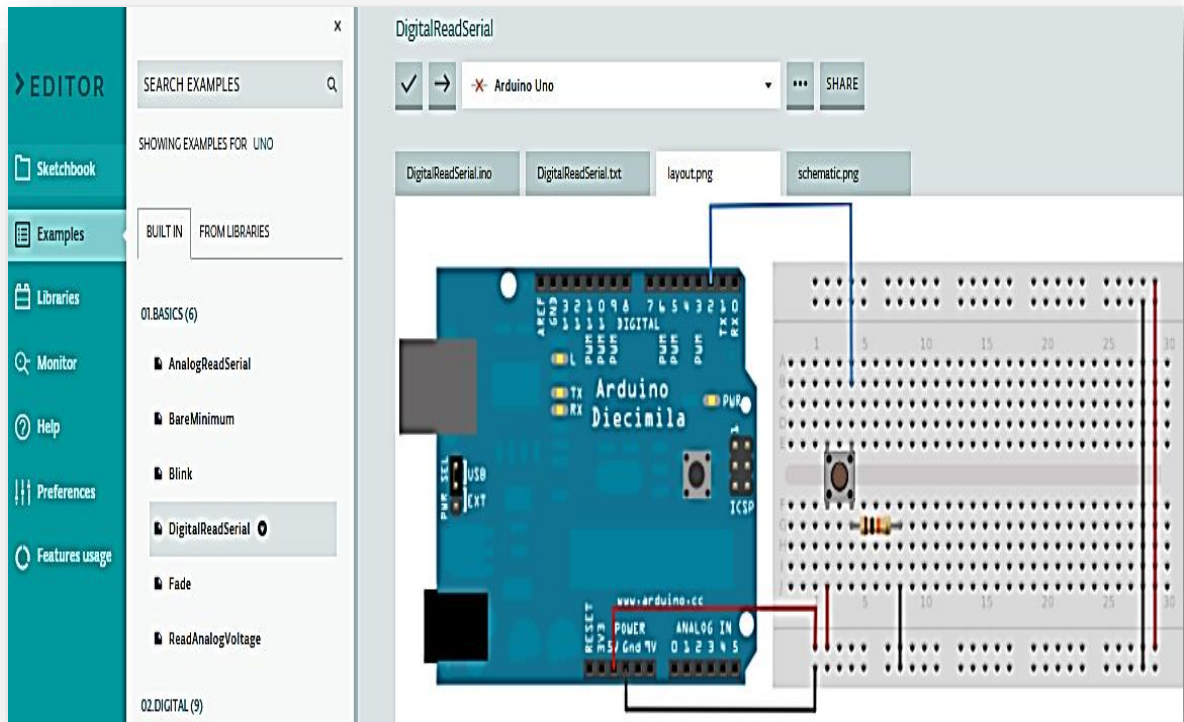


Εικόνα 2.15: Διαχειριστής βιβλιοθηκών στο Arduino IDE

2.8.2 Arduino Web Editor

Ο καθησυχασμός της εταιρίας δεν σταματά στο Arduino IDE. Θέλοντας να βοηθήσει και να καλύψει ακόμα και τους πιο απαιτητικούς χρήστες, δημιουργεί ένα παρόμοιο ολοκληρωμένο περιβάλλον ανάπτυξης, που φέρει το όνομα Arduino Web Editor. Αυτό δίνει την δυνατότητα στον χρήστη να μπορεί να δημιουργήσει τον κώδικα που επιθυμεί και στην συνέχεια να ανεβάσει το πρόγραμμά του σε οποιαδήποτε συσκευή Arduino έχει συνδέσει στον υπολογιστή, μέσω ενός

προγράμματος περιήγησης (browser). Οι browsers που είναι συμβατοί είναι ο Firefox, το Chrome, το Safari και ο Edge, ενώ η εταιρία προτείνει να χρησιμοποιηθεί ο Chrome. Εφόσον ο Arduino Web Editor φιλοξενείται στο διαδίκτυο, γίνεται προσβάσιμος 24 ώρες την μέρα από οποιαδήποτε συσκευή μπορεί να συνδεθεί στο internet, με μία συνεχή αναβάθμιση των ήδη υπαρχόντων χαρακτηριστικών, καθώς και την άμεση υποστήριξη νέων πλακετών. Το μόνο που απαιτείται από τον χρήστη είναι να δημιουργήσει έναν λογαριασμό, δίνοντας ένα e-mail και password, και στην συνέχεια να εγκαταστήσει δωρεάν το Arduino Web Editor plugin, το οποίο δίνει την δυνατότητα να ανεβαίνουν τα sketches μέσω του προγράμματος περιήγησης στην εκάστοτε πλακέτα. Στην συνέχεια, εφόσον ο χρήστης δημιουργήσει τα δικά του sketches, αποθηκεύονται αυτόματα στον λογαριασμό που πραγματοποιήθηκε η εγγραφή, ενώ μπορεί να τα κατηγοριοποιήσει σε φακέλους για καλύτερη και πιο οργανωμένη πρόσβαση. Το σημείο που βρίσκονται αποθηκευμένα είναι το Sketchbook, που βρίσκεται αριστερά στην σελίδα, ενώ στην συνέχεια εμφανίζονται τα παραδείγματα (Examples), με διάφορες κατηγορίες επ'αυτών, που εκτός από τον έτοιμο κώδικα διαθέτουν ένα txt αρχείο που αναφέρει τι αποτέλεσμα έχει το συγκεκριμένο παράδειγμα, το διάγραμμα συνδεσμολογίας που παρέχει τεράστια βοήθεια στα πρώτα βήματα ενός χρηστή, καθώς και μία εικόνα παρουσίασης του κυκλώματος.



Εικόνα 2.10: DigitalReadSerial- Arduino Web Editor

Πηγή: <https://create.arduino.cc/editor/>

Μία μεγάλη διευκόλυνση που παρέχει το Arduino Web Editor είναι οι βιβλιοθήκες (Libraries), όπου παρέχονται στον χρήστη, δίχως ανάγκη εγκατάστασή τους. Επίσης, μέσω του Library Manager, δίνεται η δυνατότητα να ξεχωρίσει ο χρήστης όποιες επιθυμεί, και να χαρακτηριστούν ως αγαπημένες. Σε περίπτωση που ο χρήστης θελήσει να μοιραστεί το πρόγραμμά του, ο παραλήπτης το μόνο που έχει να κάνει είναι να αντιγράψει τον κώδικα στο Sketchbook, δίχως να σπαταλήσει χρόνο στην εγκατάσταση των βιβλιοθηκών, αφού βρίσκονται πάντα διαθέσιμες. Ωστόσο, αν ο χρήστης δημιουργήσει ένα δικό του library, παρέχεται η δυνατότητα εισαγωγής (import) αυτής, καλύπτοντας έτσι όλες τις περιπτώσεις που τυχόν θα αντιμετωπίσει ο εκάστοτε χρήστης.

Η σειριακή οθόνη (Serial Monitor) που εμφανίζεται στην συνέχεια, επικοινωνεί σειριακά με την πλακέτα που έχει συνδέσει ο χρήστης. Όταν έχουν συνδεθεί παραπάνω από δύο συσκευές, τότε ο χρήστης δύναται να επιλέξει με ποια θα κάνει χρήση της σειριακής επικοινωνίας. Τα δεδομένα που διοχετεύει η συσκευή μέσω του Serial Monitor, εμφανίζονται ακριβώς δεξιά από την επιλογή «Monitor», όπως

και η δυνατότητα αποστολής εντολής (Send) καθώς και η συχνότητα της σειριακής (baud).



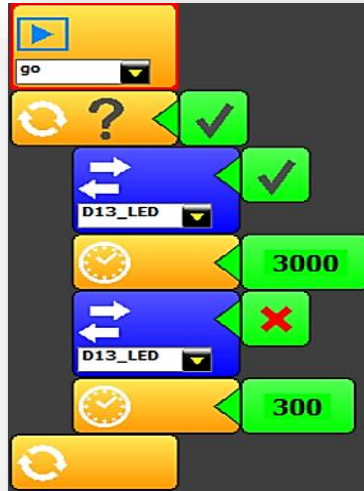
Εικόνα 2.11: Εμφάνιση δεδομένων στο Serial Monitor

Πηγή: https://create.arduino.cc/projecthub/Arduino_Genuino/getting-started-with-arduino-web-editor-on-various-platforms-4b3e4a

Μέσω της επιλογής «Help» εμφανίζονται χρήσιμες πληροφορίες για το Arduino, καθώς και υπερσυνδέσμους με βίντεο που προσφέρουν ένα εύκολο δρόμο για τα πρώτα βήματα του χρήστη στον τομέα αυτό. Στην επιλογή «Preference» παρέχονται κάποιες ρυθμίσεις, όπως για παράδειγμα αλλαγή θέματος του Editor, ενώ στο «Features usage» εμφανίζεται ο χώρος αποθήκευσης που μπορεί να κάνει ο χρήστης, ο αριθμός των προγραμμάτων που δεν πρέπει να ξεπερνά τους 100, καθώς και το όριο των 200 δευτερολέπτων την μέρα στην μεταγλώττιση του Sketch (για την δωρεάν έκδοση)[44].

2.8.3 MiniBlox

Είναι γεγονός πως η τεχνολογία κάνει αλματώδη πρόοδο στις μέρες μας, δημιουργώντας την ανάγκη για περισσότερα άτομα να ασχοληθούν έμπρακτα για την περαιτέρω ανάπτυξή της στο μέλλον. Με γνώμονα αυτό το γεγονός, αποκαλύπτεται η ανάγκη για δημιουργία σταθερών βάσεων στο προγραμματισμό, που θα λαμβάνει χώρα στην πρωτοβάθμια και δευτεροβάθμια εκπαίδευση. Όμως, σε τέτοιες ηλικίες ο προγραμματισμός φαντάζει ένα πολύ δύσκολο κομμάτι στην εκπαίδευση, κάνοντάς τον μία αρκετά δύσκολη πρόκληση, αφενός στον εκπαιδευτή και αφετέρου στους μαθητές. Την λύση, όσο αναφορά για τα πρώτα βήματα στην ρομποτική, στο Arduino, καθώς και σε άλλες πλατφόρμες, έρχεται να προσφέρει το Miniblox. Το Miniblox είναι ένα ολοκληρωμένο περιβάλλον γραφικού προγραμματισμού που αναπτύχθηκε από τον Julián da Silva Gillig, κάνοντας χρήση της γλώσσας C++ και την βιβλιοθήκη αυτής, την wxWidgets. Τα λειτουργικά συστήματα που μπορούν να εκτελέσουν το Miniblox είναι τα Windows και τα Linux, ενώ η τελευταία έκδοση v0.83 προσφέρεται μόνο για τα Windows, καθώς είναι υπό κατασκευή για τα Linux. Τα κομμάτια γραφικού κώδικα, δηλαδή τα block, έχουν διαμορφωθεί σε XML και στην εμφάνισή τους θα μπορούσαν να χαρακτηριστούν ως κομμάτια ενός παζλ, τα οποία αν ενωθούν σωστά, θα βγει και ένα αποτέλεσμα, ή αλλιώς ένα πρόγραμμα[45][46]. Στην [Εικόνα 2.17] εμφανίζεται ένα πρόγραμμα με χρήση των blocks, που έχει ως αποτέλεσμα να ανάβει το Led στην θέση 13 του Arduino για 3 δευτερόλεπτα και στην συνέχεια να παραμένει σβηστό για 0,3 δευτερόλεπτα.



Εικόνα 2.17: Πρόγραμμα στο MiniBlox με χρήση blocks

Στην [Εικόνα 2.18] εμφανίζεται ο κώδικας που δημιουργείται με την χρήση των blocks, αφού το MiniBlox παρέχει αυτή την δυνατότητα στον χρήστη.

```
#include <mbq.h>
#include <PingIRReceiver.h>

void setup()
{
  initBoard();
  while(true)
  {
    digitalWrite(D13_LED, true);
    delay(3000);
    digitalWrite(D13_LED, false);
    delay(300);
  }
}

void loop()
{
}
```

Εικόνα 2.18: Πρόγραμμα στο MiniBlox με χρήση κώδικα

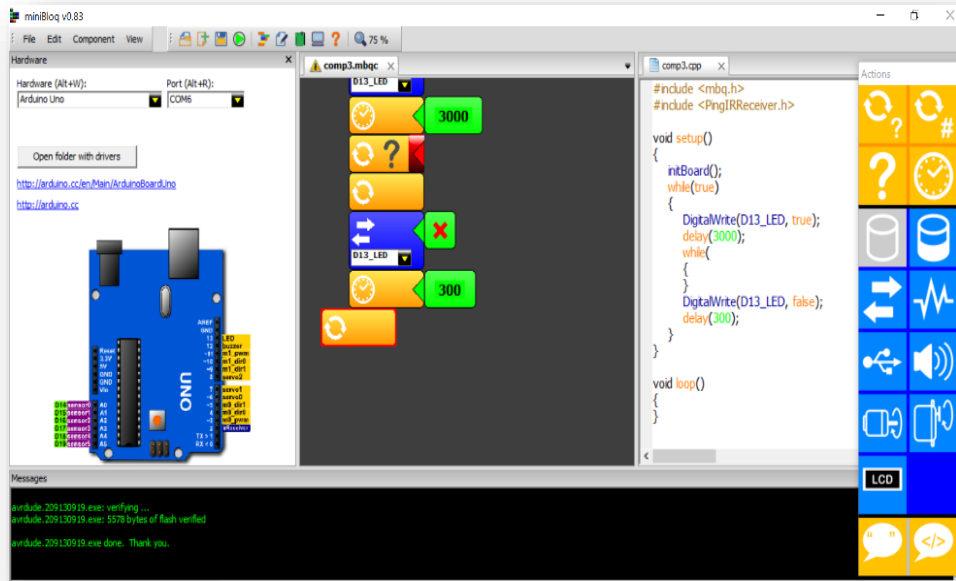
Με το άνοιγμα του συγκεκριμένου ολοκληρωμένου περιβάλλοντος ανάπτυξης εμφανίζεται στον χρήστη ένα πλαίσιο στην δεξιά πλευρά της οθόνης με το όνομα «Actions» [Εικόνα 16], όπου παρέχεται η δυνατότητα επιλογής των blocks που επιθυμεί.



Εικόνα 2.16: Actions

Παρουσιάζεται μια μεγάλη ποικιλία στα αυτοματοποιημένα κομμάτια κώδικα καλύπτοντας αν όχι όλες, ένα μεγάλο μέρος των απαιτήσεων του χρήστη. Στην επίσημη σελίδα εμφανίζεται η λεπτομερής ανάλυση του εκάστοτε block καθώς και παραδείγματα που ωθούν τον χρήστη στη δημιουργία των πρώτων του προγραμμάτων[47].

Σχεδιασμός και ανάπτυξη συστήματος αυτοματισμού για μελισσοκομικές εφαρμογές



Εικόνα 2.19: Περιβάλλον MiniBlox

Κεφάλαιο 3: Σύστημα αυτοματοποιημένου ελέγχου μελισσιών

3.1 Εισαγωγή

Στο συγκεκριμένο κεφάλαιο παρουσιάζεται το αυτοματοποιημένο σύστημα που αναπτύχθηκε στο πλαίσιο της συγκεκριμένης μεταπτυχιακής διατριβής, για τον απομακρυσμένο έλεγχο βαθμίδων μελισσοκομίας.

3.2 Σύνοψη συστήματος

Με την πάροδο των αιώνων, η τεχνολογία αναπτύσσεται με ραγδαίους ρυθμούς, ώστε να διευκολυνθεί όσο γίνεται περισσότερο η ανθρώπινη καθημερινότητα. Είναι κοινώς αποδεκτό, ότι με την τεχνολογική πρόοδο, ο άνθρωπος επιδιώκει να δώσει λύσεις σε ένα ευρύ φάσμα προβλημάτων που τον ταλανίζουν. Όπως προαναφέρθηκε, σε προηγούμενο κεφάλαιο, στην Ελλάδα, η μελισσοκομία αποτελεί έναν κλάδο της γεωργίας με ιδιαίτερη δυναμική, που καλύπτει σχεδόν το 1% της ακαθάριστης αξίας της γεωργικής παραγωγής.

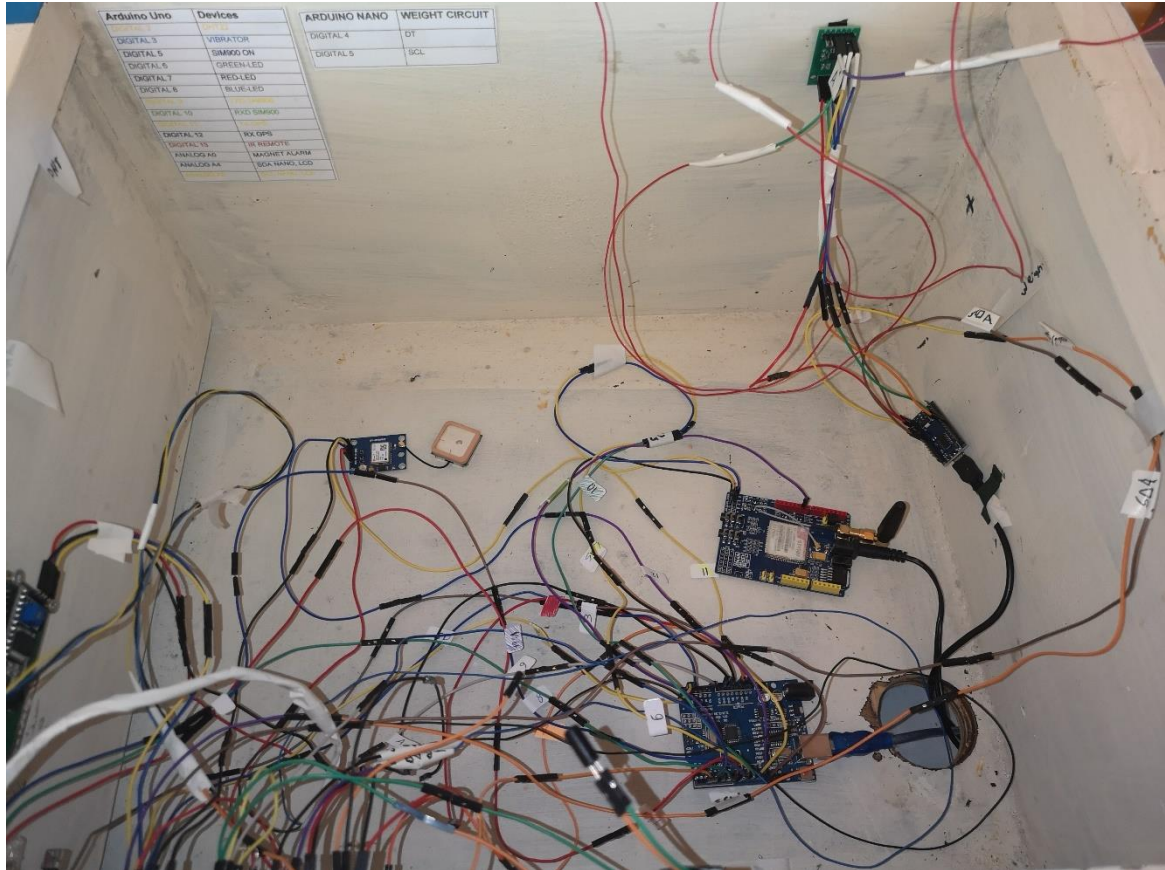
Στην παρούσα διπλωματική εργασία, λοιπόν, θα επικεντρωθούμε στον πρωτογενή τομέα και συγκεκριμένα στο τομέα της μελισσοκομίας, καθώς είναι εξαιρετικής σημασίας να διευθετηθούν, με τη χρήση της τεχνολογίας, σοβαρά ζητήματα που ανακύπτουν στην καθημερινότητα ενός μελισσοκόμου. Είναι γεγονός, ότι αρκετά συχνά κρίνεται απαραίτητη η μεταφορά των μελισσιών σε πιο απομακρυσμένες περιοχές, όπως για παράδειγμα δάση, προκειμένου να παραχθεί μέλι. Αυτό έχει ως άμεση συνέπεια, την αδυναμία του παραγωγού για εύκολη πρόσβαση στα μελίσσια. Επομένως, ο παραγωγός αντιμετωπίζει δυσχέρειες στο να γνωρίζει καθημερινά το μέγεθος του παραγόμενου προϊόντος, το πληθυσμό της κυψέλης, την θερμοκρασία και την υγρασία του περιβάλλοντα χώρου κλπ. Ταυτόχρονα, αναδύεται ο κίνδυνος κλοπής της κυψέλης.

Προκειμένου να επιλυθούν τα προαναφερθέντα ζητήματα, αναπτύχθηκε ένα αυτοματοποιημένο σύστημα απομακρυσμένου ελέγχου της κατάστασης μελισσιών, που διαθέτει αισθητήρες, ώστε να δίνει την δυνατότητα στον παραγωγό να ελέγχει τόσο την εσωτερική κατάσταση των μελισσιών όσο και τον περιβάλλοντα χώρο εξ'αποστάσεως. Η συγκεκριμένη εφαρμογή απαρτίζεται από αισθητήρες βάρους, θερμοκρασίας και υγρασίας. Ταυτόχρονα υπάρχουν αισθητήρες (δόνησης και

μαγνητικού συναγερμού) που είναι αρμόδιοι για τον οπλισμό του συναγερμού, προκειμένου να καταστεί δυνατή η ασφάλεια της κυψέλης. Εφόσον παραβιαστεί η κυψέλη αρχικά θα πραγματοποιείται κλήση στο κινητό του χρήστη και εν συνεχεία μέσω του GPS θα υπάρχει η δυνατότητα εντοπισμού της. Τέλος θα πραγματοποιείται έλεγχος του βάρους μέσα στην ημέρα, όπου θα ειδοποιείται ο χρήστης μέσω μηνύματος, όταν το βάρος υπερβεί κάποιο υψηλό ή χαμηλό όριο.



Εικόνα 3.1 : Ολοκληρωμένο σύστημα (εξωτερικά)



Εικόνα 3.2 : Ολοκληρωμένο σύστημα (εσωτερικά)

3.3 Ανάλυση της δομής και της λειτουργίας του συστήματος

Όπως αναφέρθηκε προηγουμένως, στο συγκεκριμένο σύστημα χρησιμοποιούνται διάφοροι αισθητήρες, οι οποίοι πρόκειται να αναλυθούν εκτενέστερα.

Για την δημιουργία του προαναφερθέντος πειράματος χρησιμοποιούνται:

- Arduino Uno
- Arduino Nano
- Lcd1602 οθόνη
- I2C Serial Interface Module
- Αισθητήρες βάρους

- Ενισχυτής αισθητήρα βάρους (HX711)
- GSM Module SIM900
- GPS Module
- IR Wireless Remote Control
- Αισθητήρας δόνησης
- Μαγνητικός αισθητήρας συναγερμού «παγίδα»
- Αισθητήρας θερμοκρασίας και υγρασίας (DHT22 module)
- RGB Led
- Bread Board
- Καλώδια
- Αντιστάσεις

3.3.1 Arduino Uno

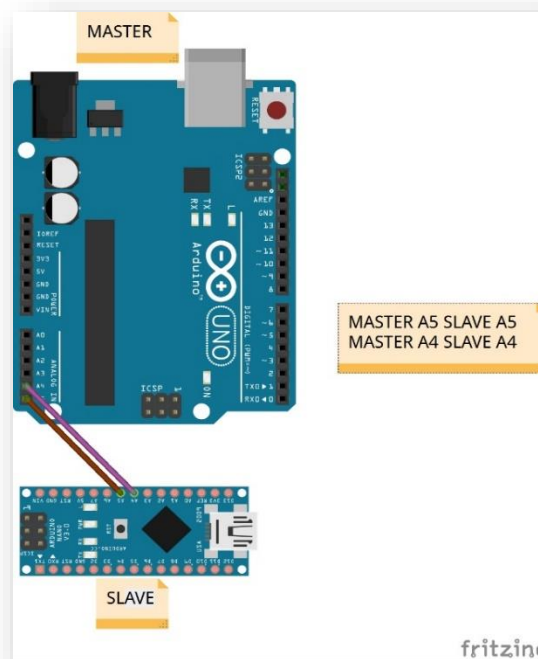
Το Arduino Uno θα μπορούσε να παρομοιαστεί ως ο «εγκέφαλος» της εφαρμογής, καθώς είναι επιφορτισμένος με τη διαχείριση όλων των υπολοίπων συσκευών, όπως για παράδειγμα το Arduino Nano, οι διάφοροι αισθητήρες κτλ. Πιο συγκεκριμένα, θα είναι υπεύθυνο για την σωστή λειτουργία της συσκευής, την αποστολή εντολών στο υπόλοιπο σύστημα και την επεξεργασία των τιμών των διαφόρων αισθητήρων. Τέλος θα κατέχει το ρόλο του «Master» στην παρούσα εφαρμογή.

3.3.2 Arduino Nano

Εξαιτίας των μεγάλων απαιτήσεων της εφαρμογής κρίθηκε αναγκαία η χρήση ενός «Slave» που θα είναι αρμόδιο για την ενεργοποίηση του κυκλώματος του βάρους, εφόσον ζητηθεί από τον Master. Αρμόδιο για αυτή τη διαδικασία είναι το Arduino Nano και η επιλογή του έγινε με γνώμονα αφενός το μικρό του μέγεθος και αφετέρου το ελάχιστο χρηματικό του κόστος.

3.3.2.1 Συνδεσμολογία Arduino Uno & Nano

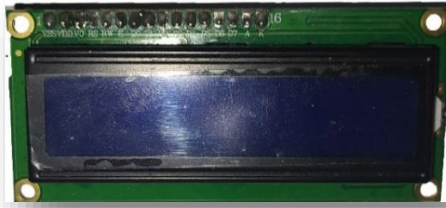
Η σύνδεση μεταξύ των δύο αυτών συσκευών γίνεται μέσω του πρωτοκόλλου επικοινωνίας I2C (Inter-Integrated Circuit). Για την επίτευξη αυτής της επικοινωνίας είναι απαραίτητα δύο καλώδια. Το πρώτο ονομάζεται SDA (Serial Data), είναι υπεύθυνο για την μεταφορά των δεδομένων μεταξύ του Master και του Slave και βρίσκεται στην θέση A4(Analog 4) στην εκάστοτε συσκευή, ενώ το δεύτερο καλώδιο χρησιμοποιείται για τον συγχρονισμό των συσκευών, φέρει το όνομα SCL (Serial Clock) και βρίσκεται στο πινάκι A5 (Analog 5)[48].



Εικόνα 3.3 : Μία βαθμίδα Arduino Uno και μια βαθμίδα Nano I2C Protocol

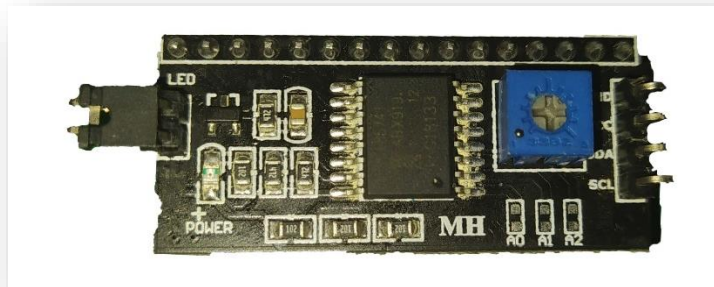
3.3.3 Lcd 1602 & I2C Serial Interface

Είναι προφανές πως κρίνεται αναγκαία η τοποθέτηση μιας οθόνης Lcd, ώστε να επιτυγχάνεται η ενημέρωση του χρήστη μέσω κατάλληλων μηνυμάτων. Στην παρούσα εφαρμογή γίνεται εμφάνιση των μετρήσεων του βάρους, της θερμοκρασίας, της υγρασίας. Ταυτόχρονα μέσω μηνύματος γίνεται ξεκάθαρο στο χρήστη, αν ο συναγερμός είναι οπλισμένος ή όχι. Η συγκεκριμένη οθόνη είναι τύπου 1602 δηλαδή ,μπορεί να εμφανίσει 16 χαρακτήρες και έχει δύο γραμμές για την παρουσίαση αυτών.



Εικόνα 3.4 : Lcd 1602

Λόγω της πολυπλοκότητας της σύνδεσης της οθόνης αρκετές φορές γίνεται η ένωσή της με έναν I2C Serial Interface, αναγκάζοντάς την να μετατραπεί σε Slave αφού γίνεται η χρήση του I2C πρωτοκόλλου. Έτσι δίνεται η δυνατότητα στον χρήστη να την συνδέσει με 4 μόνο καλώδια, αφού τα δύο χρησιμοποιούνται για την I2C επικοινωνία ενώ τα υπόλοιπα για το ρεύμα (VCC) και την γείωση (GND) της.

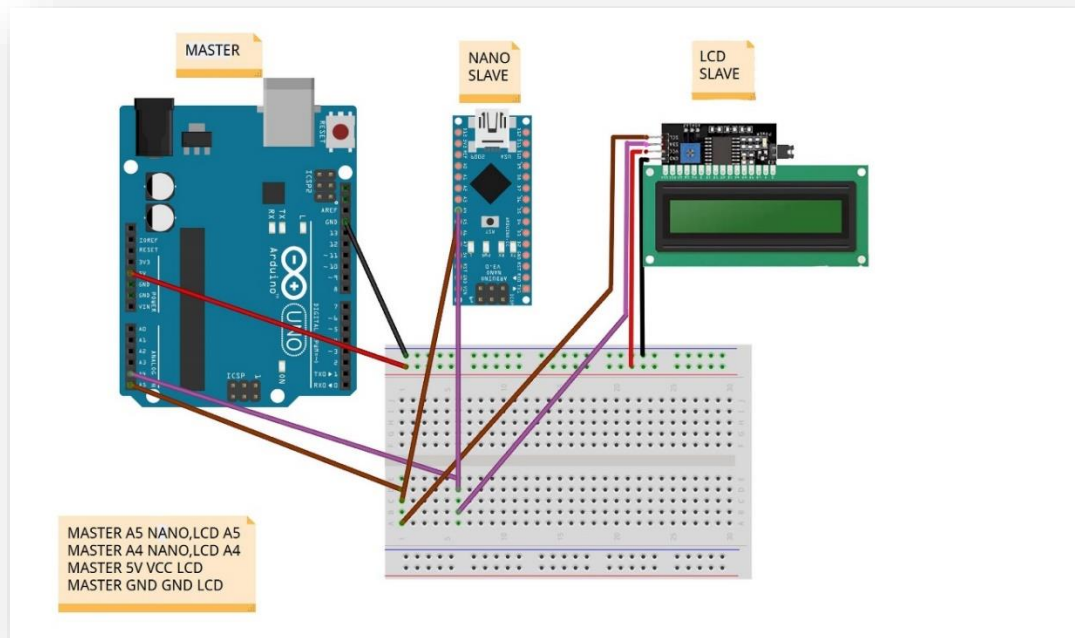


Εικόνα 3.5 : I2C Serial Interface Module

3.3.3.1 Συνδεσμολογία Arduino Uno & LCD (with I2C Serial Interface)

Έχοντας ενώσει την οθόνη με το I2C Serial Interface η σύνδεσή της με το Arduino Uno είναι αρκετά απλή. Όπως έχει αναφερθεί θα υπάρξει μία επικοινωνία τύπου I2C μεταξύ των συσκευών όπου το SCL θα συνδεθεί με το πινάκι A5 και το SDA με το A4, ενώ το πινάκι VCC θα συνδεθεί με το 5V του Arduino Uno όπως και το GND με το GND. Εφόσον έχουμε σύνολο δύο συσκευές Slave κρίνεται αναγκαία η

χρήση ενός Bread Board έτσι ώστε να υπάρχει εύκολα πρόσβαση στα κοινά πινάκια που χρησιμοποιούνται.



Εικόνα 3.6 : Ελεγκτής Arduino Uno με 2 βαθμίδες I2C Slaves

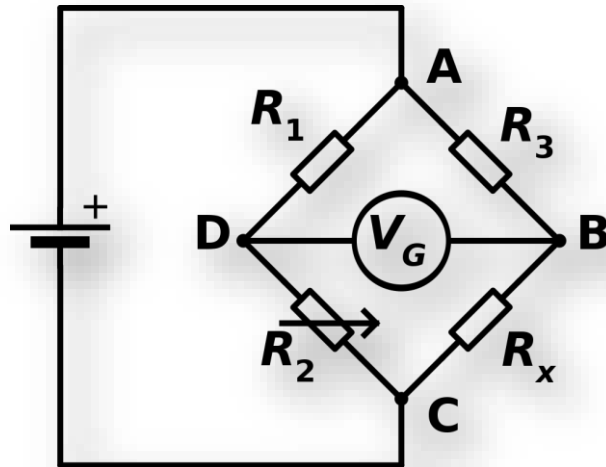
3.3.4 Αισθητήρες βάρους & HX711

Ο υπολογισμός του βάρους αποτελεί ένα σημαντικό και αναπόσπαστο κομμάτι για την δημιουργία της εφαρμογής. Για την κατασκευή της ζυγαριάς μπορεί να χρησιμοποιηθεί ένας, δύο ή τέσσερις αισθητήρες βάρους, όπου ο καθένας έχει την δυνατότητα να υπολογίζει μέχρι 50kg. Εφόσον, το βάρος των κυψελών πολλές φορές φτάνει πάνω από τα 100kg, αποτελεί προϋπόθεση η δημιουργία κυκλώματος με τέσσερις αισθητήρες βάρους, με μέγιστο βάρος υπολογισμού τα 200kg.



Εικόνα 3.7 : Αισθητήρας βάρους

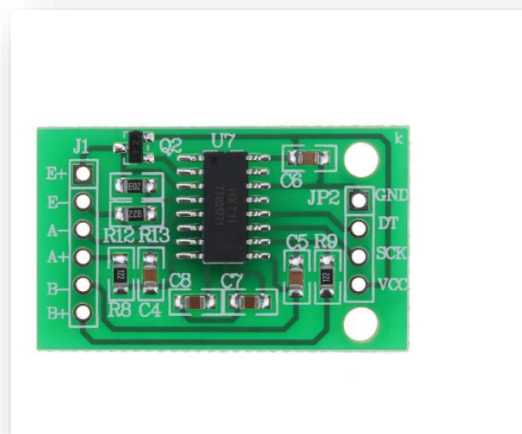
Το συγκεκριμένο κύκλωμα βασίζεται την γέφυρα Wheatstone, η οποία αρχικά χρησιμοποιούταν για την μέτρηση άγνωστων τιμών αντιστάσεων και ως μέσο υπολογισμού οργάνων μέτρησης (πχ βολτόμετρο). Ακόμα και σήμερα, όπου μέσω των ψηφιακών πολύμετρων προσφέρεται ένας αρκετά ευκολότερος τρόπος μέτρησης αντιστάσεων, η γέφυρα Wheatstone χρησιμοποιείται για την εύρεση πολύ μικρών τιμών αντίστασης της τάξεως των milli-Ohm. Το κύκλωμά της αποτελείται από δύο σειρές παράλληλες διευθετημένες από αντιστάσεις που συνδέονται μεταξύ μιας σταθερής τάσης και της γείωσης. Όταν βρίσκονται σε ισορροπία δηλαδή δεν ασκείται κάποια δύναμη βάρους ώστε να αλλάξει η τιμή της αντίστασης, τότε παράγεται μηδενική διαφορά τάσης μεταξύ των δυο παράλληλων κλάδων. Επιπρόσθετα η τοποθέτηση των αντιστάσεων (τον ρόλο των αντιστάσεων λαμβάνουν οι αισθητήρες βάρους) σε ένα κύκλωμα Wheatstone προσιδιάζει σε σχήμα διαμαντιού. Όταν τοποθετείται κάποιο βάρος αλλάζει και η αντίσταση αυτών των αισθητήρων με αποτέλεσμα να αλλάξει και η διαφορά τάσης. Υπολογίζοντας αυτή την διαφορά τάσης δίνεται η δυνατότητα να υπολογιστεί το βάρος που τοποθετήθηκε[49][50].



Εικόνα 3.8 : Γέφυρα Wheatstone Bridge

Πηγή: <https://commons.wikimedia.org/w/index.php?curid=2888809>

Σε θεωρητικό επίπεδο θα ήταν δυνατό να χρησιμοποιηθεί ένας αναλογικός ακροδέκτης ώστε να μετρηθεί η τιμή του βάρους απευθείας από τους αισθητήρες βάρους. Ωστόσο η διαφορά της τάσης είναι τόσο μικρή που το Arduino δεν μπορεί να μετατρέψει το σήμα από αναλογικό σε ψηφιακό. Την λύση στο πρόβλημα αυτό την προσφέρει ο ενισχυτής αισθητήρα βάρους (HX711), ο οποίος προσφέρει μεγάλη ακρίβεια στις αλλαγές των τιμών των αισθητήρων[50].



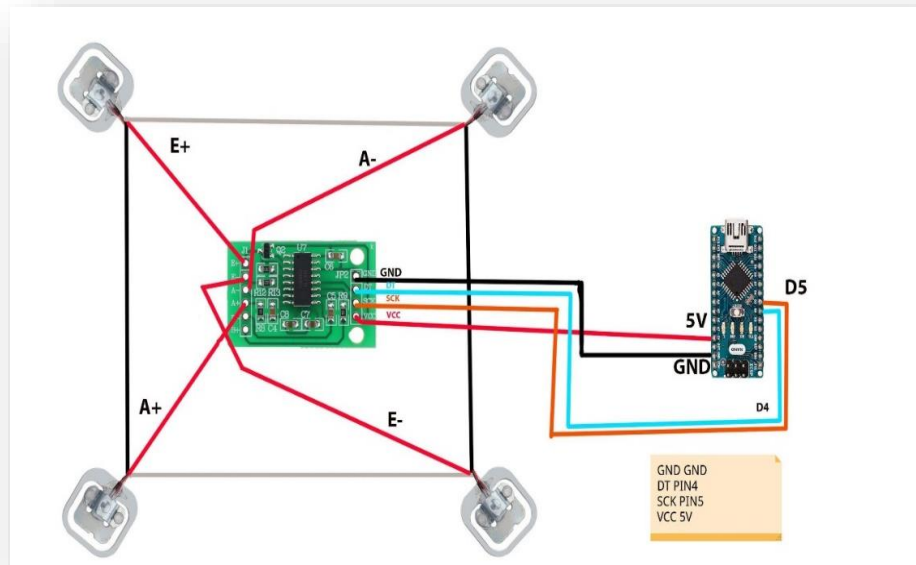
Εικόνα 3.9 : Ενισχυτής αισθητήρα βάρους (HX711)

3.3.4.1 Συνδεσμολογία Αισθητήρων βάρους και HX711

Ο κάθε αισθητήρας βάρους διαθέτει 3 καλώδια, μαύρο, κόκκινο και άσπρο. Κρίνεται απαραίτητο να γίνει μέτρηση της αντίστασης αυτών των καλωδίων, ώστε να εξαιρεθούν τυχόν κατασκευαστικά λάθη του προϊόντος. Η αντίσταση μεταξύ του μαύρου καλωδίου και του άσπρου είναι περίπου 2kOhm, ενώ η αντίσταση μεταξύ μαύρου με κόκκινου και άσπρου με κόκκινου καλωδίου φέρει τιμή 1 kOhm. Εφόσον δεν βρεθεί κάποια διαφορετική τιμή αντίστασης στους 4 αισθητήρες, γίνεται η τοποθέτηση αυτών σε σχήμα ορθογωνίου.

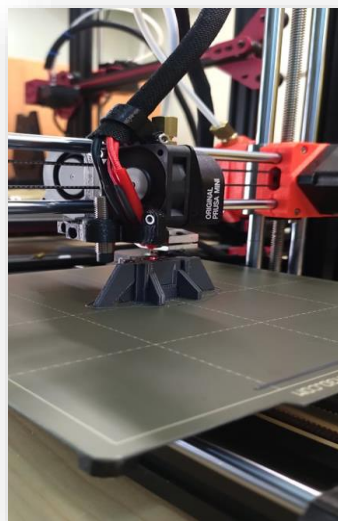
Στην συνέχεια, επιλέγοντας τον αισθητήρα που βρίσκεται στην κάτω αριστερά γωνία (A+) συνδέουμε το μαύρο καλώδιο του με το μαύρο καλώδιο από τον πάνω αισθητήρα (E+) και το άσπρο καλώδιο του με το άσπρο καλώδιο του δεξιού αισθητήρα. Ενώνοντας και τα υπόλοιπα μαύρα και άσπρα καλώδια με τα ίδια χρώματα των αισθητήρων, είναι σημαντικό οι διαγώνιοι αισθητήρες να φέρουν το ίδιο όνομα με αντίθετη πόλωση (A+ A- & E+ E-). Η διαφορετική τοποθέτηση των αισθητήρων με διαφορετική πόλωση μεταξύ των πλευρών, για παράδειγμα στην θέση του E+ να μπει ο αισθητήρας E- , δεν αποτελεί πρόβλημα στο κύκλωμα και εν συνεχεία στην μέτρηση του βάρους.

Οι αισθητήρες βάρους που φέρουν το όνομα «E» είναι τα καλώδια για την παροχή ρεύματος στους αισθητήρες, ενώ αντίστοιχα οι «A» αποτελούν τις εισόδους μέτρησης αυτών. Αφού ο κάθε αισθητήρας φέρει και από ένα όνομα στο κύκλωμα, μετά τοποθετείται στο εκάστοτε πινάκι του HX711 που φέρει το όνομα του. Εφόσον τοποθετηθούν και οι 4 αισθητήρες, ενώνεται το «DT» του αισθητήρα ενίσχυσης με το πινάκι «D4» (Digital 4) του Arduino Nano, το «SCK» με το πινάκι «D5» και αντίστοιχα τα πινάκια VCC & GND μεταξύ των συσκευών.



Εικόνα 3.10 : Κύκλωμα υπολογισμού βάρους

Δυστυχώς οι συγκεκριμένοι αισθητήρες βάρους δεν διαθέτουν κάποια βάση ώστε να είναι εύκολη η τοποθέτησή τους στην κατασκευή. Την λύση έρχεται να προσφέρει ο τρισδιάστατος εκτυπωτής (3D Printer). Με τον 3D Printer δίνεται η δυνατότητα στον χρήστη, μέσω προγραμμάτων σχεδίασης, να δώσει σάρκα και οστά στα σχέδιά του, ακόμα και μέσα από το σπίτι του.



Εικόνα 3.11 : 3D printer. Πηγή: 3D Maniacs

Σχεδιασμός και ανάπτυξη συστήματος αυτοματισμού για μελισσοκομικές εφαρμογές

Παρόλα αυτά υπάρχουν σε πολλές ιστοσελίδες έτοιμα σχέδια που δίνουν την δυνατότητα στον κάτοχο ενός 3D Printer, να τα δημιουργήσει δίχως να κατασκευάσει κάποιο σχέδιο.



Εικόνα 3.12 : Τοποθέτηση αισθητήρων σε σχέδιο από 3D printer

Πηγή σχεδίου: <https://www.thingiverse.com/thing:2624188>



Εικόνα 3.13 : Η βάση ζύγισης

3.3.4.2 Κώδικας της διαδικασίας ζύγισης

Προϋπόθεση για την αποτελεσματικότητα της λειτουργίας ζύγισης, είναι η εύρεση της τιμής βαθμονόμησης της ζυγαριάς, η οποία έχει οριστεί στο πρόγραμμα ως calibration value. Η συγκεκριμένη σταθερή τιμή, χαρακτηρίζει την σχέση μεταξύ της αλλαγής της τάσης στην γέφυρα Wheatstone που χρησιμοποιείται από τον αντίστοιχο αισθητήρα μάζας με τη δύναμη που ασκεί το αντικείμενο που έχει τοποθετηθεί σε αυτόν, με σκοπό τελικά τον υπολογισμό του βάρους του παρόντος αντικειμένου. Ο ολοκληρωμένος κώδικας για την εύρεση του Calibration_value είναι:

```
1 #include <HX711_ADC.h> // Βιβλιοθήκη για τον αισθητήρα ενίσχυσης

//Δήλωση των pins
2 const int HX711_dout = 4; //HX711 dout pin
3 const int HX711_sck = 5; //HX711 sck pin
4 HX711_ADC LoadCell(HX711_dout, HX711_sck); // Δημιουργία
   αντικειμένου LoadCell
5 unsigned long t = 0;

6 void setup() {
7   Serial.begin(57600); // Έναρξη σειριακής επικοινωνίας
8   delay(10); // Καθυστέρηση 10 ms
9   Serial.println();
10  Serial.println("Begining..."); //Εμφάνιση μηνύματος

11  LoadCell.begin(); // Έναρξη λειτουργίας LoadCell

12  if (LoadCell.getTareTimeoutFlag() || LoadCell.getSignalTimeoutFlag()) {
   //Συναρτήσεις ελέγχου συνδεσμολογίας
13  Serial.println("Wrong.Check wiring and pins");
14  while (1);
```

```
}  
15 else {  
16 LoadCell.setCalFactor(1.0); // Αρχικοποίηση της τιμής calculate factor  
17 Serial.println("Complete");  
}  
18 while (!LoadCell.update()); //Έλεγχος μετατροπής δεδομένων  
19 if (LoadCell.getSPS() < 7) {  
20 Serial.println("Samples Per second is very low. Check wiring and pins");  
}  
21 else if (LoadCell.getSPS() > 100) {  
22 Serial.println("Samples Per second is very high. Check wiring and pins");  
}  
  
23 calibrate(); //Έναρξη συνάρτησης υπολογισμού του calibration value  
}  
  
24 void loop() {  
25 static boolean newDataReady = 0;  
26 const int serialPrintInterval = 10; //Τιμή υπεύθυνη για την μείωση εμφάνισης  
    δειγμάτων στο serial  
// Έλεγχος για νέα δεδομένα / Έναρξη εμφάνισης δεδομένων  
27 if (LoadCell.update()) newDataReady = true;  
  
28 if (newDataReady) {  
29 if (millis() > t + serialPrintInterval)    { //Εμφάνιση δεδομένων ανά  
    serialPrintInterval millisecond  
30 float i = LoadCell.getData(); // Η τιμή του βάρους σε γραμμάρια  
31 Serial.print("Load_cell output val: ");  
32 Serial.println(i);  
33 long tareoffset = LoadCell.getTareOffset(); Η επιστροφή της μη  
    επεξεργάσιμης τιμής του απόβαρου.  
34 Serial.println("tareoffset = ");
```

```
35 Serial.println(tareoffset);
36 newDataReady = 0;
37 t = millis(); Επιστρέφεται ο αριθμός των millisecond του τρέχοντος
    προγράμματος.
}
}

// Λήψη χαρακτήρων από το serial
38 if (Serial.available() > 0) { //Έλεγχος λειτουργίας της σειριακής επικοινωνίας
39 char inByte = Serial.read(); //Διάβασμα χαρακτήρων από το serial
40 if (inByte == 't') LoadCell.tareNoDelay(); // Μηδενισμός του απόβαρου

}

// Έλεγχος ολοκλήρωσης απόβαρου
41 if (LoadCell.getTareStatus() == true) {
42 Serial.println("Tare complete");

}
}

43 void calibrate() {
44 Serial.println("****");
45 Serial.println("Calibrate...");
46 Serial.println("Please remove any weight from the Load Cell.");
47 Serial.println("Send 't' to the serial monitor for calibration tare.");

48 boolean _resume = false;
49 while (_resume == false) {
50 LoadCell.update();
51 if (Serial.available() > 0) {
```

```
52 if (Serial.available() > 0) {
53   char inByte = Serial.read();
54   if (inByte == 't') LoadCell.tareNoDelay();
55 }
56 }
57 if (LoadCell.getTareStatus() == true) { //Συνάρτηση ελέγχου ολοκλήρωσης
58   απόβαρου
59   Serial.println("Tare complete");
60   _resume = true;
61 }
62 }
63
64 Serial.println("Place the known weight.");
65 Serial.println("Send the value of weight (px 100.0) to serial monitor.");
66
67 float known_weight = 0;
68 _resume = false;
69 while (_resume == false) {
70   LoadCell.update();
71   if (Serial.available() > 0) {
72     known_weight = Serial.parseFloat(); // Η μεταβλητή known_weight παίρνει
73     την πρώτη δεκαδική τιμή από το serial
74     if (known_weight != 0) {
75       Serial.print("Known mass is: ");
76       Serial.println(known_weight);
77       _resume = true;
78     }
79   }
80 }
81
82 LoadCell.refreshDataSet(); // Ανανέωση του πίνακα τιμών
83 float newCalibrationValue = LoadCell.getNewCalibration(known_weight);
84 //Τοποθέτηση νέας τιμής του CalibrationValue
```

```
72 Serial.print("newCalibrationValue: ");
73 Serial.print(newCalibrationValue);
74 Serial.println("End calibration");

}
```

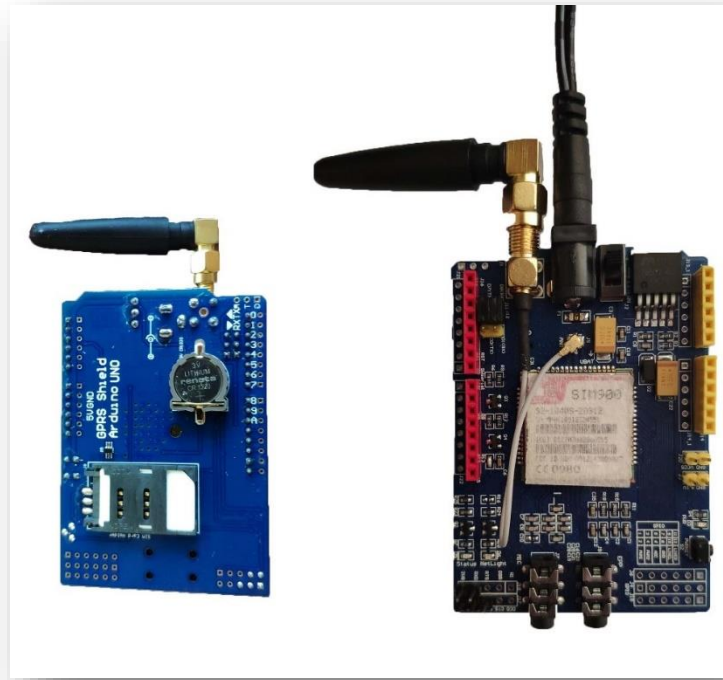
3.3.4.3 Επεξήγηση κώδικα ζύγισης

- Εντολή 4 (HX711_ADC LoadCell(HX711_dout, HX711_sck)): Δημιουργία αντικειμένου LoadCell(με τους ψηφιακούς ακροδέκτες HX711_dout, HX711_sck) της κλάσης HX711_ADC
- Εντολή 12 (LoadCell.getTareTimeoutFlag()) : Επιστρέφει τιμές 0 εφόσον δεν παρουσιάζεται πρόβλημα στον υπολογισμό του απόβαρου, και 1 για τυχόν λάθος συνδεσμολογία της συσκευής.
- Εντολή 12 (LoadCell.getSignalTimeoutFlag()) : Επιστρέφει «TRUE» εφόσον παρουσιάζεται καθυστέρηση στην δημιουργία δεδομένων από τον ακροδέκτη «dout». Αντίθετα επιστρέφει «FALSE» όταν δεν παρουσιάζεται πρόβλημα στην εξαγωγή των δεδομένων.
- Εντολή 18 (LoadCell.update()) : Γίνεται η μετατροπή των δεδομένων και εφόσον είναι έτοιμη επιστρέφει την τιμή «1» . Όταν ο υπολογισμός του απόβαρου είναι έτοιμος επιστρέφεται η τιμή «2», ενώ σε οποιαδήποτε άλλη περίπτωση επιστρέφεται η τιμή «0».
- Εντολή 19 (LoadCell.getSPS()) : Επιστρέφει την τιμή της δειγματοληψίας ανά δευτερόλεπτο (Sample Per Second), η οποία σύμφωνα με τον κατασκευαστή, παίρνει τιμές 10 ή 80.
- Εντολή 29 (Millis()) : Επιστρέφεται ο αριθμός των millisecond του τρέχοντος προγράμματος. Συχνά χρησιμοποιείται για την εκτέλεση εντολών κώδικα μετά από καθορισμένο χρόνο που έχει θεσπίσει ο χρήστης, δίχως να προκαλείται καθυστέρηση στην εκτέλεση του προγράμματος.

- Εντολή 41 (LoadCell.getTareStatus()) : Επιστρέφει την τιμή «TRUE» όταν η συνάρτηση tareNoDelay() ολοκληρωθεί
- Εντολή 70 (LoadCell.refreshDataSet()) : Ανανέωση του πίνακα τιμών για να δημιουργηθεί η βεβαιότητα πως ο υπολογισμός του βάρους είναι σωστός.

3.3.5 SIM900 GPRS/GSM

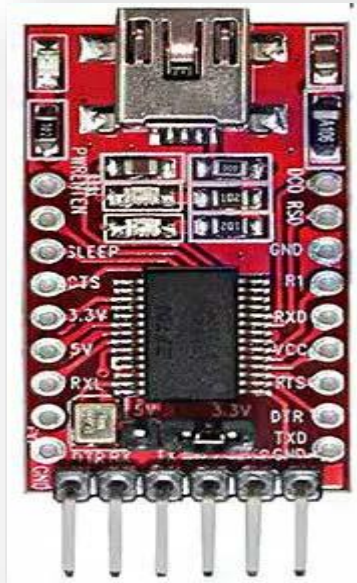
Το SIM900 GSM (Global System for Mobile Communications) /GPRS(General Packed Radio Service) είναι το κύκλωμα που παρέχει την δυνατότητα να υπάρξει τηλεπικοινωνία μεταξύ του χρήστη και της εφαρμογής. Έχει την ικανότητα να συνδέεται στο Internet μέσω του GPRS δικτύου, να στέλνει και να λαμβάνει μήνυμα καθώς και τηλεφωνήματα. Στην παρούσα κατασκευή το SIM900 θα είναι αρμόδιο να καλεί τον χρήστη όταν σπλίσει ο συναγερμός καθώς και να στέλνει τις απαραίτητες μετρήσεις όταν το επιθυμεί ο χρήστης. Προκειμένου να έχει την δυνατότητα να συνδεθεί στο δίκτυο τηλεφωνίας είναι απαραίτητη η χρήση μιας κάρτας SIM, η οποία θα τοποθετηθεί στο πίσω μέρος του. Δίπλα στο μέρος της SIM υπάρχει υποδοχή για την ένταξη μια μπαταρίας τύπου CR1220, όπου θα διατηρεί την ώρα και την ημερομηνία ακόμα και όταν κλείσει η συσκευή ,εφόσον την ρυθμίσει σωστά ο χρήστης. Επιπρόσθετα, λόγω της απαίτησης της συσκευής για περαιτέρω παροχή ενέργειας, πρέπει να τοποθετηθεί στην υποδοχή DC jack, τροφοδοσία της τάξεων από 5-12V και 2 A[51].



Εικόνα 3.14: SIM900 (πίσω-μπροστά)

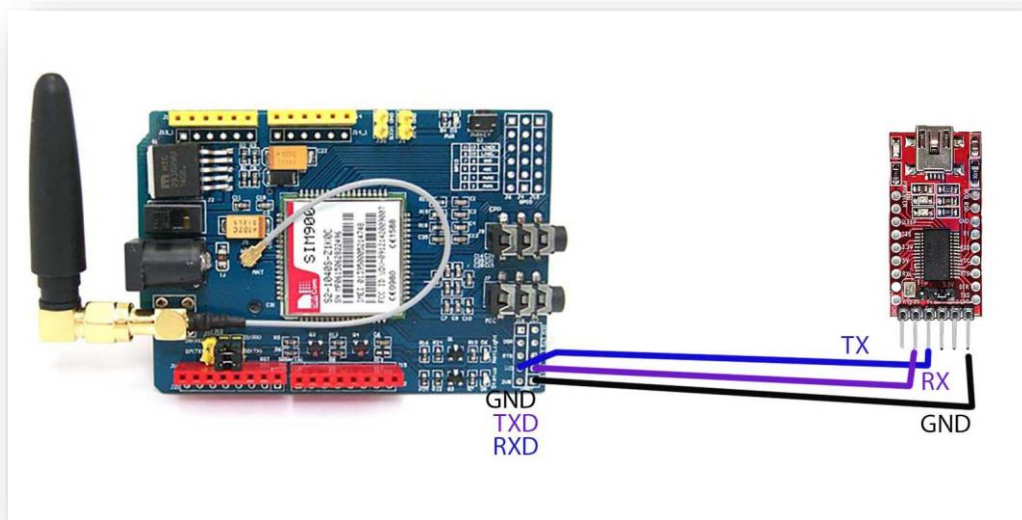
3.3.5.1 Συνδεσμολογία SIM900 & FTDI FT232RL USB

Συνήθως όταν κατασκευάζεται μία αυτοματοποιημένη συσκευή εμφανίζονται πολλαπλά εμπόδια μέχρι την ομαλή της λειτουργία. Μπορεί να προκύψουν κατασκευαστικά προβλήματα ή ακόμα και λανθασμένες συνδεσμολογίες. Για να εξαιρεθούν οι πιθανότητες αυτών των εμποδίων πολλές φορές γίνεται έλεγχος στο κάθε κομμάτι της συσκευής πριν την ένωσή της με την κατασκευή αυτοματισμού. Θέλοντας, στην παρούσα περίπτωση, να ελεγχθεί ο SIM900 γίνεται χρήση του FTDI USB.



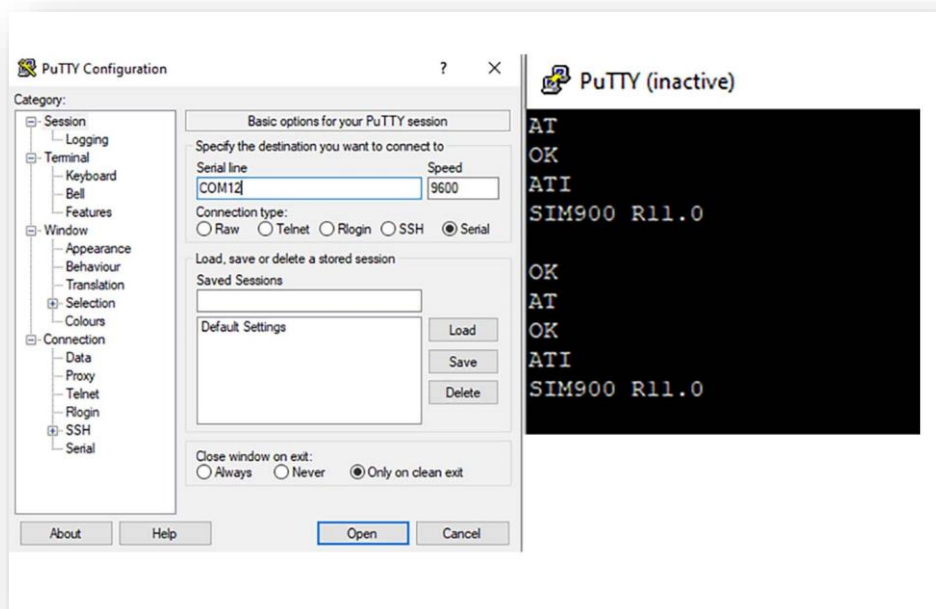
Εικόνα 3.15 : FTDI FT232RL USB

Η συγκεκριμένη συσκευή δίνει την ικανότητα στον χρήστη να συνδέσει την εκάστοτε συσκευή με τον υπολογιστή. Ενώνοντας τα πινάκια RXD, TXD του SIM900 με τα αντίστοιχα TX, RX του FTDI καθώς και το GND μεταξύ των συσκευών επιτυγχάνεται η σειριακή επικοινωνία.



Εικόνα 3.16 : Συνδεσμολογία SIM900-FTDI USB

Κάνοντας χρήση του προγράμματος PuTTY ολοκληρώνεται η αλληλεπίδραση μεταξύ του SIM900 και του χρήστη με τις κατάλληλες προγραμματιστικές εντολές. Αρχικά γίνεται η επιλογή θύρας USB στην οποία έχει τοποθετηθεί το FTDI USB, μέσω του προγράμματος καθώς και πατώντας την επιλογή «Serial». Στην συνέχεια, επιλέγοντας την επιλογή «Open» αρχίζει η σειριακή επικοινωνία ανοίγοντας ένα παράθυρο που γράφονται οι εντολές. Στέλνοντας την εντολή «AT» καλούμε το SIM900 να απαντήσει. Εφόσον σταλεί η απάντηση «OK», τότε όλα λειτουργούν κανονικά. Τέλος πληκτρολογώντας την εντολή «ATI» εμφανίζεται η ταυτοποίηση του προϊόντος.



Εικόνα 3.17 : PuTTY & AT commands

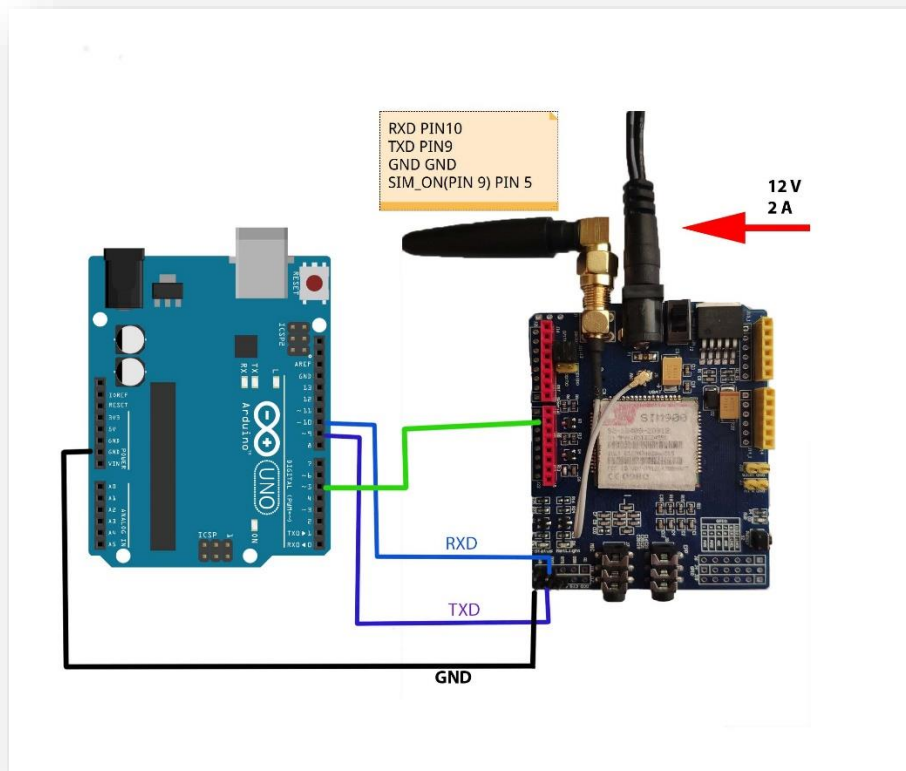
Εφόσον έχει επιτευχθεί η σωστή επικοινωνία μεταξύ του SIM900 με τον υπολογιστή, μπορεί να ρυθμιστεί η ώρα και η ημερομηνία. Αρχικά στέλνεται η εντολή «AT+CLTS=1», ώστε να ενημερωθεί η συσκευή ότι θα διαμορφωθεί η ημερομηνία και η ώρα. Εφόσον σταλεί η απάντηση «OK» αποστέλλεται η εντολή «AT+CCLK= “dd/MM/yy, hh:mm:ss+/-zz” » με τις κατάλληλες τιμές στην ημερομηνία και την ώρα καθώς και την time zone της χώρας[52].

```
AT+CLTS=1
OK
AT+CCLK="07/03/21,19:59:00+03"
ERROR
AT+CCLK="07/03/21,19:59:00+03"
OK
AT+CLTS=1
OK
AT+CCLK?
+CCLK: "07/03/21,19:59:31+03"
```

Εικόνα 3.18 : Ενημέρωση ημερομηνίας και ώρας

3.3.5.2 Συνδεσμολογία SIM900 & Arduino Uno

Έχοντας ελέγξει την επικοινωνία του SIM900, συνδέεται με τον εγκέφαλο της συσκευής, το Arduino Uno. Συνδέεται το RXD, TXD του SIM900 με τα πινάκια 10,9 αντίστοιχα και τα GND μεταξύ τους. Έτσι επιτυγχάνεται, με την κατάλληλη χρήση εντολών κώδικα, η σειριακή επικοινωνία των δύο συσκευών. Τέλος για να ενεργοποιείται προγραμματιστικά η συσκευή SIM900, συνδέεται ο ακροδέκτης D5 του Arduino με το D9 του SIM900.



Εικόνα 3.19 : Σύνδεση Arduino Uno-SIM900

3.3.6 GPS Module

Η κλοπή των μελισσιών, όπως έχει ήδη αναφερθεί, παρουσιάζεται ως μείζον πρόβλημα στην καθημερινότητα ενός μελισσοκόμου. Εφόσον η τοποθέτηση των κυψελών γίνεται σε απομονωμένες περιοχές, η εγκατάσταση μιας σειρήνας που θα ηχεί όταν παραβιάζεται το μελίσσι, δεν θα επιφέρει κάποιο θετικό αποτέλεσμα. Η βέλτιστη λύση κάτω από αυτές τις συνθήκες είναι να δίνεται η δυνατότητα στον μελισσοκόμο να εντοπίσει την κυψέλη, αφού πρώτα ενημερωθεί για την παραβίασή της. Για την εμφάνιση της τοποθεσίας της συσκευής, χρησιμοποιείται το GPS Module που έχει εγκατασταθεί σε αυτή.

Το GPS Module είναι προγραμματισμένο να γνωρίζει που βρίσκονται οι δορυφόροι GPS, καθώς και να υπολογίζει το πόσο μακριά βρίσκονται από αυτό. Οι δορυφόροι διαδίδουν πληροφορίες σχετικά με την θέση τους και τον τρέχοντα χρόνο, μέσω ραδιοφωνικών σημάτων. Η συσκευή καθορίζει την απόσταση με τον δορυφόρο, υπολογίζοντας τον χρόνο που χρειάστηκαν τα σήματα να φτάσουν σε αυτή.

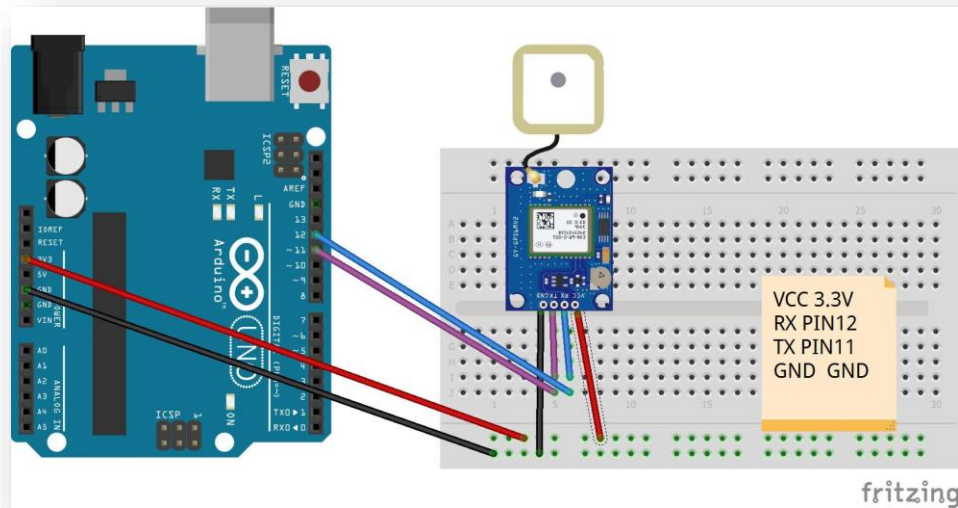
Κατέχοντας αυτές τις πληροφορίες από τουλάχιστον 3 δορυφόρους μπορεί να εντοπίσει την τοποθεσία της συσκευής[53].



Εικόνα 3.20 : NEO-6M GPS Module με κεραία

3.3.6.2 Συνδεσμολογία GPS Module & Arduino Uno

Το GPS διαθέτει 4 υποδοχές ακροδεκτών, οι δύο από αυτές αποτελούν την αποστολή και την λήψη δεδομένων TX-RX, και συνδέονται με τους ψηφιακούς ακροδέκτες 11 & 12 αντίστοιχα του Arduino Uno. Έτσι επιτυγχάνεται η σειριακή επικοινωνία μεταξύ των συσκευών. Το chip που διαθέτει το GPS, λειτουργεί στα 2.7-3.6V όμως υπάρχει εγκαταστημένος ένα ρυθμιστής τάσης που δίνει τη δυνατότητα σύνδεσης του ακόμα και με 5V τροφοδοσία. Παρ' όλα αυτά, για ασφάλεια συνδέεται το πινάκι VCC από το GPS με τάση 3V3 στο Arduino. Τέλος συνδέεται και η γείωση μεταξύ των συσκευών(GND-GND). Η συσκευή χρειάζεται λίγο χρόνο μέχρι να βρεθεί η τοποθεσία. Όταν υπάρχει αλληλεπίδραση με τους δορυφόρους αναβοσβήνει το Led που διαθέτει[53].



Εικόνα 3.21 : Συνδεσμολογία Arduino Uno & GPS Module

3.3.7 IR Wireless Remote Control

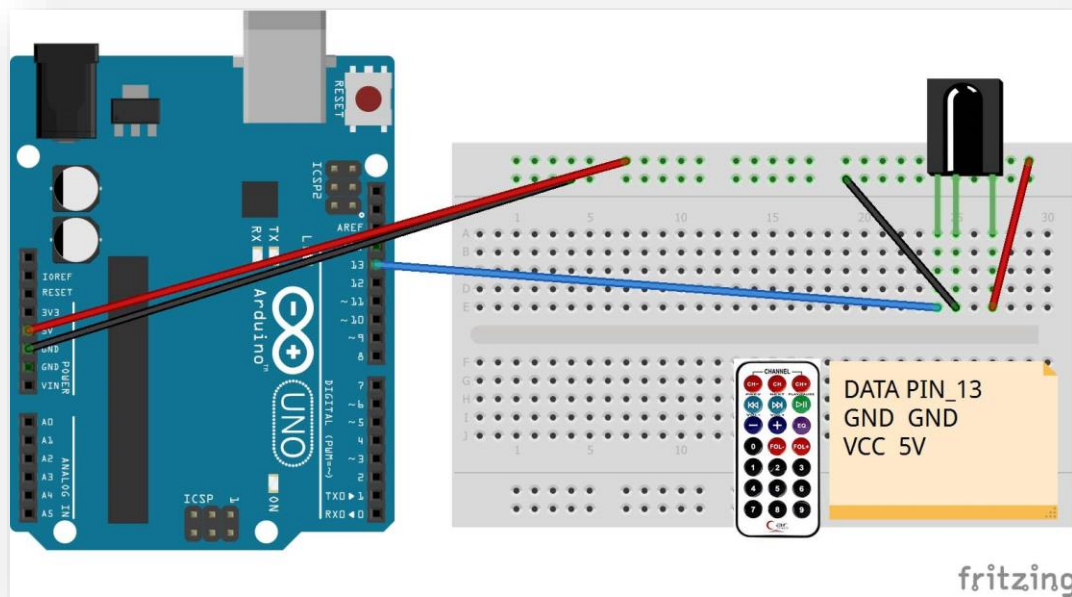
Η συγκεκριμένη συσκευή είναι αρμόδια για την επικοινωνία με τον χρήστη όταν βρίσκεται κοντά σε αυτή. Ο IR αποδέκτης λαμβάνει σήματα από το τηλεκοντρόλ και με κατάλληλη αποκωδικοποίηση αυτών των σημάτων δίνεται η δυνατότητα να μεταβάλλεται η κατάσταση του τελικού πειράματος. Στην παρούσα κατάσταση θα δίνεται η ικανότητα στον χρήστη να λαμβάνει της παρούσες τιμές των αισθητήρων θερμοκρασίας-υγρασίας, βάρους καθώς και να ενεργοποιείται, απενεργοποιείται ο συναγερμός.



Εικόνα 3.22 : IR Receiver & Wireless Remote Control

3.3.7.1 Συνδεσμολογία IR Wireless Remote Control

Η συνδεσμολογία που απαιτείται είναι αρκετά απλή. Ο IR δέκτης αποτελείται από τρία «ποδαράκια», όπου το αριστερό είναι αρμόδιο για την λήψη του κωδικού που θα σταλθεί από το τηλεκοντρόλ και εν συνεχεία στον μικροελεγκτή μέσω του πινακίου D13 . Ο μεσαίος ακροδέκτης αποτελεί την γείωση και ο δεξιός την παροχή ρεύματος.



Εικόνα 3.23 : Συνδεσμολογία Arduino Uno & IR Wireless Remote Control

3.3.7.2 Κώδικας IR Wireless Remote Control

Πριν την εγκατάσταση του παρόντος μηχανισμού στο αυτοματοποιημένο σύστημα θα πρέπει να βρεθούν οι τιμές των σημάτων που θα στέλνονται από το τηλεκοντρόλ, ώστε να μπορούν να χρησιμοποιηθούν στο τελικό πρόγραμμα. Για την επίτευξη αυτού του στόχου χρησιμοποιήθηκε ο εξής κώδικας:


```
#include <IRremote.h> // Βιβλιοθήκη για το IRremote

int IRpin = 11; // Σύνδεση στο ψηφιακό pin11

IRrecv irrecv(IRpin); // Δημιουργία αντικειμένου irrecv της κλάσης IRrecv

decode_results results; // Δημιουργία αντικειμένου results της κλάσης
decode_results

void setup()

{

Serial.begin(9600); // Έναρξη σειριακής επικοινωνίας

irrecv.enableIRIn(); // Έναρξη αντικειμένου irrecv

}

void loop()

{

if (irrecv.decode(&results)) // Έλεγχος αποστολής σήματος

{

unsigned int value = results.value; // Αποθήκευση ληφθέντα κωδικού στο
results.value

Serial.println(value); // Εμφάνιση κωδικών σε δεκαδική μορφή

irrecv.resume(); // Προετοιμασία δέκτη για λήψη της επόμενης τιμής

}

}
```

Οι τιμές που προκύπτουν σε δεκαδική μορφή είναι:

Button	Code(DEC)	Value = results.value
PLAY/PAUSE	16769565	57885
1	16724175	12495
2	16718055	6375
3	16743045	31365
4	16716015	4335
5	16726215	14535
6	16734885	23205

Πίνακας 3.1 : Τιμές από τον IR Receiver

3.3.8 Αισθητήρας δόνησης (vibrator sensor)

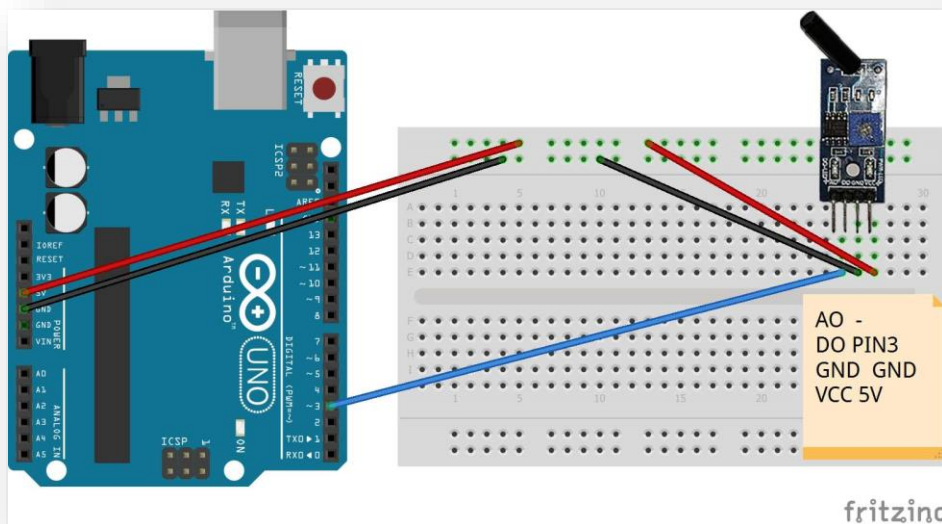
Σε μία πιθανή απόπειρα κλοπής του μελισσιού, προφανώς θα πραγματοποιηθεί μετακίνηση της κυψέλης, με αποτέλεσμα τη δημιουργία κάποιου «τραντάγματος» στην συσκευή. Ο κατάλληλος αισθητήρας για αυτή την κατάσταση είναι ο αισθητήρας δόνησης. Μόλις ο αισθητήρας ενεργοποιηθεί μέσω δόνησης, θα ενημερώσει το Arduino Uno που θα είναι συνδεδεμένο μαζί του, με αποτέλεσμα τελικά την όπλιση του συναγερμού.



Εικόνα 3.24 : Αισθητήρας δόνησης

3.3.8.1 Συνδεσμολογία αισθητήρα δόνησης

Ο vibrator sensor διαθέτει 4 «ποδαράκια», AO(Analog Out),DO(Digital Out), GND (Ground) και VCC. Αρχικά, συνδέουμε το DO με το D3 του Arduino και την παροχή ρεύματος(VCC) με το 5V. Εφόσον επιτευχθεί και η γείωση μεταξύ του αισθητήρα και του Υπο(GND-GND), θα υπάρξει σύνδεση και θα ανάψει ένα πράσινο led δείχνοντας πως ο αισθητήρας διαρρέεται από ρεύμα. Πάνω στον αισθητήρα είναι εγκατεστημένο το ποτενσιόμετρο, όπου ρυθμίζεται η ευαισθησία που θα διαθέτει ο αισθητήρας στις δονήσεις που θα προκληθούν. Τέλος όταν προκληθεί κάποια δόνηση που θα οπλίσει τον αισθητήρα τότε ανάβει προσωρινά ένα led αριστερά στην συσκευή.

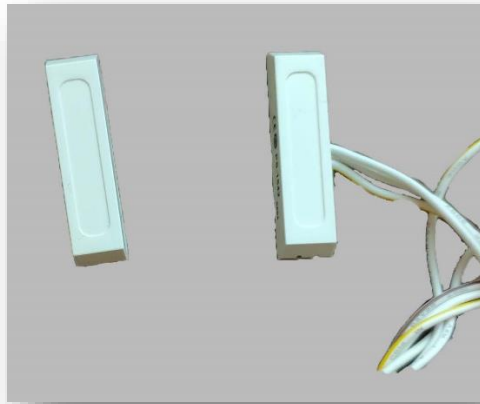


Εικόνα 3.25 : Συνδεσμολογία αισθητήρα δόνησης

3.3.9 Μαγνητικός αισθητήρας συναγερμού

Επόμενη αιτία ενεργοποίησης του συναγερμού αποτελεί ο μαγνητικός αισθητήρας. Αυτός ο αισθητήρας απαρτίζεται από δύο μαγνητικούς πόλους οι οποίοι όταν αποκολληθούν αλλάζει κατάσταση το κύκλωμα. Η εγκατάσταση του αισθητήρα πραγματοποιήθηκε στο καπάκι της κυψέλης, όπου στην περίπτωση παραβίασης θα πραγματοποιηθεί κλήση στον χρήστη πως υπάρχει κίνδυνος κλοπής. Ο

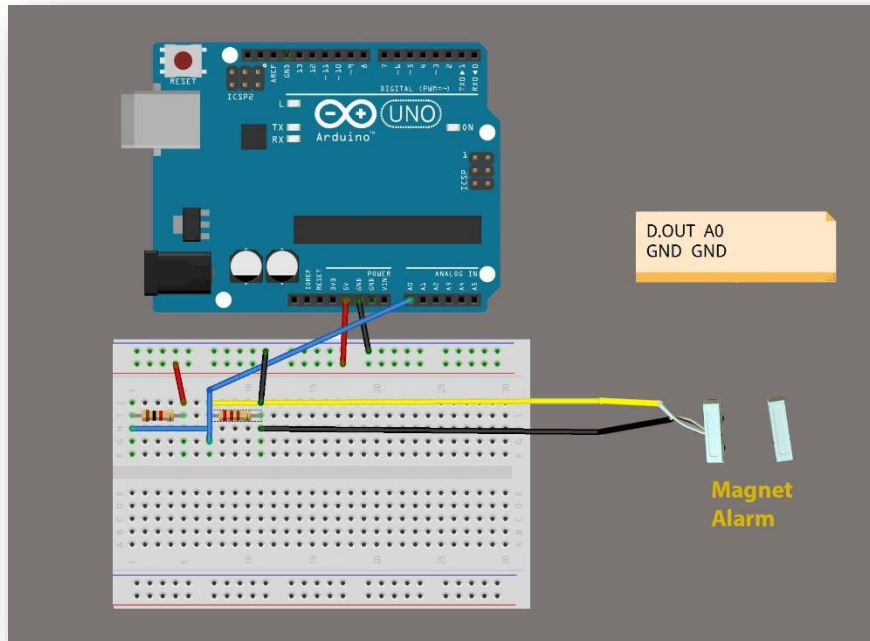
συγκεκριμένος αισθητήρας είναι αρκετά διαδεδομένος σε συστήματα συναγερμού, καθώς έχει μικρό κόστος και εύκολη εγκατάσταση.



Εικόνα 3.26 : Αισθητήρας μαγνητικής παγίδευσης

3.3.9.1 Συνδεσμολογία μαγνητικού αισθητήρα συναγερμού

Ο αναφερόμενος αισθητήρας διαθέτει τρία καλώδια, τα δύο εκ των οποίων αποτελούν την κατάσταση του κυκλώματος, αν είναι N.O (Normally Open) ή N.C(Normally closed). Στην κατάσταση N.O, όταν οι πόλοι του μαγνήτη είναι συνδεδεμένοι (δηλαδή στην κατάσταση ηρεμίας), τότε το κύκλωμα είναι ανοιχτό, ενώ όταν υπάρχει συμβάν το κύκλωμα αλλάζει κατάσταση σε κλειστό. Στην παρούσα κατασκευή επιλέχθηκε αυτή η κατάσταση του κυκλώματος, όπου το χρώμα του καλωδίου είναι άσπρο με κίτρινο και συνδέεται στο A0 (Analog) του Arduino αφού πρώτα μειωθεί η ένταση του ηλεκτρικού ρεύματος με δύο αντιστάσεις 1kΩm και 220Ω. Το τρίτο καλώδιο αποτελεί την γείωση όπου ενώνεται με το GND του Arduino.

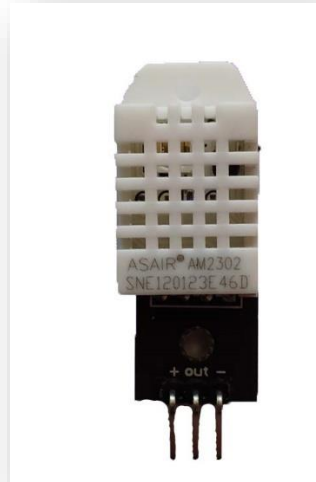


Εικόνα 3.27 : Συνδεσμολογία αισθητήρα μαγνητικού συναγερμού

3.3.10 Αισθητήρας θερμοκρασίας και υγρασίας (DHT22 module)

Η θερμοκρασία που επικρατεί στη κυψέλη είναι ζωτικής σημασίας για το μέλισσι, καθώς ανάλογα με την υπάρχουσα θερμοκρασία εξαρτάται η επιβίωση και η αναπαραγωγή του. Ως αποτέλεσμα αυτού η θερμοκρασία αποτελεί μία αρκετά σημαντική ένδειξη για την υγεία των μελισσών. Επιπρόσθετα, οι ασθένειες που έχουν να αντιμετωπίσουν οι μέλισσες, έχουν ένα πολύ ισχυρό σύμμαχο στην εξάπλωσή τους, την υγρασία. Έτσι, η υγρασία συγκροτεί μία ακόμα ενδιαφέρουσα ένδειξη και εξίσου σημαντική για την εικόνα που επικρατεί σε ένα μέλισσι. Είναι ξεκάθαρο πως η ένδειξη της θερμοκρασίας και της υγρασίας θα αποτελέσει μέγιστη βοήθεια στην καθημερινότητα του παραγωγού, για αυτό έχει εγκατασταθεί στην παρούσα εφαρμογή ένας αισθητήρας που υπολογίζει με μεγάλη ακρίβεια την θερμοκρασία και την υγρασία του μελισσιού. Εφόσον σταλθεί το κατάλληλο μήνυμα από τον χρήστη στην εφαρμογή, τότε θα λάβει ως απάντηση την ένδειξη της θερμοκρασίας και της υγρασίας εκείνη την συγκεκριμένη στιγμή. Στην περίπτωση που ο παραγωγός βρίσκεται κοντά στην συσκευή έχει την δυνατότητα μέσω του IR Remote Control να επικαλεστεί τις τιμές αυτές οι οποίες θα προβληθούν στην

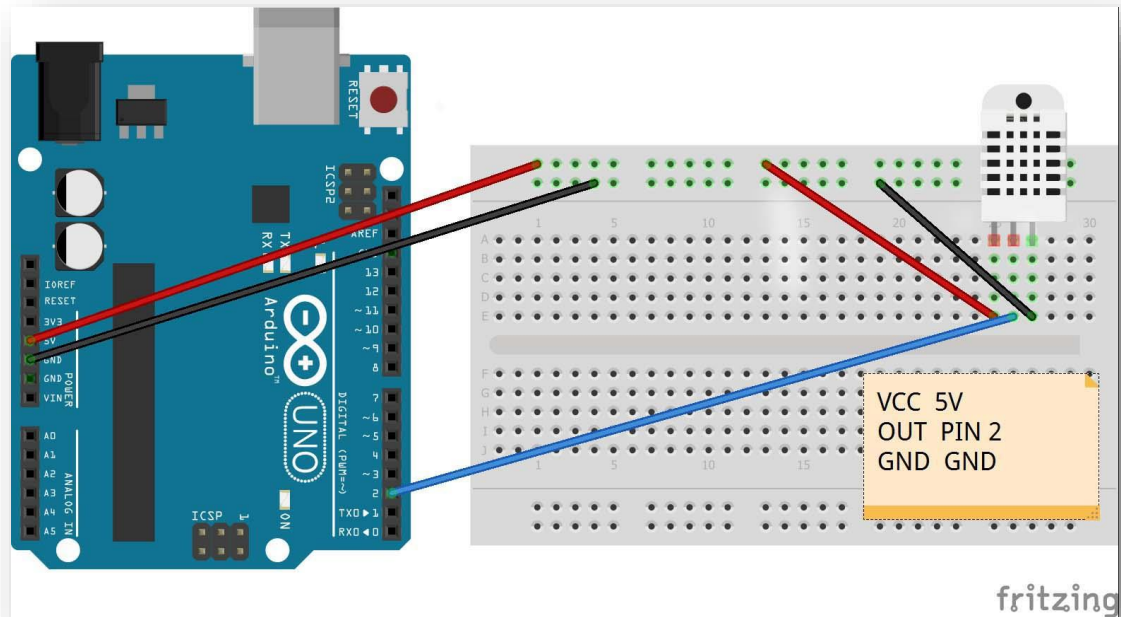
συνέχεια στην οθόνη. Αυτό έχει ως αποτέλεσμα να παρέχεται στον μελισσοκόμο μια αρκετά σημαντική πληροφορία από το μελίσσι, εύκολα, γρήγορα, δίχως να απαιτείται η παρουσία του στον τόπο εγκατάστασης του μελισσιού.



Εικόνα 3.28 : Αισθητήρας θερμοκρασίας και υγρασίας (DHT22 module)

3.3.10.1 Συνδεσμολογία DHT22 module

Ο συγκεκριμένος αισθητήρας αποτελείται από 3 ακροδέκτες. Υπεύθυνος για την μεταφορά των μετρήσεων είναι ο μεσαίος με την ένδειξη «OUT» ο οποίος συνδέεται με το D2 του Arduino. Για την τροφοδοσία του αισθητήρα ενώνεται ο αριστερός ακροδέκτης «+» με 5V, και ο τελευταίος με την γείωση της συσκευής.



Εικόνα 3.29 : Συνδεσμολογία DHT22 module & Arduino Uno

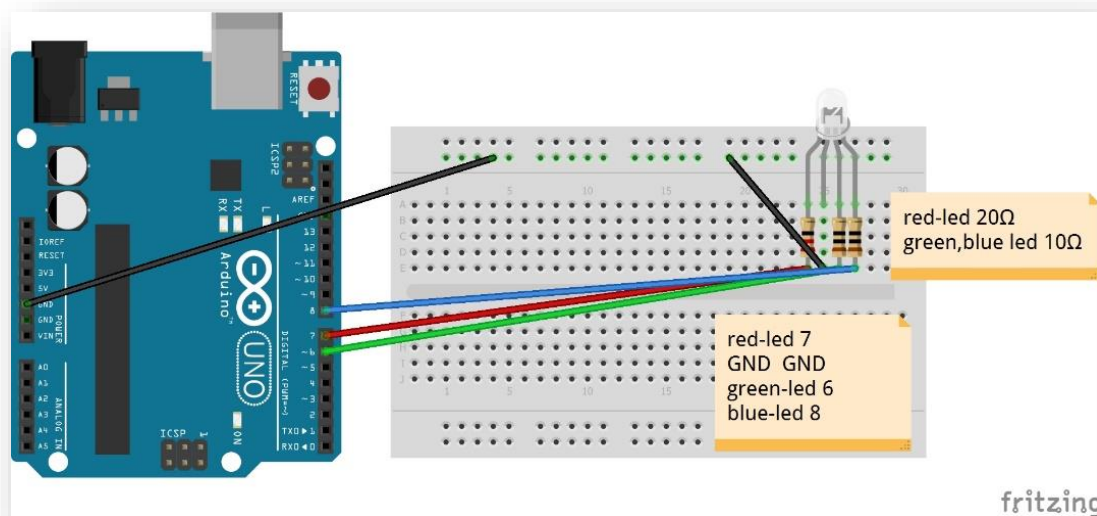
3.3.11 RGB Led

Το RGB Led θα μπορούσε να ερμηνευτεί ως το σύνολο τριών led σε ένα, μόνο που το κάθε ένα από αυτά, διαθέτει και ένα χρώμα(κόκκινο, πράσινο και μπλε). Τροφοδοτώντας αυτά τα led με διαφορετικές αναλογικές τιμές δίνεται η δυνατότητα στον χρήστη να δημιουργήσει ένα τεράστιο φάσμα χρωμάτων. Η επιλογή του παρόντος τύπου led γίνεται να γνώμονα την ενημέρωση του χρήστη(όταν αλλάζει η κατάσταση του αυτοματισμού) μέσω φωτεινών ενδείξεων. Συγκεκριμένα όταν ο συναγερμός απενεργοποιηθεί μέσω του τηλεκοντρόλ, τότε θα υπάρξει πράσινη ένδειξη, ενώ όταν ενεργοποιηθεί κόκκινη. Στην περίπτωση που ο χρήστης θελήσει να ενημερωθεί για τις τιμές θερμοκρασίας, βάρους και υγρασίας τότε η ένδειξη είναι μπλε.

3.3.11.1 Συνδεσμολογία RGB Led

Το ξεχωριστό αυτό led διαθέτει 4 ακροδέκτες. Ο αριστερός είναι αρμόδιος για την κόκκινη ένδειξη και ενώνεται με το D7, ο τρίτος για την πράσινη προβολή, που συνδέεται με το D6, ενώ ο τέταρτος για την μπλε ένδειξη το οποίο συνδέεται με το

D8 του Arduino αντίστοιχα. Προτού όμως συνδεθούν με την κεντρική συσκευή πρέπει να μειωθεί η τάση του ρεύματος αλλιώς θα καταστραφεί το εκάστοτε led. Για το πράσινο και μπλε led η προτεινόμενη τάση ρεύματος είναι τα 3.5V και χρησιμοποιείται αντίσταση της τάξεως των 10Ω, ενώ το κόκκινο έχει προτεινόμενη τάση ρεύματος στα 2.2V και έτσι τοποθετείται μεγαλύτερη τιμή αντίστασης, η οποία είναι στα 20Ω. Το δεύτερο ποδαράκι αποτελεί την γείωση η οποία συνδέεται με το Arduino Uno στο GND.



Εικόνα 3.30 : Συνδεσμολογία RGB Led & Arduino Uno

3.4. Κώδικας προγραμματισμού συστήματος

Μετά την ολοκλήρωση των συνδεσμολογιών μεταξύ των προαναφερθέντων εξαρτημάτων, πραγματοποιείται η σύνταξη του κώδικα που θα εισαχθεί στο Arduino Nano ,που επιτελεί τον ρόλο του Slave, καθώς και στο Arduino Uno που διαθέτει τον ρόλο του Master αντίστοιχα.

3.4.1 Κώδικας Slave (Nano)

- 1 `#include <HX711_ADC.h> //Βιβλιοθήκη αισθητήρα ενίσχυσης`
- 2 `#include<Wire.h> //Βιβλιοθήκη επίτευξης I2C επικοινωνίας`


```
//Δήλωση Ακροδεκτών
```

```
3  const int HX711_dout = 4; //HX711 dout digital pin
```

```
4  const int HX711_sck = 5; //HX711 sck digital pin
```

```
5  HX711_ADC  LoadCell(HX711_dout,  HX711_sck);  //  Δημιουργία  
    αντικειμένου LoadCell
```

```
6  void setup() {
```

```
7  Serial.begin(9600); // Έναρξη σειριακής επικοινωνίας
```

```
8  delay(10); // Καθυστέρηση 10 ms
```

```
9  Serial.println();
```

```
10 Serial.println("Begining...");
```

```
11 float calibrationValue; // Υπολογιστική τιμή ζύγισης
```

```
12 calibrationValue = 22.73; //Η υπολογιστική τιμή ζύγισης
```

```
13 LoadCell.begin(); // Έναρξη λειτουργίας LoadCell
```

```
14 unsigned long newoffset = 8666294; // Η μη επεξεργασμένη τιμή του  
    απόβαρου
```

```
15 LoadCell.setTareOffset(newoffset); // Ορισμός τιμής απόβαρου
```

```
16 LoadCell.refreshDataSet(); // // Ανανέωση του πίνακα τιμών
```

```
17 if (LoadCell.getTareTimeoutFlag() || LoadCell.getSignalTimeoutFlag()) {  
    //Συναρτήσεις ελέγχου συνδεσμολογίας
```

```
18 Serial.println("Wrong.Check wiring and pins");
```

```
19 while (1);
```

```
}
```

```
20 else {
```

```
21 LoadCell.setCalFactor(calibrationValue); // Ορισμός τιμής CalibrationValue
22 Serial.println("Complete...");
    }

23 while (!LoadCell.update()); ///Έλεγχος μετατροπής δεδομένων
24 Serial.print("Calibration value: ");
25 Serial.println(LoadCell.getCalFactor()); // Εμφάνιση τιμής CalibrationValue
26 Serial.print("HX711 sampling rate: ");
27 Serial.println(LoadCell.getSPS()); // Εμφάνιση Samples Per Second
28 if (LoadCell.getSPS() < 7) {
29 Serial.println("Samples Per second is very low. Check wiring and pins");
    }

30 else if (LoadCell.getSPS() > 100) {
31 Serial.println("Samples Per second is very high. Check wiring and pins");
    }

32 Wire.begin(8); //Έναρξη της I2C επικοινωνίας με διεύθυνση «8»

33 Wire.onReceive(requestEvent); //Όταν ληφθεί αίτημα από τον Master
    εκτελείται η συνάρτηση «requestEvent»
}

34 void loop() {
35 delay(100);
}

//Η συνάρτηση που εκτελείται όταν ζητηθεί από τον Master

36 void requestEvent() {

37 String weight = Weight(); // Εκτελείται η συνάρτηση Weight() και
    αποθηκεύεται στην μεταβλητή weight
38 Serial.println("the length is");
39 Serial.println(weight.length()); //Εμφάνιση μεγέθους της μεταβλητής weight
```

Σχεδιασμός και ανάπτυξη συστήματος αυτοματισμού για μελισσοκομικές εφαρμογές

```
40 if(weight.length(<9)){ // Έλεγχος αποστολής τιμών
41 weight = weight +"xx"; // Αύξηση γραμματοσειράς weight
42 Serial.print(weight); // Εμφάνιση γραμματοσειράς weight
    }

43 Wire.write(weight.c_str()); // Αποστολή χαρακτήρων στον Master
}

// Συνάρτηση υπολογισμού βάρους

44 String Weight() {

45 float i = 0; // Αρχικοποίηση τιμής

46 if(LoadCell.update()) { // Έλεγχος μετατροπής δεδομένων
47 LoadCell.refreshDataSet(); // Ανανέωση πίνακα τιμών
48 i = LoadCell.getData()/1000; //Μετατροπή τιμής βάρους σε κιλά
49 Serial.print("weight==");
50 Serial.println(i); //Εμφάνιση νέας τιμής βάρους
51 String weight = float_to_String (i,3); // Μετατροπή μεταβλητής float σε String
52 Serial.println(weight);
53 return weight +"kg"; // Επιστροφή τιμής weight ενωμένη με kg

    }

54 else {

55 return("error_kg"); // Επιστροφή μηνύματος error_kg
    }

}
```

// Συνάρτηση μετατροπής float σε String

```
56 String float_to_String (float value, int x){  
  
57 float a = value;  
58 String data = ""; // Αρχικοποίηση τιμής  
59 data = String (a,x); // Μετατροπή float a σε String με x ψηφία  
60 return data; //Επιστροφή τιμής  
}
```

3.4.1.1 Διευκρινήσεις κώδικα Slave (Nano)

- Εντολή 12 (calibrationValue = 22.73) : Η συγκεκριμένη τιμή βρέθηκε στο υποκεφάλαιο «Κώδικας υπολογιστικής τιμής ζυγαριάς».
- Εντολή 14 (unsigned long newoffset = 8666294;): Η μη επεξεργασμένη τιμή του απόβαρου βρέθηκε στο υποκεφάλαιο «Κώδικας υπολογιστικής τιμής ζυγαριάς»
- Εντολή 40 (if(weight.length(<9))) : Το μέγιστο συνολικό βάρος, με βάση τον κατασκευαστή, που μπορεί να μετρηθεί είναι 250kg. Εφόσον έχει επιλεγθεί να εμφανίζονται ακόμα τρία ψηφία μετά τον χαρακτήρα «.» , καθώς και η γραμματοσειρά «kg», τότε ο μέγιστος συνολικός αριθμός χαρακτήρων(char) που μπορεί να δημιουργηθεί είναι 9.
- Εντολή 43 (Wire.write(weight.c_str())) : Εκτελείται η συνάρτηση «weight» και στέλνεται ο αριθμός των Byte που αιτήθηκε ο Master. Η συνάρτηση «weight» δημιουργεί ένα String, όπου είναι απαραίτητη η χρήση της συνάρτησης c_str(), η οποία με την σειρά της μετατρέπει τα περιεχόμενα ενός String σε έναν πίνακα χαρακτήρων. Με αυτή την μορφή είναι ικανή η αποστολή τους στον Master.
- Εντολή 56 (String float_to_String (float value, int x)) : Αποτελεί την συνάρτηση που μετατρέπει μία float τιμή (value) σε String με «x» δεκαδικά ψηφία να εμφανίζονται στην τελική γραμματοσειρά.

3.4.2 Κώδικας Master (UNO)

```
1 #include <dht.h> // Βιβλιοθήκη αισθητήρα DHT
2 #include <LiquidCrystal_I2C.h> // Βιβλιοθήκη LCD οθόνης
3 #include <SoftwareSerial.h> // Βιβλιοθήκη επίτευξη σειριακής επικοινωνίας
4 #include <IRremote.h> // Βιβλιοθήκη IRremote
5 #include <TinyGPS++.h> // Βιβλιοθήκη GPS
6 #include <Wire.h> // Βιβλιοθήκη επίτευξης I2C επικοινωνίας

7 LiquidCrystal_I2C lcd(0x27, 16, 2); // Δημιουργία αντικειμένου lcd
8 SoftwareSerial SIM900(9, 10); // Δημιουργία αντικειμένου SIM900(RX,TX)
9 SoftwareSerial GPS_devis(11,12); // Δημιουργία αντικειμένου
GPS_devis(RX,TX)

10 TinyGPSPlus gps; // Δημιουργία αντικειμένου gps
11 dht DHT; // Δημιουργία αντικειμένου DHT

// Ορισμός των PINS //

12 #define DHT22_PIN 2
13 #define vs 3
14 #define ledgreen 6
15 #define ledred 7
16 #define ledblue 8
17 #define RECV_PIN 13
18 #define trap_sensor A0

19 IRrecv irrecv(RECV_PIN); // Δημιουργία αντικειμένου irrecv
20 decode_results results; // Δημιουργία αντικειμένου results
21 // Κωδικοί IR remote
22 #define change 100 // Αλλαγή συχνότητας ληφθέντων κωδικών
```

```
23 #define codePlay 57885 + change // Κωδικός από button PLAY/PAUSE
    (ALARM OFF)
24 #define code1 12495 + change // Κωδικός από button no. 1 (DHT22)
25 #define code2 6375 + change // Κωδικός από button no. 2 (Weight)
26 #define code3 31365 + change // Κωδικός από button no. 3 (ALARM )

27 float t; // Θερμοκρασία
28 float h; // Υγρασία
29 char incomingChar; // Αποθήκευση χαρακτήρων

30 // Μεταβλητές για την συνάρτηση millis()
31 const unsigned long timecall = 50000;
32 unsigned long previousTime = 0;
33 const unsigned long timeGPS = 10000;
34 unsigned long previousTimeGPS = 0;
35 const unsigned long DayCheck = 3500000;
36 unsigned long previousDayCheck = 0;

37 boolean case_flag; // Flag ενεργοποίησης ή απενεργοποίησης συναγερμού
38 String StringWeight= "";
39 String printWeight= "";
40 String gps_value = "";

41 void setup() {

42 case_flag = false;

43 Serial.begin(9600); // Έναρξη σειριακής επικοινωνίας
44 delay(100);
45 SIM900_On(); // Άνοιγμα SIM900
46 GPS_devisе.begin(9600); // Έναρξη λειτουργίας GPS_devisе
```

```
47 delay(100);
48 Wire.begin(); //Έναρξη της I2C επικοινωνίας
49 delay(100);
50 irrecv.enableIRIn(); // Έναρξη του IR receive
51 delay(1000);

// Δήλωση συσκευών
52 pinMode(vs, INPUT);
53 pinMode(trap_sensor, INPUT);
54 pinMode(ledgreen, OUTPUT);
55 pinMode(ledred, OUTPUT);
56 pinMode(ledblue, OUTPUT);

// Αρχική τιμή led σε LOW
57 digitalWrite(ledgreen, LOW);
58 digitalWrite(ledred, LOW);
59 digitalWrite(ledblue, LOW);

60 delay(5000);
61 AlarmOff (); // Συνάρτηση απενεργοποίησης συναγερμού
62 delay(1000);

63 SIM900.begin(19200); // Έναρξη λειτουργίας SIM900
64 delay(15000); // Καθυστέρηση συσκευής ώστε να συνδεθεί ο SIM900 στο
    δίκτυο
65 if(SIM900.available()>0) // Έλεγχος ενεργοποίησης SIM900
{
66 Serial.print("SIM900 ready...");
67 MylcdMethod("SIM900", 0, "READY", 0); // Εμφάνιση μηνυμάτων στη LCD
68 RGB ('g',5000); // Συνάρτηση ενεργοποίησης 'g' led
69 lcd.noBacklight(); // Απενεργοποίηση φωτισμού LCD
```

```
}  
70 else  
{  
71 MylcdMethod("No connection", 0, "SIM900", 0); // Εμφάνιση μηνυμάτων στη  
    LCD  
72 RGB ('r',5000); // Συνάρτηση ενεργοποίησης 'r' led  
73 lcd.noBacklight();  
74 }  
  
75 SIM900.print("AT+CMGF=1\r"); // Ορισμός συσκευής σε λειτουργία SMS  
76 delay(100);  
// Set module to send SMS data to serial out upon receipt  
77 SIM900.print("AT+CNMI=2,2,0,0,0\r"); // Ορισμός συσκευής σε αποστολή  
    δεδομένων SMS στο serial  
78 delay(100);  
  
}  
  
79 void loop() {  
  
80 int trap_sensor = analogRead(A0); // Διάβασμα αναλογικού σήματος  
81 int vibratorSensor = digitalRead(vs); // Διάβασμα ψηφιακού σήματος  
  
82 SIM900.listen(); // Ενεργοποίηση σειριακής επικοινωνίας SIM900  
  
83 if((millis() > DayCheck + previousDayCheck ))  
{  
84 previousDayCheck = millis();  
85 if (CLOCK('2','0')){ // Έλεγχος ώρας
```



```
86 String printWeight= Request_Master(); // Λήψη δεδομένων από τον Slave
    Nano
87 Serial.println(printWeight);

88 Serial.println("its time");
89 if ((printWeight.toInt() > 10) || (printWeight.toInt() < 5)) // Έλεγχος τιμής
    βάρους
{
90 Serial.println("sms sending");
91 sendSMS("Danger " +printWeight); // Αποστολή SMS στον χρήστη
}
}

92 else {
93 Serial.println("no time");
}
}

//----- ALARM -----//

94 if (((vibratorSensor == 0) || (trap_sensor >100)) &&(case_flag==true)) //
    Έλεγχος για ενεργοποίηση συναγερμού
{

95 MylcdMethod("Danger ", 0, "Calling", 0);

96 if (millis() > timecall + previousTime)
{
```

```
97 call(); // Συνάρτηση κλήση στον χρήστη
98 previousTime = millis();

}

99     delay(2000);
100     lcd.noBacklight();
101     }

102     if (irrecv.decode(&results)) { // Έλεγχος κωδικών από τον IR remote

103         unsigned int value = results.value + change; // Εισαγωγή κωδικών
            του IR σε μεταβλητή

// Έλεγχος τιμών value
104         switch (value) {

105             case codePlay: //<---- Alarm off ----->
106                 AlarmOff ();
107                 case_flag = false;
108                 break;

109             case code1: //<---- Θερμοκρασία-Υγρασία ----->
110                 MyTemperature(); // Συνάρτηση υπολογισμού θερμοκρασίας-
                    υγρασίας
111                 MylcdMethod("Humidity: ", h, "Temp: ", t); // Εμφάνιση υγρασίας-
                    θερμοκρασίας σε LCD
112                 RGB ('b',4000);
```

```
113     lcd.noBacklight();
114     case_flag = false;
115     break;

116     case code3: //<---- Alarm ON ----->

117         MylcdMethod("Bye user ", 0, "Alarm is on ", 0);
118         RGB ('r',6000);
119         lcd.noBacklight();
120         case_flag = true;
121         break;

122     case code2: //<---- Weight ----->
123         String printWeight= Request_Master();
124         delay(100);

125         MylcdMethod("Weight:"+printWeight, 0,"ALARM IS OFF" , 0); //
        Εμφάνιση βάρους στην LCD
126         RGB ('b',6000);
127         lcd.noBacklight();
128         case_flag = false;
129         break;

}

130     Serial.println(value); // Εμφάνιση τιμής value
131     irrecv.resume(); // Λήψη επόμενης τιμής value
}
```

```
132     if (SMSRequest()){ // Έλεγχος λήψης SMS από χρήστη
133     if (readData()) { // Διάβασμα δεδομένων από WEIGHT,GPS,DHT22

134     String dataMessage3 = "";

135     Serial.println(gps_value);
136     delay(1000);

137     SIM900.println("AT + CMGS = \"+306984035200'\"); // Ορισμός
        αριθμού τηλεφώνου αποστολής SMS
138     delay(100);

139     String dataMessage = ("Temperature: " + String(t) + "*C " + "
        Humidity: " + String(h) + "%" + " "); // Αποθήκευση τιμών του DHT22
140     String dataMessage2=(" Weight: " +printWeight + " "); // Αποθήκευση
        τιμών Weight
141     dataMessage3 = (gps_value); // Αποθήκευση τοποθεσίας GPS

// Εμφάνιση τιμών στην σειριακή
142     Serial.println("dataMessage==");
143     Serial.println(dataMessage);
144     Serial.println(dataMessage2);
145     Serial.println(dataMessage3);

// Αποστολή τιμών σε SMS
146     SIM900.println(dataMessage);
147     delay(100);
148     SIM900.println(dataMessage2);
```

```
149     delay(100);
150     SIM900.println(dataMessage3);
151     delay(100);

// Δήλωση χαρακτήρα τέλους a ^Z, ASCII code 26, στο SMS
152     SIM900.println((char)26);
153     delay(100);
154     SIM900.println();

155     delay(5000); // Καθυστέρηση συσκευής για αποστολή SMS
}

156     delay(10);
}
}

// Συνάρτηση αποστολής SMS
157     void sendSMS(String sms)
158     {

159         SIM900.println("AT + CMGS = \"+306984035200\");
160         delay(100);

161         SIM900.println(sms);
162         delay(100);

163         SIM900.println((char)26);
164         delay(100);
```

```
165     SIM900.println();
166     delay(5000);

}

// Συνάρτηση κλήσης τηλεφώνου
167     void call()
{

168     SIM900.println("ATD+ +306984035200;"); // Ορισμός κλήσης
        τηλεφώνου
169     delay(20000);
170     SIM900.println("ATH"); // Τέλος κλήσης
171     Serial.println("calling");

}

// Συνάρτηση διαβάσματος τιμών των συσκευών
172     boolean readData() {

173     MyTemperature();

174     printWeight = Request_Master();
175     GPS_deviser.listen(); // Ενεργοποίηση σειριακής επικοινωνίας
        GPS_deviser
176     previousTimeGPS=millis();
177     gps_value = GPS_Method(); // Αποθήκευση τοποθεσίας σε
        μεταβλητή
178     Serial.print("Humidity: ");
179     Serial.print(h);
180     Serial.print(" %\t");
```

```
181     Serial.print("Temperature: ");
182     Serial.print(t);
183     Serial.print(" *C ");
184     delay(10);
185     return true;
}

// Συνάρτηση λήψης SMS χρήστη
186     boolean SMSRequest() {

187         //Έλεγχος λήψης SMS "STATE"

188         if (SIM900.available() > 0) {
189             incomingChar = SIM900.read(); // Διάβασμα χαρακτήρα από
                σειριακή επικοινωνία με SIM900
190             if (incomingChar == 'S') {
191                 delay(10);
192                 Serial.print(incomingChar);
193                 incomingChar = SIM900.read();
194                 if (incomingChar == 'T') {
195                     delay(10);
196                     Serial.print(incomingChar);
197                     incomingChar = SIM900.read();
198                     if (incomingChar == 'A') {
199                         delay(10);
200                         Serial.print(incomingChar);
201                         incomingChar = SIM900.read();
202                         if (incomingChar == 'T') {
203                             delay(10);
204                             Serial.print(incomingChar);
205                             incomingChar = SIM900.read();
```



```
{  
  
224     int chk = DHT.read22(DHT22_PIN); // Διάβασμα τιμών DHT22  
225     delay(2000);  
226     h = DHT.humidity; // Τιμή υγρασίας  
227     t = DHT.temperature; // Τιμή θερμοκρασίας  
  
228     if (isnan(h) || isnan(t)) { // Έλεγχος τιμών  
229         Serial.println("Failed to read from DHT sensor!");  
230         return false;  
}  
  
}  
  
// Συνάρτηση απενεργοποίησης συναγερμού  
231     void AlarmOff ()  
{  
232         MylcdMethod("Wellcome user ", 0, "Alarm is off", 0);  
233         RGB ('g',5000);  
234         lcd.noBacklight();  
  
}  
  
235     // Συνάρτηση ελέγχου RGB Led  
236     void RGB (char color,int delay_time )  
{  
237         switch (color) {  
  
238         case 'r': // Ενεργοποίηση κόκκινου led  
  
239             digitalWrite(ledred, HIGH);  
240             delay(delay_time);  

```

```
241     digitalWrite(ledred, LOW);
242     break;

243     case 'b': // Ενεργοποίηση μπλε led

244     digitalWrite(ledblue, HIGH);
245     delay(delay_time);
246     digitalWrite(ledblue, LOW);
247     break;

248     case 'g': // Ενεργοποίηση πράσινου led
249     digitalWrite(ledgreen, HIGH);
250     delay(delay_time);
251     digitalWrite(ledgreen, LOW);
252     break;

}
}

//Συνάρτηση εύρεση τοποθεσίας μέσω GPS
253     String GPS_Method(){
254     previousTimeGPS = millis() ;
255     while(true){
256     if (GPS_devis.available() > 0) { // Έλεγχος λειτουργίας συσκευής
        GPS_devis
257     gps.encode(GPS_devis.read()); // Διάβασμα & αποκωδικοποίηση
        τιμών GPS
258     if (gps.location.isUpdated()) { // Έλεγχος ανανέωσης τιμών GPS
259     delay(100);
```

```
260     double Latitude = gps.location.lat(); // Αποθήκευση της τιμής Latitude
261     double Longitube = gps.location.lng(); // Αποθήκευση της τιμής
        Longitube

262     String Lat= double_to_String (Latitude,6); // Αποθήκευση της τιμής
        Latitude με ακρίβεια 6 ψηφίων
263     String Lng= double_to_String (Longitube,6); // Αποθήκευση της τιμής
        Longitube με ακρίβεια 6 ψηφίων

264     String gps_message = ("https://www.google.com/maps/place/" + Lat
        + "," + Lng); // Εμφάνιση τοποθεσίας σε ψηφιακό χάρτη
265     previousTimeGPS = millis() ;
266     return gps_message; // Επιστροφή Link τοποθεσίας
267     break;
}

}

// Έλεγχος μη εύρεσης τοποθεσίας
268     else if(millis() - previousTimeGPS > timeGPS)
{

269     String gps_message = "no connection with gps" ;
270     //Serial.println(gps_message);
271     previousTimeGPS=millis();
272     return gps_message;
273     break;
}
}
}

// Συνάρτηση μετατροπής double σε String σε 'x' ψηφία
274     String double_to_String (double value, int x)
```

```
{  
  
    275     double a = value;  
    276     String data = "";  
    277     data = String(a,x);  
  
    278     return data;  
}  
  
// Συνάρτηση ενεργοποίησης SIM900  
    279     void SIM900_On(){  
    280     digitalWrite(5, HIGH);  
    281     delay(2000);  
    282     digitalWrite(5, LOW);  
    283     delay(1000);  
}  
  
// Συνάρτηση υπολογισμού ώρας  
    284     boolean CLOCK(char x, char z) {  
  
    285     int k = 0;  
    286     int len = 50;  
    287     char timearr[len]; // Πίνακας χαρακτήρων  
  
    288     for (int y=0;y<len;y++){ // Αρχικοποίηση τιμών πίνακα  
  
    289     timearr[y] = 0;  
    290     }  
    291     SIM900.println("AT+CCLK?"); // Εμφάνιση ώρας από SIM900  
    292     delay(1000);  
  
// Τοποθέτηση χαρακτήρων στον πίνακα
```

```
293     for (int i=0;i<len;i++){
294         timearr[i] = SIM900.read(); // Αποθήκευση χαρακτήρων από την
           σειριακή του SIM900
295         if (timearr[i] == ',') // Έλεγχος χαρακτήρων
296             {
297                 k = i;
298             }
299     }

// Αποθήκευση χαρακτήρων ώρας
300     char h1 = timearr[k+1];
301     char h2 = timearr[k+2];

// Έλεγχος χαρακτήρων ώρας
302     if( (h1==x) && (h2== z)
{

303         return true;
}

304     else {
305         return false;
}
}

// Συνάρτηση αποστολής τιμών από Nano Slave
306     String Request_Master()

{
```

```
307     String StringWeight = ""; // Αρχικοποίηση τιμής
308     Wire.requestFrom(8,9); // Απαίτηση αποστολής 9 bytes από το
        Nano slave με διεύθυνση #8

309     while (Wire.available()) {

310         char c = Wire.read(); // Αποθήκευση χαρακτήρων από I2C
            επικοινωνία
311         if(c == 'x') // Έλεγχος τιμής
{
312         StringWeight = "";
313         break;

}

314     StringWeight+= c; // Αποθήκευση χαρακτήρων από Nano Slave
315     printWeight =StringWeight; // Αποθήκευση χαρακτήρων σε String

}

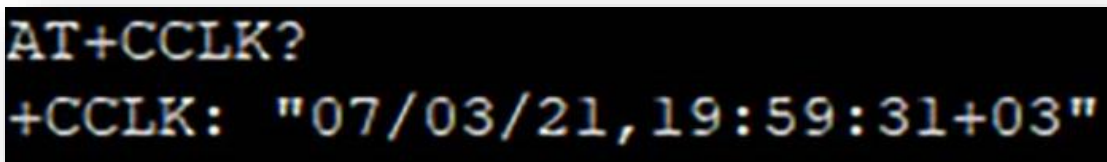
316     return printWeight; // Τελική τιμή βάρους

}
```

3.4.2.2 Διευκρινήσεις κώδικα Master (Uno)

- Εντολή 7 (LiquidCrystal_I2C lcd(0x27, 16, 2)) : Δημιουργία αντικειμένου «lcd» από την κλάση LiquidCrystal_I2C, με διεύθυνση 0x27, 16 στήλες και 2 γραμμές.
- Εντολή 82 (SIM900.listen()) : Στον κώδικα χρησιμοποιούνται μέσω της βιβλιοθήκης SoftwareSerial δύο σειριακές επικοινωνίες. Όμως δεν είναι δυνατόν να γίνεται ταυτόχρονη χρήση του SIM900 και του GPS. Η εντολή «listen» ενεργοποιεί την μία από την δύο σειριακές επικοινωνίες.

- Εντολή 85 (if (CLOCK('2','0'))): Η συνάρτηση CLOCK έχει δημιουργηθεί ώστε να ελέγχει αν η ώρα είναι 8μμ.
- Εντολή 262 (String Lat= double_to_String (Latitude,6)): Η τιμή του Latitude έχει τύπο μεταβλητής Long. Εφόσον αυτή η τιμή θα σταλθεί στον χρήστη πρέπει να μετατραπεί σε String, ώστε να είναι αναγνωρίσιμο από τον SIM900. Ο αριθμός '6' δηλώνει τον αριθμό των ψηφίων που θα μετατραπούν από την Long τιμή.
- Εντολή 295 (if (timearr[i] == ',')): Η συγκεκριμένη εντολή βρίσκεται μέσα στην συνάρτηση CLOCK. Όταν καλείται ο SIM900 να επιστρέψει την ώρα έχει την μορφή «AT+CCLK="dd/MM/yy,hh:mm:ss+/-zz"». Παρατηρείται ότι μετά τον χαρακτήρα «,» τα δύο επόμενα ψηφία δηλώνουν την παρούσα ώρα, που χρησιμοποιείται στην συνέχεια στο πρόγραμμα.



```
AT+CCLK?  
+CCLK: "07/03/21,19:59:31+03"
```

Εικόνα 3.31 : Ένδειξη ώρας SIM900

3.5.1 Android Εφαρμογή

Η παρούσα διπλωματική εργασία δημιουργήθηκε με γνώμονα την διευκόλυνση της καθημερινότητας ενός μελισσοκόμου. Αυτό έχει ως αποτέλεσμα την αλληλεπίδραση του με την τεχνολογία, επιφέροντας σημαντικές και γρήγορες λύσεις με το πάτημα ενός κουμπιού. Παρ' όλα αυτά έχει παρατηρηθεί πως αρκετοί άνθρωποι που ασχολούνται με τον πρωτογενή τομέα δεν διαθέτουν τις απαραίτητες τεχνολογικές γνώσεις, με αποτέλεσμα να εμφανίζεται ένα σοβαρό εμπόδιο στην σχέση μεταξύ του χρήστη και της συσκευής. Την λύση στον πρόβλημα προσφέρει η δημιουργία μιας Android εφαρμογής, όπου το μόνο που χρειάζεται ο εκάστοτε χρήστης είναι το πάτημα ενός κουμπιού. Εφόσον γίνει αυτό, αναπαράγεται ένα φωνητικό μήνυμα, που ενημερώνει τον χρήστη για την

επικοινωνία της εφαρμογής με την συσκευή. Στην συνέχεια λαμβάνει ως απάντηση τις μετρήσεις του βάρους, της θερμοκρασίας, της υγρασίας καθώς και την τοποθεσία της κυψέλης. Ταυτόχρονα παρουσιάζεται ένας δορυφορικός χάρτης δείχνοντας την ακριβή τοποθεσία αλλά και ένα εικονίδιο, που με την ενεργοποίηση του, ανοίγει αυτόματα το Google Maps, προσφέροντας περαιτέρω πληροφορίες στον χρήστη. Με αυτό το φιλικό περιβάλλον της εφαρμογής καθώς και με την εύκολη λειτουργία της εκμηδενίζονται οι αποστάσεις μεταξύ του παραγωγού και της συσκευής.

3.5.2 MIT App Inventor & Κώδικας

Η υλοποίηση μιας εφαρμογής Android αποτελεί ένα αρκετά δύσκολο επίτευγμα καθώς απαιτεί μία αρκετά καλή γνώση προγραμματισμού σε συνδυασμό με την εμπειρία ενός προγραμματιστή. Δημοφιλέστερη, στην δημιουργία Android εφαρμογών, είναι η γλώσσα προγραμματισμού Java, η οποία έχει χαρακτηριστεί ως μία αρκετά απαιτητική γλώσσα. Τον εκμηδενισμό του βαθμού δυσκολίας στην υλοποίηση μιας εφαρμογής προσφέρει το **MIT App Inventor**. Το MIT App Inventor είναι ένα περιβάλλον γραφικού προγραμματισμού το οποίο είναι προσβάσιμο από όλες τις ηλικίες προσφέροντας αμέτρητες δυνατότητες στις εκάστοτε εφαρμογές. Στόχος του είναι η οδήγηση των ανθρώπων και κυρίως των νέων, στην δημιουργία της τεχνολογίας και όχι μόνο στην κατανάλωσή της[54].

Το γραφικό του περιβάλλον περιέχει δύο όψεις, το front end και το back end. Το front end αποτελεί την γραφική αλληλεπίδραση του χρήστη με την εφαρμογή. Στην αριστερή πλευρά της οθόνης εμφανίζονται οι επιλογές που διαθέτει ο σχεδιαστής. Οι λειτουργίες που χρησιμοποιήθηκαν στην εφαρμογή είναι οι εξής:

- **TextBox:** Δημιουργεί ένα πλαίσιο κειμένου όπου στην προκειμένη περίπτωση γράφεται το μήνυμα που θα σταλεί στο SIM900 (STATE).
- **Button:** Εμφανίζεται ένα κουμπί, όπου όταν ενεργοποιηθεί από τον χρήστη εκτελείτε μία λειτουργία. Στην παρούσα εφαρμογή, στέλνεται το μήνυμα που έχει συνταχθεί στο TextBox.
- **TableArrangement:** Δημιουργείται ένας πίνακας όπου ο σχεδιαστής εισάγει τις λειτουργίες που επιθυμεί. Στον

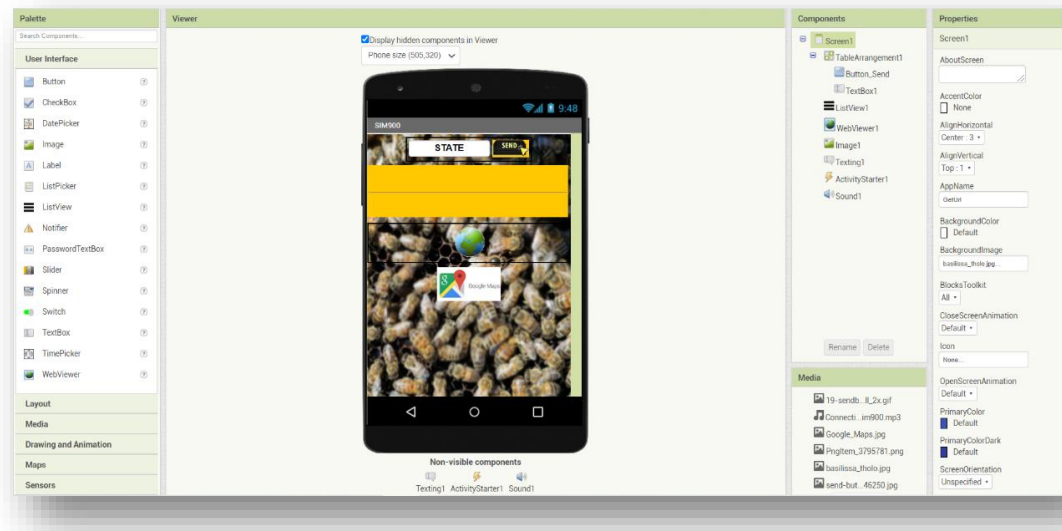
συγκεκριμένο πίνακα έχουν τοποθετηθεί οι λειτουργίες TextBox και Button, ώστε να εμφανίζονται στην ίδια νοητή γραμμή του πίνακα.

- ListView: Κατασκευάζεται μία λίστα όπου θα λαμβάνουν μέρος τα μηνύματα που στάλθηκαν από τον SIM900.
- WebViewer: Εμφανίζεται η ιστοσελίδα που έχει κληθεί από τις εντολές στο back end.
- Image: Εισαγωγή εικόνας στην εφαρμογή, η οποία όταν πατηθεί από τον χρήστη θα μεταφερθεί στο Google Maps
- Texting: Είναι αρμόδιο για την αποστολή και λήψη των μηνυμάτων
- ActivityStarter: Είναι υπεύθυνο για το άνοιγμα της παρούσας διεύθυνσης τοποθεσίας, σε μία άλλη εφαρμογή του χρήστη, στο google maps.
- Sound: Αναπαράγει ένα ηχητικό μήνυμα στο χρήστη, όταν πατηθεί το κουμπί αποστολής μηνύματος στον SIM900.

Στο δεξιό μέρος της οθόνης βρίσκονται τα συστατικά (Components) της εφαρμογής καθώς και οι ειδικότητες για το κάθε ένα ξεχωριστά. Οι παραμετροποιήσεις των ειδικοτήτων για την κάθε επιλογή λειτουργίας είναι:

- Screen: Εισαγωγή εικόνας στο Background (BackgroundImage)
- TableArrangement: Columns = 2, Height = 35 pixels, Width = 230 pixels, Rows = 1 , Visible = On
- Button: FontSize 14.0 , Height & Width = Automatic , Image = (Ανέβασμα εικόνας «Send»)
- TextBox: FontSize = 14.0, Height = Fill parent, Width = 40 percent, Text = «STATE», Visible = On
- ListView: Height = 20 percent, Width = Fill parent, TextSize 22 , Visible = Off
- WebViewer: FollowLinks = On, Height = Fill parent, Width = Fill parent, Visible = Off

- Image: Clickable = On, Height = Automatic, Width = Automatic, Picture = (Ανέβασμα εικόνας «Google Maps»), Visible = Off
- Texting: ReceivingEnabled = Always
- ActivityStarter: Action = android.intent.action.VIEW
- Sound: Source = (Εισαγωγή audio clip τύπου mp3)



Εικόνα 3.32 : MIT App Inventor front end

Εφόσον έχει γίνει μία πλήρης καθοδήγηση του front end, σειρά έχει το back end, το οποίο αναφέρεται στην λειτουργία ενός server, μιας βάσης δεδομένων που λειτουργούν στο παρασκήνιο της εφαρμογής, με αποτέλεσμα την παράδοση πληροφοριών στον εκάστοτε χρήστη. Στο back end βρίσκονται κομμάτια γραφικού κώδικα υπό την μορφή «παζλ», τα blocks, τα οποία εφόσον συνδεθούν σωστά παράγουν την λειτουργία της εφαρμογής.

Στην εικόνα 3.31 παρουσιάζεται ο κώδικας υπό την μορφή γραφικού προγραμματισμού, ο οποίος αποτελείται από τρία βασικά μέρη.

Ειδικότερα στο Α μέρος γίνεται η αρχικοποίηση των τιμών. Οι μεταβλητές «msg», «Words» είναι αρμόδιες για την δημιουργία λιστών, ενώ οι «SpaceChar», «Word» και «URL» έχουν την τιμή « ».

Το Β μέρος ενεργοποιείται από την στιγμή που ο χρήστης πατήσει το κουμπί «Button_Send» και ευθύνεται για την αποστολή του μηνύματος προς τον αριθμό που κατέχει το SIM900. Συγκεκριμένα αποτελείται από τα εξής βήματα :

1. Ορίζεται ως μήνυμα το κείμενο που έχει γραφτεί στο TextBox1
2. Ορίζεται ως αριθμός τηλεφώνου το String "+3069XXXXXX"
3. Ορίζεται ως ήχος το αρχείο "Connectionwithsim900.mp3"
4. Αναπαράγεται ο ήχος που έχει οριστεί
5. Στέλνεται το μήνυμα

Το Γ μέρος εκτελείται εφόσον ληφθεί μήνυμα από το SIM900. Το μήνυμα που αποστέλεται από την συσκευή αποτελείται από κείμενο, καθώς και από ένα link που εμφανίζει την τοποθεσία του πειράματος. Σκοπός του παρόντος μέρους είναι η απομόνωση του link, με αποτέλεσμα να δίνεται η δυνατότητα στον WebViewer εμφανίσει την τοποθεσία αυτή. Τα βήματα αυτού του μέρους είναι:

1. Τοποθετείται στην λίστα «msg» το String «SIM900» και το μήνυμα που λήφθηκε.
2. Ενεργοποιείται η εμφάνιση της λίστας (ListView) στο front end.
3. Εισέρχονται τα στοιχεία της λίστας msg στην λίστα εμφάνισης (ListView).
4. Διασπάτε το μήνυμα που έχει ληφθεί (messageText), σε ξεχωριστά κομμάτια, κάθε φορά που εμφανίζεται ένα κενό (SpaceChar) στο συγκεκριμένο μήνυμα. Στην συνέχεια εισέρχονται στην κενή λίστα «Words».
5. Εμφανίζεται μια δομή επανάληψης όπου για κάθε «είδος» (item) της λίστας «Words» ελέγχεται αν υπάρχει στο συγκεκριμένο μέρος του μηνύματος, η λέξη «http».
6. Εφόσον βρεθεί το link τότε εμφανίζεται στο front end το WebViewer, όπου και ενεργοποιείται το συγκεκριμένο URL.
7. Εμφανίζεται η εικόνα Google_Maps

Στο τελευταίο μέρος (Δ) δίνεται η δυνατότητα στον χρήστη να ανοίξει το link που περιέχει την τοποθεσία της συσκευής και σε άλλη εφαρμογή, όπως το Google

Mars, απλά πατώντας πάνω στην εικόνα που εμφανίζεται. Τα βήματα για την υλοποίηση αυτή είναι:

1. Ορισμός λειτουργίας ActivityStarter1 όπου το DataUri παίρνει την τιμή του link τοποθεσίας
2. Εκτελείται η λειτουργία ActivityStarter1

The image displays a sequence of MIT App Inventor code blocks. Section A shows the initialization of global variables: 'msg' (empty list), 'Words' (empty list), 'SpaceChar' (space character), 'Word' (empty string), and 'URL' (empty string). Section B is a 'when Button_Send .Click' event handler that sets the message, phone number, and sound source, then plays the sound and sends the message. Section C is a 'when Texting1 .MessageReceived' event handler that processes the received message: it adds the message to a list with a 'SIM900' prefix and a newline, sets the list's visibility and elements, splits the message into words at the space character, and iterates through each word. If a word contains 'http', it sets the WebView's visibility, home URL, and Google Maps' visibility. Section D is a 'when Google_Maps .Click' event handler that sets the ActivityStarter1's DataUri to the selected word and starts the activity.

A

B

C

D

Εικόνα 3.33 : MIT App Inventor program

Κεφάλαιο 4: Επίλογος

4.1 Εισαγωγή

Στο τελευταίο κεφάλαιο της παρούσας διπλωματικής εργασίας αρχικά γίνεται μία σύνοψη αυτής, καταγράφονται τα προβλήματα που προέκυψαν κατά την εκπόνησή της, οι αντίστοιχες λύσεις που επινοήθηκαν, τα συμπεράσματα που εξάχθηκαν καθώς και προτάσεις για μελλοντική εξέλιξη του συστήματος που αναπτύχθηκε.

4.2 Σύνοψη εργασίας

Στην παρούσα διπλωματική εργασία πραγματοποιήθηκε η σχεδίαση και ανάπτυξη ενός συστήματος αυτοματισμού για την εποπτεία και τον έλεγχο των συνθηκών που επικρατούν σε μελισσοκομικές μονάδες. Το συγκεκριμένο σύστημα καταγράφει τις συνθήκες που επικρατούν σε ένα μελίσσι μέσω κατάλληλων αισθητήρων, και μέσω κεντρικού ελεγκτή επεξεργάζεται τα αντίστοιχα δεδομένα και παράγει κατάλληλα σήματα ελέγχου. Επιπρόσθετα μέσω κατάλληλης εφαρμογής που αναπτύχθηκε καθίσταται δυνατός ο απομακρυσμένος έλεγχος λειτουργίας από συσκευή ευφυούς κινητού τηλεφώνου.

4.3 Προβλήματα και λύσεις

Στην παρούσα διπλωματική εργασία, λοιπόν, εμφανίστηκαν προκλήσεις που έπρεπε να αντιμετωπιστούν. Μία εξ αυτών ήταν στις μετρήσεις του βάρους. Τα καλώδια των αισθητήρων βάρους είναι αρκετά λεπτά και ευαίσθητα με αποτέλεσμα να χρειάζεται προσοχή στην συγκόλλησή τους. Στην προκειμένη περίπτωση η συγκόλληση δεν έγινε σωστά την πρώτη φορά, με αποτέλεσμα να μην έχει γίνει σωστή συνδεσμολογία, δίνοντας λάθος μετρήσεις στην σειριακή επικοινωνία του Arduino. Όμως, τα προβλήματα με τη διαδικασία ζύγισης δεν τελείωσαν εκεί. Όταν είχαν συνδεθεί όλα τα απαραίτητα εξαρτήματα στην τελική κατασκευή, παρ' όλο που είχε ελεγχθεί εξονυχιστικά το κάθε κομμάτι του κώδικα για την εκάστοτε λειτουργία, την στιγμή που καλούνταν το Arduino Uno να δώσει την τιμή του βάρους και την τοποθεσία, τότε δίνονταν στον χρήστη μία λανθασμένη

μέτρηση του βάρους. Για την λύση του προβλήματος αυτού, ενσωματώθηκε στην τελική συσκευή το Arduino Nano, που ήταν αποκλειστικά αρμόδιο για την μέτρηση του βάρους και την αποστολή της τιμής αυτού, στον Master όταν αυτή ζητηθεί.

Το επόμενο εμπόδιο που αντιμετωπίστηκε αφορούσε την SIM900 συσκευή. Έχοντας δημιουργήσει τον κώδικα για την συσκευή, καθώς και την απαιτούμενη συνδεσμολογία, όταν καλούνταν να σταλθεί μήνυμα ή να πραγματοποιηθεί κλήση στον χρήστη, δεν πραγματοποιούνταν τίποτα από τα δύο. Αυτό συνέβαινε επειδή δεν υπήρχε η απαραίτητη τροφοδοσία της συσκευής. Τοποθετώντας ένα τροφοδοτικό στον SIM900, για την παροχή περαιτέρω ρεύματος, τότε ήταν ικανό να πραγματοποιήσει την εντολές που καλούσαν να εκτελέσει.

Ένα επιπρόσθετο σημαντικό πρόβλημα που επιλύθηκε ήταν μεταξύ της GPS συσκευής και του SIM900, καθώς και οι δύο συσκευές απαιτούν την έναρξη μιας σειριακής επικοινωνίας με το Arduino Uno. Κάνοντας χρήση της βιβλιοθήκης SoftwareSerial, δίνεται η δυνατότητα για επιτευχθεί σειριακή επικοινωνία στους ψηφιακούς ακροδέκτες, αλλά δεν είναι επιτρεπτό να επιτευχθεί ταυτόχρονη λήψη δεδομένων. Για την αντιμετώπιση του προβλήματος δημιουργήθηκαν συναρτήσεις για την κάθε λειτουργία της τελικής συσκευής, ώστε να διαχωριστούν οι πολυάριθμες εντολές των συσκευών που απαιτούν την χρήση της βιβλιοθήκης. Στην συνέχεια μέσω της συνάρτησης «listen()» δίνεται η πρόσβαση στην συσκευή που την καλεί (πχ. SIM900.listen()) να λαμβάνει δεδομένα, αναγκάζοντας τις συσκευές που χρησιμοποιούν την SoftwareSerial να μένουν σε αναμονή.

4.4 Συμπεράσματα

Στην παρούσα διπλωματική εργασία καταβλήθηκε προσπάθεια να δημιουργηθεί ένα σύστημα ικανό μπορεί να υποστηρίξει τη μελισσοκομία.

Όσον αφορά τις χρήσιμες μετρήσεις των αισθητήρων, εκτός από την σωματική κούραση που πρέπει να υποστεί για τον χρονοβόρο έλεγχο των κυψελών, απαιτείται μεγάλη σπατάλη χρηματικού ποσού μέχρι να βρεθεί ο μελισσοκόμος στην τοποθεσία των κυψελών. Με λίγα λόγια τα έξοδα εργασίας και μεταφοράς αποτελούν μεγάλο πλήγμα στον χώρο της μελισσοκομίας. Ιδιαίτερα σημαντικό οικονομικό πλήγμα αποτελούν επίσης οι κλοπές των κυψελών. Οι κλοπές δεν αποτελούν ένα σπάνιο φαινόμενο στην ζωή του παραγωγού, καθώς οι κυψέλες βρίσκονται σε απομονωμένα μέρη. Το να αποτραπεί η κλοπή είναι σχεδόν

ακατόρθωτο γιατί ακόμα και να ηγήσει μια σειράνα μέσω αυτοματοποιημένης κατασκευής δεν θα εμποδίσει τον επίδοξο κλέφτη να ολοκληρώσει την πράξη του. Παρ' όλα αυτά μέσω της τελικής συσκευής, είναι δυνατή η εύρεση των κλοπιμαίων δείχνοντας το στίγμα στο δορυφορικό χάρτη με αρκετή ακρίβεια. Συμπερασματικά, δεν θα ήταν υπερβολικό να ειπωθεί ότι οι αυτοματισμοί στο τομέα της μελισσοκομίας αποτελούν μία απαραίτητη προϋπόθεση τόσο για την δυνητική αύξηση του παραγόμενου προϊόντος, όσο και για τη μείωση του κόστους.

4.5 Προτάσεις για μελλοντική εξέλιξη

Θα ήταν αδύνατον να δημιουργηθεί μια συσκευή στην οποία να μην υπάρχουν περιθώρια εξέλιξης και βελτίωσης στο μέλλον. Η πρόοδος της τεχνολογίας επιφέρει νέες λύσεις και δυνατότητες σε οποιαδήποτε κατασκευή, καλώντας τον δημιουργό συχνά σε παραμετροποιήσεις προσφέροντας νέες, καλύτερες επιδώσεις.

Η κατασκευή που δημιουργήθηκε στην παρούσα διπλωματική προσφέρει λύσεις και ευκολίες στον εκάστοτε χρήστη. Παρ' όλα αυτά έχοντας ως προϋπόθεση την δημιουργία μιας κατασκευής χαμηλού κόστους, αυτό έχει ως αποτέλεσμα να δημιουργούνται περιθώρια βελτίωσης.

Μια βελτίωση που θα μπορούσε να επιβληθεί στο παρόν έργο είναι η τοποθέτηση διαφορετικού μοντέλου Arduino. Συγκεκριμένα η χρήση ενός Arduino Mega, θα μπορούσε να επιφέρει καλύτερα αποτελέσματα, καθώς δεν θα απαιτούνταν να γίνει εγκατάσταση ενός δεύτερου Arduino ως Slave. Θα ήταν αρκετά ικανή στο να διαχειριστεί όλες τις απαραίτητες λειτουργίες της συσκευής, καταναλώνοντας λιγότερο ποσοστό ρεύματος καθώς και λιγότερες εντολές κώδικα.

Ως μία μελλοντική εξέλιξη θα μπορούσε να χαρακτηριστεί και η προσθήκη ενός πάνελ ηλιακού συλλέκτη. Το ηλιακό πάνελ θα έδινε την δυνατότητα στο έργο να αποκτήσει ενεργειακή αυτονομία καθώς θα φόρτιζε την υποτιθέμενη μπαταρία της συσκευής, όταν θα ήταν αυτό δυνατό. Επίσης θα ήταν απαραίτητη η ένδειξη του ποσοστού της μπαταρίας, ώστε να πληροφορεί τον χρήστη για το πότε είναι αναγκαία η αλλαγή αυτής.

Επίσης, η προσθήκη επιπλέον αισθητήρων θα μπορούσε να βελτιώσει περαιτέρω την συσκευή. Η τοποθέτηση εξωτερικού αισθητήρα θερμοκρασίας και αισθητήρα βροχής θα παρουσιάζεσαι μία πλήρη εικόνα καιρικών συνθήκων που θα ήτα πολύ χρήσιμη για την ζωή των μελισσών. Όσο αφορά το κομμάτι της ασφάλειας, θα μπορούσε να εγκατασταθεί μία κάμερα, όπου όταν παραβιαστεί η κυψέλη, να ενεργοποιείται και να τραβά φωτογραφίες οι οποίες θα αποθηκεύονται σε μία κάρτα SD. Αυτό θα έδινε την δυνατότητα στον χρήστη να κατέχει αποδεικτικό υλικό που θα μπορούσε να χρησιμοποιήσει στην συνέχεια.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] Μ. Δ. Υφαντίδης, (2005), Η σύγχρονη μελισσοκομία ως επιστήμη και πράξη
- [2] Γ. Τουφεξής, (1922), Μελισσοκομία
- [3] Μ. Δ. Δρούλου, Μέλισσα και Θρησκεία. Διαθέσιμο στο διαδίκτυο: http://theologoi-kritis.sch.gr/index.php?option=com_content&view=article&id=821:2014-11-12-21-37-53&catid=65:2008-12-29-20-40-10&Itemid=54 (προσπεράστηκε στις 7/11/20)
- [4] Κ. Παπαδάκη-Ήρινα, Η Μέλισσα ως ιερό σύμβολο μέσα στους αιώνες. Διαθέσιμο στο διαδίκτυο: <https://melissokomianet.gr/i-melissa-os-iero-sumvolo-mesa-stous-aiones/> (προσπεράστηκε στις 7/11/20)
- [5] Κ. Ρωμαίος (1969), Ελλάδα, μεγάλη εικονογραφημένη λαογραφία, γεωγραφία, ιστορία, Τόμος Β΄
- [6] Clement, H. (2007). Σύγχρονη μελισσοκομία. Αθήνα: ΨΥΧΑΛΟΥ.
- [7] Νικ. Ι. Νικολαΐδης, (2011), Μελισσοκομία: Σύγχρονες Μέθοδοι Εντατικής Εκμετάλλευσης
- [8] Α. Θ. Θρασύβουλος, (2012), Πρακτική μελισσοκομία
- [9] Αριστοτέλης, Πολιτικά, Τόμος Α
- [10] Α. Μόσχος, Γιατί οι μέλισσες αγαπούν τα εξάγωνα;. Διαθέσιμο στο διαδίκτυο: <https://melissokomianet.gr/melisses-eksagona/> (προσπεράστηκε στις 10/11/20)
- [11] Ε. Αλυσσανδράκης, (2007), Μελισσοκομία
- [12] Χ. Μανάβη, Α. Παπαδόπουλος, Γ. Μήτσιου-Πολυχρονίδου (2019), Το μέλι στην τοπική οικονομία από την αρχαιότητα ως σήμερα
- [13] Α. Βαφία, Η πρόπολη. Διαθέσιμο στο διαδίκτυο: http://melissokomia.com/aggeliki_vafia/propolis.html (προσπεράστηκε στις 10/11/20)
- [14] W. Ritter, (2007), Ασθένειες των μελισσών
- [15] Υπουργείο Αγροτικής Ανάπτυξης και Τροφίμων, Πληροφορίες για τη μελισσοκομία. Διαθέσιμο: http://www.minagric.gr/images/stories/docs/agrotis/MeliMelissokomia/plirofories_gia_melissokomia.pdf (προσπεράστηκε στις 10/11/20)

- [16] European Commission (2020). Honey Market Presentation. Ανακτήθηκε από: [HONEY MARKET PRESENTATION EUROPEAN COMMISSION.pdf](https://ec.europa.eu/food/food/qualityandnutrition/quality/honey-market-presentation-european-commission.pdf).
- [17] H. Barragan, [The Untold History of Arduino \(arduinohistory.github.io\)](https://github.com/arduinohistory)
- [18] J. Maeda, (1999), Design By Numbers
- [19] J. Lahart, (2009), «Taking an Open-Source Approach to Hardware». The Wall Street Journal
- [20] J. Nussey, (2018), Arduino for dummies, 2η έκδοση
- [21] Arduino main page. Διαθέσιμο: <http://www.arduino.cc/> (προσπεράστηκε στις 12/12/20)
- [22] Σπ. Ρομπότης, (2014), Εφαρμογές μικροελεγκτών AVR, πτυχιακή εργασία, Τμήμα Ηλεκτρολόγων μηχανικών
- [23] What is Arduino?. Διαθέσιμο στο διαδίκτυο: <https://www.arduino.cc/en/Guide/Introduction#why-arduino> (προσπεράστηκε στις 15/12/2020)
- [24] Most Popular. Διαθέσιμο στο διαδίκτυο: <https://store.arduino.cc/arduino-genuino/most-popular> (προσπεράστηκε στις 15/12/2020)
- [25] ARDUINO MEGA 2560 REV3. Διαθέσιμο στο διαδίκτυο: <https://store.arduino.cc/arduino-mega-2560-rev3> (προσπεράστηκε στις 16/12/2020)
- [26] ARDUINO DUE. Διαθέσιμο στο διαδίκτυο: <https://store.arduino.cc/arduino-due> (προσπεράστηκε στις 16/12/2020)
- [27] ARDUINO NANO. Διαθέσιμο στο διαδίκτυο: <https://store.arduino.cc/arduino-nano> (προσπεράστηκε στις 16/12/2020)
- [28] ARDUINO LEONARDO WITH HEADERS. Διαθέσιμο στο διαδίκτυο: <https://store.arduino.cc/arduino-leonardo-with-headers> (προσπεράστηκε στις 17/12/2020)
- [29] ARDUINO MKR WIFI 1010. Διαθέσιμο στο διαδίκτυο: <https://store.arduino.cc/arduino-mkr-wifi-1010> (προσπεράστηκε στις 17/12/2020)
- [30] ARDUINO MOTOR SHIELD REV3. Διαθέσιμο στο διαδίκτυο: <https://store.arduino.cc/arduino-motor-shield->

- [rev3?queryID=06c16a5599d96b19afebe1464a8826bd](https://store.arduino.cc/arduino-ethernet-shield-2?queryID=06c16a5599d96b19afebe1464a8826bd) (προσπεράστηκε στις 17/12/2020)
- [31] ARDUINO ETHERNET SHIELD 2. Διαθέσιμο στο διαδίκτυο: <https://store.arduino.cc/arduino-ethernet-shield-2?queryID=06c16a5599d96b19afebe1464a8826bd> (προσπεράστηκε στις 18/12/2020)
- [32] ARDUINO EDUCATION SHIELD. Διαθέσιμο στο διαδίκτυο: <https://store.arduino.cc/arduino-education-shield?queryID=09182a65688ec4496eae53e7fbb4ad3a> (προσπεράστηκε στις 18/12/2020)
- [33] ARDUINO MKR RGB SHIELD. Διαθέσιμο στο διαδίκτυο: <https://store.arduino.cc/arduino-mkr-rgb-shield?queryID=2985bccb93c329d2d169fac35369bafb> (προσπεράστηκε στις 19/12/2020)
- [34] Arduino seminars ANALOGREAD. Διαθέσιμο στο διαδίκτυο: <https://arduino-seminars.gr/analogread/>. (προσπεράστηκε στις 19/12/2020)
- [35] Βυσανσιώτης, Σ. & Καρβουνίδου, Α. ΥΛΟΠΟΙΗΣΗ ΑΣΥΡΜΑΤΗΣ ΕΠΙΚΟΙΝΩΝΙΑΣ ΜΕ ΧΡΗΣΗ ΜΙΚΡΟΕΛΕΓΚΤΩΝ (Διπλωματική εργασία). ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΗΠΕΙΡΟΥ. ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ Τ.Ε <https://apothetirio.lib.uoi.gr/xmlui/bitstream/handle/123456789/5433/1426.pdf?sequence=1>
- [36] Β. Νούσης, (2019), Arduino για αρχάριους με έμφαση στη χρήση του στο εργαστήριο Φυσικών Επιστημών
- [37] Master Reader/Slave Sender. Διαθέσιμο στο διαδίκτυο: <https://www.arduino.cc/en/Tutorial/LibraryExamples/MasterReader> (προσπεράστηκε στις 20/12/2020)
- [38] Arduino 101: Timers and Interrupts. Διαθέσιμο στο διαδίκτυο: <https://www.robotshop.com/community/forum/t/arduino-101-timers-and-interrupts/13072> (προσπεράστηκε στις 20/12/2020)
- [39] SPI library. Διαθέσιμο στο διαδίκτυο: <https://www.arduino.cc/en/reference/SPI> (προσπεράστηκε στις 20/12/2020)
- [40] J. Blum, (2013), « Exploring Arduino®: Tools and Techniques for Engineering Wizardry»

- [41] E. Savasgard, (2014), «Arduino 101 Beginners Guide»
- [42] M.Banzi, (2011), « Getting Started with Arduino»
- [43] Arduino Libraries. Διαθέσιμο στο διαδίκτυο: <https://www.arduino.cc/en/hacking/libraries> (προσπεράστηκε στις 21/12/2020)
- [44] Arduino Web Editor. Διαθέσιμο στο διαδίκτυο: https://create.arduino.cc/projecthub/Arduino_Genuino/getting-started-with-arduino-web-editor-on-various-platforms-4b3e4a (προσπεράστηκε στις 14/1/2021)
- [45] MiniBlox. Διαθέσιμο στο διαδίκτυο: <https://en.wikipedia.org/wiki/Miniblog> (προσπεράστηκε στις 18/1/2021)
- [46] Επίσημη σελίδα MiniBlox. Διαθέσιμο στο διαδίκτυο: <http://blog.miniblog.org/2011/03/what-is-miniblog.html> (προσπεράστηκε στις 18/1/2021)
- [47] Root: A robot to teach coding from 4 to 99 years old. Διαθέσιμο στο διαδίκτυο: <http://blog.miniblog.org/> (προσπεράστηκε στις 20/1/2021)
- [48] [48] BASICS OF THE I2C COMMUNICATION PROTOCOL. Διαθέσιμο στο διαδίκτυο: <https://www.circuitbasics.com/basics-of-the-i2c-communication-protocol/> (προσπεράστηκε στις 20/2/21)
- [49] [49] Γέφυρα Ουίτστοουν. Διαθέσιμο στο διαδίκτυο: https://el.wikipedia.org/wiki/%CE%93%CE%AD%CF%86%CF%85%CF%81%CE%B1_%CE%9F%CF%85%CE%90%CF%84%CF%83%CF%84%CE%BF%CE%BF%CF%85%CE%BD (προσπεράστηκε στις 2/3/21)
- [50] [50] Wheatstone Bridge. Διαθέσιμο στο διαδίκτυο: <https://www.electronics-tutorials.ws/blog/wheatstone-bridge.html> (προσπεράστηκε στις 2/3/21)
- [51] [51] Send Receive SMS & Call with SIM900 GSM Shield & Arduino. Διαθέσιμο στο διαδίκτυο <https://lastminuteengineers.com/sim900-gsm-shield-arduino-tutorial/> (προσπεράστηκε στις 8/3/21)
- [52] [52] GSM SIM900 Arduino Daily SMS Message. Διαθέσιμο στο διαδίκτυο: <https://www.instructables.com/GSM-SIM900-Arduino-Daily-SMS-Message/> (προσπεράστηκε στις 10/3/21)

- [53] [53] Interface ublox NEO-6M GPS Module with Arduino. Διαθέσιμο στο διαδίκτυο: <https://lastminuteengineers.com/neo6m-gps-arduino-tutorial/> (προσπεράστηκε στις 20/3/21)
- [54] [54] MIT App Inventor About Us. Διαθέσιμο στο διαδίκτυο: <https://appinventor.mit.edu/about-us> (προσπεράστηκε στις 15/4/21)