



ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΔΙΑΤΜΗΜΑΤΙΚΟ ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ
ΣΠΟΥΔΩΝ
«ΠΛΗΡΟΦΟΡΙΚΗ ΚΑΙ ΥΠΟΛΟΓΙΣΤΙΚΗ ΒΙΟΙΑΤΡΙΚΗ»

**Κατασκευή ηλεκτρονικής διάταξης για συλλογή και επεξεργασία
βιοσημάτων που σχετίζονται με την ανίχνευση του
Μαθητικού – Φοιτητικού Άγχους**

Κόντος Ευστάθιος

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Υπεύθυνος

Δελημπασης Κωνσταντίνος

Λαμία, 2018



UNIVERSITY OF THESSALY

SCHOOL OF SCIENCE

INFORMATICS AND COMPUTATIONAL BIOMEDICINE

**Creation of an electronic layout for collection and processing of
bio-signals related to detection of student's stress**

Kontos Efstathios

Master thesis

Delimpasis Konstantinos

Lamia

2018



ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΔΙΑΤΜΗΜΑΤΙΚΟ ΜΕΤΑΠΤΥΧΙΑΚΟ ΠΡΟΓΡΑΜΜΑ ΠΛΗΡΟΦΟΡΙΚΗ ΚΑΙ

ΥΠΟΛΟΓΙΣΤΙΚΗ ΒΙΟΙΑΤΡΙΚΗ

ΚΑΤΕΥΘΥΝΣΗ

«ΥΠΟΛΟΓΙΣΤΙΚΗ ΙΑΤΡΙΚΗ ΚΑΙ ΒΙΟΛΟΓΙΑ»

Κατασκευή ηλεκτρονικής διάταξης για συλλογή και επεξεργασία

βιοσημάτων που σχετίζονται με την ανίχνευση του

Μαθητικού – Φοιτητικού Άγχους

Κόντος Ευστάθιος

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Επιβλέπων

Δελήμπασης Κωνσταντίνος

Λαμία, 2018

«Υπεύθυνη Δήλωση μη λογοκλοπής και ανάληψης προσωπικής ευθύνης»

Με πλήρη επίγνωση των συνεπειών του νόμου περί πνευματικών δικαιωμάτων, και γνωρίζοντας τις συνέπειες της λογοκλοπής, δηλώνω υπεύθυνα και ενυπογράφως ότι η παρούσα εργασία με τίτλο [«Κατασκευή ηλεκτρονικής διάταξης για συλλογή και επεξεργασία βιοσημάτων που σχετίζονται με την ανίχνευση του Μαθητικού – Φοιτητικού Άγχους»] αποτελεί προϊόν αυστηρά προσωπικής εργασίας και όλες οι πηγές από τις οποίες χρησιμοποίησα δεδομένα, ιδέες, φράσεις, προτάσεις ή λέξεις, είτε επακριβώς (όπως υπάρχουν στο πρωτότυπο ή μεταφρασμένες) είτε με παράφραση, έχουν δηλωθεί κατάλληλα και ευδιάκριτα στο κείμενο με την κατάλληλη παραπομπή και η σχετική αναφορά περιλαμβάνεται στο τμήμα των βιβλιογραφικών αναφορών με πλήρη περιγραφή. Αναλαμβάνω πλήρως, ατομικά και προσωπικά, όλες τις νομικές και διοικητικές συνέπειες που δύναται να προκύψουν στην περίπτωση κατά την οποία αποδειχθεί, διαχρονικά, ότι η εργασία αυτή ή τμήμα της δεν μου ανήκει διότι είναι προϊόν λογοκλοπής.

Ο ΔΗΛΩΝ

Ημερομηνία

Υπογραφή

**Κατασκευή ηλεκτρονικής διάταξης για συλλογή και επεξεργασία
βιοσημάτων που σχετίζονται με την ανίχνευση του
Μαθητικού – Φοιτητικού Άγχους**

Κόντος Ευστάθιος

Τριμελής Επιτροπή:

Κωνσταντίνος Δελημπασης, Επίκουρος Καθηγητής

Χαρης Σανδαλίδης , Επίκουρος Καθηγητής

Αθανάσιος Λουκόπουλος, Επίκουρος Καθηγητής

Επιστημονικός Σύμβουλος:

Ονοματεπώνυμο.....

Περίληψη

Στην εργασία αυτή παρουσιάζεται μία διάταξη για την καταγραφή σε πραγματικό χρόνο χαρακτηριστικών βιοσημάτων που υποδεικνύουν άγχος. Για τη συλλογή των βιομετρικών σημάτων χρησιμοποιήσαμε δύο αισθητήρες: Ο πρώτος μετρά και καταγράφει καρδιακούς παλμούς ενώ ο επόμενος την εφίδρωση του δέρματος. Και οι δύο αισθητήρες είναι απλοί στη λειτουργία τους και εφαρμόζονται στις άκρες των δακτύλων του ενός χεριού. Συνδυάζοντας αυτές τις δύο μετρήσεις μπορούμε να έχουμε ενδείξεις για την αύξηση του stress ενός μαθητή / φοιτητή. Οι τιμές που απεικονίζουμε με καθορισμένη δειγματοληψία καταγράφουν αντιδράσεις του οργανισμού που αφορούν αλλαγές στο καρδιακό ρυθμό και την εφίδρωση του ανθρώπινου δέρματος. Η βίωση έντονων συναισθηματικών καταστάσεων έχει αποδειχθεί ότι επηρεάζει άμεσα αυτά τα δύο χαρακτηριστικά. Έχουν γίνει πολλές προσεγγίσεις κατά το παρελθόν που αφορούν τη καταγραφή του άγχους υποκειμένων. Στην παρούσα διπλωματική, καταγράφουμε απεικονίζουμε και συγκρίνουμε παίρνοντας προσωπικά δείγματα και μετρήσεις από 8 εθελοντές. Στην εργασία αρχικά γίνεται μια αναφορά στα συναισθήματα στη δομή τους και στη συναισθηματική νοημοσύνη. Αμέσως μετά ορίζουμε το άγχος στη ζωή μας που εμπεριέχει το μαθητικό φοιτητικό άγχος στις εξετάσεις και σε κάθε είδους αξιολόγηση στην εκπαίδευση. Στο ίδιο κεφάλαιο υπάρχει αναφορά στη συναισθηματική υπολογιστική και στη φυσιολογική προσέγγιση αναγνώρισης συναισθημάτων. Σε επόμενο κεφάλαιο περιγράφονται η βιοανάδραση και οι βιοαισθητήρες με αναφορές στη φυσιολογία της καρδιάς και του δέρματος. Στο 3^ο κεφάλαιο υπάρχει εισήγηση για τη πλατφόρμα του Arduino, καθώς και τη γλώσσα προγραμματισμού processing. Η μεθοδολογία αποτελεί το 4^ο κεφάλαιο και εμπεριέχει βασικές κατασκευές και

κυκλώματα των αισθητήρων καθώς και τη προσαρμογή τους στα χέρια μας. Εδώ επίσης περιγράφουμε τους τρόπους μέτρησης και καταγραφής καθώς και τα αποτελέσματα της συσκευής μας. Στο τέλος υπάρχει το παράρτημα με διαγράμματα από τις αποθηκευμένες καταγραφές μετρήσεων των υποκειμένων, ο κώδικας των Arduino και ο κώδικας στη processing για την οπτικοποίηση των αποτελεσμάτων .

ABSTRACT

In this project, we record real-time features that indicate stress.

To identify these features (biometric signals), we used two sensors. The first one counts and records heartbeats while the second one the sweat of our skin. Both sensors have simple functions and are applied to the tips of the fingers of one of our hands.

By combining these two measurements we get indications of the stress levels of students. These measurements can identify reactions of the human body that affect the rate of the heart and the sweat on our skin. It has been shown that intense emotional states affect these two characteristics. There have been many searches about this subject in the past years. We, in our try to make some conclusions, we record and compare by taking personal samples and measurements by 8 volunteers.

This project starts by a reference at the emotions and their structure, and at the emotional intelligence. Afterwards we define the word anxiety and more specifically at the anxiety in our everyday life like the student's anxiety at his/hers exams or any kind of assessment in his/her education. In the same chapter we analyze the emotional computing and the normal sense of emotion recognition. In the next chapter we refer to bioreactions and biosensors and the physiology of the heart and the skin. In the 3rd chapter there is a proposition at the Arduino platform as well as the computer language processing. The methodology consists the 4th chapter and contains basic constructions and circuits of the sensors as well as their application in our hands. Moreover we describe the ways we can measure the subjects and the results of our device. Lastly there is an appendix with diagrams from our results, the Arduino code and the processing code.

Ευχαριστίες

Η παρούσα διπλωματική εργασία εκπονήθηκε στα πλαίσια του μεταπτυχιακού στο τμήμα Πληροφορικής και Υπολογιστικής Βιοιατρικής του Πανεπιστημίου Θεσσαλίας

Είχα την ευκαιρία να ασχοληθώ με ένα ενδιαφέρον θέμα που συνδέει άμεσα τον Εργασιακό μου βίο (Εκπαιδευτικός σε σχολική μονάδα της Β΄θμιας Εκπαίδευσης) με τη τεχνολογία και τη πληροφορική.

Θα ήθελα να ευχαριστήσω ιδιαίτερος το καθηγητή μου Κο Δελήμπαση Κωνσταντίνο που αποτελούσε πολύτιμο βοηθό καθ' όλη τη διάρκεια της εργασίας μου.

Τέλος θα ήθελα να ευχαριστήσω την οικογένεια μου και ιδιαιτέρως τη σύζυγό μου Μαρία Ολυμπία για την υποστήριξη που μου προσέφερε κατά τη διάρκεια της 2^{ης} φοιτητικής μας ζωής.

ΠΕΡΙΕΧΟΜΕΝΑ

| | |
|--|----|
| ΚΕΦΑΛΑΙΟ 1 ΣΥΝΑΙΣΘΗΜΑΤΑ ΚΑΙ ΑΓΧΟΣ | 14 |
| 1.1 Συναισθήματα | 15 |
| 1.2 Κατηγορίες και Δομή συναισθημάτων | 17 |
| 1.3 Συναισθηματική νοημοσύνη..... | 17 |
| 1.4 Συναισθήματα Μαθητή - Φοιτητή..... | 18 |
| 1.5 Η θεωρία της συμπεριφοράς του Watson..... | 18 |
| 1.6 Η θεωρία της ψυχανάλυσης των Freud, Jung, Adler και Erikson | 19 |
| 1.7 Η γενετική-γνωστική θεωρία του Piaget | 20 |
| 1.8 Άγχος | 21 |
| 1.8.1 Άγχος στη ζωή μας..... | 22 |
| 1.8.2 Άγχος εξετάσεων και Αξιολόγησης..... | 22 |
| 1.9 Συναισθηματικοί υπολογιστές – συναισθηματική υπολογιστική | 23 |
| 1.9.1 Τομείς της συναισθηματικής υπολογιστικής | 24 |
| 1.9.2 Μοντέλα ΣυναισθηματικήςΥπολογιστικής (AffectiveLearningModels) | 25 |
| 1.9.3 Τεχνολογικό υπόβαθρο της συναισθηματικής υπολογιστικής | 26 |
| 1.10 Αναγνώριση συναισθημάτων μέσω της ομιλίας | 28 |
| 1.10.1 Αναγνώριση συναισθημάτων μέσω ανάλυσης προσώπου | 29 |
| 2 ΚΕΦΑΛΑΙΟ 2 ΒΙΟΑΝΑΔΡΑΣΗ ΒΙΟΣΗΜΑΤΑ ΦΥΣΙΟΛΟΓΙΑ ΚΑΡΔΙΑΣ ΚΑΙ ΔΕΡΜΑΤΟΣ . | 31 |

| | | |
|-------|---|----|
| 2.1 | Βιοανάδραση | 32 |
| 2.2 | Βιοσήματα ή Βιολογικά σήματα | 33 |
| 2.3 | Κατηγορίες Βιοανάδρασης | 34 |
| 2.4 | Αξιοποίηση δεδομένων βιοανάδρασης και εξαγωγή συμπερασμάτων | 35 |
| 2.5 | Η φυσιολογία της καρδιάς..... | 36 |
| 2.5.1 | Καρδιακός ρυθμός..... | 37 |
| 2.6 | Ανατομία του Δέρματος..... | 39 |
| 2.6.1 | Επιδερμίδα..... | 40 |
| 2.6.2 | Κυρίως δέρμα | 40 |
| 2.6.3 | Υπόδερμα ή Υποδόριο Λίπος..... | 41 |
| 2.6.4 | Ιδρωτοποιοί Αδένες..... | 42 |
| 2.7 | Γενική Περιγραφή της Γαλβανικής Απόκρισης Δέρματος | 43 |
| 3 | ΚΕΦΑΛΑΙΟ 3 ΕΙΣΑΓΩΓΗ ΣΤΟ ARDUINO ΚΑΙ ΣΤΗ PROCESSING | 47 |
| 3.1 | Εισαγωγή στο Arduino | 48 |
| 3.2 | Arduino Family | 49 |
| 3.3 | Τι είναι η γλώσσα προγραμματισμού processing | 59 |
| 3.4 | Σύγκριση με άλλες γλώσσες..... | 61 |
| 3.5 | Βασικά στοιχεία της γλώσσας..... | 62 |
| 3.6 | Δημιουργία stand-alone εφαρμογών | 64 |

| | | |
|-------|--|-----------|
| 3.7 | Βιβλιοθήκες και εργαλεία | 64 |
| 4 | ΚΕΦΑΛΑΙΟ 4 ΜΕΘΟΔΟΛΟΓΙΑ – ΑΠΟΤΕΛΕΣΜΑΤΑ | 66 |
| 4.1 | Κατασκευή και Προσαρμογή του συστήματος μετρήσεων..... | 67 |
| 4.2 | Γενικός σχεδιασμός..... | 69 |
| 4.3 | Ο αισθητήρας Pulse Sensor | 70 |
| 4.4 | Ο αισθητήρας GSR | 77 |
| 4.5 | Η Προσαρμογή των βιοαισθητήρων στο χέρι μας..... | 84 |
| 4.6 | Γραφική αναπαράσταση βιοσημάτων..... | 84 |
| 4.7 | Προβλήματα που παρουσιάστηκαν..... | 90 |
| 4.8 | Συμπεράσματα..... | 91 |
| 4.9 | Μελλοντικοί στόχοι..... | 91 |
| | Βιβλιογραφία..... | 93 |
| 5 | ΠΑΡΑΡΤΗΜΑΤΑ..... | 95 |
| 5.1 | Διαγράμματα των 8 υποκειμένων από τις αποθηκευμένες τιμές των αισθητήρων..... | 95 |
| 5.2 | Κώδικας GSR..... | 101 |
| 5.2.1 | Arduino GSR Code..... | 101 |
| 5.2.2 | Processing Code GSR..... | 101 |
| 5.3 | Κώδικας Pulse Sensor..... | 104 |

| | | |
|-------|----------------------------------|-----|
| 5.3.1 | Pulse Sensor Arduino Code | 104 |
| 5.3.2 | PulseSensor Processing Code..... | 109 |
| 5.4 | DATA SHEETS..... | 117 |

ΚΕΦΑΛΑΙΟ 1

ΚΕΦΑΛΑΙΟ 1 ΣΥΝΑΙΣΘΗΜΑΤΑ ΚΑΙ ΑΓΧΟΣ

1.1 Συναισθήματα

Η συναισθηματική εμπειρία κατασκευάζεται μέσω μίας “συνεργασίας” ανάμεσα στο γνωστικό και το συναισθηματικό κομμάτι. Συνήθως μπορεί το ένα από τα δύο να υπερισχύσει. Τα συναισθήματα, εκτός από τη ρύθμιση του εσωτερικού κόσμου, δηλαδή την ενδοπροσωπική συμπεριφορά, παίζουν ρόλο και στις αλληλεπιδράσεις με τον εξωτερικό κόσμο, δηλαδή τις διαπροσωπικές σχέσεις (intrapersonal, interpersonal behavior). Εξάλλου τα συναισθήματα παίρνουν νόημα μέσω των διαπροσωπικών σχέσεων και δημιουργούν προσδοκίες για τις διαπροσωπικές επαφές στο μέλλον. Τα συναισθήματα έχουν κυρίαρχη θέση στην αλληλεπίδρασή μας με το περιβάλλον. Επηρεάζουν την αντιληπτική διαδικασία και επιφέρουν αλλαγές στη συμπεριφορά, στις εκφράσεις του προσώπου όπως και στη στάση σώματος. Το συναίσθημά είναι μία υποκειμενική αντίδραση απέναντι σε ένα γεγονός, η οποία εκφράζεται μέσα από φυσιολογικές, εμπειρικές και συμπεριφορικές αλλαγές. Τα συναισθήματα διαθέτουν αρχέγονα βιολογικά θεμέλια, πρόκειται για έναν πανάρχαιο κώδικα που μας κληροδότησαν οι απώτεροι πρόγονοί μας (Goleman 1995), είναι συνδεδεμένα με τον αποκαλούμενο “παλαιό” εγκέφαλο, ως μετέπειτα εξέλιξη απλών αρχέγονων μορφών αντιδράσεων του ανθρώπινου θυμικού σε διάφορες καταστάσεις (κυρίως κινδύνου). Κύρια ιδιότητα τους είναι η παγκοσμιότητα και η καθολικότητά τους. Το συναίσθημα πυροδοτεί τη δράση και προκαλεί την απαραίτητη εγρήγορση στο άτομο προκειμένου να αντιμετωπίσει τις προκλήσεις. Αφενός το άτομο σκέφτεται και αφετέρου νιώθει, συνυπάρχουν η συναισθηματική νοημοσύνη και η λογική, ο συγκινησιακός και ο λογικός νους. Τα συναισθήματα και η σκέψη είναι δηλαδή αλληλεξαρτώμενα. Η διανοητική ζωή και δραστηριότητα είναι πολύπλευρη συγκινησιακά. Η εξισορρόπηση θετικών και αρνητικών συναισθημάτων είναι ο τρόπος εξασφάλισης της ψυχικής ευεξίας (Goleman 1995). Τα συναισθήματα συνεπώς είναι αλληλεξαρτώμενα με την αντίληψη και την επηρεάζουν με τους ακόλουθους τρόπους κατά το Hoffman: α) θέτουν σε λειτουργία ή παρεμποδίζουν την επεξεργασία πληροφοριών, ενδεχομένως ακόμη και να επιβάλλουν στην επεξεργασία μια επιλεκτικότητα, β) οργανώνουν την ανάκληση πληροφοριών για καταστάσεις του παρελθόντος, οι οποίες συνδέονται με αυτά τα συναισθήματα, γ) συντελούν στη διαμόρφωση συναισθηματικά φορτισμένων κατηγοριών και σχημάτων. Δεν μπορεί να γίνει λόγος για απλή γραμμική σχέση αιτιώδους συνάφειας, αλλά για

αλληλοσυμπλήρωση στο πλαίσιο της ενιαίας αναπτυξιακής πορείας του ατόμου, για μηχανισμούς που αλληλοσυμπληρώνονται (Barbalet, J. (1998).

Δεν είναι εύκολο να δοθεί ένας ολοκληρωμένος ορισμός για το συναίσθημα (emotion) εξαιτίας της πληθώρας των συναισθηματικών καταστάσεων που μπορεί να βιώσει κάθε άτομο. Σε μια προσπάθεια απόδοσης ορισμού θα μπορούσε κανείς να πει ότι πρόκειται για την ψυχική κατάσταση στην οποία το άτομο νιώθει θετικά ή αρνητικά αισθήματα αναλόγως με τις εκάστοτε συνθήκες. Πολλοί μελετητές θεωρούν ότι συναισθήματα αποτελούν η χαρά, το ενδιαφέρον, η λύπη, η έκπληξη, η ντροπή, ο θυμός, άλλοι όμως κάνουν αναφορά σε μικτές ψυχικές καταστάσεις όπως για παράδειγμα η αγάπη που περιλαμβάνει την χαρά και την αποδοχή, ή τη φιλία που περιλαμβάνει την αμοιβαιότητα αποδοχής και χαράς. Η φύση των συναισθημάτων αποτελείται από πέντε κύρια χαρακτηριστικά (Damasio, A. R. 2003):

- Τα ερεθίσματα που τα δημιουργούν (εξωτερικές εικόνες, παραστάσεις),
- οι αντιδράσεις του νευρικού συστήματος,
- οι αλλαγές στις φυσιολογικές λειτουργίες του οργανισμού (παλμοί, πίεση αίματος),
- οι εξωτερικές μεταβολές του ατόμου (φωνή, συμπεριφορά) και
- οι εμπειρίες του προσώπου σαν αποτέλεσμα της βίωσης των συγκεκριμένων συναισθημάτων.

Κατά τον Goleman (1995) τα κύρια συναισθήματα είναι η λύπη, η χαρά, η έκπληξη, η αηδία, ο θυμός και ο φόβος. Υφίσταται όμως και η οικογένεια των συναισθημάτων όπως π.χ. το συναίσθημα της αγάπης εμπεριέχει και την αποδοχή, την αφοσίωση, την τρυφερότητα και ο θυμός εμπεριέχει τον εκνευρισμό, την οργή την εχθρότητα κ.α. δηλαδή ποικίλα συναισθήματα με μεταλλάξεις και προσμείξεις. Οι Leutenber & Khalsa-Korb (1999) διεύρυναν τον κατάλογο από τα 16 στα 63 συναισθήματα. Όταν απευθυνόμαστε σε μικρά παιδιά τα 16 συναισθήματα που χρησιμοποιούμε είναι η αγάπη, η απογοήτευση, η απόρριψη, ο εγωισμός, η ειλικρίνεια, η ενοχή, η επιθετικότητα, η ευγνωμοσύνη, η ζήλια, ο θυμός, η μοναξιά, η ντροπή, η περηφάνια, η συγκίνηση, ο φόβος και η χαρά. Σε πιο μεγάλα παιδιά και σε εφήβους χρησιμοποιούμε πιο σύνθετα συναισθήματα όπως ευθυμία, ευγνωμοσύνη, κατήφεια, κακία, κόπωση, κολακεία, οδύνη, ματαιώση, προδοσία, πλήξη, συγκλονισμός, σύγχυση, συγκίνηση κ.α.

1.2 Κατηγορίες και Δομή συναισθημάτων

Τα συναισθήματα κατηγοριοποιούνται σε ατομικά, κοινωνικά, βιολογικά και ανώτερα. Με αφετηρία τα ατομικά συναισθήματα (αυτοσυναισθήματα) που έχουν ως σημείο αναφοράς το ίδιο το άτομο και τι νιώθει μπορούμε να φτάσουμε στο θετικό ή αρνητικό αυτοσυναισθήματα. Σε ακραίες καταστάσεις αυτό μπορεί να καταλήξει σε σύμπλεγμα ανωτερότητας ή κατωτερότητας αντιστοίχως. Επιπρόσθετα έχουμε τα κοινωνικά συναισθήματα που έχουν ως πεδίο αναφοράς τον άλλον και τα συναισθήματα που δημιουργούνται μέσα από συναναστροφή αυτή. Όταν η επικοινωνία είναι εφικτή προξενεί θετικά κοινωνικά συναισθήματα όπως: αποδοχή, φιλία, συμπάθεια που είναι απαραίτητα για την ψυχική υγεία του ατόμου. Αντιθέτως τα αρνητικά κοινωνικά συναισθήματα προκαλούν τη διάσπαση της ψυχικής ηρεμίας του ατόμου οδηγώντας το στο θυμό, την οργή, την επιθετικότητα κ.τ.λ. Όσον αφορά τα βιολογικά συναισθήματα που προξενούνται στο άτομο αυτά σχετίζονται με βιολογικές λειτουργίες όπως η πείνα, η δίψα, ο κορεσμός και όποια συναισθήματα δημιουργούνται στο άτομο ως συνέπεια αυτών. Τέλος, τα ανώτερα συναισθήματα οποία χωρίζονται σε ηθικά, θεωρητικά, θρησκευτικά, καλαισθητικά, εθνικά, ανθρωπιστικά, και πατριωτικά είναι συνδεδεμένα με ανώτερες γνωστικές καταστάσεις και τον αξιακό κώδικα.(Creemers, B.P.M. (1994).

1.3 Συναισθηματική νοημοσύνη

Η Συναισθηματική Νοημοσύνη (Emotional Intelligence, Emotional Quotient -EQ) είναι μια καινούργια σχετικά έννοια, την οποία πρότειναν οι επιστήμονες, για να περιγράψει χαρακτηριστικά που κατά τα φαινόμενα παίζουν βασικό ρόλο στην επιτυχία των ατόμων στην εργασία και τη ζωή τους γενικότερα.

Ο όρος συναισθηματική νοημοσύνη αναφέρεται πρώτη φορά επισήμως από τους Mayer, DiPaolo και Salovey (1990) και επαναπροσδιορίστηκε από τους Salovey και Sluyter (1997) τον όρο καταλήγοντας στην ύπαρξη τεσσάρων δεξιοτήτων, της πρόσβασης, της αντίληψης, της ρύθμισης και της κατανόησης και των συναισθημάτων. Ο όρος συναισθηματική νοημοσύνη έχει ως σκοπό να τονίσει την άμεση σχέση που υπάρχει ανάμεσα στις γνωστικές λειτουργίες και στο συναίσθημα. Ο Goleman στο βιβλίο του

«Συναισθηματική νοημοσύνη», υποστηρίζει πως η επιτυχία στη ζωή δεν είναι κατ' ανάγκην συνάρτηση του δείκτη νοημοσύνης αλλά της ικανότητας για αυτοπεποίθηση, αυτογνωσία, αυτοέλεγχο, κατανόηση και ενδιαφέρον. Η μελέτη του έδειξε πως τα κυκλώματα του εγκεφάλου διαμορφώνονται από την παιδική κιόλας ηλικία όπου αναπτύσσονται τέτοιου είδους δεξιότητες. Η θεωρία του Goleman (1998) συνδέθηκε με την θεωρία του Gardner(1983) περί πολλαπλής νοημοσύνης και όχι αδίκως καθώς και αυτός καταπιάστηκε με το θέμα των ενδοατομικών και διατομικών σχέσεων των παιδιών με τον εαυτό τους και τους γύρω τους. Κατά το Maslow, για να οδηγηθεί το πρόσωπο στην αυτοπραγμάτωση που είναι ο κυρίαρχος στόχος του πρέπει να ικανοποιηθούν οι βασικές ανάγκες του σε νερό, τροφή, στέγη και μετά οι συναισθηματικές του ανάγκες. Οι πνευματικές ανάγκες του αναπτύσσονται αφού έχουν πρώτα ικανοποιηθεί οι συναισθηματικές ανάγκες του.

1.4 Συναισθήματα Μαθητή - Φοιτητή

Η προέλευση των συναισθημάτων, το πώς αυτά γεννιούνται, αποτελεί ένα αναπτυξιακό ζήτημα. Δεν υπάρχει γέννηση των συναισθημάτων από το μηδέν, ούτε υπάρχουν ώριμα συναισθήματα σε λανθάνουσα μορφή κατά τους πρώτους μήνες της ζωής που θα εκδηλωθούν μετέπειτα στην πλήρη μορφή τους. Ανακύπτουν μέσω μιας διαδικασίας διαφοροποίησης των προ-συναισθηματικών αντιδράσεων, των προδρομικών συναισθημάτων, μέσω της ενδυνάμωσης του νοήματος και μέσω των διαρκώς μεταβαλλόμενων καταστάσεων διέγερσης. (Leal, M. R. M. (2002). Επομένως η θεώρηση των συναισθημάτων γίνεται μέσω μιας αναπτυξιακής διαδικασίας, κάνοντας σύνδεση των πρώιμων εκφάνσεων και των ώριμων μορφών.

Τα αρχικά στάδια της ζωής είναι αυτά που καθορίζουν την ανάπτυξη των συναισθηματικών ικανοτήτων και οι μεταγενέστερες δεξιότητες χτίζονται πάνω στις πρώτες (Goleman 1995).

1.5 Η θεωρία της συμπεριφοράς του Watson.

Ο Watson Αμερικανός ψυχολόγος που βασίστηκε στα πειράγματα του Pavlov και τη γνωστή θεωρία των εξαρτημένων ανακλαστικών συσχέτισε την αντίδραση με την παρουσία ενός επαναλαμβανόμενου ερεθίσματος καθιερώνοντας την θεωρία της συμπεριφοράς (Behaviorism). Την εν λόγω θεωρία μετέφερε και έθεσε σε εφαρμογή στις

συναισθηματικές εμπειρίες του ανθρώπου. Διεξάγοντας το περίφημο πείραμά του με τον Albert, 11 μηνών, απέδειξε ότι ο φόβος είναι αποτέλεσμα μάθησης. Μέσω των αντιδράσεων του παιδιού οδηγήθηκε στο συμπέρασμα ότι τα συναισθήματα που αναπτύσσονται από το άτομο είναι εξαρτώμενα από τα ερεθίσματα που υφίστανται στο περιβάλλον του κατά την ανάπτυξή του. Επομένως, αντικείμενα συνδεδεμένα με ευχάριστα γεγονότα δημιουργούν ευχάριστα συναισθήματα. Αντίστοιχα, αρνητικά γεγονότα προκαλούν δυσάρεστα συναισθήματα και κατά συνέπεια το άτομο απομακρύνεται από αυτά και τα αποφεύγει.

1.6 Η θεωρία της ψυχανάλυσης των Freud, Jung, Adler και Erikson

Οι βασικότερες θεωρίες που αναπτύχθηκαν στον χώρο της Ψυχανάλυσης όσον αφορά την συναισθηματική ανάπτυξη είναι η ψυχαναλυτική θεωρία του Sigmund Freud, η Αναλυτική θεωρία του C.Jung, η Ατομική θεωρία του A.Adler και η Βιοκοινωνική θεωρία του E.Erikson. Βάσει της ψυχαναλυτικής θεωρίας του Freud, το βρέφος δε διαθέτει συναισθήματα στην αρχή της ζωής του. Κατά τον δεύτερο - τρίτο μήνα όπου και αναπτύσσονται οι λειτουργίες του Εγώ (μνήμη, αντίληψη, αυτογνωσία, λογική σκέψη) γίνεται η ανάπτυξη των συναισθημάτων όπως η ενοχή, η υπερηφάνεια, η ντροπή. Στην ανάπτυξη των συναισθημάτων, το Εγώ διαδραματίζει βασικό ρόλο. Έως εκείνο το σημείο, οι συναισθηματικές καταστάσεις που βιώνονταν από το βρέφος βρίσκονταν σε αδιαφοροποίητη μορφή στο ασυνείδητο. Παρά την αμφισβήτηση που δέχτηκε από αρκετούς επιστήμονες, ο Freud ήταν εκείνος που παρουσίασε τα στάδια της ψυχοσεξουαλικής ανάπτυξης του ατόμου υποστηρίζοντας ότι το πιο έντονο συναίσθημα του παιδιού κατά την ανάπτυξη του είναι ο έρωτας. Σύμφωνα με την Αναλυτική θεωρία του Jung, η οποία και υιοθετεί ορισμένα σημεία από τη θεωρία του Freud, υποστηρίζεται η ύπαρξη του συνειδητού και ασυνείδητου μέρους της ψυχικής δομής του ατόμου. Υπάρχουν δύο κατηγορίες στις οποίες χωρίζεται το ασυνείδητο, στο προσωπικό και στο κοινωνικό ασυνείδητο. Από τις δύο αυτές κατηγορίες προκύπτουν δύο τύποι ανθρώπου, ο εσωστρεφής και ο εξωστρεφής. Οι τύποι αυτοί, αναλόγως με το ποια λειτουργία επικρατεί (συναίσθημα, νόηση, ενόραση, αντίληψη) διαμορφώνονται οι αντίστοιχοι. Ο εσωστρεφής συναισθηματικός, διανοητικός, ενορατικός, αντιληπτικός και ο εξωστρεφής συναισθηματικός, διανοητικός, ενορατικός, αντιληπτικός τύπος. Κάθε

τύπος κυριαρχείται από έντονα συναισθήματα αλλά η διαφορά έγκειται στο ότι ο εσωστρεφής δεν τα εκφράζει ενώ ο εξωστρεφής τα εκφράζει. Η Ατομική θεωρία του Adlerστηρίχθηκε στην άποψη ότι ο άνθρωπος από την στιγμή που γεννιέται επιζητά την κοινωνική αναγνώριση και την υπερίσχυση έναντι των συνομηλίκων του. Έτσι οδηγείται στη βίωση θετικής αυτοεικόνας (θετικό αυτοσυναίσθημα). Αυτό το συναίσθημα είναι πρωταρχικό κίνητρο στην ζωή του. Αντιθέτως, το αίσθημα της μειονεξίας κατά πάσα πιθανότητα θα έχει αρνητικές συνέπειες στην ψυχοσύνθεσή του (νευρώσεις). Συνεπώς, οποιοδήποτε μειονέκτημα παρουσιαστεί οδηγεί το πρόσωπο να αναπληρώσει το κενό ή τη μειονεξία του μέσω της προβολής ενός θετικού γνωρίσματος της προσωπικότητας του. Ανάλογο παράδειγμα αποτελεί ο Beethoven, που παρά το μειονέκτημα πως ήταν κωφός, διέπρεψε δείχνοντας το σπουδαίο ταλέντο του στην μουσική. Ο Erikson στο βιβλίο του «Η παιδική ηλικία και η κοινωνία» τονίζει πως η προσωπικότητα του ατόμου αρχίζει να διαμορφώνεται από πολύ νωρίς και συνεχίζει να διαμορφώνεται μέχρι τα γηρατειά. Σε κάθε φάση της ζωής του το άτομο αντιμετωπίζει «κρίσεις», συγκρούσεις εσωτερικές τις οποίες πρέπει να επιλύσει. Αυτές ορίζονται από τον Erikson ως «κοινωνικο-συναισθηματικές κρίσεις του Εγώ». Το κάθε στάδιο αποτελούν δίπολα συναισθηματικών καταστάσεων, οι οποίες είναι αντίθετες μεταξύ τους (Αυτονομία-Αμφιβολία, Εμπιστοσύνη-Δυσπιστία, Φιλοπονία-Κατωτερότητα, Πρωτοβουλία-Ενοχή, Οικειότητα-Απομόνωση, Πληρότητα-Απελπισία, Παραγωγικότητα-Στασιμότητα). Μπορούμε εύκολα να αντιληφθούμε τη διαδρομή της συναισθηματικής ανάπτυξης του παιδιού μέσα από τα στάδια.

1.7 Η γενετική-γνωστική θεωρία του Piaget

Ο Piaget συνδέει τις γνωστικές και συναισθηματικές λειτουργίες του ανθρώπου στη θεωρία του. Αυτά συντελούν τα αποκαλούμενα ψυχογνωστικά σχήματα τα οποία ενεργοποιούνται με τη λήψη κατάλληλου ερεθίσματος. Σε οποιαδήποτε πράξη πραγματοποιεί το άτομο, πάντοτε εμπεριέχονται συναισθηματικά κίνητρα. Συνεπώς τα συναισθήματα δρουν σε συνεργασία με τις γνωστικές λειτουργίες και καθώς περνάει ο χρόνος μετασχηματίζονται.

1.8 Άγχος

Ο Selye (1956, 1973), που εισήγαγε την έννοια του στρες ως βιολογικό φαινόμενο, το χαρακτήρισε ως τη «γενικευμένη αντίδραση του οργανισμού σε κάθε είδους πίεσης». Ο Spielberger (1972, 1979) ισχυρίστηκε ότι το στρες αποτελεί τη σωματική και ψυχολογική αντίδραση σε εξωτερικά ερεθίσματα από τα οποία απειλείται η ισορροπία του οργανισμού. Με την εμφάνιση αυτών των «απειλητικών» ερεθισμάτων, προξενείτε η βιολογική απόκριση του οργανισμού μέσα από μια διαδικασία διαδοχικών αντιδράσεων που λαμβάνουν χώρα στο νευρικό και το ενδοκρινικό σύστημα, με συνέπεια την εκδήλωση αντιδράσεων του σώματος όπως η αύξηση παλμών, η ταχύπνοια, η εφίδρωση κ.α. Πέραν όμως των σωματικών εκδηλώσεων, το στρες έχει και εκδηλώσεις ψυχολογικές, εκ των οποίων, μία είναι και το άγχος. Λόγω των αρκετών εννοιολογικών ορισμών του άγχους, συναντώνται στη βιβλιογραφία αντικρουόμενοι μεταξύ τους ορισμοί. Παρόλα αυτά, έναν ευρέως αποδεκτό για το άγχος ορισμό, αποτελεί ο ορισμός του ως «μια δυσάρεστη συναισθηματική κατάσταση ή συνθήκη, η οποία χαρακτηρίζεται από υποκειμενικά αισθήματα φόβου, έντασης και ανησυχίας όπως και από αυξημένη δραστηριότητα του αυτόνομου νευρικού συστήματος». Μπορεί οι περισσότερες απόπειρες ορισμού του άγχους να εστιάστηκαν κατά κύριο λόγο στις αρνητικές πλευρές του, όμως η παρουσία άγχους δεν αποτελεί αποκλειστικά αρνητικό παράγοντα στη ζωή ενός ατόμου. Αντιθέτως, είναι μία προειδοποίηση απέναντι στον κίνδυνο που θέτει τον οργανισμό σε εγρήγορση, προκειμένου να αντεπεξέλθει αποτελεσματικά και εγκαίρως στην όποια απειλητική για το άτομο περίσταση. Εν ολίγοις, η παρουσία του άγχους είναι αναγκαία στην εξέλιξη της ανθρώπινης ζωής. Επομένως δε χρειάζεται κανείς να αποφεύγει το άγχος αλλά να εντοπίσει και να καθορίσει σε τι βαθμό μπορεί να το ανεχτεί προκειμένου να λειτουργεί ως χρήσιμη κινητήρια δύναμη, καθώς, ουσιαστικά, η παρουσία του έχει σαν αποτέλεσμα να δρα κανείς στο μέγιστο των δυνατοτήτων του, επιτυγχάνοντας τη βέλτιστη επίδοση. Βάσει του νόμου YerkesDodson (1908), το καλύτερο δυνατό επίπεδο διέγερσης είναι αναγκαίο για να περατωθεί σωστότερα ένα έργο, όπως οι εξετάσεις. Όμως εάν το άγχος ή το επίπεδο διέγερσης ξεπερνά το βέλτιστο, οδηγεί σε μειωμένη επίδοση.

1.8.1 Άγχος στη ζωή μας

Παρότι το άγχος είναι ένα καθολικό ανθρώπινο συστατικό στοιχείο, χρειάζεται η ύπαρξη ορισμένων παραγόντων πρόκλησης προκειμένου να εκδηλωθεί και να βιωθεί. Αρκετές φορές όμως, για την εκδήλωσή του δεν αρκούν μονάχα αυτοί οι παράγοντες, αλλά πρέπει να βρεθεί και το κατάλληλο έδαφος. Εν τέλει, το ζήτημα του άγχους, είναι όπως και όλα τα ψυχοπαθολογικά φαινόμενα, μια δυναμική σχέση «περιβάλλοντος-εδάφους». (Harter, S., (1992).

1.8.2 Άγχος εξετάσεων και Αξιολόγησης

Επειδή το άγχος αποτελεί ένα συναίσθημα που εξαρτάται σε μεγάλο βαθμό από τη συνθήκη μέσα στην οποία εμφανίζεται, πρώτοι οι Mandler και Sarason (1952) επικεντρώθηκαν αποκλειστικά στην εμφάνιση του άγχους υπό συνθήκες εξέτασης. Ο γενικός ορισμός για το άγχος εξέτασης, που πρότεινε ο Dusek το 1980 ήταν ο εξής «Ένα δυσάρεστο συναίσθημα, το οποίο χαρακτηρίζεται από φυσιολογικά (σωματικά) και συμπεριφορικά επακόλουθα και βιώνεται, κυρίως, κατά τη διάρκεια μιας επίσημης εξέτασης ή και σε άλλες περιστάσεις αξιολόγησης». Η πλειοψηφία των αρχικών θεωριών του άγχους εξέτασης υιοθετεί τη διάκριση ανάμεσα στα δύο συστατικά του άγχους: ένα γνωστικό συστατικό, που είναι η ανησυχία, και ένα θυμικό συστατικό που είναι η συναισθηματικότητα. Αντίθετα με τις πρώτες μηχανιστικές προσεγγίσεις των μελετητών που θεώρησαν το άγχος εξέτασης μία σχετικά αδιαφοροποίητη έννοια, τα τελευταία δεκαπέντε χρόνια, τείνει το άγχος εξέτασης να αντιμετωπίζεται ως πολυδιάστατη και σύνθετη έννοια, που περιλαμβάνει μία σειρά αλληλοσχετιζόμενων γνωστικών, συμπεριφορικών και θυμικών συστατικών και αντιδράσεων. (Best, R. (2003)

1.9 Συναισθηματικοί υπολογιστές – συναισθηματική υπολογιστική

Εννοιολογικός ορισμός

Συναισθηματική υπολογιστική(affective computing) είναι ο κλάδος έρευνας και ανάπτυξης της τεχνητής νοημοσύνης που ασχολείται με το σχεδιασμό συσκευών και συστημάτων που έχουν τη δυνατότητα αναγνώρισης, ερμηνείας και επεξεργασίας των ανθρώπινων συναισθημάτων και αποσκοπεί στη δημιουργία μιας μηχανής η οποία θα μπορεί να ερμηνεύσει τη συναισθηματική κατάσταση του ανθρώπου και να δράσει αναλόγως. Πρόκειται για ένα διεπιστημονικό τομέα που υπάγεται στον ερευνητικό χώρο της επικοινωνίας ανθρώπου-μηχανής και βρίσκεται μεταξύ των πεδίων της τεχνητής νοημοσύνης, της πληροφορικής της γνωστικής επιστήμης και της ψυχολογίας. Προκειμένου να δημιουργηθεί μια αυτοματοποιημένη μηχανή η οποία θα δύναται να αναγνωρίσει ανθρώπινα συναισθηματικά επίπεδα, χρειάζεται η αρχιτεκτονική των υπολογιστικών εφαρμογών να συνυπολογίσει τη δυνατότητα παροχής εισόδου με πολλαπλή μορφή από τους ανθρώπους στους υπολογιστές, να ξεπεραστεί η πεπερασμένη διεπαφή του τύπου παράθυρο-ποντίκι-δείκτης και να χρησιμοποιηθούν μέσα πιο διαισθητικά και πιο κοντά στις συνήθειες των ανθρώπων(PicardR.W. 1997).

1.9.1 Τομείς της συναισθηματικής υπολογιστικής

Για να μπορεί μια μηχανή να έχει έξυπνες αντιδράσεις στα συναισθήματα του εκάστοτε χρήστη χρειάζεται να έχει τη δυνατότητα παρατήρησής του, κατόπιν να προχωρά σε ερμηνεία των δεδομένων της παρατήρησης, στη συγκέντρωσή τους, την εξαγωγή συμπερασμάτων και εν τέλει να μπορεί να έχει την κατάλληλη αντίδραση για κάθε περίπτωση (Moridis, C.N. & Economides, A.A., 2008)

Έτσι, είναι εφικτό η αλληλεπίδραση μηχανής-ανθρώπου να κρατά σε υψηλά επίπεδα το ενδιαφέρον του χρήστη και να δημιουργεί μια αξιοσημείωτη εμπειρία που πλησιάζει πιο πολύ την ανθρώπινη λογική, είναι λεκτικά αποτελεσματικότερη και περισσότερο ικανοποιητική. Στη συνέχεια παρατίθενται οι κύριες περιοχές στις οποίες η συναισθηματική υπολογιστική βρίσκει εφαρμογή:

- Εντοπισμός συναισθηματικών πληροφοριών. Η ανίχνευση πληροφοριών που φέρουν συναισθήματα είναι δυνατόν να εκτελείται με τη χρήση αισθητήρων οι οποίοι κάνουν καταγραφή των δεδομένων που αφορούν τη συναισθηματική κατάσταση του χρήστη και παρακολουθώντας τη συμπεριφορά του, δίχως να επικεντρώνεται στην ερμηνεία των εν λόγω δεδομένων. Η ανίχνευση βασίζεται σε ανάλογα δεδομένα με εκείνα που εξετάζουν και οι άνθρωποι για να διαπιστώσουν τη συναισθηματική κατάσταση στην οποία βρίσκεται ο απέναντί τους, όπως π.χ. μια κάμερα καταγράφει τις εκφράσεις του προσώπου, τις χειρονομίες και τη στάση του σώματος, κάποιοι άλλοι αισθητήρες καταγράφουν δεδομένα που αφορούν την ανθρώπινη φυσιολογία, σαν τους καρδιακούς παλμούς, τη θερμοκρασία ή την αγωγιμότητα του δέρματος
- Αναγνώριση συναισθημάτων. Η αναγνώριση των συναισθημάτων είναι εφικτή με την ανάλυση ποικίλων δεδομένων σχετικά με τον χρήστη. Με την ανάλυση των χαρακτηριστικών της ομιλίας, π.χ. την προσωδία και τον τόνο, δίνεται η πληροφορία για τη διάθεση και το ύφος του ατόμου. Για να γίνει η ερμηνεία χρησιμοποιούνται προχωρημένοι αλγόριθμοι μάθησης και ταξινόμησης, οι οποίοι σε πρώτη φάση εκπαιδεύονται σε ορισμένα δεδομένα αναφοράς, αποτελέσματα ως επί το πλείστον της αναγνώρισης και ταξινόμησης των αντίστοιχων συναισθηματικών καταστάσεων από τους ανθρώπους.

- Συναισθηματικά μηχανικά συστήματα. Το σημαντικότερο ίσως κομμάτι της συναισθηματικής υπολογιστικής αποτελεί ο σχεδιασμός υπολογιστικών συστημάτων, τα οποία να μπορούν ή να επιδείξουν έμφυτες συναισθηματικές ικανότητες, ή να πείσουν ότι έχουν συναισθήματα. Τα συγκεκριμένα συστήματα περιέχουν μια διεπαφή (interface) αλληλεπίδρασης με τον χρήστη η οποία μπορεί να είναι φωνητική ή γραπτής μορφής ή με την μορφή εικονικού πράκτορα (virtual agent). Οι εν λόγω πράκτορες είναι λογισμικές οντότητες που διαθέτουν δυνατότητες προσομοίωσης συναισθημάτων και εμφανίζονται με μια ομιλούσα κεφαλή (talking head) ή με ολόκληρη γραφική σωματική υπόσταση. Αυτή η προσέγγιση αποσκοπεί στον εμπλουτισμό και τη διευκόλυνση της αλληλεπίδρασης μηχανής-ανθρώπου μέσω ενός ανθρωπόμορφου γραφικού περιβάλλοντος. Στον εικονικό πράκτορα ενσωματώνεται η εκφραστική ανάλυση, που αποτελεί τη δυνατότητα του αντίληψης και ερμηνείας της συναισθηματικής κατάστασης του χρήστη ή τουλάχιστον ορισμένων ενδείξεών της. Η ικανότητα των αληθοφανών εικονικών πρακτόρων να δίνουν εκφραστική ανατροφοδότηση στο χρήστη είναι μια κρίσιμη πτυχή προκειμένου να υποστηρίξουν τη φυσικότητα της αλληλεπίδρασής τους.
- Κατανόηση και προσομοίωση συναισθημάτων. Η κατανόηση συναισθημάτων (emotion understanding) αφορά στην ικανότητα μιας συσκευής να ανιχνεύσει τη συναισθηματική πληροφορία, να την αποθηκεύσει, να την επεξεργαστεί και να δημιουργήσει ένα πρότυπο συναισθηματικής συμπεριφοράς του προσώπου ενδιαφέροντος. Απώτερος σκοπός είναι η μηχανή, αναλόγως την κατάσταση, το περιβάλλον, τις συνθήκες και τις ενδείξεις που συγκεντρώνει από το χρήστη, να έχει την αρμόζουσα αντίδραση σε κάθε περίπτωση (Venkatesh, V., Morris, M. G., Davis, G. B., Davis, F. D. (2003).

1.9.2 Μοντέλα Συναισθηματικής Υπολογιστικής (Affective Learning Models)

Τα αποτελέσματα της εφαρμογής συστημάτων συναισθηματικής υπολογιστικής ερμηνεύονται με βάση διάφορα μοντέλα, εκ των οποίων τα επικρατέστερα είναι του Russell και του Kort.

Russell's Circumplex Model

Βάσει του μοντέλου του Russell, τα συναισθήματα κατανέμονται σε ένα κύκλο στο κέντρο του οποίου βρίσκεται ένα σύστημα συντεταμένων. Στον Υ-άξονα αναπαρίσταται

ο βαθμός της διέγερσης- εξασθένησης ενός συναισθήματος και ο Χ-άξονας αναπαριστά την κλιμάκωση του συναισθήματος αυτού από αρνητικό συναίσθημα σε θετικό. Αναλόγως την ένταση και την επίδραση του κάθε συναισθήματος (θετική ή αρνητική), αυτό τοποθετείται μέσα στον ερμηνευτικό κύκλο του Russell καταλήγοντας σε μια γωνία. Για τη γωνία αυτή δίνεται ένας χαρακτηρισμός στο συναίσθημά, όπως ευχαριστημένος, χαρούμενος, ενοχλημένος, εκνευρισμένος, κ.τ.λ. Το συγκεκριμένο μοντέλο εστιάζει στην υποκειμενική εμπειρία, συνεπώς τα συναισθήματα εντός του συστήματος αυτού μπορεί να μην τοποθετούνται με τον ίδιο τρόπο ακριβώς για όλα τα άτομα.(Russell, J. A. (1980)

Kort's Learning Spiral Model

Ο Kort παρουσίασε ένα σπειροειδές μοντέλο μάθησης που αποτελείται από τέσσερα τεταρτημόρια, στο οποίο η συναισθητική κατάσταση ενός προσώπου προσδοκάται να επηρεάζει συστηματικά το πώς αυτό επεξεργάζεται νέες πληροφορίες. Στο πρώτο τεταρτημόριο ο μαθητής έχει θετικά συναισθήματα καθώς κατορθώνεται εύκολα η απαιτούμενη γνώση. Όταν αρχίζουν να εμφανίζονται διαφορές ανάμεσα στις πληροφορίες και στη δομή της γνώσης του μαθητή, ο εκπαιδευόμενος μεταβαίνει στο δεύτερο τεταρτημόριο, στο οποίο αν και επιτυγχάνεται η νέα γνώση, εμφανίζονται αρνητικές, συναισθηματικές επιδράσεις. Στην περίπτωση αυτή ο μαθητής εμφανίζει συναισθήματα, όπως σύγχυση και απογοήτευση. Καθώς ο μαθητής επιχειρεί να λύσει το πάζλ και αποτυγχάνει, μπορεί να κινηθεί στο τρίτο τεταρτημόριο, το τεταρτημόριο της μη γνώσης και της αρνητικής επίδρασης, περιπτώσεις δηλαδή όπου ο μαθητής βιώνει καταστάσεις όπως η απογοήτευση. Όταν ο μαθητής ξεπεράσει τις παρανοήσεις, κινείται στο τέταρτο τεταρτημόριο, το οποίο χαρακτηρίζεται από μη γνώση και θετική επίδραση. Εκεί ο μαθητής δεν είναι ακόμα σίγουρος για το πώς πρέπει να προχωρήσει, εντούτοις, αποκτά καινούργιες γνώσεις και ψάχνει καινούργιες ιδέες. Στη συνέχεια, περνάει στο πρώτο τεταρτημόριο. Έτσι ολοκληρώνεται ένας συναισθηματικός κύκλος γύρω από τη μάθηση (KortB, ReillyR, PicardR., (2001).

1.9.3 Τεχνολογικό υπόβαθρο της συναισθηματικής υπολογιστικής

Υπολογιστικά μοντέλα συναισθημάτων

Τα υπολογιστικά μοντέλα συναισθημάτων χωρίζονται στα μοντέλα που ο σκοπός τους είναι η αναγνώριση και κατανόηση της συναισθηματικής συμπεριφοράς του ανθρώπου και σε αυτά που ο στόχος τους είναι η προσομοίωση της συμπεριφοράς του ανθρώπου από εικονικούς/συνθετικούς χαρακτήρες. Για να μπορέσει να μοντελοποιηθεί η ανθρώπινη συμπεριφορά με στόχο την δημιουργία συνθετικών πρακτόρων που να ενεργούν έξυπνα, ανεξαρτήτως συνθηκών ή εννοιολογικού πλαισίου, κρίνεται απαραίτητο η ύπαρξη και κατανόηση της γνωσιακής αρχιτεκτονικής (cognitive architecture). Προσφάτως, πολλές αρχιτεκτονικές (όπως πάλι η SOAR και η ACT-R καθώς και η ICARUS και CLARION) έχουν επεκταθεί και μοντελοποιούν μηχανισμούς αντίληψης και δράσης καθώς και συναισθηματικές καταστάσεις διαθέσεις και κίνητρα (Picard, R., Papert, S., Bender, W., Blumberg, B., Breazeal, C., Cavallo, D., et al. (2004)

Ευφυείς πράκτορες

Ένα από τα πιο βασικά μοντέλα συναισθημάτων που βρίσκει εφαρμογή η συναισθηματική υπολογιστική είναι οι πράκτορες. Πράκτορας (agent) καλείται μια οντότητα η οποία αντιλαμβάνεται το περιβάλλον μέσα στο οποίο βρίσκεται μέσω αισθητήρων (sensors), και δρα εντός αυτού με τη βοήθεια μηχανισμών δράσης (effectors), για την επίτευξη στόχων. Για να καταστεί σαφές ποιοι πράκτορες υπάρχουν στον τομέα της συναισθηματικής υπολογιστικής, όπως και με ποιο τρόπο συνδέονται με υπολογιστικά μοντέλα συναισθήματος δίνεται μια σύντομη κατηγοριοποίηση των τύπων των πρακτόρων. Οι βιολογικοί πράκτορες χρησιμοποιούν τις αισθήσεις τους προκειμένου να αντιληφθούν τον κόσμο γύρω τους, τις γνώσεις για να εξάγουν συμπεράσματα για αυτόν, και τα διάφορα μέρη του σώματος τους για να εφαρμόσουν τις ενέργειες που προκύπτουν από τη συλλογιστική τους. Πέρα από τους βιολογικούς πράκτορες υπάρχουν και οι τεχνητοί, ρομποτικοί, και υπολογιστικοί πράκτορες. Οι ρομποτικοί πράκτορες έχουν σαν αισθητήρες και μηχανισμούς δράσης μηχανικά ή ηλεκτρονικά μέρη και δρουν στον πραγματικό κόσμο. Οι υπολογιστικοί πράκτορες διαιρούνται σε ευφυείς λογισμικούς πράκτορες (intelligent software agents ή softbots) που είναι προγράμματα και δρουν σε ένα υπολογιστικό σύστημα και σε πράκτορες που επιδεικνύουν δραματουργικές συμπεριφορές τεχνητής ζωής (artificial life agents). Οι πράκτορες

ακολουθούν δύο κύριες θεωρήσεις όσον αφορά τον τρόπο λειτουργίας τους: τη χαλαρή θεώρηση (weaknotion of agency) κατά την οποία είναι αναπροσδιοριζόμενοι (self-contained) και την ισχυρή θεώρηση (strongnotionofagency) όπου πρέπει να έχουν γνώση (knowledge), πεποιθήσεις (beliefs) επιθυμίες (desires), προθέσεις (intetions) και υποχρεώσεις (obligations). Στο κομμάτι της επικοινωνίας ανθρώπου-μηχανής και ειδικότερα της συναισθηματικής υπολογιστικής, μεγαλύτερη έμφαση δίνεται στα μοντέλα πρακτόρων που υιοθετούν την ισχυρή θεώρηση. Μερικά από τα βασικά χαρακτηριστικά που κάνουν τους πράκτορες εξελιγμένα υπολογιστικά μοντέλα είναι τα ακόλουθα:

- αυτονομία (autonomy)
- Αντιδραστικότητα (re-activeness)
- κοινωνικότητα (social ability) και
- προνοητικότητα (pro-activeness).

Μια ειδική κατηγορία ευφυών πρακτόρων αποτελεί ο πράκτορας BDI (Belief-Desire-Intention), ένα υπολογιστικό μοντέλο το οποίο αναπτύχθηκε για τον προγραμματισμό λογισμικών πρακτόρων. Το συγκεκριμένο μοντέλο έχει στενή σύνδεση με τους ευφυείς εικονικούς χαρακτήρες (intelligent virtual agents), όμως δεν αρκεί από μόνο του να εξασφαλίσει όλα τα χαρακτηριστικά που έχουν σχέση με μια τέτοια οντότητα. Εντούτοις, υπάρχουν υβριδικά μοντέλα τα οποία υιοθετούν την προσέγγιση BDI μαζί με συστήματα τα οποία επιτρέπουν τη διαχείριση διαλόγου (dialogue management systems) και παρέχουν τη δυνατότητα αλληλεπίδρασης με ανθρώπινους χρήστες, μέσω διάφορων μέσων, όπως η ομιλία και οι μη-λεκτικές εκφράσεις.(Breazeal, C. And Aryananda, L. (2002).

1.10 Αναγνώριση συναισθημάτων μέσω της ομιλίας

Ένα σύστημα αναγνώρισης συναισθημάτων μέσω της φωνής περιλαμβάνει τα εξής τρία κύρια μέρη,

- επεξεργασία του σήματος,
- υπολογισμό χαρακτηριστικών και
- ταξινόμηση.

Η επεξεργασία του σήματος εμπεριέχει την ψηφιοποίηση του καταγεγραμμένου σήματος, ενδεχομένως ακουστική προ-επεξεργασία όπως το φιλτράρισμα και τον κατακερματισμό του σήματος εισόδου σε σημαντικές μονάδες. Ο υπολογισμός των χαρακτηριστικών αποσκοπεί στο να βρεθούν οι ιδιότητες εκείνες των ψηφιοποιημένων και προ-επεξεργασμένων ηχητικών σημάτων που χαρακτηρίζονται από συναισθήματα και η αναπαράστασή τους σε n -διάστατο διάνυσμα χαρακτηριστικών γνωρισμάτων. Μέχρι σήμερα δεν υπάρχει κάποια κοινή παραδοχή για το πια χαρακτηριστικά είναι πιο βασικά. Συνεπώς, οι πιο πολλές προσεγγίσεις υπολογίζουν μια πληθώρα χαρακτηριστικών και κατόπιν εφαρμόζουν αλγόριθμους επιλογής χαρακτηριστικών για να ελαττωθούν οι διαστάσεις των δεδομένων εισόδου. Ύστερα από τον υπολογισμό των χαρακτηριστικών, κάθε συναισθηματική μονάδα αντιπροσωπεύεται από ένα ή περισσότερα διανύσματα χαρακτηριστικών γνωρισμάτων κάνοντας το πρόβλημα της αναγνώρισης συναισθημάτων ένα γενικό πρόβλημα ταξινόμησης. Η ταξινόμηση είναι είτε στατική (Lerner, J. & Keltner, D., 2000) ή δυναμική. Στη δυναμική ταξινόμηση συνεκτιμάται και η χρονική συμπεριφορά των διανυσμάτων, η οποία είναι πολύ βασική για την αναγνώριση των συναισθημάτων. Προκειμένου να γίνει αναγνώριση συναισθημάτων σε πραγματικό χρόνο, οι δυνατότητες απευθείας σύνδεσης των αλγορίθμων χρειάζεται να χρησιμοποιούνται κάθε στάδιο. Επιπρόσθετα, οι μέθοδοι είναι πιο εύρωστες στις αλλαγές του θορύβου και άλλων παρεμβολών.

1.10.1 Αναγνώριση συναισθημάτων μέσω ανάλυσης προσώπου

Οι εκφράσεις του προσώπου είναι τα βασικά μη-λεκτικά εργαλεία της ανθρώπινης επικοινωνίας παρέχοντας το ισχυρότερο και πιο ευέλικτο φυσικό μέσο επικοινωνίας αποτελώντας ένδειξη της συναισθηματικής φόρτισης του ατόμου εστιάζοντας στα βασικά χαρακτηριστικά, δηλαδή τα μάτια, τα φρύδια, το στόμα και τη μύτη.

Η ανάλυση των εκφράσεων του προσώπου ήταν η πρώτη μέθοδος αναγνώρισης συναισθημάτων και εφαρμόζεται έκτοτε ευρέως σε πολλές μελέτες αφού είναι η προτιμητέα μέθοδος για ενιαία συνδυαστικά συστήματα αναγνώρισης και έκφρασης συναισθημάτων (LeDoux, J., (1999).

Η κλασική τακτική αναγνώρισης των συναισθημάτων μέσω της έκφρασής του προσώπου είναι ίδια με εκείνη της αναγνώρισης τους δια της ομιλίας. Στην περίπτωση

της οπτικής πληροφορίας η απεικόνιση είναι απλώς μια φωτογραφία στην οποία το υποκείμενο απεικονίζεται στο υψηλότερο εκφραστικό επίπεδο. Οι σημερινές τεχνικές πάντως έχουν την τάση να εμπλέκουν πληροφορία δυναμικής υφής για την κατανόηση των συναισθημάτων μέσω των εκφράσεων του προσώπου. Η αναγνώριση συναισθημάτων από την οπτική πληροφορία έχει ένα πιο ισχυρό υπόβαθρο, που προέρχεται από μελέτες στο τομέα της νευροφυσιολογίας, σε σύγκριση με την αναγνώρισή τους από ηχητικά δεδομένα.

Τα χαρακτηριστικά του προσώπου χωρίζονται σε στατικά – π.χ. το χρώμα του δέρματος-, σε αργά μεταβαλλόμενα –π.χ. η υφή, η οποία μεταβάλλεται με την ανάπτυξη ρυτίδων- και σε κινούμενα –π.χ. βλεφαρίδες, φρύδια κτλ. Η ανίχνευση της θέσεως αυτών των χαρακτηριστικών από χρονικά σταθερές φωτογραφίες είναι ο στόχος των στατικών προσεγγίσεων της ανάλυσης εκφράσεων. Εντούτοις, υπάρχει σημαντική ένδειξη ότι η αναγνώριση εκφράσεων από τον άνθρωπο στηρίζεται πιο πολύ σε πληροφορία δυναμικής υφής απ ό τι σε στατικές απεικονίσεις. Η θεμελιώδης διαφορά μεταξύ ανθρώπου και υπολογιστή είναι η ακρίβεια εντοπισμού των προσώπων και των χαρακτηριστικών τους στο χώρο. Το οπτικό σύστημα του ανθρώπου είναι εξαιρετικά αποτελεσματικό σ αυτόν τον τομέα. Αντιθέτως στους υπολογιστές τα σφάλματα εντοπισμού του προσώπου και των χαρακτηριστικών του, λειτουργούν προσθετικά και αρκετές φορές καλύπτουν την ουσιαστική πληροφορία κίνησης που δίνεται από τις ακολουθίες. Συμπερασματικά, οι στατικές προσεγγίσεις οι οποίες είναι λιγότερο επιρρεπείς στον εντοπισμό των χαρακτηριστικών του προσώπου διατηρούν την αξία τους ως προς την αναγνώριση εκφράσεων. Επιπλέον τα στάδια προ επεξεργασίας που αφορούν στον εντοπισμό του προσώπου, των βασικών χαρακτηριστικών του και σημείων στη περιοχή αυτών των χαρακτηριστικών είναι μεγάλης σημασίας σε κάθε περίπτωση. Οι δυναμικές προσεγγίσεις επιτυγχάνουν επιπρόσθετα και την παρακολούθηση της κίνησης των χαρακτηριστικών με μεθόδους εκτίμησης κίνησης και μοντελοποίησης των μυών και των ενεργειών τους. (J. K. Aggarwal, Q. Cai, (1999)

ΚΕΦΑΛΑΙΟ 2

2 ΚΕΦΑΛΑΙΟ 2 ΒΙΟΑΝΑΔΡΑΣΗ ΒΙΟΣΗΜΑΤΑ ΦΥΣΙΟΛΟΓΙΑ
ΚΑΡΔΙΑΣ ΚΑΙ ΔΕΡΜΑΤΟΣ

2.1 Βιοανάδραση

Η πλειοψηφία των φυσιολογικών λειτουργιών του οργανισμού μας είναι αυτόνομες ή ασυνείδητες. Η βιοανάδραση αποτελεί μια διαδικασία που δίνει τη δυνατότητα στον άνθρωπο να γνωρίσει και να αλλάξει τις λειτουργίες αυτές για να ενισχύσει την υγεία και την απόδοση του. Χρειάζεται ακριβή όργανα μετρήσεων για την καταγραφή σωματικών λειτουργιών, τα οποία δίνουν πληροφορίες "ανατροφοδότησης" στο χρήστη. Οι πληροφορίες αυτές σε συνδυασμό με μεταβολές στον τρόπο σκέψης, τα συναισθήματα και τη συμπεριφορά μπορούν να επιφέρουν τις επιθυμητές αλλαγές και βελτιώσεις. Καθώς περνά ο χρόνος, οι αλλαγές αυτές μπορούν να συνεχιστούν δίχως να απαιτείται η συνεχόμενη χρήση του οργάνου μέτρησης. (Association for Applied Psychophysiology and Biofeedback (AAPB), Biofeedback Certification International Alliance (BCIA), International Society for Neuro feedback and Research (ISNR), 2008)

Σήμερα, γνωρίζουμε ότι η διέγερση του δέρματος και της καρδιάς (αύξηση παλμών) αποτελούν αυτόνομους μηχανισμούς του οργανισμού οι οποίοι ενεργοποιούνται όταν βιώνονται κάποια συναισθήματα. Η χαμηλή διέγερση του δέρματος συνδυαστικά με χαμηλούς καρδιακούς παλμούς υποδηλώνει χαλάρωση και ηρεμία. Η υψηλή διέγερση του δέρματος συνδυαστικά με υψηλούς καρδιακούς παλμούς υποδηλώνει έντονη συναισθηματική κατάσταση, εγρήγορση ενδεχομένως προκαλούμενη από φόβο, άγχος, στρες κ.τ.λ. Η υψηλή διέγερση του δέρματος ή η αύξηση των καρδιακών παλμών υποδηλώνουν ανησυχία, συναισθηματική διέγερση και αφύπνιση χωρίς κατ' ανάγκη να σημαίνει πως το άτομο βρίσκεται στο αποκορύφωμα της συναισθηματικής του κατάστασης. Αναμφίβολα, κάθε οργανισμός αντιδρά διαφορετικά, βιώνοντας διάφορα συναισθήματα. Παραδείγματος χάριν, ένα υποκείμενο όταν αισθάνεται λίγο αγχωμένο ίσως να έχει ταχυπαλμία δίχως αυτό να συνοδεύεται από αντιστοίχως μεγάλες τιμές

εφίδρωσης ή το αντίστροφο. Πάντως σε όποια περίπτωση, οι μηχανισμοί που χρησιμοποιεί ο κάθε οργανισμός σαν αντίδραση στις διάφορες συναισθηματικές καταστάσεις που βιώνει είναι ίδιοι (Zeidner, M., (2007).

2.2 Βιοσήματα ή Βιολογικά σήματα

Σήματα ονομάζονται τα αποτελέσματα μέτρησης ενός φυσικού μεγέθους, το οποίο περιέχει και μεταφέρει κάποια πληροφορία. Εάν η πληροφορία είναι ιατρικής φύσεως, τότε το εν λόγω σήμα είναι βιοσήμα ή βιολογικό σήμα. Πρακτικά, τα Βιοσήματα είναι έξοδοι βιολογικών διεργασιών ενός ζώντος οργανισμού. Τύποι βιοσημάτων:

- ηλεκτρικά
- μηχανικά
- χημικά

Ως προς τη διαστατικότητα τους χαρακτηρίζονται ως

- Μονοδιάστατα
- Πολυκαναλικά ή μη
- Δισδιάστατα, Τριδιάστατα, Τετραδιάστατα κλπ που αφορούν κυρίως εικόνες

Καθώς τα βιοσήματα προέρχονται από δυναμικές και πολύπλοκες βιολογικές λειτουργίες, η λήψη ενός βιοσήματος δεν είναι από μόνη της επαρκής. Χρειάζεται επεξεργασία του σήματος για να εξαχθεί η πληροφορία που περιέχει. Τα βιοσήματα συνήθως είναι κρυμμένα μαζί με άλλα σήματα και θόρυβο. Αρχικά γίνεται η μετατροπή του από αναλογικό σε ψηφιακό, το οποίο λαμβάνει χώρα συνήθως στον αισθητήρα, ή στο κύκλωμα συλλογής του σήματος και περιλαμβάνει τη χρονική δειγματοληψία και την κβάτωση των τιμών του. Κάποιες φορές είναι επιθυμητή η προεπεξεργασία του σήματος σε αναλογική μορφή. Η περαιτέρω επεξεργασία του (biosignal processing) αποσκοπεί στο φιλτράρισμα του σήματος από τον υφιστάμενο θόρυβο, στην ενίσχυση συγκεκριμένων χαρακτηριστικών του και τέλος στην εξαγωγή και επιλογή λίγων χαρακτηριστικών που θα χρησιμοποιηθούν στην περαιτέρω ανάλυση του.

2.3 Κατηγορίες Βιοανάδρασης

Υπάρχουν πολλές εφαρμογές βιοανάδρασης. Ο ευκολότερος τρόπος κατηγοριοποίησής τους είναι με βάση το βιοσήμα ή/και τον αισθητήρα που χρησιμοποιείται για τη συγκεκριμένη εφαρμογή, δίνοντας έτσι τις ακόλουθες κατηγορίες.

- Ηλεκτρομυογραφικό: Ηλεκτρομυογράφημα είναι η γραφική αναπαράσταση της ηλεκτρικής δραστηριότητας των μυών του σκελετού και των νεύρων είτε σε κατάσταση ηρεμίας ή σε κατάσταση διέγερσης. Τοποθετείται βελόνα μιας χρήσης εντός του μυ και καταγράφεται η ηλεκτρική του δραστηριότητα σε ηρεμία και προσπάθεια.
- Ηλεκτροδερμικό: Η τεχνική αυτή ανιχνεύει την αλλαγή στην αγωγιμότητα του δέρματος κατά την διέλευση ηλεκτρικού ρεύματος σταθερής και πολύ χαμηλής τάσης μέσω αυτού. Αποσκοπεί στον εντοπισμό άγχους, καθώς όταν αγχωνόμαστε ιδρώνουμε πιο πολύ.
- Θερμογραφικό: Στη θερμογραφική βιοανάδραση χρησιμοποιείται ένα δερματικό θερμόμετρο διαρκούς καταγραφής που συνδέεται με μία ειδική συσκευή η οποία καταγράφει την θερμοκρασία του δέρματος με ακρίβεια εκατοστού του βαθμού Κελσίου.
- Ηλεκτροεγκεφαλογραφικό: Ηλεκτροεγκεφαλογράφημα είναι η καταγραφή της ηλεκτρικής δραστηριότητας του εγκεφάλου. Η καταγραφή των ηλεκτρικών σημάτων του εγκεφάλου πραγματοποιείται με ηλεκτρόδια που τοποθετούνται στην επιφάνεια του κρανίου. Συνεπώς, αυτό που καταγράφεται είναι ηλεκτρικά σήματα από τον φλοιό του εγκεφάλου.
- Πληθυσμογραφικό: Η φώτο-πληθυσμογραφική μέθοδος είναι μια απλή και φθηνή οπτική τεχνική που μπορεί να χρησιμοποιηθεί για να ανιχνεύσει τις μεταβολές της πυκνότητας του αίματος στο στρώμα του μικροαγγειακού και στην ιατρική βρίσκει ευρεία εφαρμογή στη κατασκευή παλμικών οξύμετρων. Η συσκευή του παλμικού οξύμετρου ρίχνει υπέρυθρο φως στο δέρμα και μετρά την απορρόφηση του φωτός από αυτό. Εντούτοις, η συγκεκριμένη μέθοδος μπορεί να χρησιμοποιηθεί και σε ένα ευρύ φάσμα προϊόντων, όπως συσκευές μέτρησης κορεσμού του οξυγόνου στο αίμα, συσκευές μέτρησης της αρτηριακής πίεσης κ.ά (JohnAllen (2007))

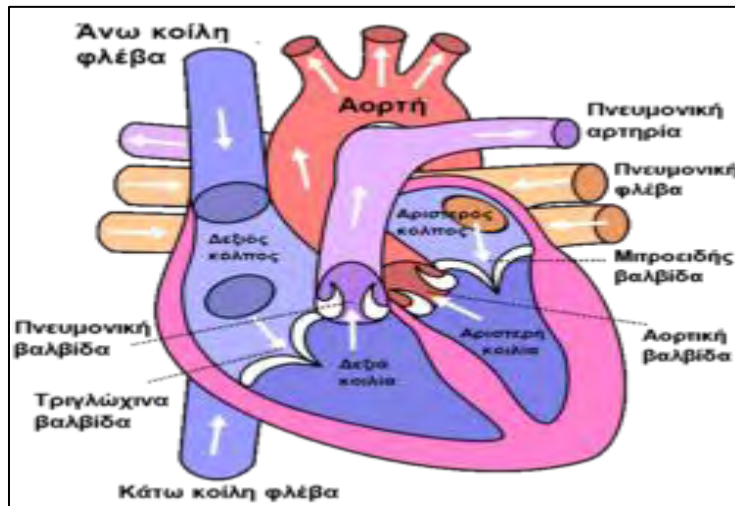
- Ηλεκτροκαρδιογραφικό: Το ηλεκτροκαρδιογράφημα κάνει καταγραφή της ηλεκτρικής δραστηριότητας της καρδιάς και χρησιμοποιείται για να μετρηθεί ο ρυθμός και η κανονικότητα των καρδιακών παλμών προκειμένου να εντοπιστεί ενδεχόμενη βλάβη στην λειτουργία της καρδιάς. Πραγματοποιείται με ειδικό μηχανήμα, τον ηλεκτροκαρδιογράφο, ο οποίος είναι ένα ευαίσθητο βολτόμετρο που καταγράφει μέσω ηλεκτροδίων διαφορές ηλεκτρικού δυναστικού στην επιφάνεια του σώματος οι οποίες προκύπτουν κατά την λειτουργία της καρδιάς.
- Αναπνευστικό: Η αναπνοή είναι μία από τις βασικότερες λειτουργίες του ανθρώπινου οργανισμού και συνδέεται άμεσα με τη διαχείριση του στρες και άλλες φυσιολογικές λειτουργίες. Ένας αισθητήρας βιοανατροφοδότησης της αναπνοής υπολογίζει την έκταση του διαφράγματος κατά την εισπνοή και την εκπνοή. Ο σκοπός είναι το άτομο να εκπαιδευτεί προοδευτικά στη σωστή λειτουργία του αναπνευστικού του συστήματος και στη χαλάρωση (NitzanMandTaitelbaumH, 2008)

2.4 Αξιοποίηση δεδομένων βιοανάδρασης και εξαγωγή συμπερασμάτων

Με τη χρήση αισθητήρων βιοανάδρασης λαμβάνουμε χρήσιμες πληροφορίες για τη συναισθηματική κατάσταση ενός εκπαιδευομένου. Επομένως, πληροφορίες όπως ο βαθμός εφίδρωσης, το πλήθος των καρδιακών του παλμών, η θερμοκρασία του σώματος κ.τ.λ. μπορούν να αποτελέσουν σημαντικό υλικό για επεξεργασία, του οποίου τα αποτελέσματα θα μπορούσαν να αξιοποιηθούν τόσο σε ατομικό επίπεδο από τον εκπαιδευόμενο όσο και σε συλλογικό επίπεδο από τους άλλους συμμαθητές του και από τον ίδιο τον εκπαιδευτή. Η συνείδηση και γνώση της συναισθηματικής μας κατάστασης καθώς και αυτής των γύρω μας κατά τη διάρκεια μιας εκπαιδευτικής διαδικασίας χρησιμεύει στον αυτοέλεγχο και στην αυτορρύθμιση παρέχοντας περισσότερες ευκαιρίες για βελτίωση της διαδικασίας και πραγματοποίηση του στόχου, δηλαδή της μάθησης. (Heckhausen, J. & Dweck, C.S. (1998)

2.5 Η φυσιολογία της καρδιάς

Η καρδιά είναι ένα μυώδες όργανο που συστέλλεται ρυθμικά λειτουργώντας σαν αντλία αίματος με τη βοήθεια του κυκλοφορικού συστήματος. Ο καρδιακός κύκλος αποτελείται από 4 αντλίες: δύο προαντλίες, τους κόλπους, και δύο προωθητικές αντλίες, τις κοιλίες. Η χρονική περίοδος από το τέλος μιας συστολής της καρδιάς μέχρι το τέλος της επόμενης συστολής, ονομάζεται καρδιακός παλμός ή κύκλος. Ο καρδιακός κύκλος, ή παλμός, περιλαμβάνει μία περίοδο συστολής, κατά την οποία το αίμα προωθείται από την καρδιά στις μεγάλες αρτηρίες, και μία περίοδο χάλασης, τη διαστολή, κατά την οποία ο καρδιακός μυς χαλαρώνει και το αίμα επιστρέφει στην καρδιά γεμίζοντας τις καρδιακές κοιλότητες. Όλος ο καρδιακός κύκλος διαρκεί περίπου 0,8 sec όταν η συχνότητα λειτουργίας της καρδιάς είναι 75 παλμοί το λεπτό 0,3 sec περίπου διαρκεί η συστολή και 0,5 sec η διαστολή. (<https://eclass.hua.gr>, 2018)



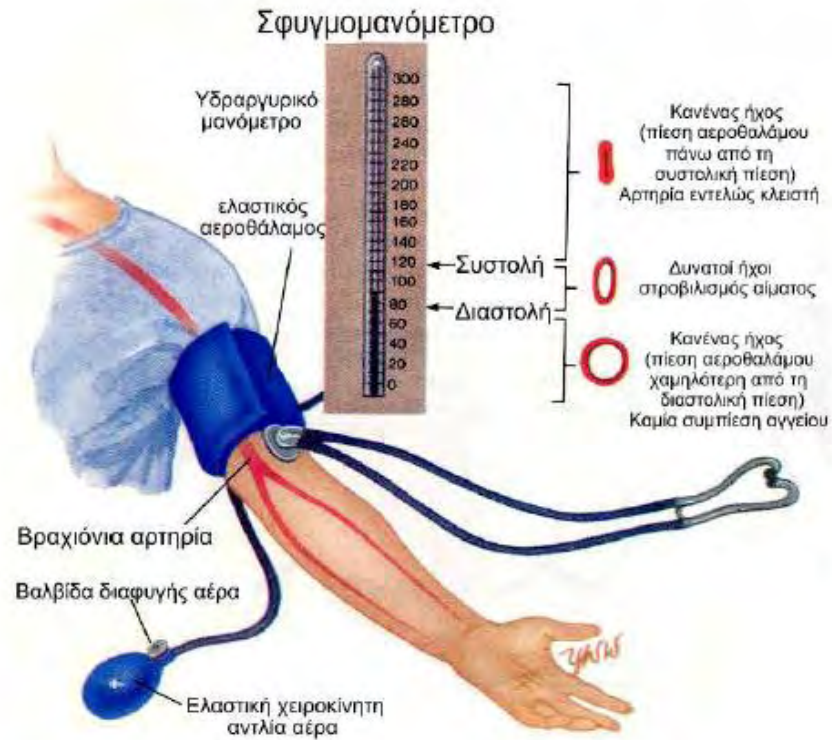
Εικόνα 1Γα κύρια μέρη της καρδιάς

Ο κάθε καρδιακός κύκλος αποτελείται από τέσσερις κύριες φάσεις. Οι δύο πρώτες ανήκουν στη συστολή της καρδιάς, κατά τη διάρκεια της οποίας η καρδιά εξωθεί το αίμα προς το υπόλοιπο σώμα, ενώ οι δύο τελευταίες ανήκουν στη διαστολή της καρδιάς, κατά τη διάρκεια της οποίας ωθείται αίμα προς την καρδιά. (<http://www.biomed.ntua.gr>, 2018)

Για το σκοπό αυτό, η καρδιά είναι εφοδιασμένη με ένα ειδικό σύστημα παραγωγής ρυθμικών ώσεων που άγονται ταχύτατα καθ' όλη την έκταση του μυοκαρδίου με τέτοιο τρόπο, ώστε πρώτα συσπώνται οι κόλποι και ακολούθως οι κοιλίες. Πολλές από τις παθήσεις της καρδιάς, και ειδικά αυτές που χαρακτηρίζονται από καρδιακές αρρυθμίες, οφείλονται σε διαταραχές αυτού του ειδικού συστήματος παραγωγής και αγωγής του ερεθίσματος. Το πρωτογενές ερέθισμα που διεγείρει τη συστολή των κόλπων ξεκινάει από μια ειδική περιοχή του δεξιού κόλπου, κοντά στην εκβολή της άνω κοίλης φλέβας, που ονομάζεται φλεβόκομβος.

2.5.1 Καρδιακός ρυθμός

Ορίζεται ως το πλήθος των κτύπων της καρδιάς ανά μονάδα του χρόνου (BPM) BeatPerMinutes . Ο καρδιακός ρυθμός είναι μεταβλητό μέγεθος. Η μεταβολή του οφείλεται στις διαφορετικές ανάγκες και λειτουργίες του σώματος όπως ύπνος – άσκηση - stress – και εξωτερικοί παράγοντες όπως καιρικές συνθήκες – φάρμακα κ.λπ. Βασικά συστατικά που εμπλέκονται στην αλλαγή του καρδιακού ρυθμού αποτελούν οι ενδορφίνες και οι ορμόνες που απελευθερώνονται στον εγκέφαλο. Ο καρδιακός ρυθμός ενός υποκειμένου σε κατάσταση ηρεμίας μπορεί να κυμαίνεται από 60 έως και 100 παλμούς , ενώ εκατέρωθεν αυτών υπάρχουν καταστάσεις βραδυκαρδίας (λιγότεροι από 60 παλμούς) και ταχυκαρδίας (περισσότεροι από 100 παλμούς εδώ να σημειώσουμε ότι στα παιδιά ο μέσος όρος παλμών είναι διαφορετικός. Σε συνθήκες άσκησης οι παλμοί ανεβαίνουν και το μέγιστο όριο παλμών κατά την Αμερικάνικη Ένωση για τον καρδιακό ρυθμό ορίζεται ως « $220 - \text{ηλικία ατόμου (έτη)}$ » Για παράδειγμα το μέγιστο όριο παλμών ενός ανθρώπου 50 ετών δεν θα πρέπει να ξεπερνά τους 170. « $220 - 50 = 170$ ». Ενα τυπικό καρδιογράφημα από μία εκ των 6 θωρακικών απαγωγών, με τα επάρματα P, QRS και T, φαίνεται στην Εικόνα 4, όπως πάρθηκε από (<http://www.easynotecards.com>) . Η μεγάλη κορυφή (QRS) αντιστοιχεί στη συστολή των κοιλιακών μυών (κύμαP: εκπόλωση κόλπων, σύμπλεγμα QRS: εκπόλωση κοιλίων, κύμα T: ανάνηψη κοιλίων).

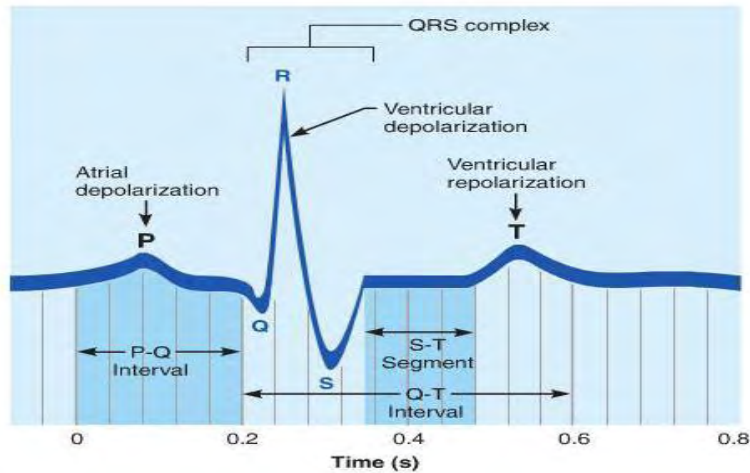


Εικόνα 2 Μέτρηση με σφυγμομανόμετρο



Εικόνα 3 Πρόσθετοι τρόποι μέτρησης του καρδιακού ρυθμού

Το καρδιακό ρυθμό μπορούμε να το μετρήσουμε σε αρκετά σημεία πάνω στο σώμα μας στο σώμα μας όπως στο χέρι βραχίονα ή καρπό , στη καρωτίδα του λαιμού , στο σημείο της καρδιάς στη κοιλιακή αρτηρία ή και στις άκρες των δακτύλων. Οι μετρήσεις μπορούν να γίνουν με την αφή με στηθοσκόπιο, ηλεκτροκαρδιογράφημα , φωτοπλεθυσμογραφία κ.α

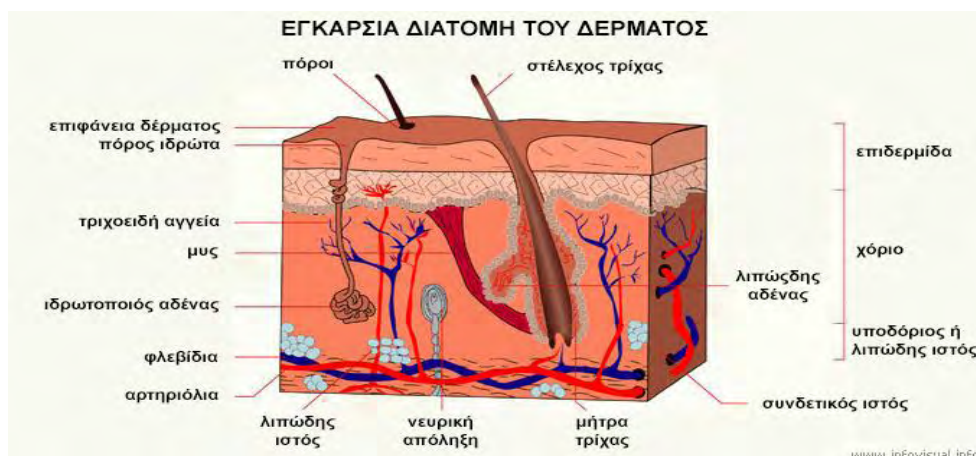


Εικόνα 4 [καρδιακός κύκλος](http://www.easynotecards.com) <http://www.easynotecards.com>

2.6 Ανατομία του Δέρματος

Η γνώση του μηχανισμού της αγωγιμότητας του δέρματος προϋποθέτει κάποια κατανόηση σχετικά με τη δομή των ιστών τόσο στην επιφάνεια του δέρματος, όσο και κάτω απ' αυτή. Το δέρμα αποτελείται από τρία διαφορετικά τμήματα από το επιφανειακό έως το πιο εσωτερικό

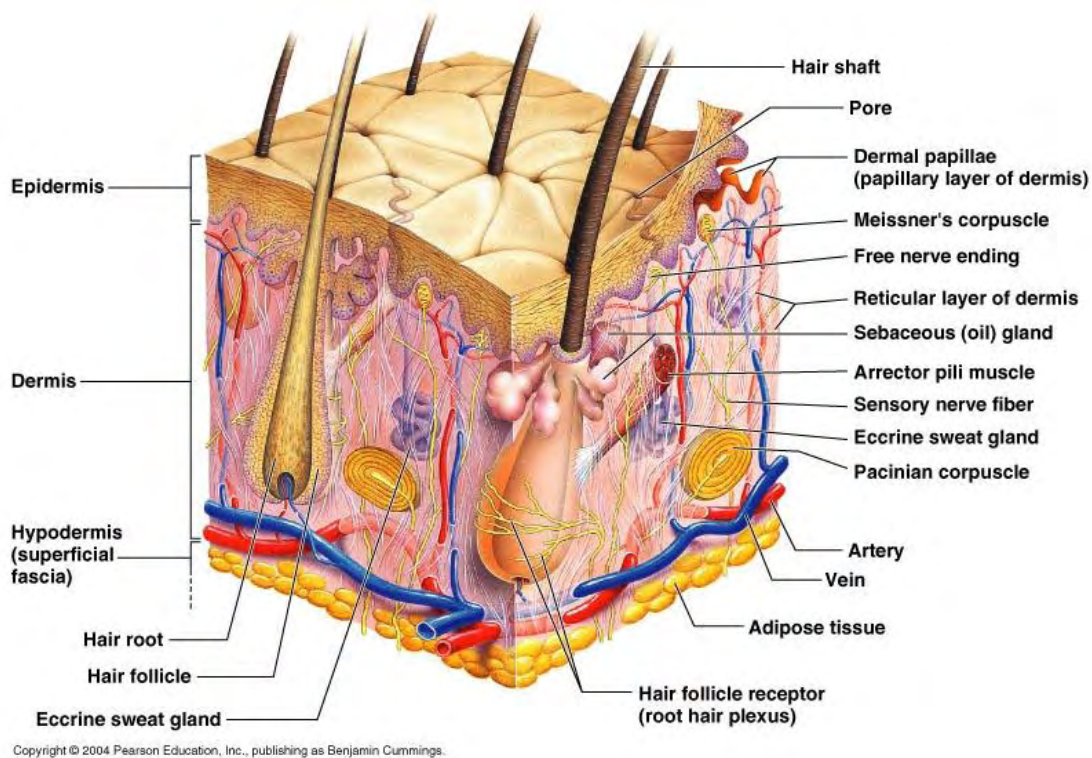
- επιδερμίδα
- χόριο ή κυρίως δέρμα
- υπόδερμα ή υποδόριο λίπος



Εικόνα 5 Εγκάρσια διατομή του δέρματος

2.6.1 Επιδερμίδα

Η επιδερμίδα βρίσκεται στην επιφάνεια του δέρματος, έχει πάχος 4mm και αποτελείται κυρίως από κερατινοκύτταρα, αλλά περιλαμβάνει και μελανοκύτταρα, κύτταρα Langerhans και κύτταρα Merkel. Τα στοιχεία του δέρματος που διαπερνούν την επιδερμίδα είναι οι τρίχες και οι ιδρωτοποιόι αδένες οι οποίοι παίζουν σημαντικό ρόλο στην γαλβανική απόκριση δέρματος



Εικόνα 6 Τμήμα επιδερμίδας [http://www.apsubiology.org/anatomy/2010/2010_Exam_Reviews/Exam_2_Review/05-01_Skin.JPG]

2.6.2 Κυρίως δέρμα

Το κυρίως δέρμα ή χόριο (corium) βρίσκεται ανάμεσα από την επιδερμίδα και το υπόδερμα και αποτελεί το στήριγμα της επιδερμίδας. Είναι πολύ παχύτερη από την επιδερμίδα, 1mm-4mm, και αποτελείται κυρίως από ίνες κολλαγόνου και ελαστίνης. Διαιρείται σε δύο υποστιβάδες, τη θηλώδη, η οποία περιέχει το αγγειακό δίκτυο και τη

δικτυωτή, στην οποία βρίσκονται τριχοφουείς αδένες, θύλακες των τριχών και εκκριτικοί αδένες.

Θηλώδης υποστιβάδα (Papillary layer):

Είναι η ανώτερη στιβάδα του χόριου. Συγκροτείται από χαλαρό συνδετικό ιστό και περιέχει μεγάλο αριθμό νευρικών ινών, τριχοειδή αγγεία, νερό και κύτταρα. Εδώ οι ίνες κολλαγόνου σχηματίζουν ένα πιο αραιό δίκτυο σε σχέση με την δικτυωτή υποστιβάδα.

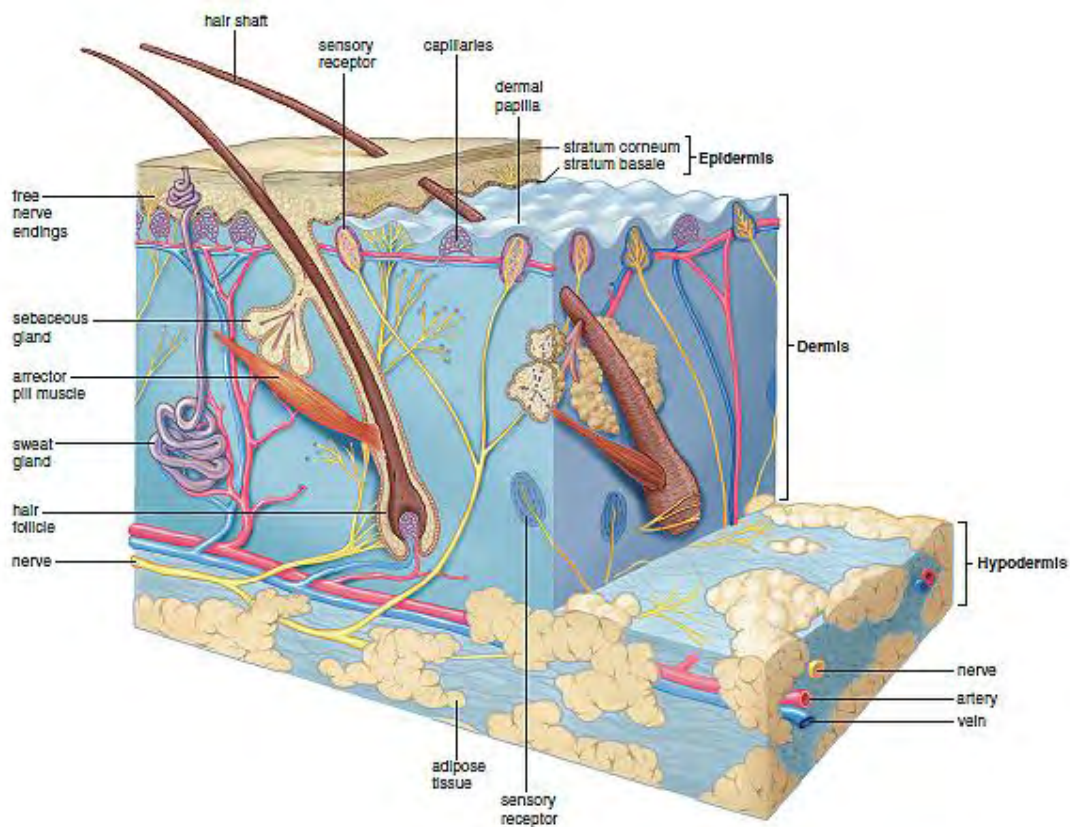
Δικτυωτή υποστιβάδα (Reticular layer):

Το βαθύτερο τμήμα του χόριου παρουσιάζει συνέχεια στη δομή με τον υποδόριο ιστό, σε αντίθεση με την επιδερμίδα που δεν έχει ομοιότητες με το χόριο. Συγκροτείται από ένα πυκνότερο και λεπτότερο δίκτυο με λιγότερες νευρικές ίνες και λιγότερα τριχοειδή αγγεία από τη θηλώδη υποστιβάδα. Σε αυτό το επίπεδο οι ίνες κολλαγόνου συγκροτούν παχιά δεμάτια, που διατάσσονται παράλληλα προς την επιφάνεια του δέρματος(<https://bioximikos.gr/topics/physiology-anatomy/100-anatomia-dermatos>)

2.6.3 Υπόδερμα ή Υποδόριο Λίπος

Ο λιπώδης ή υποδόριος ιστός είναι η τρίτη κύρια στιβάδα η οποία όμως συχνά δεν κατηγοριοποιείται στον δερματικό ιστό. Εκεί βρίσκονται τα λιπώδη κύτταρα που αποθηκεύουν το σωματικό λίπος.

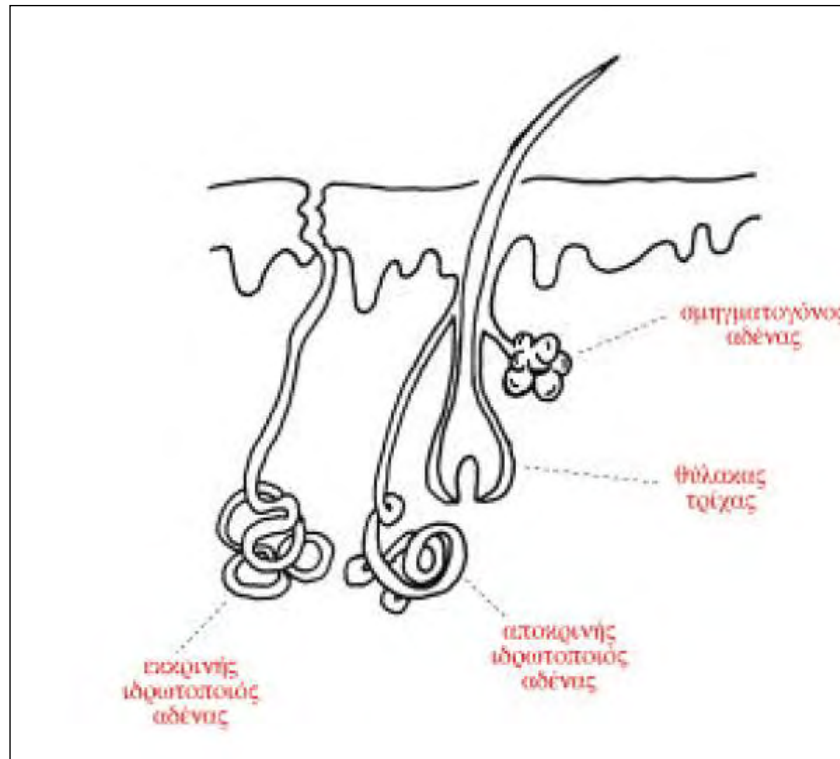
Είναι μια ελαστική στιβάδα με μεγάλη ποσότητα λιπωδών κυττάρων που απορροφούν τους κραδασμούς, προστατεύοντας τα αγγεία του αίματος και τις νευρικές απολήξεις. Έχει πάχος 4mm - 9mm κατά μέσο όρο, το οποίο διαφέρει από άνθρωπο σε άνθρωπο και εξαρτάται από την κατανομή λίπους στο σώμα. (<https://bioximikos.gr/topics/physiology-anatomy/100-anatomia-dermatos>)



Εικόνα 7 Υπόδερμα[<http://www.medicinehack.com/2011/12/skin-structure.html>]

2.6.4 Ιδρωτοποιό Αδένες

Οι ιδρωτοποιό αδένες υπάρχουν σε όλη την επιφάνεια του δέρματος, εκτός από το δέρμα των φρυδιών, το ερυθρό των χειλιών, την έσω επιφάνεια του περυγίου του αυτιού, τα κοίτη των ονύχων, το έσω πέταλο της ακροποσθίας, τη βάλανο και τα μικρά χείλη του αιδίου. Ο αριθμός τους κυμαίνεται περίπου στα 3 εκατομμύρια. Οι περισσότεροι εντοπίζονται στα πέλματα, τις παλάμες, το πρόσωπο και τη μασχάλη. Οι αδένες του Μολις παρυφές των βλεφαρίδων και οι κυψελιδοποιό αδένες του έξω ωτός είναι τροποποιημένο ιδρωτοποιό αδένες. Οι ιδρωτοποιό είναι σωληνοειδέες αδένες και διακρίνονται στους εκκριτικούς και τους απεκκριτικούς



Εικόνα 8 **Ιδρωτοποιί Αδένες** (http://edume.myds.me/00_0070_e_library/10060/001/01.pdf)

2.7 Γενική Περιγραφή της Γαλβανικής Απόκρισης Δέρματος

Η Γαλβανική Απόκριση Δέρματος (GSR) ορίζεται σαν μια αλλαγή στις ηλεκτρικές ιδιότητες του δέρματος. Η αντίσταση του δέρματος είναι συνήθως μεγάλη, από 1ΜΩ - 10ΜΩ, όμως γνωρίζουμε ότι στιγμιαίες αλλαγές της δραστηριότητας του ιδρωτοποιού αδένα προκαλούν αλλαγές στην αντίστασή του περίπου έως και 950K.

Η GSR αντικατοπτρίζει τη δραστηριότητα του ιδρωτοποιού αδένα και τις αλλαγές στο νευρικό σύστημα και τις μεταβλητές μέτρησης. Η διέγερση του Νευρικού Συστήματος εξ' αιτίας ερεθισμάτων έχει ως αποτέλεσμα τη δράση των ιδρωτοποιών που οδηγεί με τη σειρά του σε μεταβολή του επιπέδου της αγωγιμότητας. Στη παρούσα διπλωματική εργασία θα εξετάσουμε τη σχέση/σύνδεση που υπάρχει ανάμεσα στη συμπαθητική δραστηριότητα του αυτόνομου νευρικού συστήματος και στη συναισθηματική διέγερση, παρ' ότι υπάρχουν δυσκολίες στην αναγνώριση του εκάστοτε εκδηλωμένου συναισθήματος. Η αλλαγή αυτή συνεπώς στη συναισθηματική κατάσταση του υποκειμένου καταγράφεται μέσω ενός συστήματος, όπου το σήμα ουσιαστικά

καταγράφει τις μεταβολές στην αγωγιμότητα του δέρματος λόγω της λειτουργίας/δράσης των ιδρωτοποιών αδένων. (<http://www.bem.fi/book/27/27.htm>)

Η αγωγιμότητα του δέρματος είναι ένα μέτρο της ικανότητας του δέρματος να άγει ηλεκτρισμό. Μία ασθενής ηλεκτρική τάση εφαρμόζεται μέσω δύο ηλεκτροδίων, συνήθως στερεωμένων σε δύο δάχτυλα ενός χεριού, ούτως ώστε να δημιουργηθεί ένα ηλεκτρικό κύκλωμα όπου το ανθρώπινο σώμα μετατρέπεται σε μεταβλητό αντιστάτη και στη συνέχεια υπολογίζεται η σε πραγματικό χρόνο μεταβολή της αγωγιμότητας (το αντίστροφο της αντίστασης).

Η αγωγιμότητα του δέρματος αντιπροσωπεύει τις αλλαγές στο συμπαθητικό νευρικό σύστημα. Καθώς ένα άτομο αγχώνεται περισσότερο ή λιγότερο, η αγωγιμότητα του δέρματός του αυξάνει ή μειώνεται ανάλογα. Παράδειγμα, ένα οδυνηρό ερέθισμα, όπως μια κεφαλή καρφίτσας προκαλεί μια συμπαθητική ανταπόκριση από τους ιδρωτοποιούς αδένες, την αύξηση της έκκρισης τους. Αν και η αύξηση αυτή είναι γενικά πολύ μικρή, ο ιδρώτας περιέχει νερό και ηλεκτρολύτες που αυξάνουν την ηλεκτρική αγωγιμότητα, μειώνοντας έτσι την ηλεκτρική αντίσταση του δέρματος. Οι αλλαγές αυτές με τη σειρά τους επηρεάζουν το GSR. Ένα άλλο κοινό παράδειγμα είναι η αγγειοδιαστολή (διαστολή) των αιμοφόρων αγγείων στο πρόσωπο, που αναφέρεται ως το κοκκίνισμα, καθώς και η αυξημένη εφίδρωση που συμβαίνει όταν κάποιος είναι σε δύσκολη θέση. (L.Pizzamiglio, 1999)

Ο αισθητήρας δέρματος αποτελείται από δυο αγωγούς που στην κατάληξή τους έχουν ηλεκτρόδια. Αυτά εφαρμόζονται στα δάχτυλα του ενός χεριού, ώστε να τυλίγουν το δάκτυλο αρκετά σφικτά, όχι όμως τόσο ώστε να περιορίζουν την κυκλοφορία του αίματος, καθώς αυτό μπορεί να επηρεάσει την μέτρηση της GSR. Η επιφάνεια του ηλεκτροδίου πρέπει να βρίσκεται συνεχώς σε επαφή με το μαλακό τμήμα του δακτύλου ώστε να λαμβάνεται σωστά το σήμα ενώ για επιπλέον βοήθεια μπορεί να χρησιμοποιηθεί κρέμα αγωγιμότητας υπερήχων. (L.Pizzamiglio, 1999)

Η τυπική μονάδα μέτρησης για την αγωγιμότητα είναι το Siemens. Η αγωγιμότητα του δέρματος μετριέται σε micro-Siemens kS ενώ κάποια συστήματα βίο-ανάδρασης εμφανίζουν την αγωγιμότητα του δέρματος σε micro-mhos (το mhos είναι το αντίστροφο του ohm, μονάδας μέτρησης της αντίστασης), που είναι ισοδύναμο του kS. Φυσιολογικές ενδείξεις για την αγωγιμότητα του δέρματος σε ήρεμη κατάσταση είναι περίπου 2 kS.

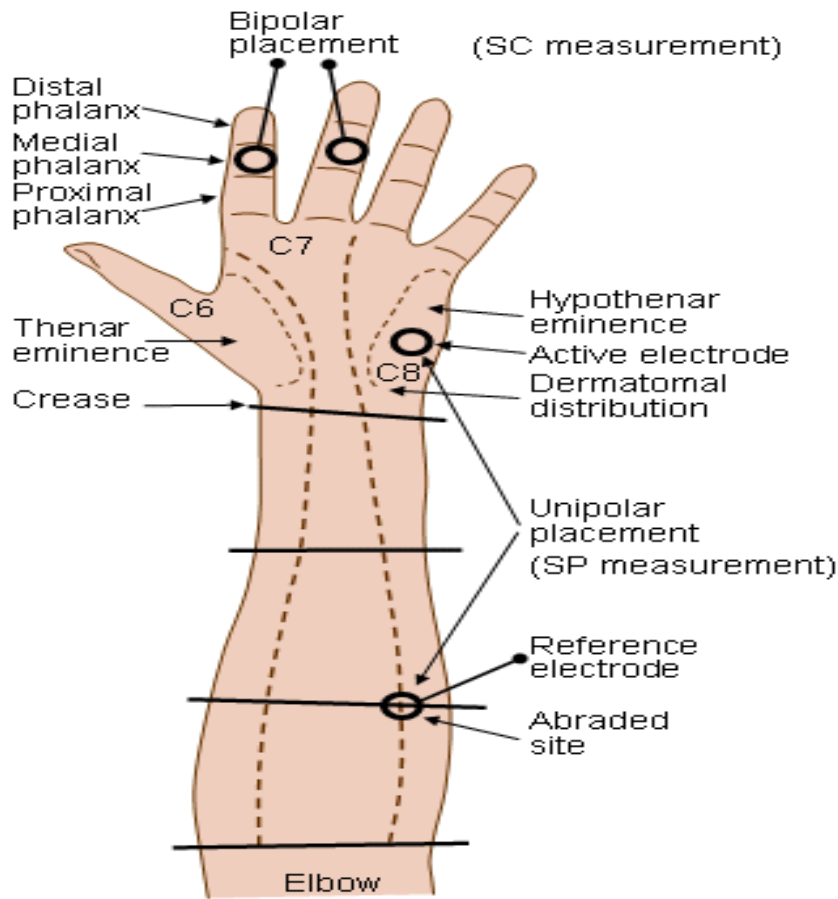
Λόγω της απόκρισης του δέρματος και του μυϊκού ιστού από εξωτερικά και εσωτερικά ερεθίσματα, η αγωγιμότητα μπορεί να ποικίλλει αρκετά micro Siemens. Όταν βαθμονομηθεί σωστά, η συσκευή μπορεί να μετρήσει αυτές τις λεπτές διαφορές. Αυτές οι μεταβολές στο αυτόνομο συμπαθητικό νευρικό σύστημα μεταβάλλουν τον ιδρώτα και τη ροή του αίματος, οι οποίες με τη σειρά του επηρεάζει την GSR. (R. Pflanzner, 2000)

Η SCR (Skin Conductivity Response δερματική απόκριση αγωγιμότητας) είναι ιδιαίτερα ευαίσθητη στα συναισθήματα σε ορισμένους ανθρώπους. Ο φόβος, ο θυμός, η λύπη, η χαρά, τα σεξουαλικά συναισθήματα είναι κάποιες από τις αντιδράσεις που μπορεί να παράγουν παρόμοιες αποκρίσεις αγωγιμότητας του δέρματος. Αυτές οι αντιδράσεις χρησιμοποιούνται ως μέρος του ανιχνευτή ψεύδους. Η αγωγιμότητα του δέρματος, η γαλβανική απόκριση του δέρματος και η ηλεκτροδερματική απόκριση είναι διαφορετικοί όροι για παρόμοιες φυσιολογικές μετρήσεις. (R. Pflanzner, 2000)

Ο τύπος του σήματος που καταγράφεται από ένα αισθητήρα αγωγιμότητας δέρματος δείχνει σχετικά γρήγορες αυξήσεις και πιο αργές μειώσεις.

Οι συνδυασμένες αλλαγές μεταξύ γαλβανικής αντίστασης του δέρματος και γαλβανικού δυναμικού του δέρματος συνθέτουν τη γαλβανική αντίδραση του δέρματος. Η γαλβανική δερματική αντίσταση (GSR) αναφέρεται στην καταγραφή της ηλεκτρικής αντίστασης μεταξύ των δύο ηλεκτροδίων, όταν ένα πολύ ασθενές ρεύμα σταθερά διέρχεται μεταξύ τους. Τα ηλεκτρόδια τοποθετούνται συνήθως περίπου δυο εκατοστά χώρια, και η καταγραφείσα αντίσταση ποικίλλει ανάλογα με τη συναισθηματική κατάσταση του υποκειμένου. Το γαλβανικό δυναμικό του δέρματος (GSP) αναφέρεται στην τάση που μετριέται μεταξύ δύο ηλεκτροδίων χωρίς οποιοδήποτε εξωτερικά εφαρμοζόμενο ρεύμα. Μετριέται με τη σύνδεση των ηλεκτροδίων σε έναν ενισχυτή τάσης. Ομοίως, αυτή η

τάση μεταβάλλεται με τη συναισθηματική κατάσταση του υποκειμένου (Plonsey, 2012)



Εικόνα 9 Προτεινόμενες θέσεις ηλεκτροδίων στην παλάμη για τη μέτρηση της GSR.
(Redrawn από Venables και Christie, 1980.)

ΚΕΦΑΛΑΙΟ 3

3 ΚΕΦΑΛΑΙΟ 3 ΕΙΣΑΓΩΓΗ ΣΤΟ ARDUINO ΚΑΙ ΣΤΗ PROCESSING

3.1 Εισαγωγή στο Arduino

Ο arduino θα λέγαμε ότι είναι ένα εργαλείο για να κατασκευάσουμε ένα υπολογιστικό σύστημα με την έννοια ότι αυτό θα ελέγχει συσκευές του φυσικού κόσμου, σε αντίθεση με τον κοινό μας ηλεκτρονικό υπολογιστή. Είναι ανοιχτού υλικού και λογισμικού και βασίζεται σε μια αναπτυξιακή πλακέτα που ενσωματώνει επάνω έναν μικροελεγκτή. Συνδέεται με τον Η/Υ για να τον προγραμματίσουμε μέσα από ένα απλό περιβάλλον ανάπτυξης. Ένας Arduino μπορεί να χρησιμοποιηθεί για να αναπτύξουμε διαδραστικά αντικείμενα, να δεχτούμε εισόδους από πληθώρα αισθητηρίων οργάνων και διακόπτες, αλλά και να ελέγχουμε διάφορα φώτα, κινητήρες και άλλες συσκευές εξόδου του φυσικού κόσμου. Τα έργα που θα κατασκευάσουμε μπορούν να είναι αυτόνομα (σε επίπεδο hardware) ή να επικοινωνούν με κάποιο software στον Η/Υ του προγραμματιστή (προγράμματα όπως Processing, python, MaxMSP). Οι πλακέτες μπορούν εύκολα να συναρμολογηθούν ακόμη και από έναν αρχάριο ή να αγοραστούν μονταρισμένες. Το περιβάλλον ανάπτυξης του λογισμικού βασίζεται στην γλώσσα προγραμματισμού Processing και την γλώσσα προγραμματισμού Wiring, οι οποίες είναι ανοιχτού κώδικα και μπορεί κάποιος να τις "κατεβάσει δωρεάν".

Το Arduino εκτός του ότι απλοποιεί την διαδικασία να δουλεύει κάποιος με μικροελεγκτές, προσδίδει και επιπλέον πλεονεκτήματα :

- i. **Φθηνός:** Οι πλακέτες του Arduino είναι εξαιρετικά φθηνές σε σχέση με άλλες πλατφόρμες μικροελεγκτών. Ειδικά δε μπορεί με τα σχηματικά που κυκλοφορούν στο Internet να κατασκευάσει κάποιος την φθηνότερη εκδοχή ενός Arduino. Ωστόσο ακόμα και αν προμηθευτεί την έτοιμη αυτή θα κοστίσει το μέγιστο 50 euro ανάλογα με την έκδοση.
- ii. **Πληθώρα Λειτουργικών Συστημάτων:** Το περιβάλλον προγραμματισμού του Arduino είναι για Windows, Macintosh OSX και για λειτουργικά συστήματα Linux.

- iii. **Απλό, ξεκάθαρο προγραμματιστικό περιβάλλον:** Το περιβάλλον προγραμματισμού ενός Arduino ενδείκνυται για αρχάριους, αλλά είναι ταυτόχρονα και ευέλικτο και για πιο προχωρημένους χρήστες.
- iv. **Ανοιχτού λογισμικού που επεκτείνεται και παραμετροποιείται:** Το software του Arduino διανέμεται με την μορφή εργαλείων ανοιχτού λογισμικού και είναι διαθέσιμο προς επέκταση για έμπειρους προγραμματιστές. Η γλώσσα προγραμματισμού του μπορεί να επεκταθεί διαμέσου των βιβλιοθηκών της C++ και οι άνθρωποι που θέλουν να ασχοληθούν περισσότερο με τους μικροελεγκτές μπορούν να μεταβούν από τον Arduino στην AVR-C. Επίσης μπορεί ο οποιοσδήποτε να προσθέσει κώδικα της AVR-C στο πρόγραμμα που έχει γράψει για το δικό του Arduino.
- v. **Ανοιχτού Υλικού το οποίο μπορεί να επεκταθεί:** Το Arduino βασίζεται στους μικροελεγκτές της Atmel. Τα σχηματικά για τα αναπτυξιακά είναι κάτω από την άδεια της CreativeCommons, επιτρέποντας σε έμπειρους σχεδιαστές να κατασκευάσουν το δικό τους αναπτυξιακό, εξελίσσοντας το ήδη υπάρχον χωρίς να έχουν νομικά προβλήματα.

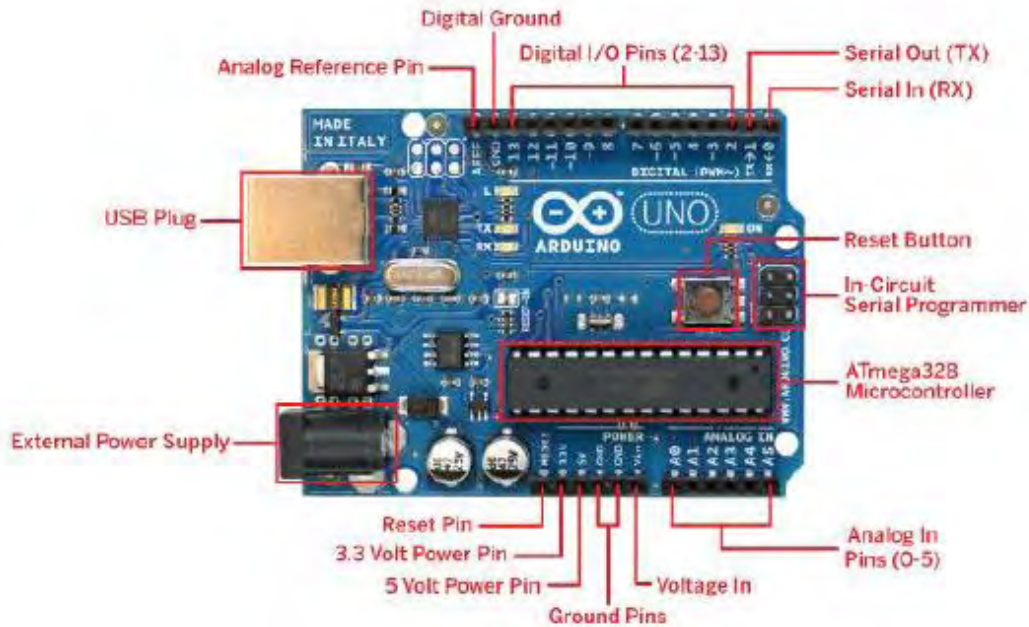
3.2 Arduino Family

Υπάρχουν πολλές διαφορετικές εκδόσεις πλακετών του Arduino, κάθε μία από αυτές έχει διαφορετικά χαρακτηριστικά και χρησιμοποιείται αναλόγως για διάφορους σκοπούς. Τα μοντέλα Arduino που κυκλοφορούν στην αγορά, παρουσιάζονται στη παρακάτω εικόνα με όλες τις διαφορετικές εκδόσεις και τα χαρακτηριστικά του (<https://www.arduino.cc/en/Products.Compare>, 2018)

Πίνακας 1 Σύγκριση Επεξεργαστών

| Όνομα | Επεξεργαστής | Τάση λειτουργίας/εισόδου | Ταχύτητα CPU | Αναλογικοί είσοδοι/έξοδοι |
|-----------|--------------|--------------------------|--------------|---------------------------|
| Uno | ATmega328P | 5V / 7-12V | 16 MHz | 6/0 |
| 101 | Intel Curie | 3.3V / 7-12V | 32 MHz | 6/0 |
| Robot | ATmega32u4 | 5V | 16 MHz | 6/0 |
| Mega 2560 | ATmega2560 | 5V / 7-12V | 16 MHz | 16/0 |
| Zero | ATSAMD21G18 | 3.3V / 7-12V | 48 MHz | 6/1 |
| Due | ATSAM3X8E | 3.3V / 7-12V | 84 MHz | 12/2 |
| Mega ADK | ATmega2560 | 5V / 7-12V | 16 MHz | 16/0 |
| Pro | ATmega168 | 3.3V / 3.35-12V | 8 MHz | 6/0 |
| | ATmega328P | 5V / 5-12V | 16 MHz | |
| Yun | ATmega32U4 | 5V | 16 MHz | 12/0 |
| | AR9331 Linux | 3.3V | 400 MHz | |
| Ethernet | ATmega328P | 5V / 7-12V | 16 MHz | 6/0 |
| Gemma | ATiny85 | 3.3V / 4-16V | 8 MHz | 1/0 |
| Lilypad | ATmega168V | 2.7-5.5V/ | 8 MHz | 6/0 |
| | ATmega328P | 2.7-5.5V | | |

Το ArduinoUno είναι η καλύτερη επιλογή για κάποιον καινούργιο χρήστη που θέλει να πειραματιστεί με τα ηλεκτρονικά κυκλώματα και τον προγραμματισμό τους. Το μοντέλο αυτό είναι το πιο συχνά χρησιμοποιούμενο και διαθέτει τις περισσότερες πωλήσεις από όλη την οικογένεια των Arduino. Χρησιμοποιεί τον επεξεργαστή ATmega 328P (8-bit), έχει τάση λειτουργίας 5V και ταχύτητα ρολογιού 16MHz. Στη συγκεκριμένη διπλωματική εργασία, επιλέχθηκε το μοντέλο Arduino unoR3 μιας και υπήρχε στη διάθεσή μας και παράλληλα καλύπτει όλες τις ανάγκες της εργασίας, τα pins που διαθέτει επαρκούν για όλες τις λειτουργίες, όπως και η μνήμη του μικροελεγκτή είναι αρκετή για να αποθηκεύει και να επεξεργάζεται τον κώδικα που χρησιμοποιήσαμε.



Εικόνα10Arduino Uno Pins

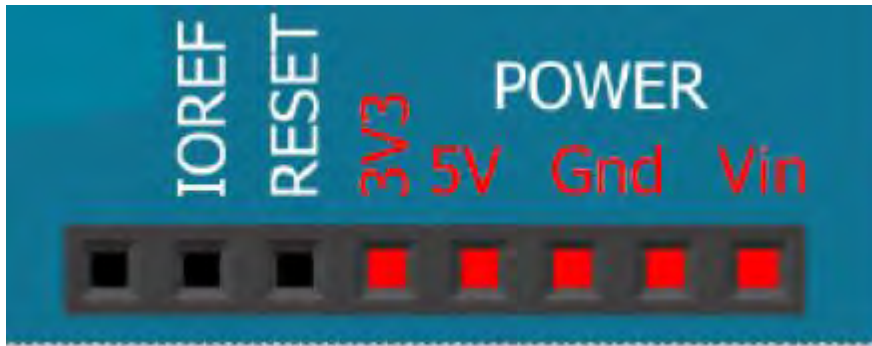
Πίνακας 2 Χαρακτηριστικά Uno (<https://store.arduino.cc/arduino-uno-rev3>, 2018)

| Μικροελεγκτής | ATmega328 |
|-------------------------------|---------------------|
| Τάση λειτουργίας | 5V |
| Τάση εισόδου (όρια) | 7-12 V |
| Ψηφιακές I / O ακίδες | 6-20 V |
| Αναλογικοί ακροδέκτες εισόδου | 14 |
| DC ρεύμα ανά I / OPin | 40 mA |
| DC ρεύμα για 3.3VPin | 50 mA |
| Μνήμη Flash | 32 KB |
| BootLoader | 0.5 KB (from Flash) |
| Μνήμη SRAM | 2 KB |
| Μνήμη EEPROM | 1 KB |
| Clock speed | 16 Mhz |
| Διαστάσεις Μήκος Πλάτος | 68,5 * 53,4 (mm) |
| Βάρος | 25 gr |

Τροφοδοσία

Το Arduino Uno μπορεί να τροφοδοτηθεί μέσω USB καλωδίωσης ή με εξωτερικό τροφοδοτικό. Η πηγή τροφοδοσίας επιλέγεται αυτόματα.

Η εξωτερική τροφοδοσία χωρίς USB μπορεί να προέλθει είτε από έναν AC-to-DC αντάπτορα είτε από μπαταρία. Ο αντάπτορας μπορεί να ενωθεί με ένα 2.1mm κεντρικό θετικό βύσμα στην υποδοχή ρεύματος (power jack) της πλακέτας. Οι αγωγοί σύνδεσης



Εικόνα 12 Τροφοδοσία

από την μπαταρία μπορούν να τοποθετηθούν στο Gnd και Vin ακροδέκτες της πλακέτας.

Η πλακέτα μπορεί να λειτουργήσει από εξωτερική πηγή των 6 έως 20 volts. Παρ όλα αυτά αν τροφοδοτηθεί με λιγότερο από 7V, ο 5V ακροδέκτης ίσως τροφοδοτήσει με λιγότερο από πέντε volt και η πλακέτα θα είναι ασταθής. Αν χρησιμοποιηθούν περισσότερα από 12V, ο ρυθμιστής τάσης μπορεί να υπερθερμανθεί και να προκαλέσει ζημιά την πλακέτα. Το προβλεπόμενο εύρος τιμών είναι 7 με 12 volts.

Οι ακροδέκτες της τροφοδοσίας είναι οι εξής:

- VIN. Η εισερχόμενη τάση του Arduino όταν τροφοδοτείται από εξωτερική πηγή. 3.3V. Μια τάση 3.3V παράγεται από τον ρυθμιστή που βρίσκετε πάνω στην πλακέτα. Η μέγιστη κατανάλωση ρεύματος είναι 50mA.
- GND. Ακροδέκτης γείωσης
- IOREF. Αυτός ο ακροδέκτης παρέχει στην πλακέτα την τάση αναφοράς με την οποία ο μικροελεγκτής λειτουργεί. Μια πλακέτα που θα προσθέσουμε (shield) θα μπορεί να διαβάσει την τάση από τον IOREF ακροδέκτη και να επιλέξει την κατάλληλη πηγή τάσης ή να ενεργοποιήσει τους επιλογείς τάσης για τις εξόδους δουλεύοντας με 5V ή 3.3V.

Μνήμη: Ο ATmega328 έχει 32 KB(με 0.5 KB να χρησιμοποιούνται για τον bootloader). Επίσης έχει 2KBSRAMκαι 1KBEEPROM(η οποία μπορεί να διαβαστεί και να γραφεί μέσω της EEPROM βιβλιοθήκης)

Είσοδοι/Εξοδοι

Κάθε ένα από τους 14 ψηφιακούς ακροδέκτες στο Uno μπορούν να χρησιμοποιηθούν σαν είσοδοι και έξοδοι κάνοντας χρήση των μεθόδων `pinMode()`, `digitalWrite()` και `digitalRead()`.

Λειτουργούν στα 5 volts. Κάθε ακροδέκτης μπορεί να διαθέσει ή να δεχτεί μέγιστο ρεύμα των 40mAκαι έχουν μια εσωτερική pull-up αντίσταση των 2050kOhms. Σε αντίθεση κάποιοι ακροδέκτες έχουν διαφορετικές ιδιότητες-λειτουργίες:

Serial: 0 (RX) και 1 (TX). Χρησιμοποιείτε για λήψη (RX) και εκπομπή (TX) TTL σειριακών δεδομένων. Αυτοί οι ακροδέκτες είναι συνδεδεμένοι με τα αντίστοιχους του



Εικόνα 13Είσοδοι/Εξοδοι

ATmega8U2 USB-to-TTLSerialchip.

External Interrupts: 2 and3. Αυτοί οι ακροδέκτες μπορούν να τροποποιηθούν για να προκαλούν διακοπές σε μία χαμηλή τιμή, σε μία ανοδική η καθοδική ακμή ή σε μία αλλαγή μίας τιμής.

PWM: 3, 5, 6, 9, 10, και 11. Παρέχουν 8-bitPWMέξοδο με την `analogWrite()` μέθοδο.

SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Αυτοί οι ακροδέκτες υποστηρίζουν SPI επικοινωνία μέσω της βιβλιοθήκης SPI.

LED: 13. Υπάρχει ένα ενσωματωμένο led ενωμένο με το ψηφιακό ακροδέκτη 13. Όταν θέτουμε στον ακροδέκτη την τιμή HIGH, το led είναι ανοικτό, ενώ όταν είναι LOW, τότε είναι κλειστό

Το UNO έχει 6 αναλογικές εισόδους, με τις ετικέτες A0 μέχρι A5, κάθε μια από αυτές



Εικόνα 14 Αναλογικές εισοδοι

παρέχει 10bit ανάλυση (δηλαδή $2^{10} = 1024$ διαφορετικές τιμές). Είναι προεπιλεγμένες να μετρούν από την γείωση μέχρι 5 volts. Είναι εφικτό να αλλάξουμε το όριο (άνω των 5volts) χρησιμοποιώντας τον AREF ακροδέκτη (τάση αναφοράς) και την `analogReference()` μέθοδο.

WI: A4 ή SDA και A5 ή SCL ακροδέκτες. Υποστηρίζουν TWI επικοινωνία χρησιμοποιώντας την Wire βιβλιοθήκη.

Reset. Κατεβάζουμε αυτή την γραμμή LOW για να κάνουμε επανεκκίνηση του μικροελεγκτή. Τυπικά χρησιμοποιείται για να δημιουργούμε ένα κουμπί επανεκκίνησης όταν έχουμε κάποιο shield τοποθετημένο και δεν μπορούμε να χρησιμοποιήσουμε αυτό του UNO.

Επικοινωνία

Το Arduino υποδιαθέτει διάφορους τρόπους επικοινωνίας με έναν υπολογιστή, με ένα άλλο Arduino ή έναν άλλο μικροελεγκτή. Ο ATmega328 παρέχει UARTTTL (5V) σειριακή επικοινωνία, η οποία είναι διαθέσιμη στους ψηφιακούς ακροδέκτες 0 (RX) και 1 (TX). Ένας ATmega16U2 μεταφέρει αυτή τη σειριακή επικοινωνία μέσω USB ενώ εμφανίζεται σαν εικονική com πόρτα στο λογισμικό του υπολογιστή. Το 16U2 firmware χρησιμοποιεί τους κλασικούς USBCOM οδηγούς και έτσι δεν χρειάζονται εξωτερικοί. Παρόλα αυτά στα Windows ένα .inf αρχείο χρειάζεται. Το λογισμικό του Arduino περιέχει μία εικονική οθόνη η οποία απλά εμφανίζει τα δεδομένα που στέλνονται από και προς την πλακέτα. Τα RX και TX leds αναβοσβήνουν όταν δεδομένα μεταφέρονται μέσω USB to serial chip και USB ένωσης στον υπολογιστή. Η βιβλιοθήκη SoftwareSerial επιτρέπει την σειριακή επικοινωνία για κάθε ένα από τους ακροδέκτες της πλακέτας. Ο

ATmega328 επίσης υποστηρίζει I2C (TWI) και SPI επικοινωνία. Το λογισμικό του Arduino περιέχει την βιβλιοθήκη Wire για να απλοποιήσει την χρήση του I2C διαύλου.

Προγραμματισμός

Το Arduino Uno μπορεί να προγραμματιστεί μέσω του λογισμικού ArduinoIDE. Ο ATmega328 έρχεται με προ-εγκατεστημένο bootloader που επιτρέπει να ανεβάσουμε το κώδικα μας χωρίς την χρήση εξωτερικού προγραμματιστή. Επικοινωνεί χρησιμοποιώντας το STK500 πρωτόκολλο. Μπορούμε επίσης να παρακάμψουμε τον bootloader και να προγραμματίσουμε τον μικροελεγκτή μέσω της ICSP (In-Circuit Serial Programming) επικεφαλίδας χρησιμοποιώντας το ArduinoISP ή κάτι παρόμοιο.

Ο firmware κώδικας του ATmega16U2 είναι διαθέσιμος. Ο ATmega16U2 είναι φορτωμένος με έναν DFU bootloader, ο οποίος μπορεί να ενεργοποιηθεί από την αντίσταση που τραβάει ο 8U2/16U2 HWB στην γείωση, κάνοντας το έτσι πιο εύκολο να μπαίνει σε DFU λειτουργία.

Μπορούμε να χρησιμοποιήσουμε το λογισμικό Flip της Atmel για Windows ή το DFU programmer για MacOSX και Linux για να φορτώσουμε firmware. Διαφορετικά μπορούμε να κάνουμε χρήση της ISP επικεφαλίδας με εξωτερικό προγραμματιστή.

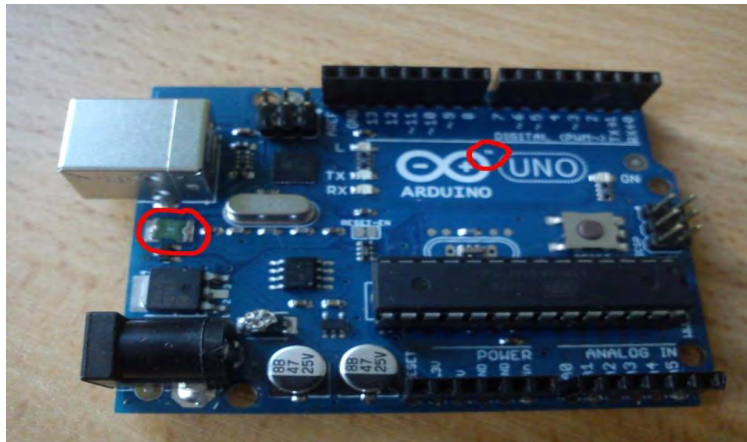
Αυτόματη Επανεκκίνηση Λογισμικού

Αντί να απαιτείται το πάτημα του πλήκτρου της επανεκκίνησης πριν από ένα ανέβασμα προγράμματος, το ArduinoUno είναι σχεδιασμένο με τέτοιο τρόπο που επιτρέπει την επανεκκίνηση από το λογισμικό που τρέχει στον συνδεδεμένο υπολογιστή. Μία από τις γραμμές ελέγχου ροής υλικού (DTR) του ATmega8U2/16U2 συνδέεται με τη γραμμή επαναφοράς των ATmega328 μέσω ενός πυκνωτή 100 nfarad. Όταν αυτή η γραμμή είναι χαμηλή, η γραμμή επανεκκίνησης πέφτει τόσο όσο το chip να κάνει επανεκκίνηση. Το λογισμικό μας επιτρέπει να ανεβάσουμε κώδικα πολύ απλά πατώντας ένα κουμπί στο περιβάλλον του λογισμικού. Αυτό σημαίνει ότι ο bootloader μπορεί να έχει μικρότερο χρονικό όριο, όπως η μείωση του DTR μπορεί να είναι καλά συντονισμένη με την έναρξη του ανεβάσματος του κώδικα. Αυτή η επιλογή έχει και άλλες επιπτώσεις. Όταν η πλακέτα μας είναι ενωμένη με έναν υπολογιστή που τρέχει MacOSX ή Linux, κάνει επανεκκίνηση κάθε φορά που γίνεται μια σύνδεση σε αυτό από το λογισμικό (μέσω USB). Για το επόμενο μισό δευτερόλεπτο ο bootloader τρέχει στο Uno. Αν και είναι προγραμματισμένο να αγνοεί τα άσχετα δεδομένα (οτιδήποτε εκτός του κώδικα που

ανεβάζουμε) θα παρακολουθήσει τα πρώτα bytes δεδομένων που στέλνονται στην πλακέτα αφότου έχει γίνει σύνδεση. Αν ένα πρόγραμμα τρέχει στην πλακέτα λαμβάνει μια απλή διαμόρφωση ή δέχεται νέα δεδομένα όταν αρχικά ανοίγει, πρέπει να σιγουρευτούμε ότι το λογισμικό με το οποίο επικοινωνεί περιμένει ένα δευτερόλεπτο για να ανοίξει η επικοινωνία προτού στείλει αυτά τα δεδομένα.

Προστασία υπερέντασης USB

Το ArduinoUno έχει μία επανεκκινήσιμη αντίσταση η οποία προστατεύει τις USB θύρες του υπολογιστή μας από υπερένταση. Παρόλο που κάθε υπολογιστής παρέχει την δική του εσωτερική προστασία, η αντίσταση παρέχει μία επιπλέον προστασία. Αν περισσότερα από 500mAεφαρμοστούν στην USB πόρτα, η αντίσταση αυτόματα θα σπάσει την σύνδεση μέχρι να απομακρυνθεί η υπερφόρτωση.

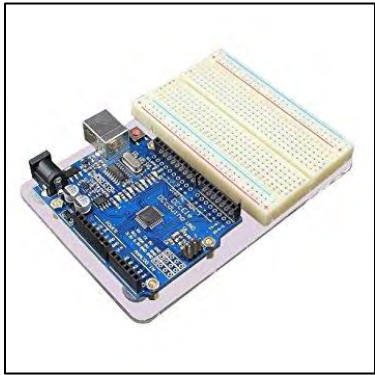


Εικόνα 15 Αντίσταση Προστασίας από Υπερφόρτωση

Φυσικά Χαρακτηριστικά

Το μέγιστο μήκος και πλάτος της πλακέτας του Uno είναι 6.858 και 5.334 εκατοστά αντίστοιχα, με την USB πόρτα και την υποδοχή ρεύματος να εκτείνονται πέρα των αρχικών διαστάσεων. Τέσσερις τρύπες για βίδες επιτρέπουν στην πλακέτα να μπορεί να τοποθετηθεί σε μία ξεχωριστή επιφάνεια ή ένα κουτί κατάλληλο για να προφυλάξεις τη πλακέτα από σκόνη και χτυπήματα. Το κουτί διαθέτει εσωτερικά βάσεις και στηρίζει τον Arduino, στο επάνω μέρος έχει οπές με πρόσβαση στα pins της πλακέτας και στο κουμπί reset, στο μπροστά μέρος έχει πρόσβαση στο βύσμα USB και στο βύσμα τροφοδοσίας.

Διαστάσεις: 7.5 x 6 x 2cm

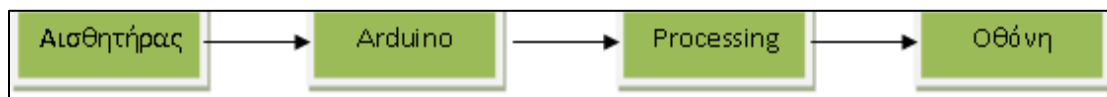


Εικόνα 17 Arduino σε Βάση με Breadboard



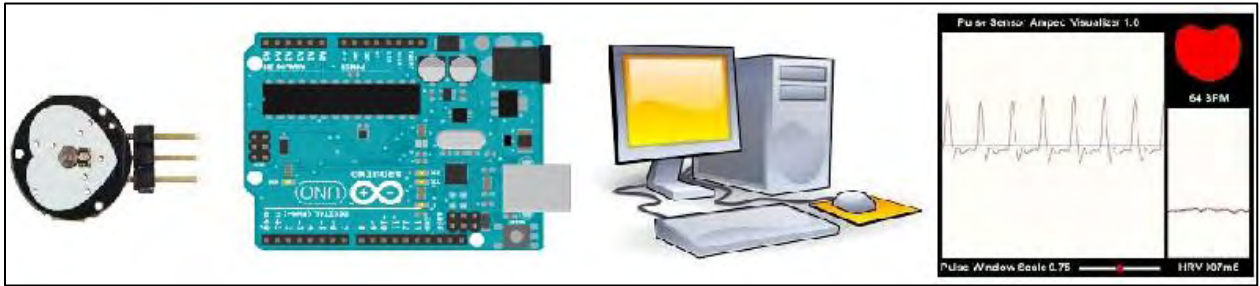
Εικόνα 16 Arduino σε Θήκη

Σε κάθε μικροελεγκτή οι μετρήσεις πραγματοποιούνται από τους αισθητήρες ενώ ο ελεγκτής- Arduino μπορεί να τις αναγνωρίζει , να τις επεξεργάζεται , να τις εμφανίζει ή και να τις αποθηκεύει. Μερικές φορές όμως θέλουμε να απεικονίζουμε πολλές μετρήσεις μαζί με διαφορετικά χρώματα ή και να αναπτύξουμε μια πιο σύνθετη διεπαφή χρήστη (User interface)και να υπάρχει ταυτόχρονα δυνατότητα επεξεργασίας σε πραγματικό χρόνο. Έτσι η απεικόνιση μετρήσεων από το Arduino στην οθόνη του Η/Υ θα μπορεί να γίνει μόνο αν οι μετρήσεις αποστέλλονται προς τον υπολογιστή .



εικόνα 18 Μεταφορά μετρήσεων και απεικόνιση στην οθόνη

Το Arduino διαβάζει τις μετρήσεις του αισθητήρα τις οποίες και στέλνει στη σειριακή θύρα USB μέσω της οποίας συνδέεται στον υπολογιστή, αυτός τις λαμβάνει και τις απεικονίζει στην οθόνη.



Εικόνα 19Block διάγραμμα συνδεσμολογίας

Στη διάταξη της παραπάνω εικόνας εκτελούνται δύο προγράμματα.

- Το πρώτο εκτελείται στο μικροελεγκτή Arduino, συλλέγει τις μετρήσεις του αισθητήρα, πραγματοποιεί την επεξεργασία που απαιτείται και στέλνει τις μετρήσεις κατάλληλα διαμορφωμένες στη σειριακή θύρα (USB) .
- Ένα δεύτερο πρόγραμμα λαμβάνει τις μετρήσεις από τη σειριακή θύρα και πραγματοποιεί τη γραφική απεικόνιση στην οθόνη.

Η γλώσσα Processing πλεονεκτεί ειδικά στη περίπτωση του arduino έχοντας τη δυνατότητα να ελέγχει τη λειτουργία του συστήματος χρησιμοποιώντας το ποντίκι ή και το πληκτρολόγιο με δικές τις έτοιμες συναρτήσεις.

3.3 Τι είναι η γλώσσα προγραμματισμού processing







Η Processing είναι μία γλώσσα προγραμματισμού ανοικτού κώδικα. Αποτελεί περιβάλλον ανάπτυξης για άτομα που θέλουν να δημιουργούν εικόνες, κινούμενα σχέδια και διάφορες αλληλεπιδράσεις. Στην αρχή αναπτύχθηκε σαν σχεδιαστικό πρόγραμμα για να διδάξει βασικές αρχές προγραμματισμού μέσα σε ένα οπτικό πλαίσιο. Στη συνέχεια εξελίχθηκε σε ένα εργαλείο δημιουργίας ολοκληρωμένων επαγγελματικών εργασιών. Η Processing βασίστηκε στη δυνατότητες γραφικών της γλώσσας προγραμματισμού Java, απλοποιώντας την χρήση και δημιουργώντας νέα χαρακτηριστικά. Σήμερα δεκάδες χιλιάδες σπουδαστές, καλλιτέχνες, σχεδιαστές, ερευνητές και χομπίστες χρησιμοποιούν τη Processing για διδασκαλία, προτυποποίηση και παραγωγή. Μερικά από τα χαρακτηριστικά της είναι:

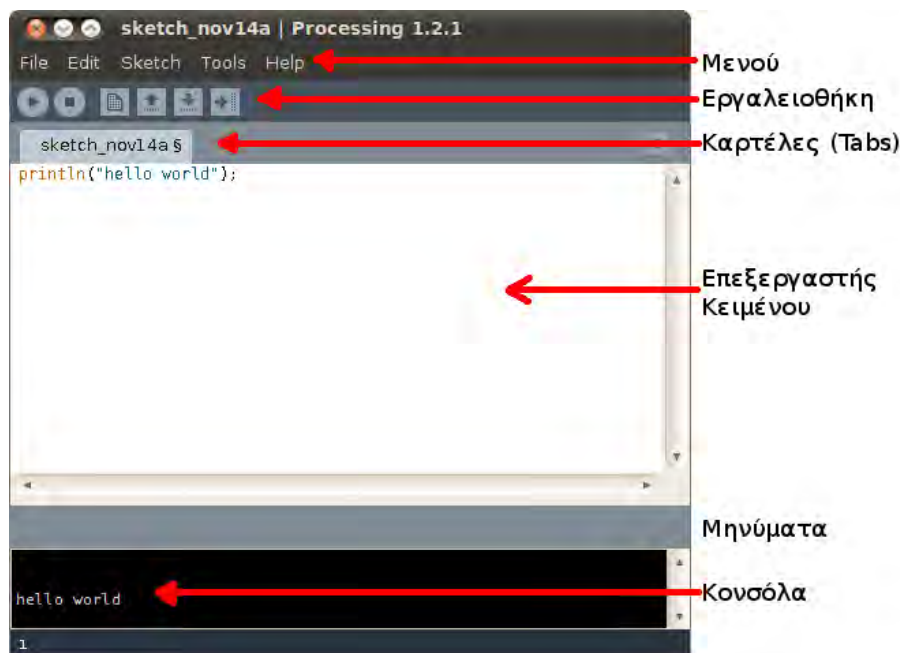
- Είναι ελεύθερο/ανοικτό λογισμικό με άδεια χρήσης GPL/LGPL.
- Είναι πολυπλατφορμική, μπορεί να τρέξει σε λειτουργικά συστήματα GNU/Linux, MacOSX και Windows.

- Δημιουργεί διαδραστικά προγράμματα χρησιμοποιώντας διδιάστατα (2D) ή τρισδιάστατα (3D) γραφικά.
- Ενσωμάτωση της OpenGL για επιτάχυνση 3D.
- Δημιουργία stand-alonedesktop εφαρμογών και web-based εφαρμογών (applets).
- Υπάρχουν αρκετές βιβλιοθήκες επέκτασης της γλώσσας, για εφαρμογές ήχου, βίντεο, τεχνητής όρασης, κ.α (<http://www.linuxinsider.gr>)

Στη παρακάτω εικόνα βλέπουμε την διεπαφή του περιβάλλοντος ανάπτυξης, ενώ στον πίνακα φαίνονται οι βασικές λειτουργίες της εργαλειοθήκης του

Πίνακας 3 Βασικές λειτουργίες

| | |
|--|---|
|  | Έλεγχος του κώδικα για λάθη/Εκτέλεση του προγράμματος. |
|  | Τερματισμός εκτέλεσης του προγράμματος. |
|  | Δημιουργία νέου έργου. |
|  | Εμφάνιση μενού με τα αποθηκευμένα έργα. Πατώντας σε ένα από αυτά ανοίγει το έργο για επεξεργασία. |
|  | Αποθήκευση του έργου. |
|  | Δημιουργία αυτόνομης εφαρμογής (<u>Export</u>). |



Εικόνα 20 Το περιβάλλον της Processing

Η δομή ενός προγράμματος Processing

Ένα τυπικό πρόγραμμα στη Processing έχει την παρακάτω δομή:

```
//δήλωση μεταβλητών
void setup()
{
  //αρχικοποίηση
}
void draw()
void setup()
{
  //κώδικας
}
```

εικόνα21 Δομή Processing

Στη Processing υπάρχουν δυο βασικές συναρτήσεις σε ένα τυπικό πρόγραμμα .

Η συνάρτηση setup() εκτελείται στην αρχή του προγράμματος και για μία μόνο φορά. Χρησιμοποιείται για τις αρχικοποιήσεις ιδιοτήτων και βιβλιοθηκών, όπως για παράδειγμα: το μέγεθος της οθόνης, το χρώμα φόντου, το φόρτωμα εικόνων κ.α. Οι μεταβλητές που δηλώνονται στη setup() δεν είναι προσβάσιμες από άλλες συναρτήσεις, συμπεριλαμβανομένης και της draw() που θα εξηγήσουμε αμέσως μετά.

Η συνάρτηση draw() εκτελείται αμέσως μετά την setup() και ο κώδικας που γράφεται μέσα στη συνάρτηση αυτή επαναλαμβάνεται συνεχώς μέχρι να τερματιστεί η εφαρμογή ή μέχρι να κληθεί η συνάρτηση noLoop().

3.4 Σύγκριση με άλλες γλώσσες

Το περιβάλλον της Processing είναι γραμμένο σε Java. Προγράμματα που έχουν γραφτεί σε Processing είναι επίσης μεταφρασμένα σε Java και μπορούν να τρέχουν σαν Java προγράμματα. Προγράμματα που είναι γραμμένα σε Java και Processing, συνήθως τρέχουν γρηγορότερα από προγράμματα που βασίζονται σε scripting γλώσσες όπως

ActionScript και Lingo, κάτι το οποίο είναι πολύ σημαντικό για εφαρμογές με γραφικά. Σημαντικές διαφορές μεταξύ της Processing και της Java είναι οι βιβλιοθήκες γραφικών της Processing και το απλοποιημένο στυλ προγραμματισμού που δεν προϋποθέτει οι χρήστες να καταλαβαίνουν πιο προηγμένες έννοιες όπως κλάσεις, αντικείμενα, ή animation (ενώ εξακολουθούν να είναι διαθέσιμες για προχωρημένους χρήστες). Τέτοιες τεχνικές λεπτομέρειες πρέπει να είναι ειδικά προγραμματισμένες σε Java, αλλά ολοκληρωμένες σε Processing, κάνοντας τα προγράμματα μικρότερα και ευκολότερα να διαβαστούν.

3.5 Βασικά στοιχεία της γλώσσας.

Βασικές εντολές παραθύρου και σχημάτων και σημείων

size (x,y,renderer); // ορίζει το μέγεθος του παραθύρου και τον renderer που θα χρησιμοποιηθεί για τον σχεδιασμό των αντικειμένων

point(x,y); // σχεδιάζει ένα σημείο στις συντεταγμένες x,y πχ point(4,5);

line (x1,y1,x2,y2); // σχεδιάζει μια γραμμή απο σημείο X1,Y1 μέχρι το X2,Y2. Πχ line(4,5,10,10);

rect(x1, y1, πλάτος, ύψος); // σχεδιάζει ένα τετράγωνο με αρχή x1,y1 και πλάτος, ύψος όσα Pixel επιθυμούμε πχ rect(4,5,200,100);

ellipse(x1, y1, πλάτος, ύψος); // σχεδιάζει μια έλλειψη/κύκλο με κέντρο x1,y1 και πλάτος, ύψος όσα Pixel επιθυμούμε πχ ellipse(4,5,200,200);

arc(); = τόξο, triangle(); τρίγωνο. **text(text, x, y)** - Τυπώνει απλό κείμενο στην οθόνη, στο σημείο (x,y). **mouseX** – Δίνει το τρέχον σημείο του mouse στον άξονα X. **mouseY** - Δίνει το τρέχον σημείο του mouse στον άξονα Y.

save(filename) – Αποθηκεύει την τελική εικόνα στο σκληρό δίσκο. Δέχεται σαν όρισμα το όνομα και τον τύπο του αρχείου. Η Processing αποθηκεύει ψηφιακές εικόνες σε μορφή TIFF, TARGA, JPEG, και PNG.

Αποτύπωση Χρωμάτων

Εντολή **fill(x);** = γεμίζει ένα σχήμα με ένα χρώμα 0=μαύρο, 255=άσπρο, 125 γκρι και ανάλογα οι ενδιάμεσες τιμές.

nofill(); = όχι γέμισμα σχήματος.

fill(0); // οι εντολές γεμίζουν με μαύρο χρώμα το τετράγωνο της επόμενης

rect(4,5,100,50); // εντολής και όσα σχήματα θα ακολουθήσουν

Εντολή **stroke(x);** και **noStroke();** = περίγραμμα σχήματος πχ **stroke(125);**
περίγραμμα γκρι.

noStroke(); = σχήμα χωρίς περίγραμμα

Όταν επιθυμώ να αποτυπώσω έγχρωμα, αναφέρομαι στα χρώματα με 3 τιμές 0-255
πρώτη για ΚΟΚΚΙΝΟ, δεύτερη για ΠΡΑΣΙΝΟ και τρίτη για ΜΠΛΕ (RGB) πχ.

fill(255,0,0); // γέμισμα με κόκκινο 255 και καθόλου πράσινο 0 και μπλέ 0 επίσης
αντίστοιχα μπορεί να δώσω **stroke(125,200,123);** και να έχω περίγραμμα με
αντίστοιχη ένταση κάθε χρώματος.

Επίσης η εντολή **background(x);** ή **background(R,G,B);** γεμίζει το φόντο με ένα
χρώμα. (μια παράμετρος σημαίνει ασπρόμαυρο και 3 παράμετροι RGB τιμές.

Βασικές ενσωματωμένες μεταβλητές(mouse , mouse, height, width)

line(mouseX,mouseY,100,123); σχεδιάζω μια γραμμή από το σημείο που βρίσκεται ο
δείκτης του Mouse...mouseX,mouseY μέχρι το σημείο 100,123.

mouseX = γραμμή στην οποία βρίσκεται το Mouse,

mouseY = στήλη που βρίσκεται το mouse.

pmouseX και **pmouseY**. όταν κινείται το mouse μας επιστρέφουν την προηγούμενη
(Previous) θέση του mouse.

width, height. μας δίνουν το πλάτος και το ύψος του καμβά μας.

Ενσωματωμένα υποπρογράμματα ανάδρασης events(keyPressed mousePressed)

```
void mousePressed(){  
  rect(mouseX,mouseY,100,20);  
}  
void keyPressed(){  
  ellipse(mouseX,mouseY,40,20);  
}
```

Το υποπρόγραμμα **MousePressed() { }** εκτελεί το μπλόκ των
εντολών που περιέχει όταν έχω κάνει κλικ στο ποντίκι.

Το υποπρόγραμμα **keyPressed() { }** όταν πατήσω κάποιο Πλήκτρο .κ.λ.π

Η μεταβλητή `mouseButton` μπορεί να χρησιμοποιηθεί για να διαπιστωθεί ποιο κουμπί του mouse έχει πατηθεί.

`mouseReleased()` - Καλείτε κάθε φορά που ένα κουμπί του mouse απελευθερώνεται.

3.6 Δημιουργία stand-alone εφαρμογών

Η Processing μας διευκολύνει να εκτελέσουμε το πρόγραμμά μας σε οποιοδήποτε σύστημα, ως αυτόνομη εφαρμογή, κάνοντας το `export` (File->Export Application). Αφού επιλέξουμε λειτουργικό σύστημα (Windows, MacOSX ή Linux), στη συνέχεια πιέζουμε το κουμπί “Export” και δημιουργούνται στο φάκελο του έργου μας δυο φάκελοι: ο φάκελος “applet” περιέχει το έργο μας σε μορφή `javaapplet`, που μπορούμε εύκολα να το ανεβάσουμε σε κάποιο `webserver` και να το τρέξουμε μέσω `webbrowser`, ενώ ο φάκελος “application.OS” (όπου 'OS' το όνομα του λειτουργικού μας συστήματος) περιέχει ένα εκτελέσιμο `script` και της απαιτούμενες βιβλιοθήκες για να τρέξουμε το έργο μας χωρίς την χρήση της Processing, σαν αυτόνομη εφαρμογή.

Το μόνο που απαιτείται να έχουμε εγκατεστημένο στο σύστημα είναι ο JRE (Java runtime environment)

3.7 Βιβλιοθήκες και εργαλεία

Η Processing συνοδεύεται από μία πληθώρα βιβλιοθηκών και εργαλείων ανοικτού κώδικα, που σκοπό έχουν να επεκτείνουν τις δυνατότητες της γλώσσας. Υπάρχουν βιβλιοθήκες για κάθε σκοπό όπως για δημιουργία τρισδιάστατων γραφικών, δημιουργία κινουμένων σχεδίων, τεχνητή όραση, μαθηματικά, επεξεργασία ήχου, δημιουργία προσομοιώσεων κ.ά. Μπορούμε να κατεβάσουμε αρκετές βιβλιοθήκες από τη διεύθυνση <http://processing.org/reference/libraries>. Επίσης υπάρχουν εργαλεία τα οποία μπορούμε να τα βρούμε στο μενού “Tools”, τα οποία μας βοηθούν για εργασίες όπως: αυτόματη μορφοποίηση κώδικα, δημιουργία γραμματοσειρών, επιλογή χρώματος, δημιουργία συμπιεσμένου αρχείου του έργου μας κ.α.

Reference - <http://processing.org/reference/> - Επεξήγηση των εντολών της Processing με μικρά παραδείγματα.

Forum - <http://forum.processing.org/> - Για συζητήσεις και επίλυση προβλημάτων

Wiki - http://wiki.processing.org/w/Main_Page – Περιλαμβάνει τεκμηρίωση, υλικό αναφοράς, παραδείγματα κ.α

ΚΕΦΑΛΑΙΟ 4

4 ΚΕΦΑΛΑΙΟ 4 ΜΕΘΟΔΟΛΟΓΙΑ – ΑΠΟΤΕΛΕΣΜΑΤΑ

4.1 Κατασκευή και Προσαρμογή του συστήματος μετρήσεων

Η κατασκευή μας θα μπορεί να πραγματοποιεί

- Μετρήσεις σε πραγματικό χρόνο και απεικόνιση σε γραφικό περιβάλλον
- Καταγραφή του πλήθους των καρδιακών παλμών του χρήστη όταν τοποθετεί το δάκτυλό του στον αισθητήρα
- Καταγραφή του βαθμού εφίδρωσης όταν ένα άτομο τοποθετεί τα δάκτυλα του στον ανάλογο αισθητήρα
- Καταγραφή μμετρήσεων εφίδρωσης και καρδιακού παλμού
- Αποθήκευση των μετρήσεων που πραγματοποιήθηκαν σε txt.files
- Συνεχής καταγραφή των δεδομένων των διαμετρήσεων

Θα υποστηρίζει επίσης χαρακτηριστικά όπως

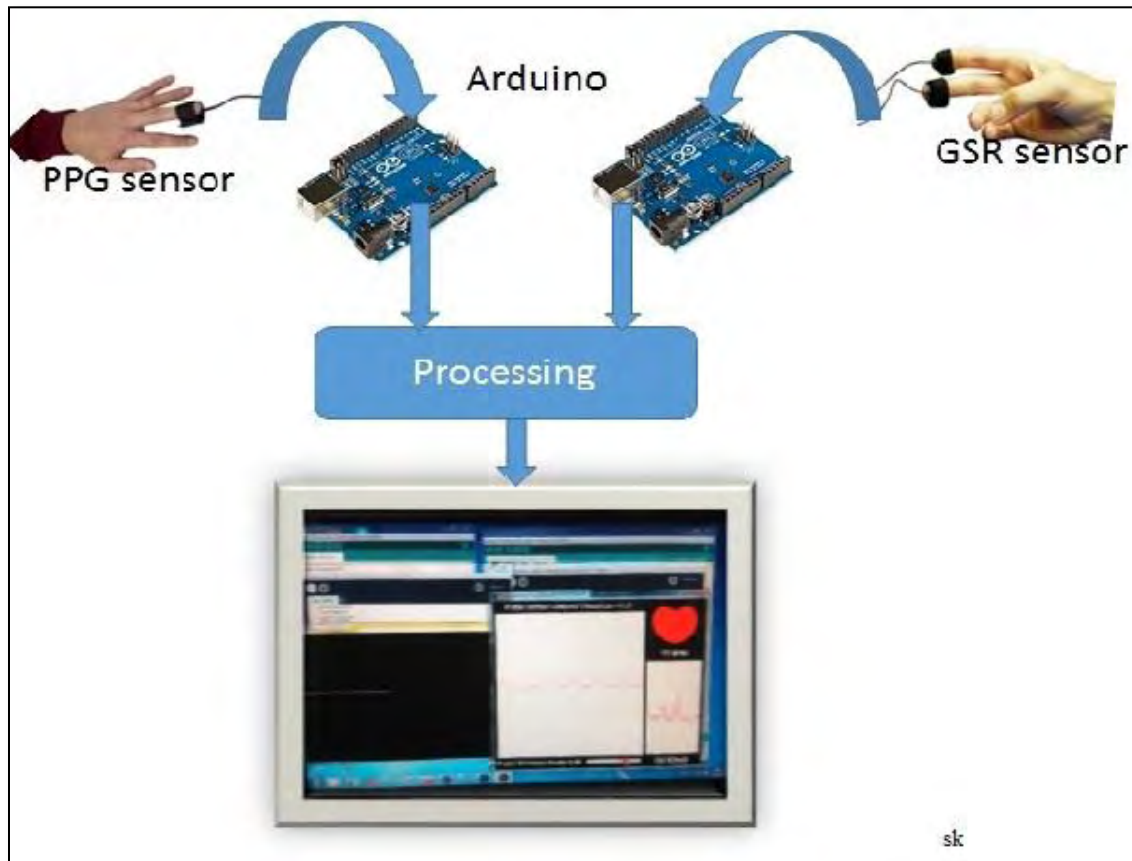
- Φθινό κόστος κατασκευής
- Μικρό μέγεθος και φορητότητα
- Ευκολία στην εγκατάσταση και τη χρήση.
- Να μην ενοχλεί/κουράζει το χρήστη
- Ευκολία στη βελτίωση/επέκταση
- Ευκολία εκτέλεσης σε διαφορετικούς υπολογιστές

Υλικά που χρειάζονται για τη λειτουργία της διάταξης

- Arduino Uno
- Ταινίες τύπου Velco για τη στερέωση των αισθητήρων στα δάκτυλα
- Αισθητήρας Pulse Sensor
- Αγωγοί Βραχυκύκλωσης
- Ένα Breadboard δοκιμών
- $R_{GSR} = 300 \text{ K}\Omega$
- LM 324 Τελεστικοί Ενισχυτές
- $R1 = 10 \text{ K}\Omega$
- $R2 = 100 \text{ K}\Omega$
- $R3 = 1 \text{ M}\Omega$
- $R4 = 1 \text{ M}\Omega$ Ποτενσιόμετρο
- $R5 = 100 \text{ K}\Omega$ Ποτενσιόμετρο

4.2 Γενικός σχεδιασμός

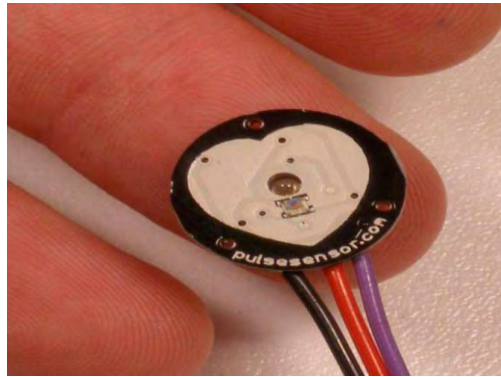
Το κύκλωμά μας βασίζεται στη συνεργασία δύο Arduino Uno και των αισθητήρων που θα περιγράψουμε παρακάτω.



Εικόνα 22 Γενικό διάγραμμα λειτουργίας και διασύνδεσης

4.3 Ο αισθητήρας Pulse Sensor

Ένας αξιόπιστος αισθητήρας για τη μέτρηση καρδιακών παλμών που ενσωματώνεται με τη πλατφόρμα Arduino είναι ο **Pulse Sensor Amped**



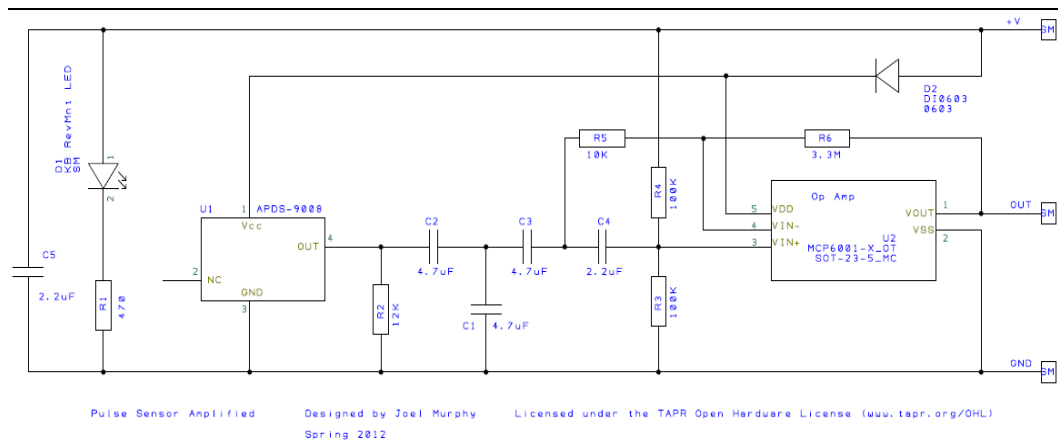
Εικόνα 23 Pulse Sensor Amped.

Ο Pulse Sensor είναι ένας καλά σχεδιασμένος αισθητήρας καρδιακού ρυθμού για Arduino. Μπορεί να χρησιμοποιηθεί πολύ εύκολα από μαθητές, καλλιτέχνες, αθλητές, κατασκευαστές, και προγραμματιστές εφαρμογών. Εύκολα μπορεί να ενσωματώνει τη μετάδοση πραγματικού καρδιακού σφυγμού στις εφαρμογές μας. Ο αισθητήρας μπορεί πολύ εύκολα να τοποθετηθεί στο δάχτυλο ή το λοβό του αυτιού. Η σύνδεσή του με το Arduino γίνεται με ακροδέκτες βραχυκύκλωσης. Διατίθεται και μια εφαρμογή open-source παρακολούθησης του σφυγμού σε πραγματικό χρόνο.



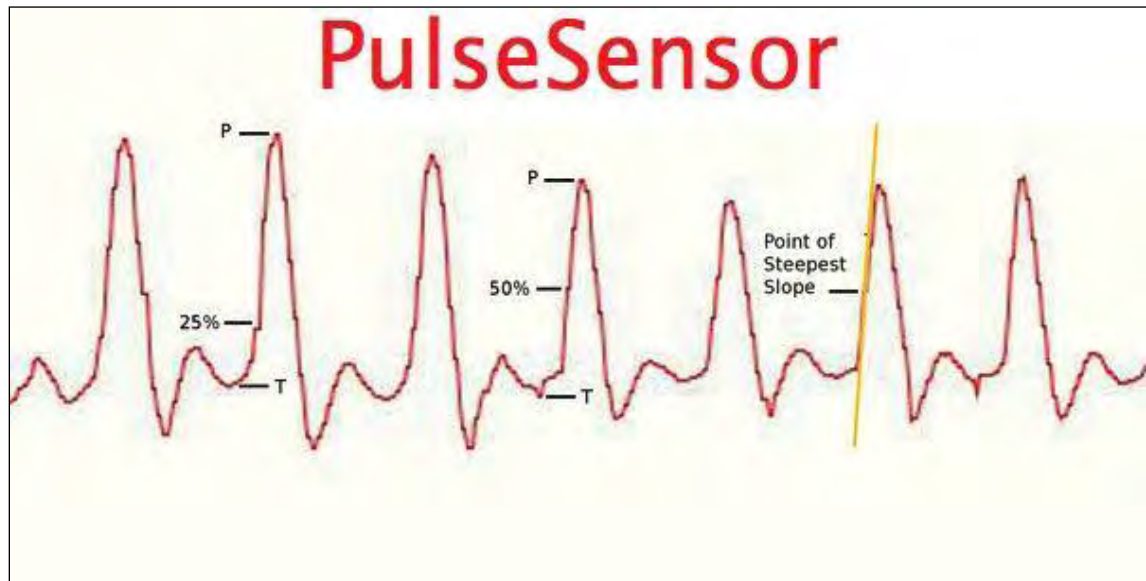
Εικόνα 24 Αισθητήρας Καρδιακού ρυθμού

Ο αισθητήρας παλμών που χρησιμοποιούμε είναι ουσιαστικά ένας φωτοπληθυσμογράφος, ο οποίος είναι μία πολύ γνωστή ιατρική συσκευή που χρησιμοποιείται για την μη επεμβατική παρακολούθηση του καρδιακού ρυθμού. Το σήμα παλμού της καρδιάς που έρχεται από ένα φωτοπληθυσμογράφο είναι μια αναλογική διακύμανση τάσης, και έχει ένα προβλέσιμο σχήμα κύματος. Η απεικόνιση του κύματος παλμού ονομάζεται photoplethysmogram, ή PPG. Έχει υλοποιηθεί με έναν αισθητήρα φωτός Avago (APDS 9008) και διαθέτει ένα Led σε πράσινο χρώμα της Εταιρίας Kingbright.



Εικόνα 25 Ηλεκτρονικό σχεδιάγραμμα του ΑισθητήραPulseSensor

Στη τελευταία έκδοση ο Pulse Sensor Amped, ενισχύει το σήμα του προηγούμενου αισθητήρα σφυγμού, και ομαλοποιεί το κύμα σφυγμού γύρω στα $V / 2$ (μεσαίο σημείο της τάσης).



Εικόνα 26 Καταγραφή Καρδιακού ρυθμού Processing

Ο Pulse Sensor Amped ανταποκρίνεται στις σχετικές αλλαγές της έντασης του φωτός. Εάν η ποσότητα του φωτός που προσπίπτει επί του αισθητήρα παραμένει σταθερή, η τιμή σήματος θα παραμείνει κοντά στο 512 (μέση στάθμη του ADC). Περισσότερο φως και το σήμα ανεβαίνει, λιγότερο φως, το αντίθετο. Φως από το πράσινο LED το οποίο συνοδεύει τον αισθητήρα ανακλάται πίσω προς αυτόν, ανάλογα με την ποσότητα του αίματος που υπάρχει στα αγγεία των ιστών κάτω από τον αισθητήρα. Η επεξεργασία του πληθυσμογραφικού αυτού σήματος μπορεί να οδηγήσει στον υπολογισμό του στιγμιαίου καρδιακού ρυθμού και στη μέτρηση του χρόνου μεταξύ των διαδοχικών παλμών, Inter Beat Interval (IBI).

Όταν οι αντλίες της καρδιάς αιματώνουν όλο το σώμα, με κάθε κτύπο υπάρχει ένα κύμα σφυγμού (κάτι σαν ένα κύμα σοκ) που ταξιδεύει κατά μήκος όλων των αρτηριών στα άκρα του πολύ τριχοειδή ιστού όπου εφαρμόζεται ο αισθητήρας παλμών. Στην πραγματικότητα το αίμα κυκλοφορεί στο σώμα πολύ πιο αργά από ό,τι το κύμα του

σφυγμού. Μία γρήγορη ανοδική αύξηση στην τιμή σήματος συμβαίνει όταν το κύμα παλμού διέρχεται κάτω από τον αισθητήρα, τότε το σήμα πέφτει κάτω από το κανονικό σημείο. Δεδομένου ότι το κύμα είναι επαναλαμβανόμενο, θα μπορούσαμε να επιλέξουμε πχ την κορυφή του σήματος, για τη μέτρηση του καρδιακού ρυθμού από το χρονικό διάστημα μεταξύ διαδοχικών κορυφών. Στην ιδανική περίπτωση, θέλουμε να βρούμε τη στιγμιαία τιμή του καρδιακού ρυθμού. Αυτό είναι σημαντικό για τον ακριβή υπολογισμό BPM, τη μεταβλητότητα του καρδιακού ρυθμού (HRV), και τη Pulse Transit Time (PTT) μέτρηση.

Είναι σημαντικό να έχουμε ένα κανονικό, αρκετά υψηλόρυθμό δειγματοληψίας. Για να επιτευχθεί αυτό, υπάρχουν 2 χρονιστές. Υπάρχει ένας 8 bit hardware χρονιστής για τον ATmega328, έτσι ώστε να διακόπτεται κάθε 2^ο χιλιοστό του δευτερολέπτου. Αυτό μας δίνει ένα ρυθμό δειγματοληψίας 500Hz, και ανάλυση χρονισμού των 2ms. Αυτό θα απενεργοποιήσει την εξόδου του PWM.

```
VoidinterruptSetup() {  
  TCCR2A = 0x02;  
  TCCR2B = 0x06;  
  OCR2A = 0x7C;  
  TIMSK2 = 0x02;  
  sei();  
}
```

Οι ρυθμίσεις παραπάνω προγραμματίζουν το δεύτερο χρονιστή να πάει σε κατάσταση CTC, και να μετράει έως $(124)_{10} = (7C)_{16}$ ξανά και ξανά. Ένας prescaler των 256 χρησιμοποιείται για να πάρει το χρονισμό σωστά, έτσι ώστε να χρειάζεται 2 χιλιοστά του δευτερολέπτου για να μετρήσει έως το 124. Μια σημαία διακοπής τίθεται κάθε φορά που φτάνει ο χρονιστής 2 το 124, καθώς μια ειδική λειτουργία που ονομάζεται μια ρουτίνα εξυπηρέτησης διακοπών (ISR) εκτελείται στην αμέσως επόμενη δυνατή στιγμή, αν και δεν έχει σημασία τι κάνει το υπόλοιπο πρόγραμμα. Με την εντολή sei() εξασφαλίζεται ότι έχουν ενεργοποιηθεί οι παγκόσμιες διακοπές. Έτσι, όταν το

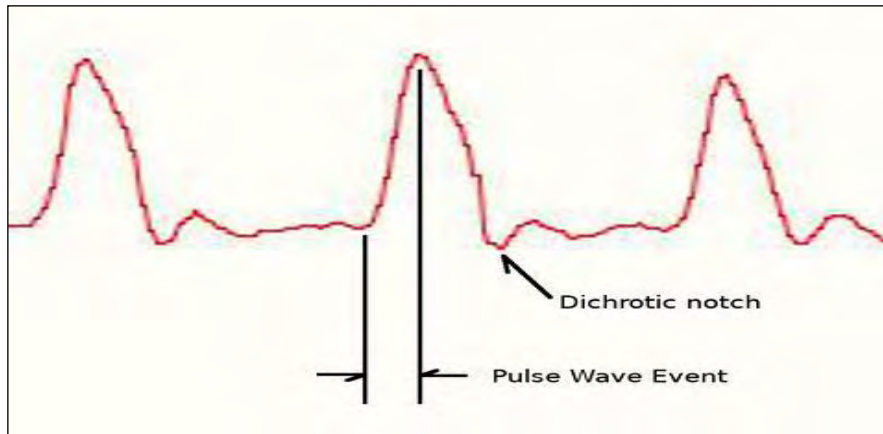
Arduino λειτουργεί με τον Pulse Sensor συνδεδεμένο στην αναλογική θύρα, συνεχώς (κάθε 2 mS) διαβάζει την τιμή του αισθητήρα και αναζητά την καρδιά να κτυπά.

```
ISR (TIMER2_COMPA_vect) {  
  Signal = analogRead(pulsePin);  
  sampleCounter += 2;  
  int N = sampleCounter - lastBeatTime;
```

Αυτή η λειτουργία καλείται κάθε 2 χιλιοστά του δευτερολέπτου. Το πρώτο πράγμα που πρέπει να κάνουμε είναι να πάρουμε ένα αναλογικό αναγνώστη του αισθητήρα παλμού. Στη συνέχεια, θα αυξήσουμε τη μεταβλητή sample Counter. που τηρεί τον χρόνο. Η μεταβλητή N θα βοηθήσει στην αποφυγή του θορύβου αργότερα. Στη συνέχεια, καταγράφουμε τις υψηλότερες και χαμηλότερες τιμές του κύματος PPG, για να πάρουμε μια ακριβή μέτρηση του πλάτους.

```
  if(Signal < thresh && N >  
    (IBI/5)*3){ if (Signal < T){ T  
    = Signal; }  
  }  
  if(Signal > thresh && Signal >  
    P){ P = Signal;  
  }
```

Οι μεταβλητές P και T κρατούν τις μέγιστες και ελάχιστες τιμές, αντίστοιχα. Η μεταβλητή thresh αρχικοποιείται σε 512 (μέση τιμή του αναλογικού φάσματος) και οι αλλαγές κατά τη διάρκεια της εκτέλεσης του χρόνου παρακολουθούνται γύρω από ένα σημείο κοντά στο 50% του πλάτους. Υπάρχει ένα χρονικό διάστημα των 3/5 του IBI που πρέπει να περαστεί πριν από την αναβάθμιση του T αυτός είναι ένας τρόπος για την αποφυγή του θορύβου και των εσφαλμένων ενδείξεων από την dichrotic notch (δίκροτη εντομή)



Εικόνα 27dichroticnotch (δίκροτη εντομή)

Τώρα, ελέγχουμε να δούμε αν έχουμε ένα παλμό.

```

if (N > 250){
if ((Signal>thresh) && (Pulse == false) && (N > ((IBI/5)*3) ){
Pulse = true;
digitalWrite(pulsePin,HIGH);
IBI = sampleCounter - lastBeatTime;
lastBeatTime = sampleCounter;
}}

```

Πριν την εξέταση ενός καρδιακού παλμού, πρέπει να περάσει ένα ελάχιστο χρονικό διάστημα. Αυτό βοηθά στην αποφυγή θορύβου στις υψηλές συχνότητες. Όταν η κυματομορφή ανεβαίνει πάνω από τη thresh μεταβλητή, και τα 3/5 του προηγούμενου IBI έχουν περάσει, τότε έχουμε ένα παλμό. Τώρα ορίζουμε τη Pulse flag και ενεργοποιούμε το pulsePin LED (led13). Στη συνέχεια, θα υπολογιστεί το χρονικό διάστημα από τον τελευταίο χτύπο όπου πήραμε IBI, και θα ενημερωθεί η lastBeatTime.

Το επόμενο bit χρησιμοποιείται για να βεβαιωθούμε ότι ξεκινάμε με μια πραγματική BPM τιμή κατά την εκκίνηση.

```

if(secondBeat){
secondBeat = false;
for(int i=0; i<=9; i++){
rate[i] = IBI;
}
}

```

```

    }
  }
  if(firstBeat){
    firstBeat = false;
    secondBeat = true;
    sei():
    return;
  }

```

Οι booleans Firstbeat και secondBeat αρχικοποιούν ως αληθινή την εκκίνηση, έτσι ώστε η πρώτη φορά που θα πάρουμε αυτό το σημείο του ISR, μας διώχνει η return;. Η BPM προκύπτει από το μέσο όρο των τελευταίων 10 τιμών του IBI.

```

word runningTotal = 0;
for(int i=0; i<=8; i++){
  rate[i] = rate[i+1];
  runningTotal += rate[i];
}
rate[9] = IBI;
runningTotal += rate[9];
runningTotal /= 10;
BPM = 60000/runningTotal;
QS = true;
}
}

```

Πρώτα, θα πάρουμε τη μεταβλητή runningTotal για να αθροίσει τα IBI, τότε τα περιεχόμενα του rate μετατοπίζονται προς τα πάνω και προστίθενται στη runningTotal. Το παλαιότερο IBI (11 χτύπους πριν) πέφτει από τη θέση 0, και το νεότερο IBI παίρνει τη θέση 9. Στη συνέχεια, υπολογίσουμε το BPM κατά το μέσο όρο του πίνακα.

```

if (Signal < thresh && Pulse == true){
  digitalWrite(13,LOW);
Pulse = false;
  amp = P - T;
  thresh = amp/2 + T;
  P = thresh;
  T = thresh;
}

```

Ο χτύπος ανακηρύχθηκε true κατά τη διάρκεια της ανοδικής αύξησης του σήματος του αισθητήρα παλμού όταν βρήκαμε το ρυθμό, πάνω, έτσι ώστε όταν το σήμα διασχίζει τη μεταβλητή thresh πηγαίνει κάτω, έτσι μπορούμε να καταλάβουμε ότι ο παλμός είναι πάνω. Τότε το πλάτος του κύματος που μόλις πέρασε μετράται, και ενημερώνεται η thresh με το νέο 50% σήμα. Ο αλγόριθμος είναι τώρα έτοιμος να βρούμε

το επόμενο beat. Υπάρχει ένα ακόμη ερώτημα που τίθεται πριν γίνει η ISR. Τι θα συμβεί αν δεν υπάρχουν beats;

```
if (N > 2500) {
    thresh = 512;
    P = 512;
    T = 512;
    firstBeat = true;
    secondBeat = false;
    lastBeatTime = sampleCounter;
}
```

Αν δεν υπάρχει παρουσία χτύπου για 2,5 δευτερόλεπτα, οι μεταβλητές που χρησιμοποιήθηκαν για να βρούμε το χτύπο της καρδιάς επαναρχικοποιούνται για τις μεταβλητές της εκκίνησης. Με τη χρήση της διακοπής του Χρονιστή2, ο αλγόριθμος τρέχει "στο παρασκήνιο" και ενημερώνει αυτόματα τις τιμές των μεταβλητών. Εδώ είναι μια λίστα χρήσιμων μεταβλητών, και κάθε πόσο συχνά ενημερώνονται.

Πίνακας 4 Ενημέρωση μεταβλητών (<https://pulsesensor.com>)

| Όνομα μεταβλητής | Χρόνος Ανανέωσης | Λειτουργία |
|------------------|-------------------|------------------------------|
| Signal | 2 msec | Πριονωτό σήμα του Αισθητήρα |
| IBI | κάθε χτύπο | Χρόνος μεταξύ χτύπων σε msec |
| BPM | κάθε χτύπο | Χτύποι ανά λεπτό |
| QS | Αληθές κάθε χτύπο | Διαγραφή από το χρήστη |
| Pulse | Αληθές κάθε χτύπο | Διαγραφή από την ISR |

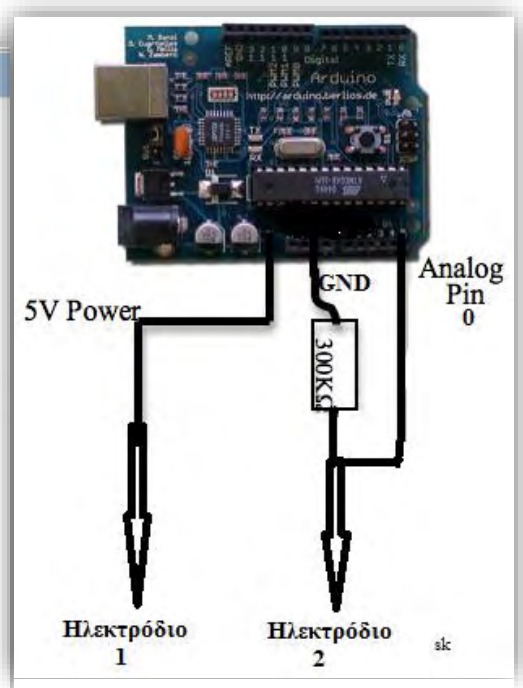
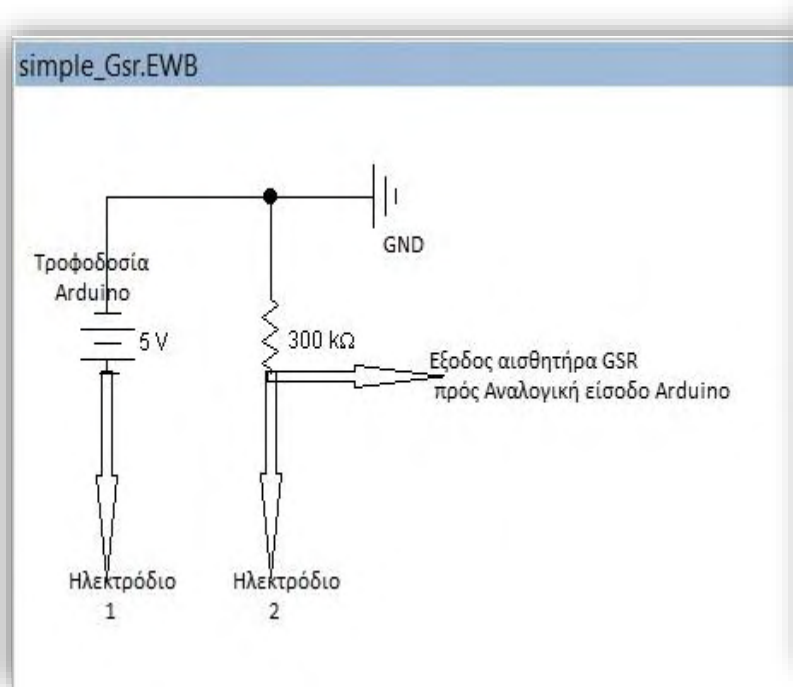
4.4 Ο αισθητήρας GSR

Για την υλοποίηση της μέτρησης της εφίδρωσης του δέρματος, που παρουσιάζεται σε πολλά άτομα λόγω αύξησης της ανησυχίας, η οποία θα μπορούσε να χαρακτηριστεί σαν αγχώδης κατάσταση, χρησιμοποιήθηκε η μέθοδος της γαλβανικής ηλεκτροδερμικής απόκρισης (galvanic skin response) (GSR). Σύμφωνα λοιπόν με την τεχνική αυτή ανιχνεύεται η αλλαγή στην αγωγιμότητα του ανθρώπινου δέρματος κατά την διέλευση μέσα από αυτό ηλεκτρικό ρεύμα σταθερής και φυσικά πολύ χαμηλής τάσης. Όπως έχει αναφερθεί και σε προηγούμενο κεφάλαιο η (GSR) ορίζεται σαν μια

αλλαγή στις ηλεκτρικές ιδιότητες του δέρματος. Η αντίσταση του δέρματος είναι συνήθως μεγάλη, από 1 - 10ΜΩ, γνωρίζουμε όμως πως οι αλλαγές των ιδρωτοποιών αδένων μπορούν να επηρεάσουν την αντίστασή του δέρματός μας έως και 1ΜΩ. Η μικρή ηλεκτρική τάση που εφαρμόζει ο αισθητήρας μας με ηλεκτρόδια στα δύο δάχτυλα του ενός χεριού, δημιουργεί ένα ηλεκτρικό κύκλωμα και το σώμα μας μετατρέπεται σε μια μεταβλητή αντιστάτη. Από τα στοιχεία αυτά του κυκλώματος υπολογίζουμε τη μεταβολή της αντίστασης R_{GSR} σε Ohm ή της αγωγιμότητας ($G_{GSR} = 1/R_{GSR}$) σε Siemens

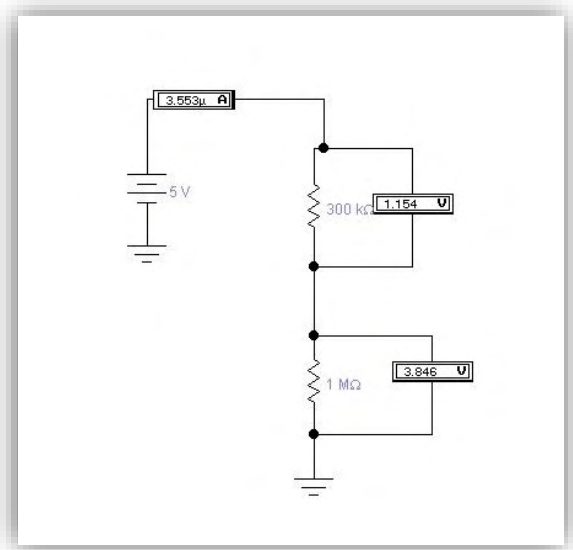
Η GSR μας δείχνει κάθε φορά τη δραστηριότητα των ιδρωτοποιών αδένων και αλλαγές στη συμπαθητική δραστηριότητα του (ΑΝΣ). Είναι γνωστό από πολλές εργασίες και μελέτες η σχέση που υπάρχει ανάμεσα στη συμπαθητική δραστηριότητα του (ΑΝΣ) αυτόνομου νευρικού συστήματος και στη συναισθηματική διέγερση, αν και υπάρχουν δυσκολίες στην αναγνώριση των συναισθημάτων που εκδηλώνουμε. Άρα αλλαγές στη συναισθηματική κατάσταση του κάθε υποκείμενου μπορούν να καταγράφονται μέσω ενός συστήματος, όπου το σήμα ουσιαστικά μετρά τις μεταβολές στην αγωγιμότητα. Για τον έλεγχο της Αντίστασης – Αγωγιμότητας το Arduino είναι προγραμματισμένο να λαμβάνει τιμές κάθε 50msec. Στη πραγματικότητα έχουμε ένα δείγμα 20 μετρήσεων ανα sec. Για τη κατασκευή του αισθητήρα μέτρησης της εφίδρωσης χρησιμοποιήσαμε μια διάταξη όπως η παρακάτω. Μία εξωτερική αντίσταση $R=300K\Omega$ η τάση των 5V από το Arduino και τρεις αγωγοί βραχυκύκλωσης μπορούν να υλοποιήσουν το πιο φθινό και εύκολο κύκλωμα για να πάρουμε αξιόπιστα τις μετρήσεις μας.

Στις παρακάτω εικόνες μπορούμε να διακρίνουμε το Ηλεκτρονικό σχεδιάγραμμα του απλού αισθητήρα GSR(εικόνα 29), τη συνδεσμολογία του με το Arduino (εικόνα 28) και τη μοντελοποίηση του αισθητήρα σαν ένα ηλεκτρικό κύκλωμα προσομοιώνοντας την αντίσταση του δέρματος με πραγματική αντίσταση. Στις δύο διαφορετικές εικόνες 30 και 31 με τη μοντελοποίηση που υπάρχουν προσομοιώνουμε το δέρμα με και χωρίς εφίδρωση αλλάζοντας τις αντίστοιχες αντιστάσεις στο διαιρέτη τάσης. Στις εικόνες 32 - 35 μπορούμε να διακρίνουμε την ωμομέτρηση των δακτύλων μας και τα αποτελέσματα από το πολύμετρό μας.

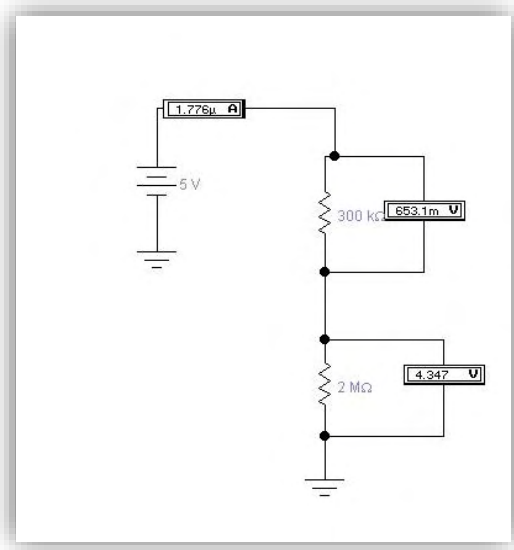


Εικόνα 28 Συνδεσμολογία απλού αισθητήρα GSR

Εικόνα 29 Ηλεκτρονικό Σχεδιάγραμμα απλού αισθητήρα GSR



Εικόνα 30 Μοντελοποίηση αισθητήρα χωρίς εφίδρωση



Εικόνα 31 Μοντελοποίηση αισθητήρα με εφίδρωση



Εικόνα 32 Ωμομέτρηση για την ανάγκη μοντελοποίησης



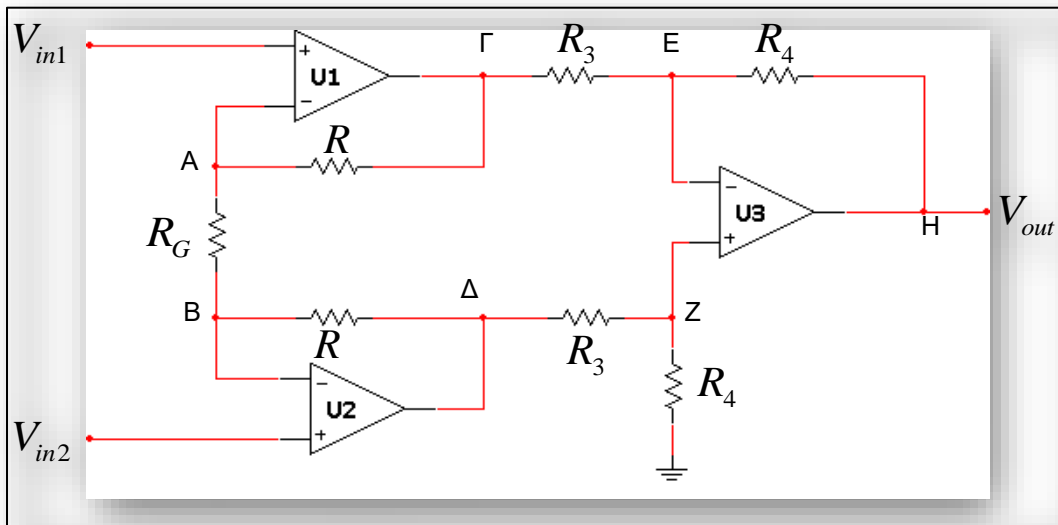
Εικόνα 34 Ωμομέτρηση χωρίς εφίδρωση



Εικόνα 33 Ωμομέτρηση με εφίδρωση

Ένας εναλλακτικός τρόπος κατασκευής αισθητήρα GSR είναι μέσω ενός ενισχυτή οργανολογίας.

Ο ενισχυτής οργανολογίας (instrumentation amplifier) είναι ένα κύκλωμα δύο εισόδων και μίας εξόδου. Η λειτουργία του είναι να ενισχύει τη διαφορά των σημάτων στις δύο εισόδους του και να απορρίπτει κοινά σήματα. Οι ενισχυτές οργανολογίας χρησιμοποιούνται σε κυκλώματα βιοενισχυτών π.χ ηλεκτροκαρδιογράφους , ηλεκτροεγκεφαλογράφους , ηλεκτρομυογράφους



Εικόνα 35 Δομή ενισχυτή οργανολογίας

Στη γενική δομή ενός ενισχυτή οργανολογίας φαίνεται στη παραπάνω εικόνα οι τελεστικοί ενισχυτές U1 και U2 δίνουν μεγάλη αντίσταση εισόδου στο κύκλωμα, ενώ ο τελεστικός ενισχυτής U3 μαζί με τις αντιστάσεις R_3 και R_4 σχηματίζουν έναν ενισχυτή διαφορών για την απόρριψη του κοινού σήματος.

Αποδεικνύεται ότι η τάση εξόδου δίνεται από την ακόλουθη σχέση:

$$V_{out} = \frac{R_4}{R_3} \left(1 + \frac{2R}{R_G} \right) (V_{in2} - V_{in1}) \quad (1)$$

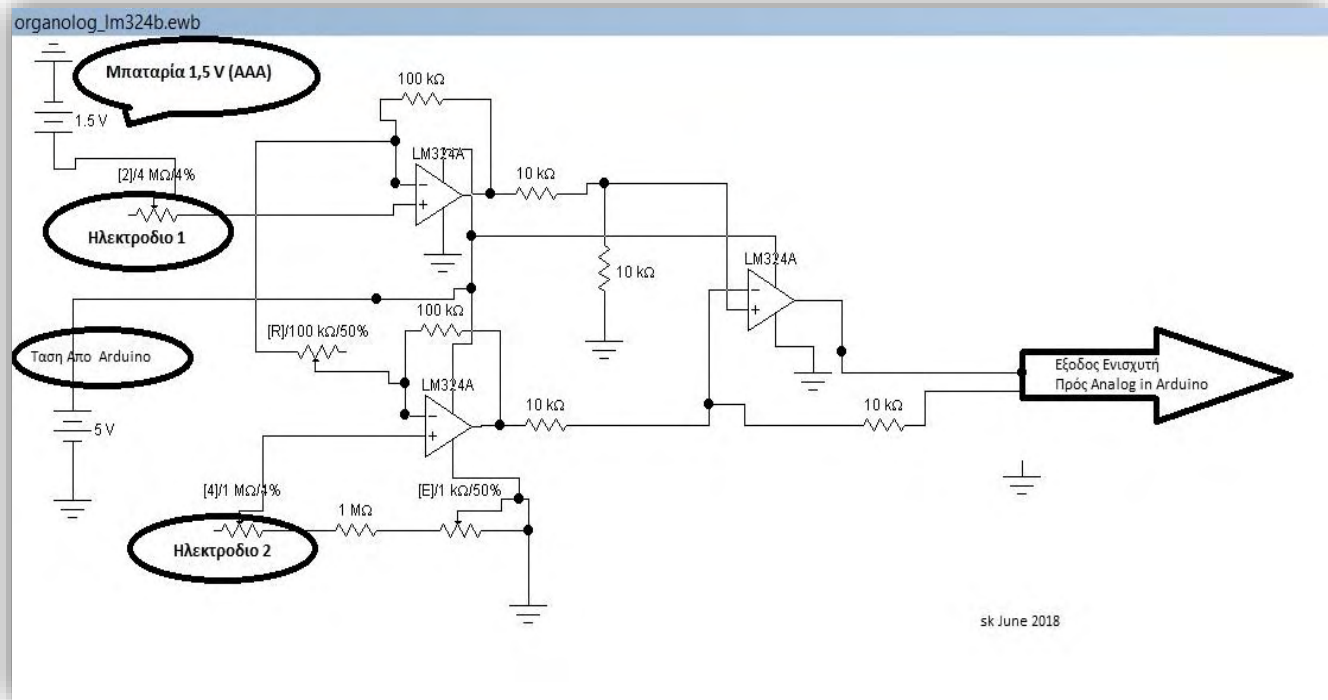
Συνεπώς στο συγκεκριμένο κύκλωμα, το διαφορικό κέρδος A_d , το οποίο ορίζεται ως ο λόγος της τάσης εξόδου V_{out} προς τη διαφορά $V_2 - V_1$ των δύο τάσεων εισόδου, δίνεται από τη σχέση:

$$A_d = \frac{R_4}{R_3} \left(1 + \frac{2R}{R_G} \right) \quad (2)$$

Συμπεραίνουμε πολύ εύκολα από τα προηγούμενα ότι από τη συνδεσμολογία του ενισχυτή οργανολογίας, προκύπτει ότι το διαφορικό κέρδος μπορεί να τροποποιηθεί μεταβάλλοντας μόνο την αντίσταση R_G . Οποιοσδήποτε άλλος συνδυασμός για μεταβολή του κέρδους θα απαιτούσε την ταυτόχρονη μεταβολή περισσότερων αντιστάσεων.

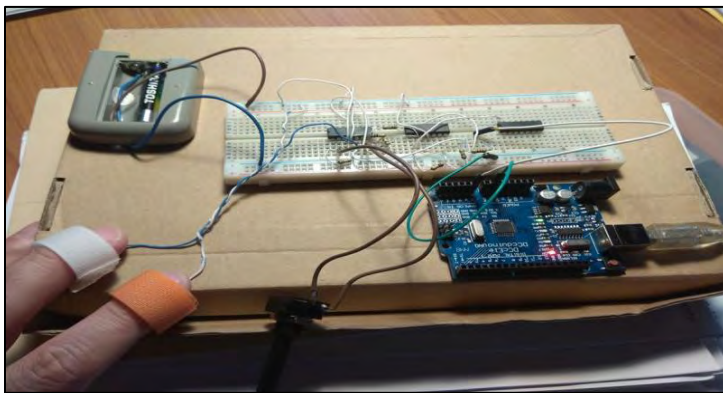
Αν στις δύο εισόδους του ενισχυτή οργανολογίας εφαρμοστεί ένα κοινό σήμα, δηλαδή $V_{in1} = V_{in2}$, τότε από την σχέση (1) προκύπτει ότι $V_{out} = 0$, δηλαδή ο ενισχυτής οργανολογίας απορρίπτει το κοινό σήμα. Επομένως, στην περίπτωση αυτή το κέρδος κοινού σήματος, είναι μηδέν και κατ' επέκταση ο λόγος απόρριψης κοινού σήματος, $CMMR$, είναι άπειρος. Στην πραγματικότητα, οι αντιστάσεις παρουσιάζουν ανοχές και το κέρδος κοινού σήματος είναι μη μηδενικό.

Εδώ μπορούμε να δούμε τη κατασκευή του ενισχυτή μας σε κατασκευή με το EWB. Για τις ανάγκες της προσομοίωσης ή αντίσταση του Δέρματος στα ηλεκτρόδια αντικαταστάθηκε με μεταβλητές αντιστάσεις (Ηλεκτρόδιο 1 και 2) Για τη κατασκευή του χρειάστηκα 3 Τ.Ε (τελεστικούς ενισχυτές) όπως φαίνεται στη εικόνα 36.



Εικόνα 36 Κατασκευή Ενισχυτή στο EWB

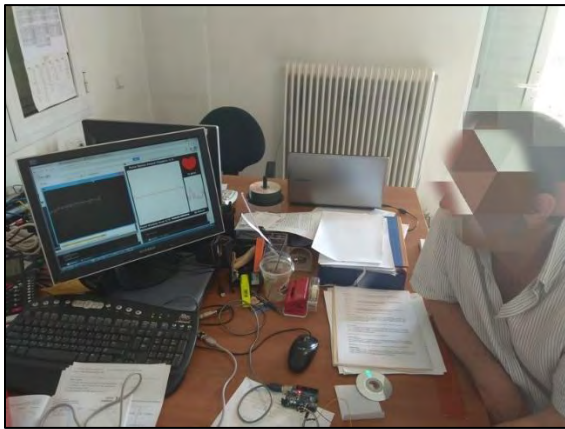
Το (LM324N) έχει 4 τελεστικούς ενισχυτές (Op Amps) .Τα "Vin1" και "Vin2" τροφοδοτούνται από τα ηλεκτρόδια που οδηγούνται στα δάκτυλα μας , το ηλεκτρόδιο 1 τροφοδοτείται με 1.5V (από μια εξωτερική μπαταρία) και το άλλο ηλεκτρόδιο έχει μια αντίσταση 1M και ένα ποτενσιόμετρο 1M που γειώνεται. Οι τελεστικοί ενισχυτές τροφοδοτούνται με 5V μέσω της τάσης που μας παρέχει το Arduino .



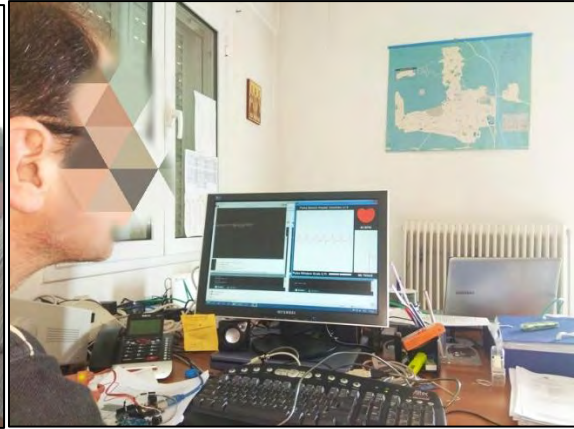
Εικόνα 37Υλοποίηση του ενισχυτή οργανολογίας σε Breadboard

4.5 Η Προσαρμογή των βιοαισθητήρων στο χέρι μας

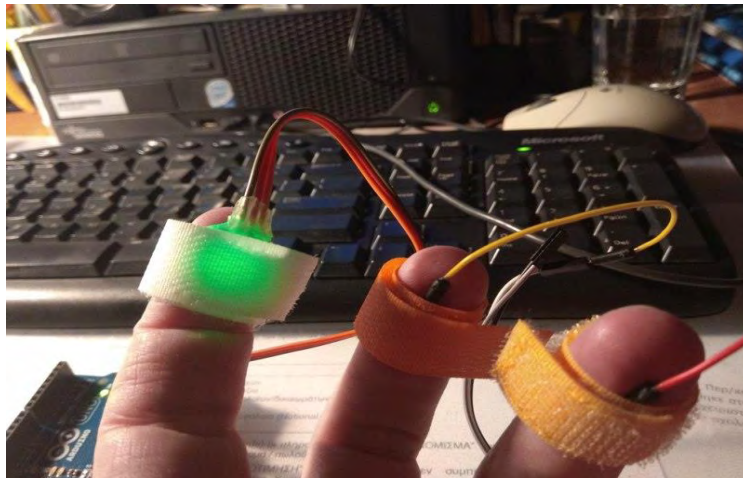
Στις παρακάτω φωτογραφίες μπορούμε να διακρίνουμε δύο άτομα κατά τη διάρκεια μετρήσεων της συσκευής μας. Στην εικόνα 38 το υποκείμενο υπόκειται σε μέτρηση από άλλο χρήστη που χειρίζεται τη συσκευή, ενώ στη διπλανή εικόνα 39 διακρίνουμε έναν από τα υποκείμενα που αναλαμβάνει μόνος να διαχειριστεί τη συσκευή καταγραφής. Στη εικόνα 40 αποτυπώνεται η προσαρμογή των αισθητήρων στα δάχτυλα του αριστερού χεριού.



Εικόνα 38 Μέτρηση υποκειμένου 1



39 Μέτρηση υποκειμένου 2

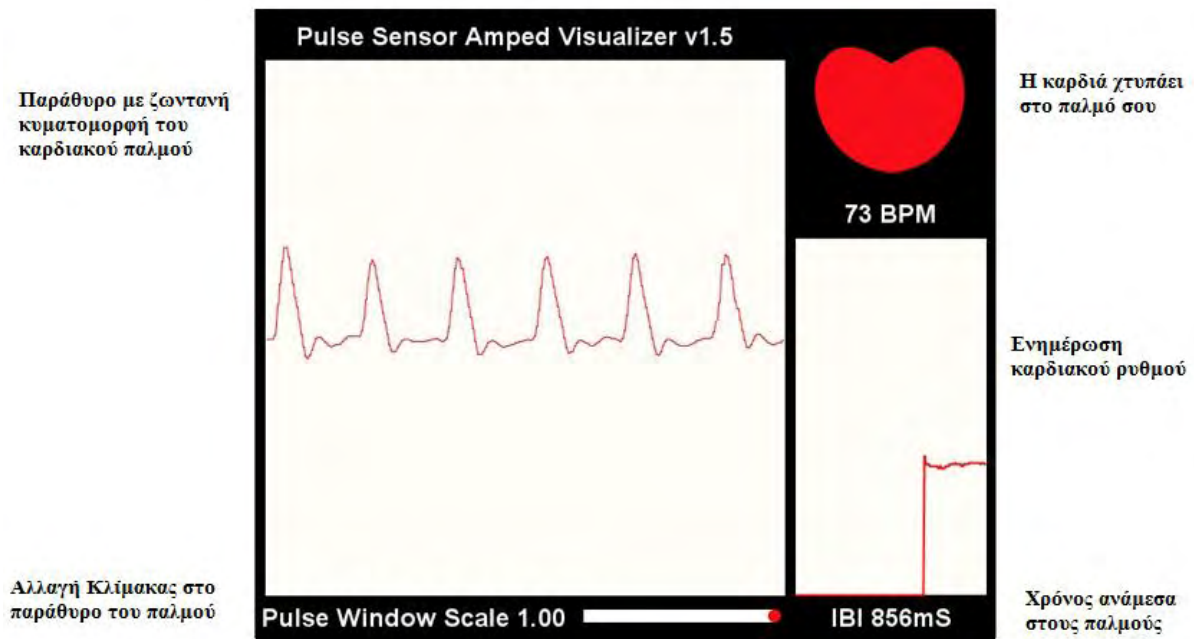


Εικόνα 40 Προσαρμογή αισθητήρων στο χέρι

4.6 Γραφική αναπαράσταση βιοσημάτων

Ο αισθητήρας Pulse sensor μπορεί να συνεργαστεί χωρίς προβλήματα με την εφαρμογή Pulse Sensor Amped Visualizer v1.5 καταγράφοντας όπως φαίνεται στη εικόνα 41 σε ένα μέγεθος παραθύρου (700×600). Στο μεγαλύτερο μέρος του παραθύρου μπορούμε να

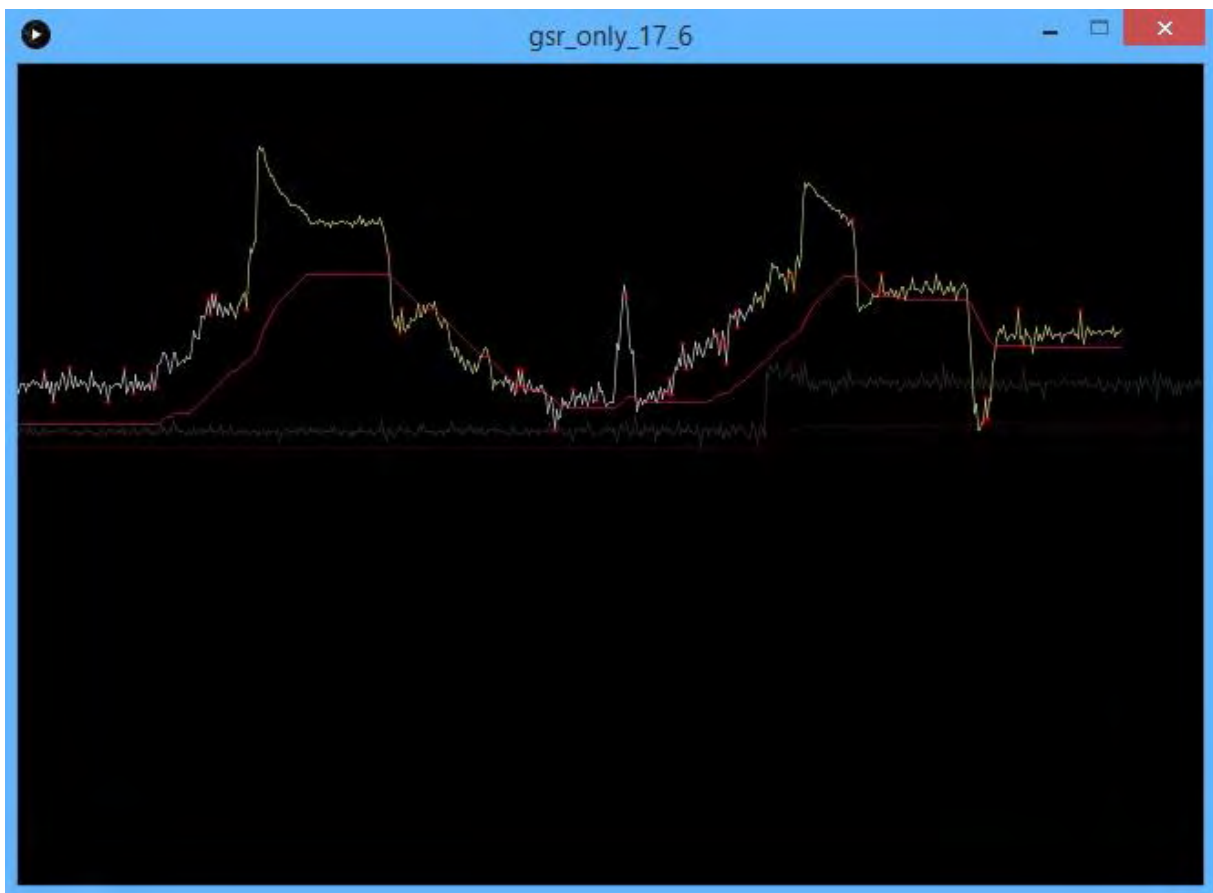
παρακολουθούμε σε πραγματικό χρόνο τη κυματομορφή του καρδιακού παλμού. Δεξιά πάνω είναι σχεδιασμένη μια καρδιά που αυξομειώνει το μέγεθος της ανάλογα με το παλμό ανά λεπτό (BPM) δίνοντας την αίσθηση μιας καρδιάς που πάλλεται στο καρδιακό ρυθμό του υποκειμένου που υπόκειται σε μέτρηση. Κάτω από τη καρδιά ένα μικρό παράθυρο που μας ενημερώνει σχεδιαστικά για το καρδιακό ρυθμό δίνοντάς μας και το χρόνο σε msec που μεσολαβεί ανάμεσα σε κάθε παλμό (IBI Inter Beat Interval). Πρόσθετη δυνατότητα για αλλαγή της κλίμακας στο παράθυρο του παλμού μας δίνει το ScaleBar που μπορούμε να το προσαρμόζουμε κατά το δοκούν.



Εικόνα 41 Pulse Sensor Amped Visualizer

Όπως φαίνεται και στη παραπάνω εικόνα, η καταγραφή του Gsr Processing δημιουργεί μέγεθος παραθύρου (650×450) και εκεί αποτυπώνει δύο γραμμές (κυματομορφές).

Ο σχεδιασμός της πρώτης γραμμής καταγράφει τα σημεία της τρέχουσας και της προηγούμενης τιμής του gsrValue (CurrentReading και lastReading) με διαφορετικά χρώματα ανάλογα με τι τιμή του gsrAverage όπως παρακάτω.



Εικόνα 42 Καταγραφή GSR

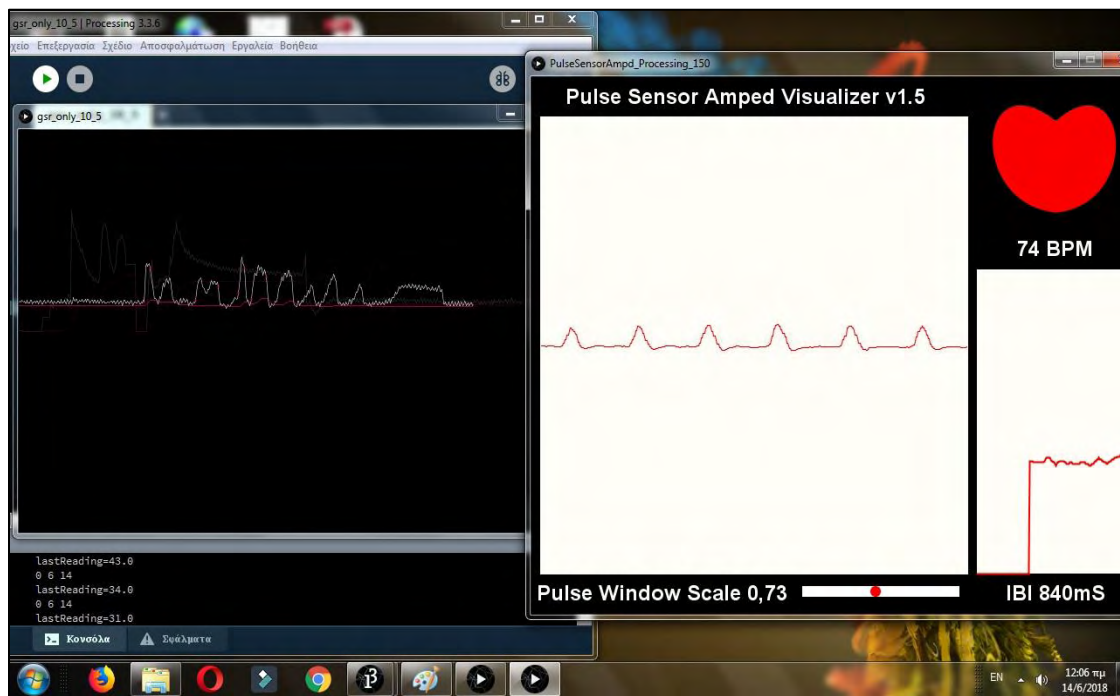
- Λευκή: $0 < \text{gsrAverage} < (\text{height}/2) \times 0,25$
- Κίτρινη: $0,25 < \text{gsrAverage} < (\text{height}/2) \times 0,5$
- Πράσινη: $0,5 < \text{gsrAverage} < (\text{height}/2) \times 0,75$
- Κόκκινη: $\text{gsrAverage} > (\text{height}/2) \times 0,75$

Ο σχεδιασμός της δεύτερης γραμμής αποτυπώνει σε κόκκινη απόχρωση στο παράθυρό μας τη τιμή του `gsrAverage` υπολογίζοντας το 3% κάθε νέας μέτρησης του αισθητήρα και προσθέτοντας το 97% της προηγούμενης `gsrAverage`, δίνοντας μια άλλη οπτική σε τιμές που έχουν έντονη διακύμανση όπως προηγουμένως. Γι αυτό και η γραμμή της `gsrAverage` παρακολουθεί τις τρέχουσες τιμές του αισθητήρα με ομαλοποιημένη σχεδίαση.

```
if(gsrRange>200){
    fill(255);
    ellipse(10,10,20,20)
```

Αριστερά πάνω στο παράθυρό μας εκτυπώνεται μια έλλειψη σε λευκό χρώμα μικρών διαστάσεων ((10,10,20,20) όταν το gsrRange ξεπεράσει τα όρια που του έχουμε θέσει. Επίσης κάθε φορά που υπάρχει μια μέτρηση, το processing δημιουργεί αρχείο τύπου txt με τις τιμές που επιθυμούμε (CurrentReading ή gsrAverage) και τη συνοδεία του χρόνου απο ένα ρολόι που τρέχει πάντα σαν διαδικασία στην εφαρμογή για την αποθήκευση των τιμών συνάρτησή του χρόνου.

Ταυτόχρονη απεικόνιση των αισθητήρων



Εικόνα 43 Ταυτόχρονη καταγραφή αισθητήρων

Το κύκλωμά μας δεν απαιτεί καμία πρόσθετη εξωτερική τροφοδοσία, όλες οι τροφοδοσίες οι οποίες απαιτούνται παρέχονται μέσα από τη USB σύνδεση των Arduino με τον Η/Υ που έχουμε στη διάθεσή μας. Τα 5 Volt είναι ικανά να υποστηρίξουν με άνεση τα δύο Arduino και τους αισθητήρες τους για τη καταγραφή των βιομετρήσεων.

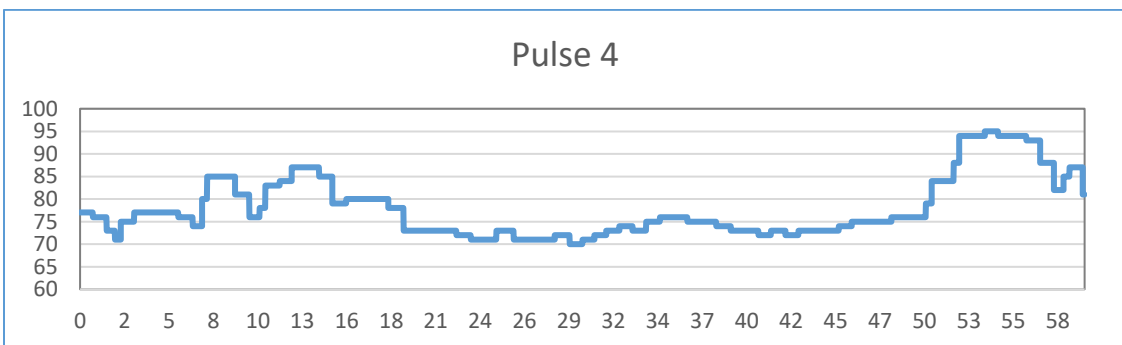
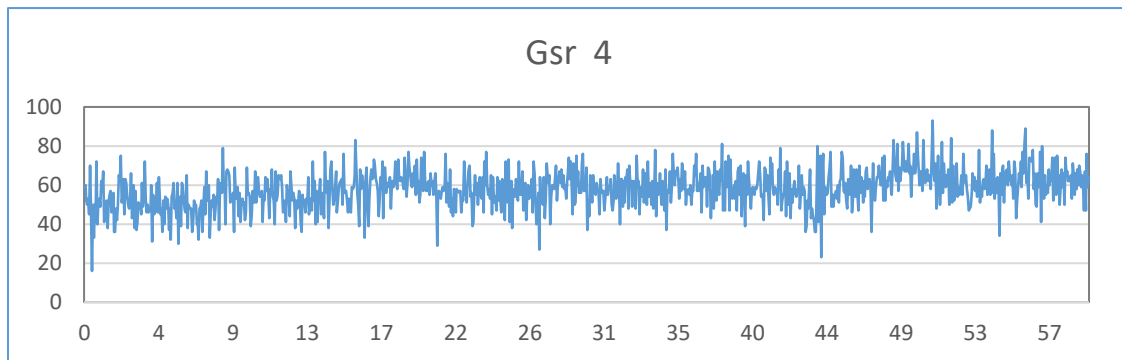
Η εφαρμογή των αισθητήρων στο χέρι μας γίνεται με ταινίες τύπου Velco. Ο αισθητήρας Pulse sensor συνδέεται στον δείκτη του αριστερού χεριού σε όλες τις δοκιμαστικές μετρήσεις των διαφορετικών υποκειμένων. Στα δύο επόμενα δάκτυλα μεσαίος και παράμεσος συνδέεται με αγωγούς βραχυκύκλωσης και ταινίες τύπου velco αισθητήρας της γαλβανικής απόκρισης. Τα καλώδια και των δύο αισθητηρίων είναι μικρά σε μήκος, για να μην προσθέτουν πρόσθετη αντίσταση, αλλά ικανά για να προσφέρουν μια άνετη μέτρηση για κάθε υποκείμενο. Ο κάθε χρήστης προκειμένου να υποβληθεί στη συγκεκριμένη διαδικασία κάθεται σε μια καρέκλα κοντά σε γραφείο και ακουμπά το χέρι του πάνω σε αυτό, εξασφαλίζοντας έτσι σταθερή βάση και κοντινή

πρόσβαση με τον Η/Υ που είναι πάνω στο ίδιο γραφείο. Κατά τη διάρκεια εκτέλεσης της διαδικασίας θα πρέπει ο χρήστης να είναι ακίνητος και να αποφεύγονται οι συνομιλίες με το χειριστή της συσκευής. Υπάρχουν περιπτώσεις που υπερβολική ξηρότητα ή λιπαρότητα του δέρματος μπορεί να οδηγήσει σε εσφαλμένες μετρήσεις.

Και οι δύο αισθητήρες θα πρέπει να είναι προσαρμοσμένοι σωστά στο σώμα μας. Χαλαρές συνδέσεις των αισθητήρων ή και υπερβολικά σφιχτές οι ταινίες γύρω από τα δάκτυλά μας οδηγούν σε πρόσθετο Θόρυβο μειωμένη ακρίβεια και τελικά σε εισαγωγή σφαλμάτων στις μετρήσεις.

Η συσκευή μας συνδέεται μέσω USB στο διαθέσιμο Η/Υ που είναι διαθέσιμος. Η εκκίνηση των εφαρμογών θα πρέπει να είναι ταυτόχρονη έτσι ώστε να υπάρξει χρονική σύγκριση αν ζητηθεί .Οι Processing εφαρμογές (όπως είδαμε και προηγουμένως) μπορούν να αποθηκεύσουν τις μετρήσεις σε txt Format (Pulse.txt και Gsr.txt) . Κάθε δοκιμασία θα πρέπει να έχει διάρκεια τουλάχιστον (60 sec)

Στα παρακάτω τυχαία διαγράμματα μπορούμε να εξάγουμε πληροφορίες για τη κατάσταση των υποκειμένων που συνεργάστηκαν για τις ανάγκες της εργασίας μας.



Σε κάθε διάγραμμα παρουσιάζονται οι αποθηκευμένες τιμές από την processing για τον αισθητήρα GSR και τον PulseSensor. Από το πρώτο αισθητήρα έχω ένα δείγμα $20(\text{δείγματα}/\text{sec}) \times 60 \text{ sec} = 1200$ τιμές ανά λεπτό, ενώ από το δεύτερο αισθητήρα έχω περίπου $100(\text{δείγματα} / \text{sec}) \times 60 \text{ sec} = 6000$ τιμές ανά λεπτό.

Για το Διάγραμμα του Gsr ο οριζόντιος άξονας ορίζει το χρόνο και ο κατακόρυφος τη διαφορετική τιμή της εφίδρωσης με περιοχή τιμής από το 20 έως και το 200. Το διάγραμμα προσαρμόζει αυτόματα τα όρια στη περιοχή γραφήματος ανάλογα με τις τιμές που έχει δεχθεί από την είσοδό του, δίνοντας έτσι μια πιο ευδιάκριτη εικόνα και ταυτόχρονα μια πιο άνετη ανάγνωσή του. Το ίδιο ισχύει και για το γράφημα που δημιουργεί το αρχείο των παλμών, με τον οριζόντιο άξονα να υποδεικνύει το χρόνο (60 sec) και στο κατακόρυφο να φιλοξενούνται οι μέγιστες και οι ελάχιστες τιμές BPM.

Για αυτό το λόγο έχουμε αποτυπώσει και διαφορετικά γραφήματα GSR και PULSE.

4.7 Προβλήματα που παρουσιάστηκαν

Τα πιο σημαντικά προβλήματα που παρουσιάστηκαν κατά τη διάρκεια των μετρήσεων

1. Πρόχειρες κατασκευές των ηλεκτροδίων GSR και PulseSensor όσο αφορά τη στήριξή τους στο χέρι. Βελτίωση μπορεί να επιφέρει ειδικό gel για καλύτερη προσαρμογή και εφαρμογή.
2. Θόρυβος στη κατασκευή του ενισχυτή οργανολογίας. Βελτίωση με πιθανή αντικατάστασή ολοκληρωμένων κυκλωμάτων από το TE AD-620
3. Χρειάζεται καλύτερη προστασία των κυκλωμάτων και των συνδέσεων των αισθητήρων. Βελτίωση θα επιφέρει η τοποθέτηση της συσκευής μας σε κουτί ηλεκτρονικών κατασκευών.
4. Λάθος τοποθέτηση των αισθητήρων από άτομα που δεν γνώριζαν τον τρόπο εφαρμογής τους. Αποφυγή του προβλήματος με τη συγγραφή ενός πίνακα με σχέδια και υποδείξεις χρήσης για αυτούς που θέλουν μόνοι τους να χειρίζονται τη συσκευή.

4.8 Συμπεράσματα

Στη παρούσα διπλωματική μας εργασία είχαμε ως αντικείμενο τη κατασκευή μιας ηλεκτρονικής διάταξης που επεξεργάζεται βιοσήματα προσπαθώντας να ανιχνεύσει το άγχος των Μαθητών – Φοιτητών και βασίζεται στη γνωστή πλατφόρμα Arduino μετρώντας γαλβανική απόκριση δέρματος και καρδιακούς παλμούς. Στη πορεία της εργασίας μας αναφερθήκαμε στο συναισθηματικό και το ιατρικό κομμάτι που αφορούσε τις μετρήσεις μας και πώς αυτές επηρεάζονται από εξωγενείς παράγοντες. Κατά τη διάρκεια της εργασίας μας και μέσα από μετρήσεις που κάναμε σε εθελοντές καταλήξαμε στα παρακάτω.

Η ταυτόχρονη μέτρηση του GSR και των καρδιακών παλμών μπορεί να αποτελέσει καλό και εύκολο μηχανισμό ανίχνευσης του stress. Οι μετρήσεις είναι εφικτές και μπορούν να δώσουν οπτικά αποτελέσματα για τη συναισθηματική κατάσταση του υποκειμένου. Η αξιολόγηση των μετρήσεων μπορεί να χρησιμοποιηθεί για τη διάγνωση έντονων στρεσογόνων καταστάσεων που βιώνουν μαθητές ή και φοιτητές κατά τη διάρκεια μίας εξέτασης, ενός test ή της προετοιμασίας για αυτά. Η μικρή τάση που εφαρμόζεται στο σώμα μας δε δημιούργησε καμία ανησυχία σε κανένα από τους εθελοντές. Οι μετρήσεις που παραθέτουμε σε διαγράμματα στο Παράρτημα για τους 8 διαφορετικούς εθελοντές φαίνεται να εκπληρώνουν τους επιθυμητούς στόχους.

4.9 Μελλοντικοί στόχοι

Μελλοντικούς στόχους αποτελούν :

1. Πρόσθετος αισθητήρας Θερμοκρασίας
2. Ενσωμάτωση όλων των αισθητήρων σε ένα arduino
3. Ασύρματη μετάδοση δεδομένων

4. Πλήρης αυτονομία με καταγραφή σε SD card και απεικόνιση σε Crystal Οθόνη
5. Καλύτεροι ακροδέκτες , με σωστότερη προσαρμογή στα σημεία εφαρμογής των δακτύλων για ελαχιστοποίηση σφαλμάτων.

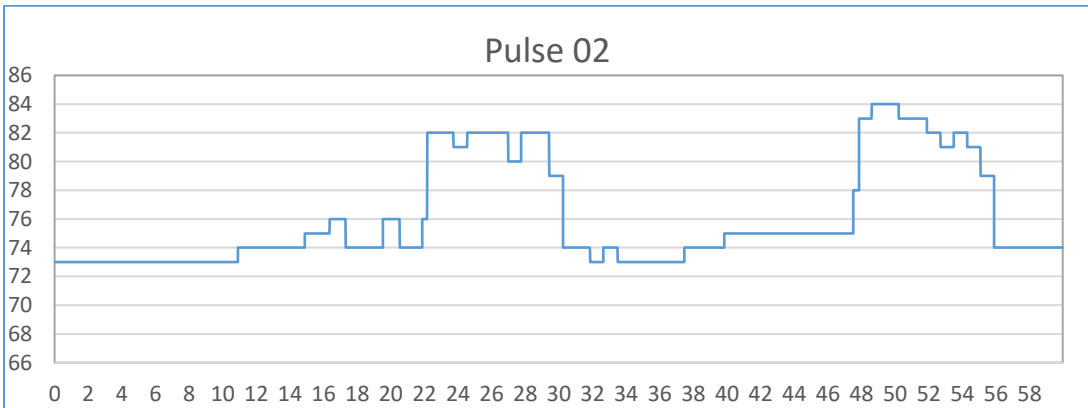
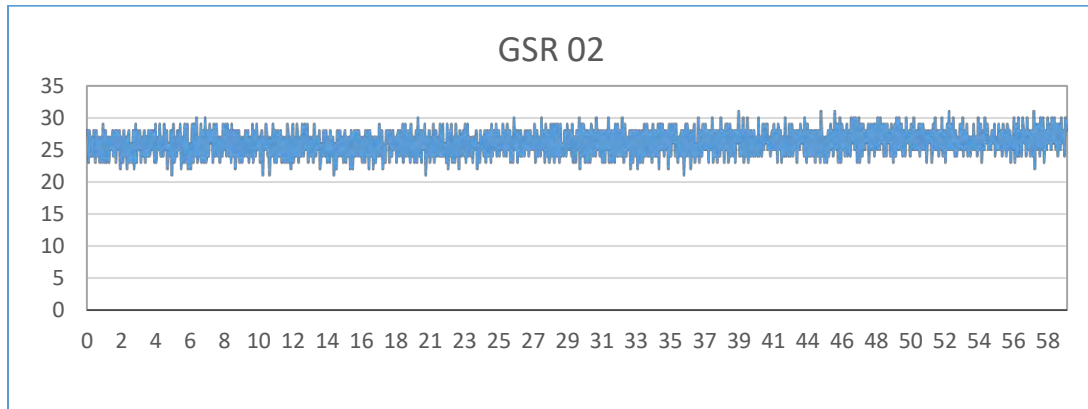
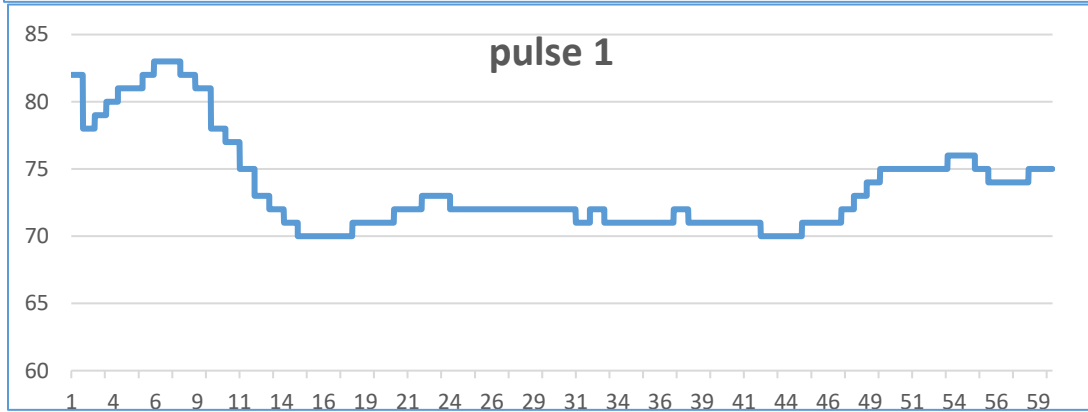
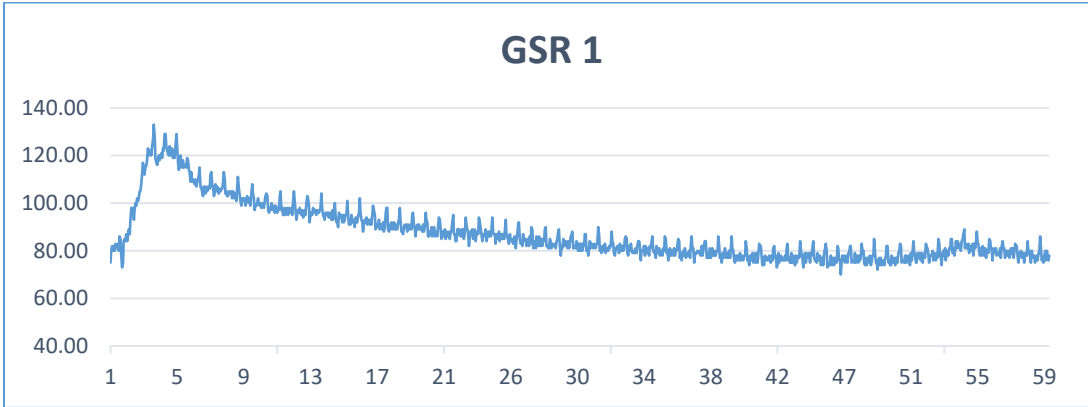
Βιβλιογραφία

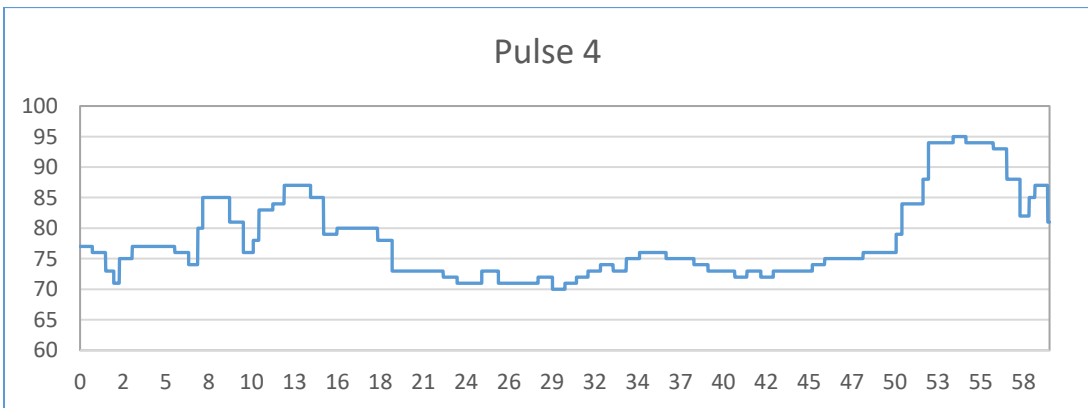
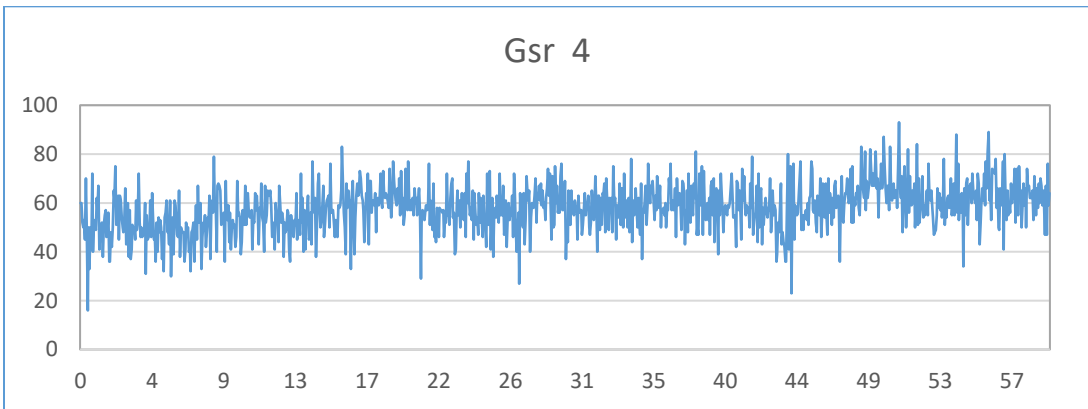
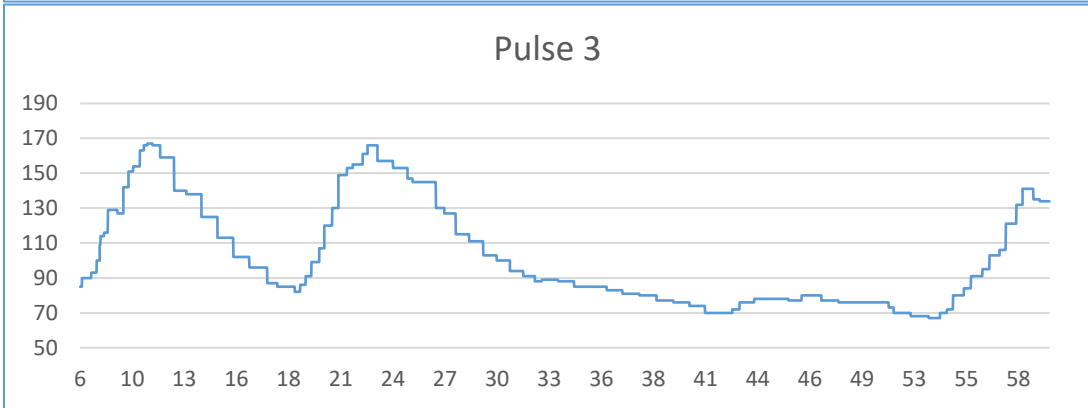
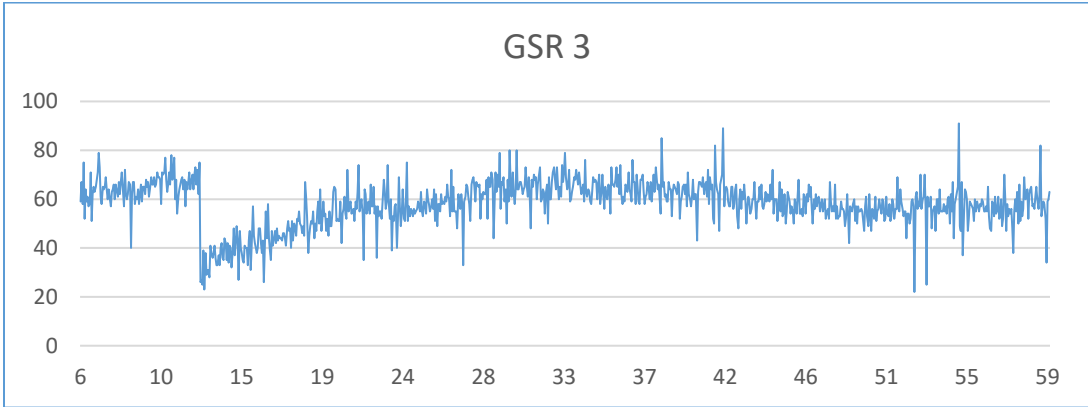
- Barbalet, J. (1998). *Emotion, Social Theory and Social Structure*. Cambridge, Cambridge University Press.
- Damasio, A. R. (2003). *Looking for Spinoza: Joy, sorrow and the feeling brain*. London:Heinemann.
- Best, R. (2003). *Struggling with the spiritual in education*. In Tenth international conference education spirituality and the whole child conference, University of Surrey Roehampton, London.
- Breazeal, C. and Aryananda, L. (2002). Recognition of affective communicative intent in robot-directed speech. *Autonomous Robots* 12 1, pp. 83–104.
- Creemers, B.P.M. (1994). *The effective classroom*. London: Cassell
- Goleman, D., (1995). *Emotional intelligence*. New York: Bantam Books.
- Harter, S., (1992). *The relationship between perceived competence, affect, and motivational orientation within the classroom: process and patterns of change*.
- Heckhausen, J. & Dweck, C.S. (1998). *Motivation and Self-Regulation Across the Life Span*. Cambridge Univ. Press.
- John Allen (2007) Photoplethysmography and its application in clinical physiological measurement, *Physiol. Meas.* 28 (2007)
- J. K. Aggarwal, Q. Cai, (1999). *Human Motion Analysis: A Review*, *Computer Vision and Image Understanding*, Vol. 73, No. 3
- Kort B, Reilly R, Picard R., (2001). *An Affective Model of Interplay between Emotions and Learning: Reengineering Educational Pedagogy-building a Learning Companion*. *Proceedings of IEEE International Conference on Advanced Learning Technologies*.
- Leal, M. R. M. (2002). *The caring relationship*. In Ninth annual international conference education spirituality and the whole child, Roehampton, London.
- Lerner, J. & Keltner, D., (2000). *Beyond valence: Toward a model of emotionspecific influences on judgment and choice*, *Cognition and Emotion*, 14(4), 473– 503.

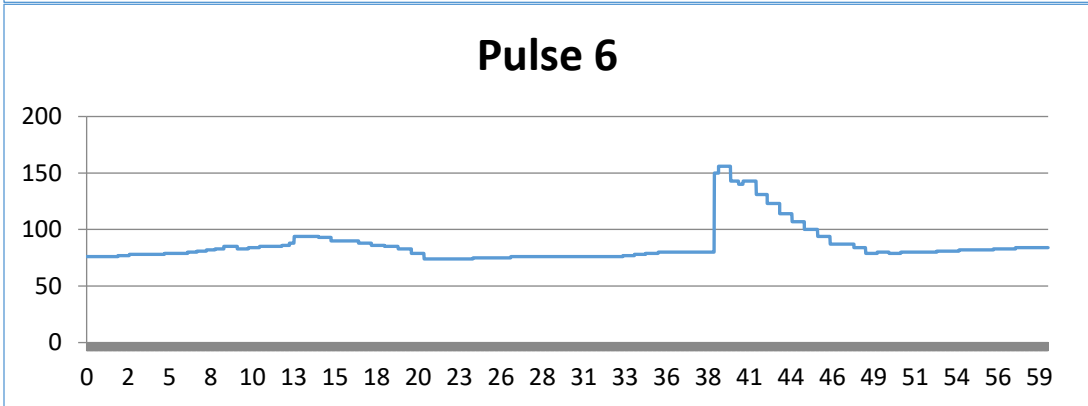
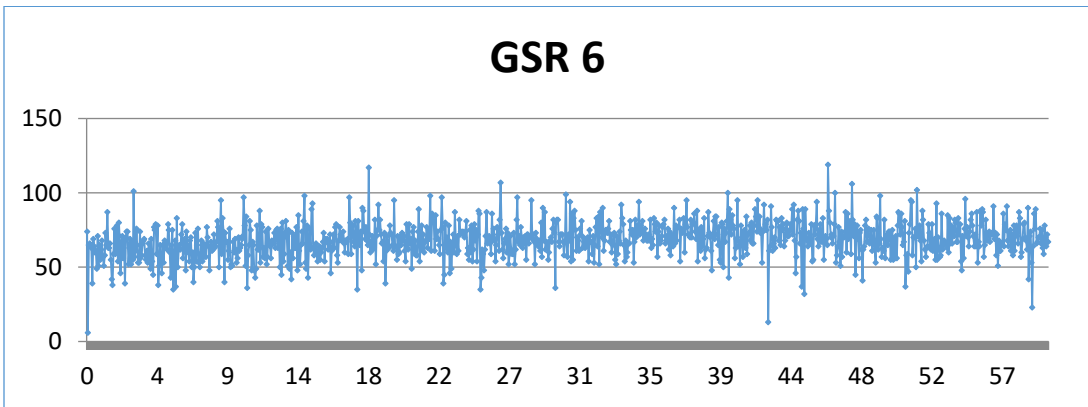
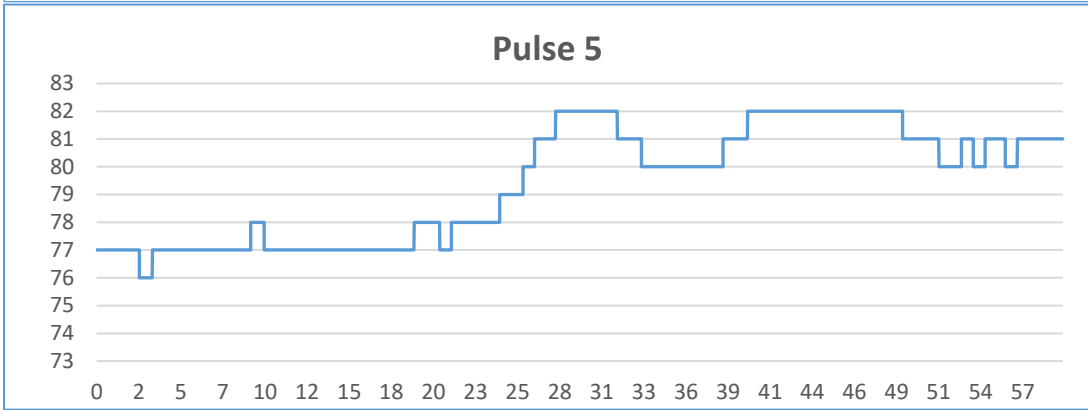
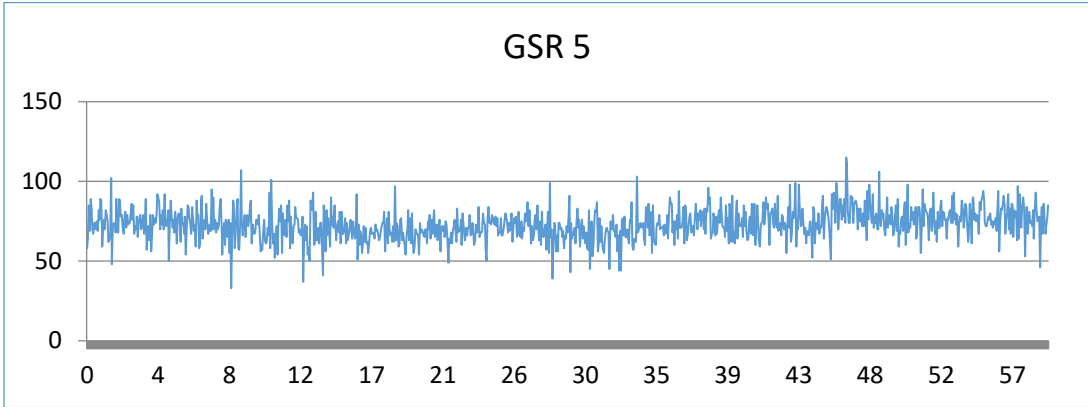
- Moridis, C.N. & Economides, A.A., (2008). Towards computer-aided affective learning systems: A literature review. *Journal of Educational Computing Research*, 39(4), 313-337.
- Nitzan M and Taitelbaum H, (2008). The measurement of oxygen saturation in arterial and venous blood *IEEE Instrum. Meas. Mag.* 11 9–15
- Pavlovic, Vladimir I.; Sharma, Rajeev; Huang, Thomas S. (1997). "Visual Interpretation of Hand Gestures for Human-Computer Interaction: A Review". *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
- Picard R.W. (1997). "Affective Computing", The MIT Press
- Picard, R., Papert, S., Bender, W., Blumberg, B., Breazeal, C., Cavallo, D., et al. (2004). Affective learning—A manifesto. *BT Technology Journal*, 22(4), 253-269.
- Russell, J. A. (1980) A circumplex model of affect. *Journal of Personality and Social Psychology*, 39 (6), 1161-1178
- Venkatesh, V., Morris, M. G., Davis, G. B., Davis, F. D. (2003). "User acceptance of information technology: Toward a unified view", *MIS Quarterly* 27(3): 425–478
- Zeidner, M., (2007). Test anxiety in educational contexts: What I have learned so far. In P. A. Schutz & R. Pekrun (Eds.), *Emotion in education* (pp. 165–184). San Diego, CA: Academic Press.
- Kort B, Reilly R, Picard R., (2001). An Affective Model of Interplay between Emotions and Learning: Reengineering Educational Pedagogy-building a Learning Companion. *Proceedings of IEEE International Conference on Advanced Learning Technologies*
- <http://www.biomed.ntua.gr>. (2018, Μάιος). Retrieved from http://www.biomed.ntua.gr/Portals/1/presentations/undergraduate/askhsh2_ECG.pdf.
- <https://biximikos.gr/topics/physiology-anatomy/100-anatomia-dermatos>. (n.d.).
- L.Pizzamiglio, G. (1999). Handbook of clinical and experimental neuropsychology. In G. & L.Pizzamiglio, *Handbook of clinical and experimental neuropsychology*.
- Plonsey, J. M. (2012). *Bioelectromagnetism: Principles and Applications of Bioelectric and Biomagnetic Fields*.
- R. Pflanzner, W. M. (2000). *Galvanic Skin Response & The Polygraph*.

5 ΠΑΡΑΡΤΗΜΑΤΑ

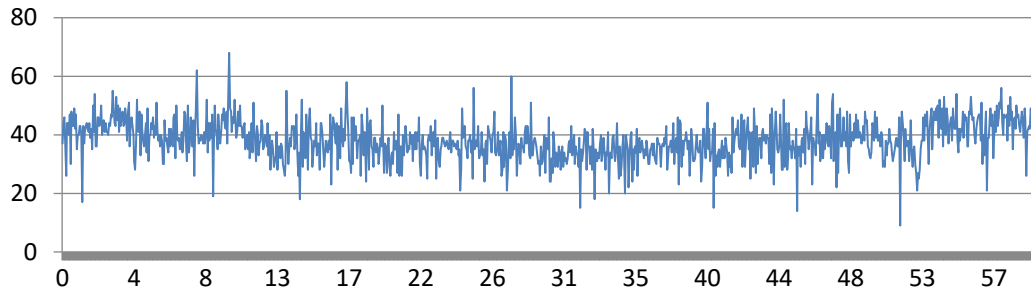
5.1 Διαγράμματα των 8 υποκειμένων από τις αποθηκευμένες τιμές των αισθητήρων



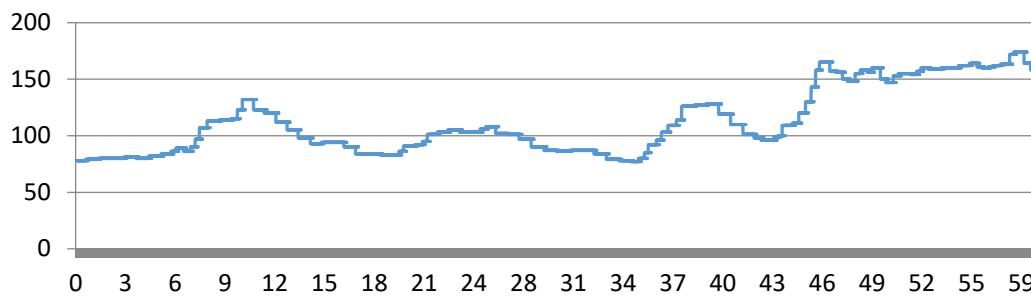




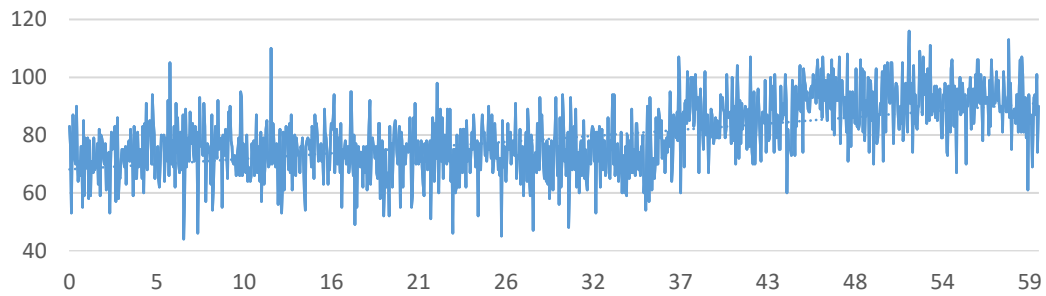
GSR 7



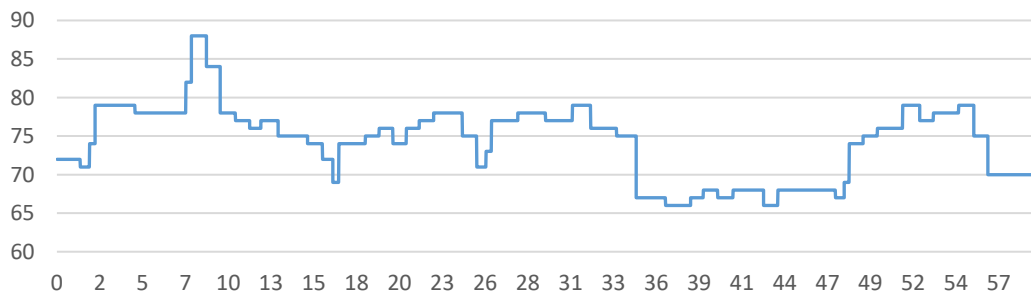
pulse 7

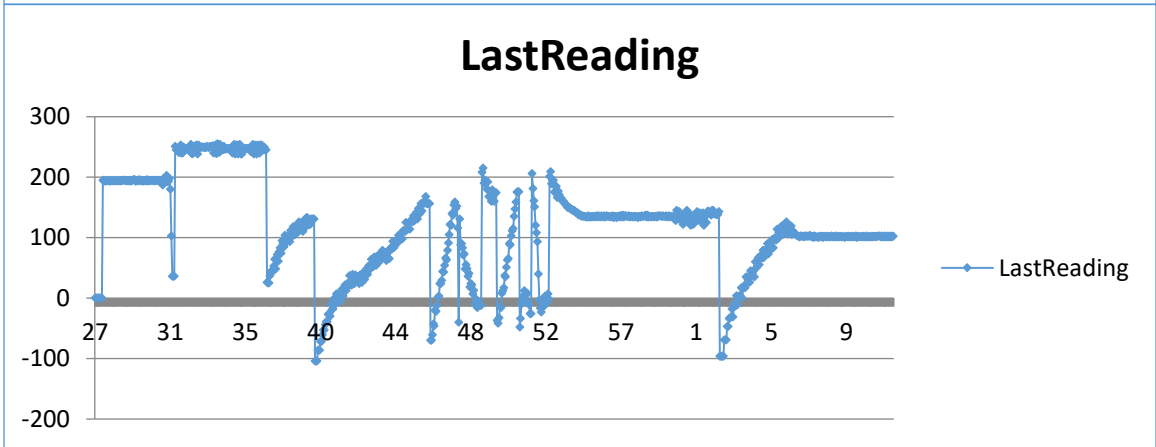
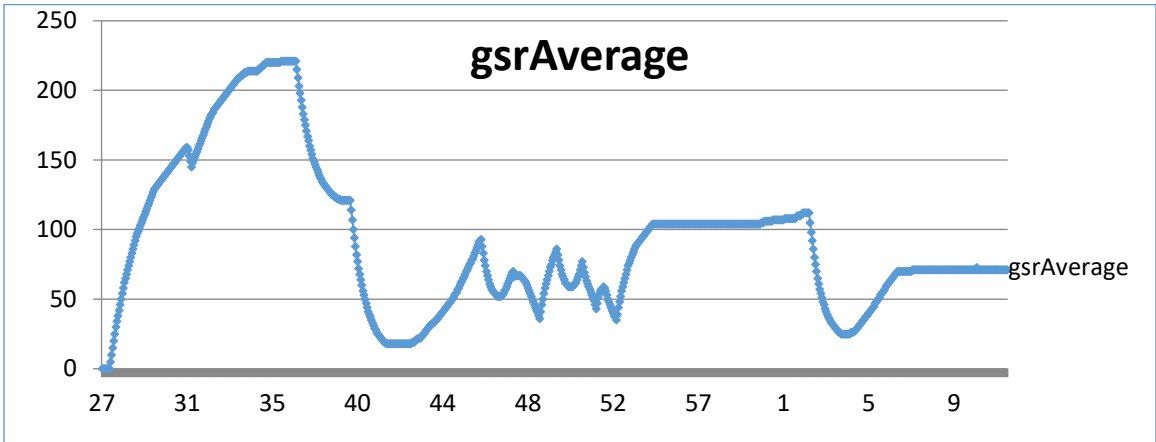


GSR 8



Pulse 8





5.2 Κώδικας GSR

5.2.1 Arduino GSR Code

```
void setup() {
  Serial.begin(9600);
}
void loop() {
  int a=analogRead(0);
  if (Serial.available() > 0) {
    byte inbyte=Serial.read();
    if(inbyte=='a'){
      Serial.print(a,BYTE);
    }
  }
}
```

5.2.2 Processing Code GSR

```
import processing.serial.*;
PrintWriter output;
PFont font;
Serial myPort;

inthPosition = 1; // the horizontal position on the graph
float currentReading;
float lastReading;
int count=0;
intzeroLinePos=0;

float gsrAverage,prevGsrAverage;
float baseLine=0;
long lastFlatLine=0;
color graphColor=color(255,255,0);
intbaselineTimer=10000;
intgsrValue;
intgsrZeroCount=0;
float gsrRange=0;
intdownhillCount=0;
intuphillCount=0;
boolean downhill;
boolean peaked=false;
float peak, valley;

void setup () {
  size(650, 450);
  // List all the available serial ports
  //println(Serial.list());
  myPort = new Serial(this, Serial.list()[2], 9600);
  //text (" lastReading =" + gsrAverage ,150,400);
  //font = loadFont("Arial-BoldMT-24.vlw");
  // textFont(font);
  //textAlign(LEFT);
  // text("lastReading = " lastReading , 450, 400);

  currentReading=0;
  lastReading=0;
  gsrAverage=0;
  background(200);//default 0=Black
  output = createWriter("Gsr.txt");
  // smooth();
}

void draw () {
```

```

// new entry for time 11-2-18
int s = second();
int m = minute();
int h = hour();
//best delay setting for gsr readings
delay(50);
text ("gsrAverage =" + gsrAverage ,250,400); //text GSR Average with Value
//text ("currentReading =" + currentReading ,250,420);
//image(myMovie, 0, 0);
println(h,m,s);// new entry 11-2-18 for time
if (gsrValue<15 &&gsrValue>-15){
  if( gsrZeroCount>10){
currentReading=0;//flatline
gsrAverage=0;
baseLine=0;
lastFlatLine=millis();
gsrZeroCount=0;
    // println("reset");

  }
gsrZeroCount++;
}
else{
currentReading=gsrValue-baseLine;
gsrZeroCount=0;
}

if(millis()-lastFlatLine>baselineTimer){
baseLine=gsrAverage;
}

//graph colors
// text (" lastReading =" + gsrAverage ,150,400);
if(gsrAverage>0 &&gsrAverage<height/2.0*.25) graphColor=color(255,255,255);
else if(gsrAverage>height/2.0*.25 &&gsrAverage<height/2.0*.5)
graphColor=color(255,250,100);
else if(gsrAverage>height/2.0*.5 &&gsrAverage<height/2.0*.75)
graphColor=color(0,255,0);
else if(gsrAverage>height/2.0*.75) graphColor=color(255,100,0);

gsrRange=peak-valley;

// at the edge of the screen, go back to the beginning:
if (hPosition>= width) {
hPosition = 0;

//cover last drawing
fill(0,200);
noStroke();
rect(0,0,width,height);
}
else {
hPosition+=1;
}

gsrAverage=smooth(currentReading, .97,gsrAverage);

//draw stuff

//spike
noStroke();
if(gsrRange>200){
fill(255);
ellipse(10,10,20,20);
}
else{
fill(0);
ellipse(10,10,20,20);
}

//graph
strokeWeight(0.5);

```

```

stroke(graphColor);
//println(graphColor);//-1-whit,1436-yell,1536-yell2,39936-red
println ("lastReading=" + lastReading );// console "lastReading";
//text (" lastReading =" + lastReading ,150,400);
line(hPosition-1, height/2.0-lastReading, hPosition, height/2.0-currentReading);
stroke(255,0,100);
line(hPosition-1,height/2.0-prevGsrAverage,hPosition,height/2.0-gsrAverage);

//draw peaks
intthres=7;

noFill();
stroke(255,0,0);
strokeWeight(2);

if (currentReading-thres>lastReading&& peaked==true){
  downhill=false;
  // println ( downhillCount);
  uphillCount++;
  downhillCount=0;
  point(hPosition-1, height/2.0-lastReading);
  valley=lastReading;
  peaked=false;
}
if(currentReading+thres<lastReading&& peaked==false){
  //println(uphillCount);
  downhill=true;
  uphillCount=0;
  downhillCount++;
  point(hPosition-1, height/2.0-lastReading);
  peak=lastReading;
  peaked=true;
}

prevGsrAverage=gsrAverage;
lastReading=currentReading;
//send 'a' for more bytes
myPort.write('a');
//output.println(lastReading);
output.println(m+ " " +s+ " " + " " +lastReading);
}

void serialEvent (Serial myPort) {
  intinByte=myPort.read();
  //0-255
  gsrValue=inByte;
}
void keyPressed(){
  if (keyCode==DOWN)zeroLinePos+=3;
  if (keyCode==UP)zeroLinePos-=3;

strokeWeight(1);
stroke(255,0,0);// def 255.0.0

line(0,zeroLinePos,2,zeroLinePos);
output.flush();
output.close();
exit(); // stop the program with another key
}
int smooth(float data, float filterVal, float smoothedVal){
  if (filterVal> 1){ // check to make sure param's are within range
    filterVal = .99;
  }
  else if (filterVal<= 0){
    filterVal = 0;
  }
  smoothedVal = (data * (1 - filterVal)) + (smoothedVal * filterVal);
  return (int)smoothedVal;
}

```

5.3 Κώδικας Pulse Sensor

5.3.1 Pulse Sensor Arduino Code

```
/* Pulse Sensor Amped 1.5 by Joel Murphy and YuryGitmanhttp://www.pulsesensor.com
*/
#define PROCESSING_VISUALIZER 1
#define SERIAL_PLOTTER 2

// Variables
intpulsePin = 0; // Pulse Sensor purple wire connected to analog pin 0
intblinkPin = 13; // pin to blink led at each beat
intfadePin = 5; // pin to do fancy classy fading blink at each beat
intfadeRate = 0; // used to fade LED on with PWM on fadePin

// Volatile Variables, used in the interrupt service routine!
volatile int BPM; // int that holds raw Analog in 0. updated every 2mS
volatile int Signal; // holds the incoming raw data
volatile int IBI = 600; // int that holds the time interval between beats!
Must be seeded!
volatile boolean Pulse = false; // "True" when User's live heartbeat is detected.
"False" when not a "live beat".
volatile boolean QS = false; // becomes true when Arduino finds a beat.

// SET THE SERIAL OUTPUT TYPE TO YOUR NEEDS
// PROCESSING_VISUALIZER works with Pulse Sensor Processing Visualizer
// https://github.com/WorldFamousElectronics/PulseSensor_Amped_Processing_Visualizer
// SERIAL_PLOTTER outputs sensor data for viewing with the Arduino Serial Plotter
// run the Serial Plotter at 115200 baud: Tools/Serial Plotter or Command+L
//static intoutputType = SERIAL_PLOTTER;
static intoutputType = PROCESSING_VISUALIZER;

void setup(){
pinMode(blinkPin,OUTPUT); // pin that will blink to your heartbeat!
pinMode(fadePin,OUTPUT); // pin that will fade to your heartbeat!
Serial.begin(115200); // we agree to talk fast!
interruptSetup(); // sets up to read Pulse Sensor signal every 2mS
// IF YOU ARE POWERING The Pulse Sensor AT VOLTAGE LESS THAN THE BOARD VOLTAGE,
// UN-COMMENT THE NEXT LINE AND APPLY THAT VOLTAGE TO THE A-REF PIN
// analogReference(EXTERNAL);
}

// Where the Magic Happens
void loop(){

serialOutput() ;

if (QS == true){ // A Heartbeat Was Found
// BPM and IBI have been Determined
// Quantified Self "QS" true when arduino finds a heartbeat
fadeRate = 255; // Makes the LED Fade Effect Happen
// Set 'fadeRate' Variable to 255 to fade LED with pulse
serialOutputWhenBeatHappens(); // A Beat Happened, Output that to serial.
QS = false; // reset the Quantified Self flag for next time
}

ledFadeToBeat(); // Makes the LED Fade Effect Happen
delay(20); // take a break
}

void ledFadeToBeat(){
fadeRate -= 15; // set LED fade value
fadeRate = constrain(fadeRate,0,255); // keep LED fade value from going into negative
numbers!
analogWrite(fadePin,fadeRate); // fade LED
}
```



```

}

//2.interrupt.ino
volatile int rate[10]; // array to hold last ten IBI values
volatile unsigned long sampleCounter = 0; // used to determine pulse timing
volatile unsigned long lastBeatTime = 0; // used to find IBI
volatile int P = 512; // used to find peak in pulse wave, seeded
volatile int T = 512; // used to find trough in pulse wave, seeded
volatile int thresh = 530; // used to find instant moment of heart beat,
seeded
volatile int amp = 0; // used to hold amplitude of pulse waveform,
seeded
volatile boolean firstBeat = true; // used to seed rate array so we startup with
reasonable BPM
volatile boolean secondBeat = false; // used to seed rate array so we startup with
reasonable BPM
void interruptSetup(){ // CHECK OUT THE Timer_Interrupt_Notes TAB FOR MORE ON INTERRUPTS
// Initializes Timer2 to throw an interrupt every 2mS.
TCCR2A = 0x02; // DISABLE PWM ON DIGITAL PINS 3 AND 11, AND GO INTO CTC MODE
TCCR2B = 0x06; // DON'T FORCE COMPARE, 256 PRESCALER
OCR2A = 0x7C; // SET THE TOP OF THE COUNT TO 124 FOR 500Hz SAMPLE RATE
TIMSK2 = 0x02; // ENABLE INTERRUPT ON MATCH BETWEEN TIMER2 AND OCR2A
sei(); // MAKE SURE GLOBAL INTERRUPTS ARE ENABLED
}

// THIS IS THE TIMER 2 INTERRUPT SERVICE ROUTINE.
// Timer 2 makes sure that we take a reading every 2 milliseconds
ISR(TIMER2_COMPA_vect){ // triggered when Timer2 counts to 124
cli(); // disable interrupts while we do this
Signal = analogRead(pulsePin); // read the Pulse Sensor
sampleCounter += 2; // keep track of the time in mS with this
variable
int N = sampleCounter - lastBeatTime; // monitor the time since the last beat to
avoid noise

// find the peak and trough of the pulse wave
if(Signal < thresh && N > (IBI/5)*3){ // avoid dichrotic noise by waiting 3/5 of
last IBI
if (Signal < T){ // T is the trough
T = Signal; // keep track of lowest point in pulse wave
}
}

if(Signal > thresh && Signal > P){ // thresh condition helps avoid noise
P = Signal; // P is the peak
} // keep track of highest point in pulse wave

// NOW IT'S TIME TO LOOK FOR THE HEART BEAT
// signal surges up in value every time there is a pulse
if (N > 250){ // avoid high frequency noise
if ( (Signal > thresh) && (Pulse == false) && (N > (IBI/5)*3) ){
Pulse = true; // set the Pulse flag when we think
there is a pulse
digitalWrite(blinkPin,HIGH); // turn on pin 13 LED
IBI = sampleCounter - lastBeatTime; // measure time between beats in mS
lastBeatTime = sampleCounter; // keep track of time for next pulse

if(secondBeat){ // if this is the second beat, if secondBeat
== TRUE
secondBeat = false; // clear secondBeat flag
for(int i=0; i<=9; i++){ // seed the running total to get a realistic
BPM at startup
rate[i] = IBI;
}
}

if(firstBeat){ // if it's the first time we found a beat,
if firstBeat == TRUE
firstBeat = false; // clear firstBeat flag
secondBeat = true; // set the second beat flag
}
}
}

```

```

sei(); // enable interrupts again
return; // IBI value is unreliable so discard it
}

// keep a running total of the last 10 IBI values
word runningTotal = 0; // clear the runningTotal variable

for(inti=0; i<=8; i++){ // shift data in the rate array
    rate[i] = rate[i+1]; // and drop the oldest IBI value
runningTotal += rate[i]; // add up the 9 oldest IBI values
}

rate[9] = IBI; // add the latest IBI to the rate array
runningTotal += rate[9]; // add the latest IBI to runningTotal
runningTotal /= 10; // average the last 10 IBI values
BPM = 60000/runningTotal; // how many beats can fit into a minute?
that's BPM!
QS = true; // set Quantified Self flag
// QS FLAG IS NOT CLEARED INSIDE THIS ISR
}

if (Signal < thresh && Pulse == true){ // when the values are going down, the beat is
over
digitalWrite(blinkPin,LOW); // turn off pin 13 LED
Pulse = false; // reset the Pulse flag so we can do it again
amp = P - T; // get amplitude of the pulse wave
thresh = amp/2 + T; // set thresh at 50% of the amplitude
P = thresh; // reset these for next time
T = thresh;
}

if (N > 2500){ // if 2.5 seconds go by without a beat
    thresh = 530; // set thresh default
    P = 512; // set P default
    T = 512; // set T default
lastBeatTime = sampleCounter; // bring the lastBeatTime up to date
firstBeat = true; // set these to avoid noise
secondBeat = false; // when we get the heartbeat back
}

sei(); // enable interrupts when youre done!
} // end isr

//.3 All Serial Handling Code
////////// All Serial Handling Code,
////////// It's Changeable with the 'outputType' variable
////////// It's declared at start of code.
void serialOutput(){ // Decide How To Output Serial.
    switch(outputType){
        case PROCESSING_VISUALIZER:
sendDataToSerial('S', Signal); // goes to sendDataToSerial function
            break;
        case SERIAL_PLOTTER: // open the Arduino Serial Plotter to visualize these data
Serial.print(BPM);
Serial.print(",");
Serial.print(IBI);
Serial.print(",");
Serial.println(Signal);
            break;
        default:
            break;
    }
}

// Decides How To OutPut BPM and IBI Data
void serialOutputWhenBeatHappens(){
    switch(outputType){
        case PROCESSING_VISUALIZER: // find it here
https://github.com/WorldFamousElectronics/PulseSensor\_Amped\_Processing\_Visualizer

```

```

sendDataToSerial('B',BPM); // send heart rate with a 'B' prefix
sendDataToSerial('Q',IBI); // send time between beats with a 'Q' prefix
    break;

    default:
        break;
}
}

// Sends Data to Pulse Sensor Processing App, Native Mac App, or Third-party Serial
Readers.
void sendDataToSerial(char symbol, int data ){
Serial.print(symbol);
Serial.println(data);
}
//4. Timer interrupt
/*
These notes put together by Joel Murphy for Pulse Sensor Amped, 2015
The code that this section is attached to uses a timer interrupt
to sample the Pulse Sensor with consistent and regular timing.
The code is setup to read Pulse Sensor signal at 500Hz (every 2mS).
The reasoning for this can be found here:
http://pulsesensor.com/pages/pulse-sensor-amped-arduino-vldot1
There are issues with using different timers to control the Pulse Sensor sample rate.
Sometimes, user will need to switch timers for access to other code libraries.
Also, some other hardware may have different timer setup requirements. This page
will cover those different needs and reveal the necessary settings. There are two
part of the code that will be discussed. The interruptSetup() routine, and
the interrupt function call. Depending on your needs, or the Arduino variant that you
use,
check below for the correct settings.

*****
ARDUINO UNO, Pro 328-5V/16MHZ, Pro-Mini 328-5V/16MHZ (or any board with ATmega328P
running at 16MHZ)

>> Timer2

Pulse Sensor Arduino UNO uses Timer2 by default.
Use of Timer2 interferes with PWM on pins 3 and 11.
There is also a conflict with the Tone library, so if you want tones, use Timer1
below.

void interruptSetup(){
// Initializes Timer2 to throw an interrupt every 2mS.
TCCR2A = 0x02; // DISABLE PWM ON DIGITAL PINS 3 AND 11, AND GO INTO CTC MODE
TCCR2B = 0x06; // DON'T FORCE COMPARE, 256 PRESCALER
OCR2A = 0X7C; // SET THE TOP OF THE COUNT TO 124 FOR 500Hz SAMPLE RATE
TIMSK2 = 0x02; // ENABLE INTERRUPT ON MATCH BETWEEN TIMER2 AND OCR2A
sei(); // MAKE SURE GLOBAL INTERRUPTS ARE ENABLED
}

use the following interrupt vector with Timer2

ISR(TIMER2_COMPA_vect)

>> Timer1

Use of Timer1 interferes with PWM on pins 9 and 10.
The Servo library also uses Timer1, so if you want servos, use Timer2 above.

void interruptSetup(){
// Initializes Timer1 to throw an interrupt every 2mS.
TCCR1A = 0x00; // DISABLE OUTPUTS AND PWM ON DIGITAL PINS 9 & 10
TCCR1B = 0x11; // GO INTO 'PHASE AND FREQUENCY CORRECT' MODE, NO PRESCALER
TCCR1C = 0x00; // DON'T FORCE COMPARE
TIMSK1 = 0x01; // ENABLE OVERFLOW INTERRUPT (TOIE1)
ICR1 = 16000; // TRIGGER TIMER INTERRUPT EVERY 2mS
sei(); // MAKE SURE GLOBAL INTERRUPTS ARE ENABLED
}

```

Use the following ISR vector for the Timer1 setup above

```
ISR(TIMER1_OVF_vect)

>> Timer0

DON'T USE TIMER0! Timer0 is used for counting delay(), millis(), and micros().
MESSING WITH Timer0 IS HIGHLY UNADVISED!

*****
IF YOU DON'T SEE THE MICROCONTROLLER YOU ARE USING, BUT YOU WANT A QUICK AND DIRTY
SOLUTION
So many new micros are coming out that it's kind of mind boggling. We will add to this
list with
code that uses interrupts when we can, but if your micro is not listed here, and you are
not willing
or able to grab a hardware timer yourself, here is a shortcut that will work.
It won't have the tight timing of a hardware interrupt, but it just might be good
enough.
We are calling this the 'Software Interrupt' version.
The code below will set up a microsecond timer and 'trigger' every 2mS (or so).
FIRST:
You will need to change the name of the function in the Interrupts tab from
'ISR(TIMER2_COMPA_vect)'
to
'void getPulse()'

THEN:
Comment out the entire interruptSetup() function in the interrupts tab in order for
this to work.
USE:
The code example below. Notice that we are using the micros() and the millis() to time
the sample rate and the fade rate.
DO NOT put any delays in the loop, or it will break the sample timing!

// FIRST, CREATE VARIABLES TO PERFORM THE SAMPLE TIMING AND LED FADE FUNCTIONS
unsigned long lastTime; // used to time the Pulse Sensor samples
unsigned long thisTime; // used to time the Pulse Sensor samples
unsigned long fadeTime; // used to time the LED fade

void setup(){
pinMode(blinkPin,OUTPUT); // pin that will blink to your heartbeat!
pinMode(fadePin,OUTPUT); // pin that will fade to your heartbeat!
Serial.begin(115200); // we agree to talk fast!
// ADD THIS LINE IN PLACE OF THE interruptSetup() CALL
lastTime = micros(); // get the time so we can create a software 'interrupt'
// IF YOU ARE POWERING The Pulse Sensor AT VOLTAGE LESS THAN THE BOARD VOLTAGE,
// UN-COMMENT THE NEXT LINE AND APPLY THAT VOLTAGE TO THE A-REF PIN
// analogReference(EXTERNAL);
} //end of setup()

//IN THE LOOP, ADD THE CODE THAT WILL DO THE 2mS TIMING, AND CALL THE getPulse()
FUNCTION.
void loop(){

serialOutput() ;

thisTime = micros(); // GET THE CURRENT TIME
if(thisTime - lastTime > 2000){ // CHECK TO SEE IF 2mS HAS PASSED
lastTime = thisTime; // KEEP TRACK FOR NEXT TIME
getPulse(); //CHANGE 'ISR(TIMER2_COMPA_vect)' TO 'getPulse()' IN THE
INTERRUPTS TAB!
}

if (QS == true){ // A Heartbeat Was Found
// BPM and IBI have been Determined
// Quantified Self "QS" true when arduino finds a heartbeat
fadeRate = 255; // Makes the LED Fade Effect Happen
// Set 'fadeRate' Variable to 255 to fade LED with pulse
fadeTime = millis(); // Set the fade timer to fade the LED
serialOutputWhenBeatHappens(); // A Beat Happened, Output that to serial.
```

```

        QS = false; // reset the Quantified Self flag for next time
    }

    if(millis() - fadeTime > 20){
    fadeTime = millis();
    ledFadeToBeat(); // Makes the LED Fade Effect Happen
    }
} // end of loop

```

5.3.2 PulseSensor Processing Code

```

/*
THIS PROGRAM WORKS WITH PulseSensorAmped_Arduino ARDUINO CODE
THE PULSE DATA WINDOW IS SCALEABLE WITH SCROLLBAR AT BOTTOM OF SCREEN
PRESS 'S' OR 's' KEY TO SAVE A PICTURE OF THE SCREEN IN SKETCH FOLDER (.jpg)
PRESS 'R' OR 'r' KEY TO RESET THE DATA TRACES
MADE BY JOEL MURPHY AUGUST, 2012
UPDATED BY JOEL MURPHY SUMMER 2016 WITH SERIAL PORT LOCATOR TOOL
UPDATED BY JOEL MURPHY WINTER 2017 WITH IMPROVED SERIAL PORT SELECTOR TOOL

THIS CODE PROVIDED AS IS, WITH NO CLAIMS OF FUNCTIONALITY OR EVEN IF IT WILL WORK
WYSIWYG
*/

import processing.serial.*; // serial library lets us talk to Arduino
PFont font;
PFontportsFont;
Scrollbar scaleBar;
PrintWriter output;

Serial port;

int Sensor; // HOLDS PULSE SENSOR DATA FROM ARDUINO
int IBI; // HOLDS TIME BETWEEN HEARTBEATS FROM ARDUINO
int BPM; // HOLDS HEART RATE VALUE FROM ARDUINO
int[] RawY; // HOLDS HEARTBEAT WAVEFORM DATA BEFORE SCALING
int[] ScaledY; // USED TO POSITION SCALED HEARTBEAT WAVEFORM
int[] rate; // USED TO POSITION BPM DATA WAVEFORM
float zoom; // USED WHEN SCALING PULSE WAVEFORM TO PULSE WINDOW
float offset; // USED WHEN SCALING PULSE WAVEFORM TO PULSE WINDOW
color eggshell = color(255, 253, 248);
int heart = 0; // This variable times the heart image 'pulse' on screen
// THESE VARIABLES DETERMINE THE SIZE OF THE DATA WINDOWS
intPulseWindowWidth = 490;
intPulseWindowHeight = 512;
intBPMWindowWidth = 180;
intBPMWindowHeight = 340;
boolean beat = false; // set when a heart beat is detected, then cleared when the BPM
graph is advanced

// SERIAL PORT STUFF TO HELP YOU FIND THE CORRECT SERIAL PORT
String serialPort;
String[] serialPorts = new String[Serial.list().length];
booleanserialPortFound = false;
Radio[] button = new Radio[Serial.list().length*2];
intnumPorts = serialPorts.length;
booleanrefreshPorts = false;

void setup() {
    size(700, 600); // Stage size
    frameRate(100);
    font = loadFont("Arial-BoldMT-24.vlw");
    textFont(font);
    textAlign(CENTER);
    rectMode(CENTER);
    ellipseMode(CENTER);
    // Scrollbar constructor inputs: x,y,width,height,minVal,maxVal

```

```

scaleBar = new Scrollbar (400, 575, 180, 12, 0.5, 1.0); // set parameters for the scale
bar
RawY = new int[PulseWindowWidth]; // initialize raw pulse waveform array
ScaledY = new int[PulseWindowWidth]; // initialize scaled pulse waveform array
rate = new int [BPMWindowWidth]; // initialize BPM waveform array
zoom = 0.75; // initialize scale of heartbeat window

// set the visualizer lines to 0
resetDataTraces();

background(0);
// DRAW OUT THE PULSE WINDOW AND BPM WINDOW RECTANGLES
drawDataWindows();
drawHeart();

// GO FIND THE ARDUINO
fill(eggshell);
text("Select Your Serial Port",245,30);
listAvailablePorts();
output = createWriter("pulse.txt");
}

void draw() {
// new entry for time
int s = second();
int m = minute();
int h = hour();
if(serialPortFound){
// ONLY RUN THE VISUALIZER AFTER THE PORT IS CONNECTED
background(0);
noStroke();
drawDataWindows();
drawPulseWaveform();
drawBPMwaveform();
drawHeart();
// PRINT THE DATA AND VARIABLE VALUES
fill(eggshell); // get ready to print text
text("Pulse Sensor Amped Visualizer v1.5",245,30); // tell them what you are
text("IBI " + IBI + "mS",600,585); // print the time between
heartbeats in mS
text(BPM + " BPM",600,200); // print the Beats Per Minute
text("Pulse Window Scale " + nf(zoom,1,2), 150, 585); // show the current scale of
Pulse Window

// DO THE SCROLLBAR THINGS
scaleBar.update (mouseX, mouseY);
scaleBar.display();

} else { // SCAN BUTTONS TO FIND THE SERIAL PORT

autoScanPorts();

if(refreshPorts){
refreshPorts = false;
drawDataWindows();
drawHeart();
listAvailablePorts();
}

for(inti=0; i<numPorts+1; i++){
button[i].overRadio(mouseX,mouseY);
button[i].displayRadio();
}

}
//output.println(m+" "+s+" "+BPM+" "+IBI);
output.println(m+" "+s+" "+BPM+" "+IBI+" "+Sensor);
} //end of draw loop

void drawDataWindows(){
// DRAW OUT THE PULSE WINDOW AND BPM WINDOW RECTANGLES

```

```

noStroke();
  fill(eggshell); // color for the window background
rect(255,height/2,PulseWindowWidth,PulseWindowHeight);
rect(600,385,BPMWindowWidth,BPMWindowHeight);
}

void drawPulseWaveform(){
  // DRAW THE PULSE WAVEFORM
  // prepare pulse data points
RawY[RawY.length-1] = (1023 - Sensor) - 212; // place the new raw datapoint at the end
of the array
  zoom = scaleBar.getPos(); // get current waveform scale value
  offset = map(zoom,0.5,1,150,0); // calculate the offset needed at this
scale
  for (inti = 0; i< RawY.length-1; i++) { // move the pulse waveform by
RawY[i] = RawY[i+1]; // shifting all raw datapoints one pixel
left
    float dummy = RawY[i] * zoom + offset; // adjust the raw data to the selected
scale
ScaledY[i] = constrain(int(dummy),44,556); // transfer the raw data array to the scaled
array
  }
  stroke(250,0,0); // red is a good color for the pulse
waveform
  noFill();
  beginShape(); // using beginShape() renders fast
    for (int x = 1; x < ScaledY.length-1; x++) {
      vertex(x+10, ScaledY[x]); //draw a line connecting the data points
    }
  endShape();
}

void drawBPMwaveform(){
  // DRAW THE BPM WAVE FORM
  // first, shift the BPM waveform over to fit then next data point only when a beat is
found
  if (beat == true){ // move the heart rate line over one pixel every time the heart
beats
    beat = false; // clear beat flag (beat flag was set in serialEvent tab)
    for (inti=0; i<rate.length-1; i++){
      rate[i] = rate[i+1]; // shift the bpm Y coordinates over one pixel
to the left
    }
  // then limit and scale the BPM value
  BPM = min(BPM,200); // limit the highest BPM value to 200
  float dummy = map(BPM,0,200,555,215); // map it to the heart rate window Y
  rate[rate.length-1] = int(dummy); // set the rightmost pixel to the new data
point value
  }
  // GRAPH THE HEART RATE WAVEFORM
  stroke(250,0,0); // color of heart rate graph
  strokeWeight(2); // thicker line is easier to read
  noFill();
  beginShape();
    for (inti=0; i< rate.length-1; i++){ // variable 'i' will take the place of pixel x
position
      vertex(i+510, rate[i]); // display history of heart rate datapoints
    }
  endShape();
}

void drawHeart(){
  // DRAW THE HEART AND MAYBE MAKE IT BEAT
  fill(250,0,0);
  stroke(250,0,0);
  // the 'heart' variable is set in serialEvent when arduino sees a beat happen
  heart--; // heart is used to time how long the heart graphic
swells when your heart beats
  heart = max(heart,0); // don't let the heart variable go into negative numbers
  if (heart > 0){ // if a beat happened recently,
strokeWeight(8); // make the heart big
  }
}

```

```

    smooth(); // draw the heart with two bezier curves
    bezier(width-100,50, width-20,-20, width,140, width-100,150);
    bezier(width-100,50, width-190,-20, width-200,140, width-100,150);
    strokeWeight(1); // reset the strokeWeight for next time
}

void listAvailablePorts(){
    println(Serial.list()); // print a list of available serial ports to the console
    serialPorts = Serial.list();
    fill(0);
    textFont(font,16);
    textAlign(LEFT);
    // set a counter to list the ports backwards
    int yPos = 0;
    int xPos = 35;
    for(int i=serialPorts.length-1; i>=0; i--){
        button[i] = new Radio(xPos,
95+(yPos*20),12,color(180),color(80),color(255),i,button);
        text(serialPorts[i],xPos+15, 100+(yPos*20));

    yPos++;
        if(yPos> height-30){
    yPos = 0; xPos+=200;
        }
    }
    int p = numPorts;
    fill(233,0,0);
    button[p] = new Radio(35, 95+(yPos*20),12,color(180),color(80),color(255),p,button);
    text("Refresh Serial Ports List",50, 100+(yPos*20));

    textFont(font);
    textAlign(CENTER);
}

void autoScanPorts(){
    if(Serial.list().length != numPorts){
        if(Serial.list().length >numPorts){
            println("New Ports Opened!");
            int diff = Serial.list().length - numPorts; // was serialPorts.length
            serialPorts = expand(serialPorts,diff);
            numPorts = Serial.list().length;
        }else if(Serial.list().length <numPorts){
            println("Some Ports Closed!");
            numPorts = Serial.list().length;
        }
        refreshPorts = true;
        return;
    }
}

void resetDataTraces(){
    for (inti=0; i<rate.length; i++){
        rate[i] = 555; // Place BPM graph line at bottom of BPM Window
    }
    for (inti=0; i<RawY.length; i++){
    RawY[i] = height/2; // initialize the pulse window data line to V/2
    }
}
// 2 Keyboard_mouse

void mousePressed(){
    scaleBar.press(mouseX, mouseY);
    if(!serialPortFound){
        for(int i=0; i<=numPorts; i++){
            if(button[i].pressRadio(mouseX,mouseY)){
                if(i == numPorts){
                    if(Serial.list().length >numPorts){
                        println("New Ports Opened!");
                        int diff = Serial.list().length - numPorts; // was serialPorts.length
                        serialPorts = expand(serialPorts,diff);
                        //button = (Radio[]) expand(button,diff);
                        numPorts = Serial.list().length;
                    }
                }
            }
        }
    }
}

```



```

        }else if(Serial.list().length < numPorts){
println("Some Ports Closed!");
numPorts = Serial.list().length;
        }else if(Serial.list().length == numPorts){
            return;
        }
refreshPorts = true;
        return;
    }else

        try{
            port = new Serial(this, Serial.list()[i], 115200); // make sure Arduino is
            talking serial at this baud rate
            delay(1000);
println(port.read());
            port.clear(); // flush buffer
            port.bufferUntil('\n'); // set buffer full flag on receipt of carriage return
            serialPortFound = true;
        }
        catch(Exception e){
println("Couldn't open port " + Serial.list()[i]);
            fill(255,0,0);
            textFont(font,16);
            textAlign(LEFT);
                text("Couldn't open port " + Serial.list()[i],60,70);
            textFont(font);
            textAlign(CENTER);
        }
    }
}
}

void mouseReleased(){
scaleBar.release();
}

void keyPressed(){

    switch(key){
        case 's': // pressing 's' or 'S' will take a jpg of the processing window
        case 'S':
saveFrame("heartLight-####.jpg"); // take a shot of that!
            break;
        case 'r':
        case 'R':
resetDataTraces();
            break;

        default:
            break;

    }
    output.flush();
    output.close();
    exit(); // stop the program with another key
}
// 2.Radio Buttons.pde

class Radio {
int _x,_y;
int size, dotSize;
    color baseColor, overColor, pressedColor;
boolean over, pressed;
int me;
    Radio[] radios;

    Radio(intxp, intyp, int s, color b, color o, color p, int m, Radio[] r) {
        _x = xp;
        _y = yp;
        size = s;

```

```

dotSize = size - size/3;
baseColor = b;
overColor = o;
pressedColor = p;
  radios = r;
  me = m;
}

booleanpressRadio(float mx, float my){
  if (dist(_x, _y, mx, my) < size/2){
    pressed = true;
    for(inti=0; i<numPorts+1; i++){
      if(i != me){ radios[i].pressed = false; }
    }
    return true;
  } else {
    return false;
  }
}

booleanoverRadio(float mx, float my){
  if (dist(_x, _y, mx, my) < size/2){
    over = true;
    for(inti=0; i<numPorts+1; i++){
      if(i != me){ radios[i].over = false; }
    }
    return true;
  } else {
    over = false;
    return false;
  }
}

void displayRadio(){
noStroke();
fill(baseColor);
ellipse(_x,_y,size,size);
if(over){
  fill(overColor);
  ellipse(_x,_y,dotSize,dotSize);
}
if(pressed){
  fill(pressedColor);
  ellipse(_x,_y,dotSize,dotSize);
}
}
}
//4.scale Bar.pde

/*
  THIS SCROLLBAR OBJECT IS BASED ON THE ONE FROM THE BOOK "Processing" by Reas and Fry
*/

class Scrollbar{
intx,y;          // the x and y coordinates
float sw, sh;    // width and height of scrollbar
float pos;       // position of thumb
float posMin, posMax; // max and min values of thumb
boolean rollover; // true when the mouse is over
boolean locked;  // true when it's the active scrollbar
float minVal, maxVal; // min and max values for the thumb

  Scrollbar (intxp, intyp, int w, int h, float miv, float mav){ // values passed from the
  constructor
    x = xp;
    y = yp;
    sw = w;
    sh = h;
    minVal = miv;
    maxVal = mav;
    pos = x - sh/2;
    posMin = x-sw/2;

```

```

posMax = x + sw/2; // - sh;
}

// updates the 'over' boolean and position of thumb
void update(int mx, int my) {
  if (over(mx, my) == true){
    rollover = true; // when the mouse is over the scrollbar, rollover is
true
  } else {
    rollover = false;
  }
  if (locked == true){
    pos = constrain (mx, posMin, posMax);
  }
}

// locks the thumb so the mouse can move off and still update
void press(int mx, int my){
  if (rollover == true){
    locked = true; // when rollover is true, pressing the mouse button will
lock the scrollbar on
  }else{
    locked = false;
  }
}

// resets the scrollbar to neutral
void release(){
  locked = false;
}

// returns true if the cursor is over the scrollbar
boolean over(int mx, int my){
  if ((mx > x-sw/2) && (mx <x+sw/2) && (my > y-sh/2) && (my <y+sh/2)){
    return true;
  }else{
    return false;
  }
}

// draws the scrollbar on the screen
void display (){

noStroke();
  fill(255);
rect(x, y, sw, sh); // create the scrollbar
  fill (250,0,0);
  if ((rollover == true) || (locked == true)){
    stroke(250,0,0);
strokeWeight(8); // make the scale dot bigger if you're on it
  }
  ellipse(pos, y, sh, sh); // create the scaling dot
strokeWeight(1); // reset strokeWeight
}

// returns the current value of the thumb
float getPos() {
  float scalar = sw / sw; // (sw - sh/2);
  float ratio = (pos-(x-sw/2)) * scalar;
  float p = minVal + (ratio/sw * (maxVal - minVal));
  return p;
}
}
//5.serial_event.pde

void serialEvent(Serial port){
try{
  String inData = port.readStringUntil('\n');
  inData = trim(inData); // cut off white space (carriage return)
}
}

```

```

    if (inData.charAt(0) == 'S'){           // leading 'S' means Pulse Sensor data packet
inData = inData.substring(1);           // cut off the leading 'S'
        Sensor = int(inData);           // convert the string to usable int
    }
    if (inData.charAt(0) == 'B'){           // leading 'B' for BPM data
inData = inData.substring(1);           // cut off the leading 'B'
        BPM = int(inData);           // convert the string to usable int
        beat = true;           // set beat flag to advance heart rate graph
        heart = 20;           // begin heart image 'swell' timer
    }
    if (inData.charAt(0) == 'Q'){           // leading 'Q' means IBI data
inData = inData.substring(1);           // cut off the leading 'Q'
        IBI = int(inData);           // convert the string to usable int
    }
} catch(Exception e) {
    // println(e.toString());
}
} // END OF SERIAL EVENT

```

5.4 DATA SHEETS

LM324, LM324A, LM324E, LM224, LM2902, LM2902E, LM2902V, NCV2902

Single Supply Quad Operational Amplifiers

The LM324 series are low-cost, quad operational amplifiers with true differential inputs. They have several distinct advantages over standard operational amplifier types in single supply applications. The quad amplifier can operate at supply voltages as low as 3.0 V or as high as 32 V with quiescent currents about one-fifth of those associated with the MC1741 (on a per amplifier basis). The common mode input range includes the negative supply, thereby eliminating the necessity for external biasing components in many applications. The output voltage range also includes the negative power supply voltage.

Features

- Short Circuited Protected Outputs
- True Differential Input Stage
- Single Supply Operation: 3.0 V to 32 V
- Low Input Bias Currents: 100 nA Maximum (LM324A)
- Four Amplifiers Per Package
- Internally Compensated
- Common Mode Range Extends to Negative Supply
- Industry Standard Pinouts
- ESD Clamps on the Inputs Increase Ruggedness without Affecting Device Operation
- NCV Prefix for Automotive and Other Applications Requiring Unique Site and Control Change Requirements; AEC-Q100 Qualified and PPAP Capable
- These Devices are Pb-Free, Halogen Free/BFR Free and are RoHS Compliant



ON Semiconductor®

www.onsemi.com



PDIP-14
N SUFFIX
CASE 646

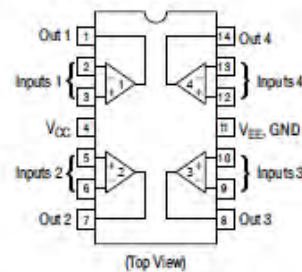


SOIC-14
D SUFFIX
CASE 751A



TSSOP-14
DTB SUFFIX
CASE 948G

PIN CONNECTIONS



ORDERING INFORMATION

See detailed ordering and shipping information in the package dimensions section on page 10 of this data sheet.

DEVICE MARKING INFORMATION

See general marking information in the device marking section on page 11 of this data sheet.

LM324, LM324A, LM324E, LM224, LM2902, LM2902E, LM2902V, NCV2902

MAXIMUM RATINGS ($T_A = +25^\circ\text{C}$, unless otherwise noted.)

| Rating | Symbol | Value | Unit |
|--|------------------------------|--|--------------------|
| Power Supply Voltages Single Supply Split Supplies | V_{CC} V_{CC}, V_{EE} | 32 ± 18 | Vdc |
| Input Differential Voltage Range (Note 1) | V_{IDR} | ± 32 | Vdc |
| Input Common Mode Voltage Range | V_{ICR} | -0.3 to 32 | Vdc |
| Output Short Circuit Duration | t_{SC} | Continuous | |
| Junction Temperature | T_J | 150 | $^\circ\text{C}$ |
| Thermal Resistance, Junction-to-Air (Note 2) | $R_{\theta JA}$ | Case 848 118 Case 751A 156 Case 948G 190 | $^\circ\text{C/W}$ |
| Storage Temperature Range | T_{stg} | -65 to +150 | $^\circ\text{C}$ |
| Operating Ambient Temperature Range | T_A | LM224 -25 to +85 LM324, LM324A, LM324E 0 to +70 LM2902, LM2902E -40 to +105 LM2902V, NCV2902 (Note 3) -40 to +125 | $^\circ\text{C}$ |

Stresses exceeding those listed in the Maximum Ratings table may damage the device. If any of these limits are exceeded, device functionality should not be assumed, damage may occur and reliability may be affected.

1. Split Power Supplies.
2. All $R_{\theta JA}$ measurements made on evaluation board with 1 oz. copper traces of minimum pad size. All device outputs were active.
3. NCV2902 is qualified for automotive use.

ESD RATINGS

| Rating | HBM | MM | Unit |
|--|------|-----|------|
| ESD Protection at any Pin (Human Body Model – HBM, Machine Model – MM) | | | |
| NCV2902 (Note 3) | 2000 | 200 | V |
| LM324E, LM2902E | 2000 | 200 | V |
| LM324DG/DR2G, LM2902DG/DR2G | 200 | 100 | V |
| All Other Devices | 2000 | 200 | V |

LM324, LM324A, LM324E, LM224, LM2902, LM2902E, LM2902V, NCV2902

ELECTRICAL CHARACTERISTICS ($V_{CC} = 5.0\text{ V}$, $V_{EE} = \text{GND}$, $T_A = 25^\circ\text{C}$, unless otherwise noted.)

| Characteristic | Symbol | LM224 | | | LM324A | | | LM324, LM324E | | | LM2902, LM2902E | | | LM2902V/NCV2902 | | | Unit |
|--|--------------------------|-------|------|----------|--------|------|----------|---------------|------|----------|-----------------|------|----------|-----------------|------|----------|------------------------------|
| | | Min | Typ | Max | Min | Typ | Max | Min | Typ | Max | Min | Typ | Max | Min | Typ | Max | |
| Input Offset Voltage $V_{CC} = 5.0\text{ V to }30\text{ V}$ $V_{ICR} = 0\text{ V to }V_{CC} - 1.7\text{ V}$ $V_O = 1.4\text{ V}$, $R_B = 0\ \Omega$ $T_A = 25^\circ\text{C}$ $T_A = T_{\text{high}}$ (Note 4) $T_A = T_{\text{low}}$ (Note 4) | V_{IO} | - | 2.0 | 5.0 | - | 2.0 | 3.0 | - | 2.0 | 7.0 | - | 2.0 | 7.0 | - | 2.0 | 7.0 | mV |
| Average Temperature Coefficient of Input Offset Voltage $T_A = T_{\text{high}}$ to T_{low} (Notes 4 and 6) | $\Delta V_{IO}/\Delta T$ | - | 7.0 | - | - | 7.0 | 30 | - | 7.0 | - | - | 7.0 | - | - | 7.0 | - | $\mu\text{V}/^\circ\text{C}$ |
| Input Offset Current $T_A = T_{\text{high}}$ to T_{low} (Note 4) | I_{IO} | - | 3.0 | 30 | - | 5.0 | 30 | - | 5.0 | 50 | - | 5.0 | 50 | - | 5.0 | 50 | nA |
| Average Temperature Coefficient of Input Offset Current $T_A = T_{\text{high}}$ to T_{low} (Notes 4 and 6) | $\Delta I_{IO}/\Delta T$ | - | 10 | - | - | 10 | 300 | - | 10 | - | - | 10 | - | - | 10 | - | $\text{pA}/^\circ\text{C}$ |
| Input Bias Current $T_A = T_{\text{high}}$ to T_{low} (Note 4) | I_B | - | -90 | -150 | - | -45 | -100 | - | -90 | -250 | - | -90 | -250 | - | -90 | -250 | nA |
| Input Common Mode Voltage Range (Note 5) $V_{CC} = 30\text{ V}$ $T_A = +25^\circ\text{C}$ $T_A = T_{\text{high}}$ to T_{low} (Note 4) | V_{ICR} | 0 | - | 28.3 | 0 | - | 28.3 | 0 | - | 28.3 | 0 | - | 28.3 | 0 | - | 28.3 | V |
| Differential Input Voltage Range | V_{ICR} | - | - | V_{CC} | - | - | V_{CC} | - | - | V_{CC} | - | - | V_{CC} | - | - | V_{CC} | V |
| Large Signal Open Loop Voltage Gain $R_L = 2.0\text{ k}\Omega$ $V_{CC} = 15\text{ V}$ for Large V_O Swing $T_A = T_{\text{high}}$ to T_{low} (Note 4) | A_{VOL} | 50 | 100 | - | 25 | 100 | - | 25 | 100 | - | 25 | 100 | - | 25 | 100 | - | V/mV |
| Channel Separation 10 kHz $\leq f \leq$ 20 kHz Input Referenced | CS | - | -120 | - | - | -120 | - | - | -120 | - | - | -120 | - | - | -120 | - | dB |
| Common Mode Rejection $R_S \leq 10\text{ k}\Omega$ | CMR | 70 | 65 | - | 65 | 70 | - | 65 | 70 | - | 50 | 70 | - | 50 | 70 | - | dB |
| Power Supply Rejection | PSR | 65 | 100 | - | 65 | 100 | - | 65 | 100 | - | 50 | 100 | - | 50 | 100 | - | dB |

4. LM224: $T_{\text{low}} = -25^\circ\text{C}$, $T_{\text{high}} = +85^\circ\text{C}$
 LM324/LM324A/LM324E: $T_{\text{low}} = 0^\circ\text{C}$, $T_{\text{high}} = +70^\circ\text{C}$
 LM2902/LM2902E: $T_{\text{low}} = -40^\circ\text{C}$, $T_{\text{high}} = +105^\circ\text{C}$
 LM2902V & NCV2902: $T_{\text{low}} = -40^\circ\text{C}$, $T_{\text{high}} = +125^\circ\text{C}$
 NCV2902 is qualified for automotive use.

5. The input common mode voltage or either input signal voltage should not be allowed to go negative by more than 0.3 V. The upper end of the common mode voltage range is $V_{CC} - 1.7\text{ V}$, but either or both inputs can go to +32 V without damage, independent of the magnitude of V_{CC} .
6. Guaranteed by design.

LM324, LM324A, LM324E, LM224, LM2902, LM2902E, LM2902V, NCV2902

ELECTRICAL CHARACTERISTICS ($V_{CC} = 5.0\text{ V}$, $V_{EE} = \text{GND}$, $T_A = 25^\circ\text{C}$, unless otherwise noted.)

| Characteristics | Symbol | LM224 | | | LM324A | | | LM324, LM324E | | | LM2902, LM2902E | | | LM2902V/NCV2902 | | | Unit |
|---|----------|-------|-----|-----|--------|-----|-----|---------------|-----|-----|-----------------|-----|-----|-----------------|-----|-----|------|
| | | Min | Typ | Max | Min | Typ | Max | Min | Typ | Max | Min | Typ | Max | Min | Typ | Max | |
| Output Voltage – High Limit $V_{CC} = 5.0\text{ V}$, $R_L = 2.0\text{ k}\Omega$, $T_A = 25^\circ\text{C}$ $V_{CC} = 30\text{ V}$ $R_L = 2.0\text{ k}\Omega$ ($T_A = T_{\text{high}}$ to T_{low}) (Note 7) $V_{CC} = 30\text{ V}$ $R_L = 10\text{ k}\Omega$ ($T_A = T_{\text{high}}$ to T_{low}) (Note 7) | V_{OH} | 3.3 | 3.5 | - | 3.3 | 3.5 | - | 3.3 | 3.5 | - | 3.3 | 3.5 | - | 3.3 | 3.5 | - | V |
| Output Voltage – Low Limit $V_{CC} = 5.0\text{ V}$, $R_L = 10\text{ k}\Omega$, $T_A = T_{\text{high}}$ to T_{low} (Note 7) | V_{OL} | - | 5.0 | 20 | - | 5.0 | 20 | - | 5.0 | 20 | - | 5.0 | 100 | - | 5.0 | 100 | mV |
| Output Source Current ($V_{IO} = +1.0\text{ V}$, $V_{CC} = 15\text{ V}$) $T_A = 25^\circ\text{C}$ $T_A = T_{\text{high}}$ to T_{low} (Note 7) | I_{O+} | 20 | 40 | - | 20 | 40 | - | 20 | 40 | - | 20 | 40 | - | 20 | 40 | - | mA |
| Output Sink Current ($V_{IO} = -1.0\text{ V}$, $V_{CC} = 15\text{ V}$) $T_A = 25^\circ\text{C}$ $T_A = T_{\text{high}}$ to T_{low} (Note 7) $(V_{IO} = -1.0\text{ V}$, $V_O = 200\text{ mV}$, $T_A = 25^\circ\text{C})$ | I_{O-} | 10 | 20 | - | 10 | 20 | - | 10 | 20 | - | 10 | 20 | - | 10 | 20 | - | mA |
| Output Short Circuit to Ground (Note 8) | I_{SC} | - | 40 | 60 | - | 40 | 60 | - | 40 | 60 | - | 40 | 60 | - | 40 | 60 | mA |
| Power Supply Current ($T_A = T_{\text{high}}$ to T_{low}) (Note 7) $V_{CC} = 30\text{ V}$ $V_O = 0\text{ V}$, $R_L = \infty$ $V_{CC} = 5.0\text{ V}$, $V_O = 0\text{ V}$, $R_L = \infty$ | I_{CC} | - | - | 3.0 | - | 1.4 | 3.0 | - | - | 3.0 | - | - | 3.0 | - | - | 3.0 | mA |

7. LM224: $T_{\text{low}} = -25^\circ\text{C}$, $T_{\text{high}} = +85^\circ\text{C}$
 LM324/LM324A/LM324E: $T_{\text{low}} = 0^\circ\text{C}$, $T_{\text{high}} = +70^\circ\text{C}$
 LM2902/LM2902E: $T_{\text{low}} = -40^\circ\text{C}$, $T_{\text{high}} = +105^\circ\text{C}$
 LM2902V & NCV2902: $T_{\text{low}} = -40^\circ\text{C}$, $T_{\text{high}} = +125^\circ\text{C}$
 NCV2902 is qualified for automotive use.

8. The input common mode voltage or either input signal voltage should not be allowed to go negative by more than 0.3 V. The upper end of the common mode voltage range is $V_{CC} - 1.7\text{ V}$, but either or both inputs can go to +32 V without damage, independent of the magnitude of V_{CC} .

Product parametric performance is indicated in the Electrical Characteristics for the listed test conditions, unless otherwise noted. Product performance may not be indicated by the Electrical Characteristics if operated under different conditions.

LM324, LM324A, LM324E, LM224, LM2902, LM2902E, LM2902V, NCV2902

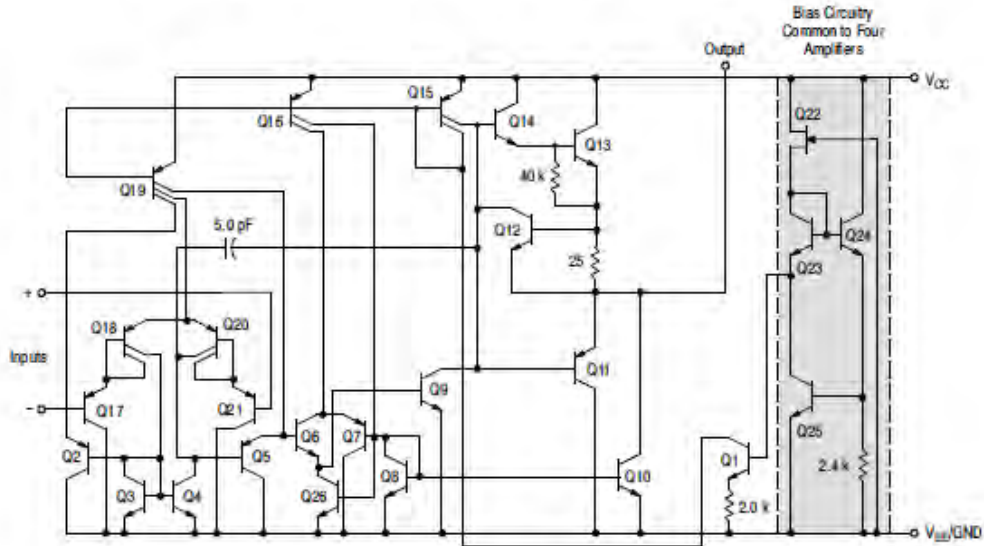


Figure 1. Representative Circuit Diagram
(One-Fourth of Circuit Shown)

LM324, LM324A, LM324E, LM224, LM2902, LM2902E, LM2902V, NCV2902

CIRCUIT DESCRIPTION

The LM324 series is made using four internally compensated, two-stage operational amplifiers. The first stage of each consists of differential input devices Q20 and Q18 with input buffer transistors Q21 and Q17 and the differential to single ended converter Q3 and Q4. The first stage performs not only the first stage gain function but also performs the level shifting and transconductance reduction functions. By reducing the transconductance, a smaller compensation capacitor (only 5.0 pF) can be employed, thus saving chip area. The transconductance reduction is accomplished by splitting the collectors of Q20 and Q18. Another feature of this input stage is that the input common mode range can include the negative supply or ground, in single supply operation, without saturating either the input devices or the differential to single-ended converter. The second stage consists of a standard current source load amplifier stage.

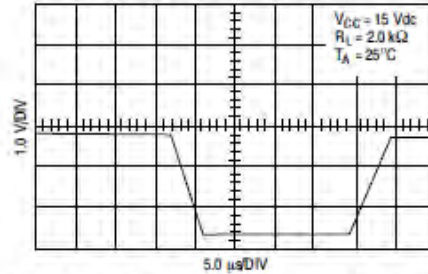


Figure 2. Large Signal Voltage Follower Response

Each amplifier is biased from an internal-voltage regulator which has a low temperature coefficient thus giving each amplifier good temperature characteristics as well as excellent power supply rejection.



Figure 3.

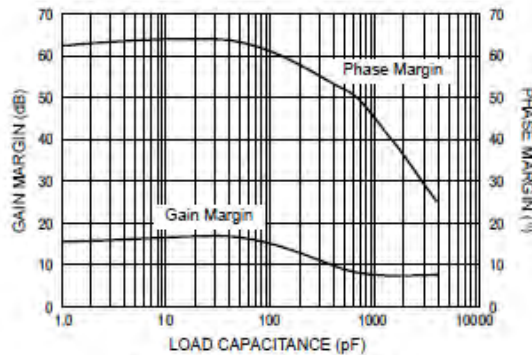


Figure 4. Gain and Phase Margin

LM324, LM324A, LM324E, LM224, LM2902, LM2902E, LM2902V, NCV2902

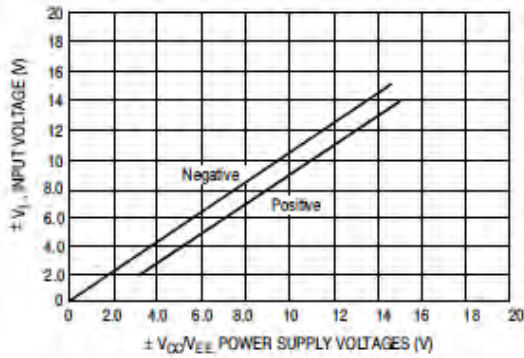


Figure 5. Input Voltage Range

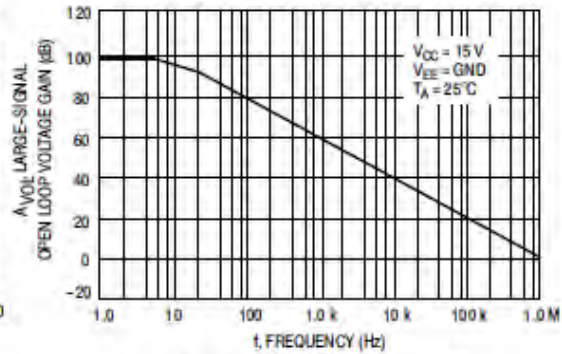


Figure 6. Open Loop Frequency

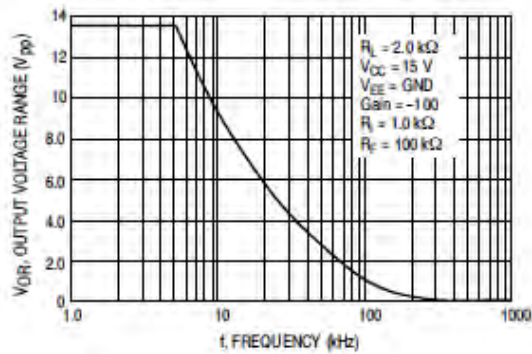


Figure 7. Large-Signal Frequency Response

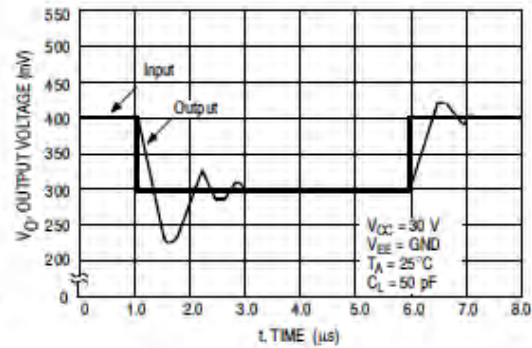


Figure 8. Small-Signal Voltage Follower Pulse Response (Noninverting)

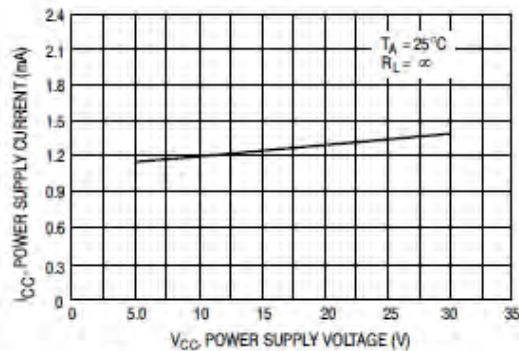


Figure 9. Power Supply Current versus Power Supply Voltage

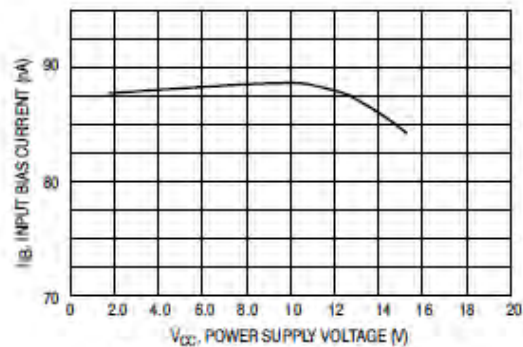


Figure 10. Input Bias Current versus Power Supply Voltage

