



**ΑΝΩΤΑΤΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΠΕΙΡΑΙΑ
ΤΕΧΝΟΛΟΓΙΚΟΥ ΤΟΜΕΑ**

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

**ΚΑΤΑΣΚΕΥΗ MP3 PLAYER ΚΑΙ ALARM CLOCK ΜΕ
ΤΕΧΝΟΛΟΓΙΑ ARDUINO**

ΔΕΔΟΥΣΗΣ ΑΔΑΜ – ΔΙΟΝΥΣΟΠΟΥΛΟΣ ΚΩΝΣΤΑΝΤΙΝΟΣ

Επιβλέπων Καθηγητής: κα Αναστασία Βελώνη

ΠΕΡΙΕΧΟΜΕΝΑ

Περιεχόμενα.....	σελ.2
Κεφάλαιο 1 – Θεωρητικό υπόβαθρο	
1.1 Ιστορία Arduino.....	σελ.4
1.2 Αρχιτεκτονική Arduino.....	σελ.4
1.2.1 Γενικές πληροφορίες για το Arduino.....	σελ.3-4
1.2.2 Το software για τον προγραμματισμό του arduino.....	σελ.5
1.2.3 Μοντέλα Arduino.....	σελ.6-7
1.2.4 Arduino Shields.....	σελ.7-8
1.2.5 Τεχνικά χαρακτηριστικά του Arduino.....	σελ.8
1.2.6 Τροφοδοσία.....	σελ.8
1.3 Δυνατότητες και Πλεονεκτήματα.....	σελ.8-9
1.4 Περιγραφή του hardware που χρησιμοποιείται για την υλοποίηση του συστήματος.....	σελ.10
1.4.1 Τεχνικά χαρακτηριστικά του Arduino Mega 2560 R3.....	σελ.10
1.4.2 Μνήμη.....	σελ.10-11
1.4.3 Τροφοδοσία.....	σελ.11-12
1.4.4 Ακροδέκτες.....	σελ.12-13
1.4.4.1.1 Ακροδέκτες Digital In/OUT.....	σελ.13-14
1.4.4.1.2 Ακροδέκτες Analog IN.....	σελ.15
1.5 Ολοκληρωμένο περιβάλλον ανάπτυξης Arduino(IDE).....	σελ.15-18
1.5.1 Βασικές δομές και λειτουργίες προγραμματισμού....	σελ.18-22
Κεφάλαιο 2- Υλικό μέρος του συστήματος.....	σελ.22-23
2.1 Tft Display Mega Shield.....	σελ.23-26
2.2 Real Time Clock DS3231.....	σελ.27-28
2.3 Mp3 Module(BY8001-16P).....	σελ.29-32
2.4 TFT 3.2” Touch Screen.....	σελ.33
2.5 Arduino Mega Proto Shield Rev3.....	σελ.34-35
Κεφάλαιο 3 – Πειραματικό Στάδιο.....	σελ.35-48
Βιβλιογραφία.....	σελ.49

ΠΕΡΙΛΗΨΗ

Η κατασκευή της παρούσας πτυχιακής εργασίας έχει ως στόχο την δημιουργία μιας συσκευής που θα λειτουργεί είτε σαν ξυπνητήρι είτε σαν συσκευή αναπαραγωγής τραγουδιών(MP3-WAV).

Η συσκευή αναπτύχθηκε στην πλακέτα **Arduino Mega 2560 R3**, η οποία προγραμματίστηκε κατάλληλα έτσι ώστε να αναπαράγει αρχεία ήχου μέσω μιας κάρτας SD που είναι τοποθετημένη μέσα στο MP3 module και να λειτουργεί ως ξυπνητήρι.

ABSTRACT

The construction of this thesis aims to create a device that will work either as an alarm clock or as a MP3 Player(supports MP3-WAV files). The device was developed on the Arduino Mega 2560 R3 board, which was properly programmed to reproduced audio files via SD card inserted in the Mp3 Module and to work as an alarm clock.

Κεφάλαιο 1 - Θεωρητικό υπόβαθρο

1.1 Ιστορία του Arduino

Το 2005, δημιουργήθηκε ένα σχέδιο προκειμένου να φτιαχτεί μία συσκευή για τον έλεγχο προγραμμάτων διαδραστικών σχεδίων από μαθητές, η οποία θα ήταν πιο φθηνή από τα άλλα πρωτότυπα συστήματα που ήταν διαθέσιμα εκείνη την περίοδο. Οι ιδρυτές Massimo Banzi και David Cueartielles ξεκίνησαν να παράγουν πλακέτες σε ένα μικρό εργοστάσιο στην Ivrea, η οποία είναι κωμόπολη της επαρχίας Τορίνο στην περιοχή Πεδεμόντιο της βορειοδυτικής Ιταλίας, στην ίδια περιοχή στην οποία στεγαζόταν η εταιρία υπολογιστών Olivetti. Σήμερα έχει κατακτήσει όλο τον κόσμο με τις δεκάδες πλακέτες ανάπτυξης που έχουν δημιουργηθεί.

1.2 Αρχιτεκτονική Arduino

Το Arduino, είναι μια υπολογιστική πλατφόρμα βασισμένη σε μια απλή μητρική πλακέτα με ενσωματωμένο μικροελεγκτή και εισόδους/εξόδους. Ο προγραμματισμός του γίνεται με τη γλώσσα Wiring η οποία είναι μια παραλλαγή της C++. Μέσα από τα περιβάλλον ανάπτυξης κώδικα IDE η γλώσσα wiring μπορεί να συνταχθεί και να υλοποιηθεί σε οποιοδήποτε λειτουργικό σύστημα. Είναι ανοιχτού κώδικα λογισμικό και μας επιτρέπει να ανπτύξουμε ένα υπολογιστικό σύστημα, το οποίο θα ελέγχει συσκευές του φυσικού κόσμου.

1.2.1 Γενικές πληροφορίες για το Arduino

Η συσκευή που κατασκευάστηκε στα πλαίσια της παρούσας εργασίας είναι βασισμένη στο μικροελεγκτή Arduino. Το Arduino, είναι μία ανοιχτού λογισμικού πλατφόρμα πρωτότυπων ηλεκτρονικών συσκευών που βασίζονται στην ευελιξία και στην ευκολία χρήσης υλικού και λογισμικού. Το Arduino, μπορεί να αλληλεπιδράσει με το περιβάλλον, κάνοντας λήψη σημάτων μέσα από μια ποικιλία αισθητήρων. Τα έργα που βασίζονται σε αυτό το μικροελεγκτή, μπορούν να είναι αυτόνομα ή να επικοινωνούν με το λογισμικό που τρέχει σε έναν υπολογιστή (π.χ. Flash, Processing, MaxMSP)



Εικόνα 1.1: Ο μικροελεγκτής Arduino Uno

1.2.2 Το software για τον προγραμματισμό του arduino










Το arduino IDE (Integrated Development Environment -Ολοκληρωμένο Περιβάλλον Ανάπτυξης) είναι ένα ειδικό πρόγραμμα βασισμένο σε java που εκτελείται στον υπολογιστή και επιτρέπει να γραφούν τα sketches για την πλακέτα arduino σε μία απλή γλώσσα που διαμορφώθηκε μέσα από την γλώσσα Processing. Πατώντας ένα κουμπί, το πρόγραμμα φορτώνει το sketch στον μικροελεγκτή του arduino. Η γλώσσα του arduino βασίζεται στη γλώσσα Wiring, μία παραλλαγή C/C++ για μικροελεγκτές αρχιτεκτονικής AVR όπως ο ATmega, και υποστηρίζει όλες τις βασικές δομές της C καθώς και μερικά χαρακτηριστικά της C++. Για compiler χρησιμοποιείται ο AVR gcc και ως βασική βιβλιοθήκη C χρησιμοποιείται η AVR libc. Λόγω της καταγωγής της από την γλώσσα προγραμματισμού C, στη γλώσσα του arduino μπορούν να χρησιμοποιηθούν ουσιαστικά οι ίδιες βασικές εντολές και συναρτήσεις, με την ίδια σύνταξη, οι ίδιοι τύποι δεδομένων και οι ίδιοι τελεστές με την C. Πέρα από αυτές όμως, υπάρχουν κάποιες ειδικές εντολές, συναρτήσεις και σταθερές που βοηθούν για την διαχείριση του ειδικού hardware του arduino. Τα βήματα με τα οποία μπορεί να προγραμματιστεί το arduino Duemilanove/UNO είναι:

1. Συνδέεται πρώτα η πλακέτα με τον υπολογιστή μέσω USB θύρας
2. Γράφεται ο κώδικας του sketch ο οποίος θα προγραμματίσει τον μικροελεγκτή ώστε να εκτελέσει τις επιθυμητές εργασίες
3. Φορτώνεται το sketch στον μικροελεγκτή του arduino μέσω της USB θύρας. Απαιτούνται μερικά δευτερόλεπτα έως ότου να τελειώσει ο μεταφορά και ο προγραμματισμός του μικροελεγκτή και στη συνέχεια το arduino κάνει μία επανεκκίνηση ώστε ο μικροελεγκτής να διαβάσει τον καινούριο κώδικα.
4. Η πλακέτα μετά την επανεκκίνηση μπορεί να εκτελέσει το καινούριο sketch το οποίο γράφτηκε και φορτώθηκε στον μικροελεγκτή.

1.2.3 Μοντέλα Arduino

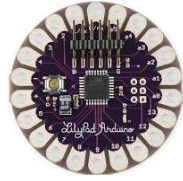
Στον **πίνακα 1** παρουσιάζονται τα πιο συνηθισμένα μοντέλα Arduino που κυκλοφορούν στο εμπόριο. Οι μεταξύ τους διαφορές βρίσκονται στη μνήμη, στη συχνότητα του μικροελεγκτή καθώς και στον αριθμό των εισόδων/εξόδων. Για την παρούσα διπλωματική εργασία, χρησιμοποιήθηκε το Arduino Mega 2560 καθώς από την ανάλυση των απαιτήσεων προέκυψε ότι η μνήμη και ο αριθμός των εισόδων/εξόδων του αρκούν για την υλοποίηση του κυκλώματος.

Πίνακας 1.1 : Μοντέλα Arduino

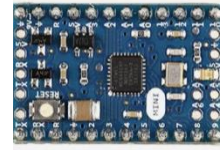
		
Arduino Uno	Arduino Leonardo	Arduino Due
		
Arduino Yun	Arduino Micro	Arduino Robot
		
Arduino Esplora	Arduino Mega Adk	Arduino Mega 2560



Arduino Ethernet



Arduino LilyPad



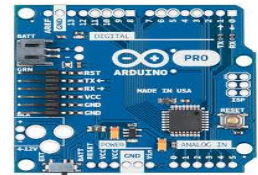
Arduino Mini



Arduino Nano



Arduino Fio








Arduino Pro

1.2.4 Arduino Shields

Τα shields, είναι πλακέτες που τοποθετούνται ακριβώς πάνω από το μικροελεγκτή Arduino και του προσδίδουν νέες δυνατότητες. Για τις ανάγκες της διπλωματικής, χρησιμοποιήθηκε ένα Ethernet shield για την αποστολή των μετρήσεων στο διακομιστή μέσω του τοπικού δικτύου. Στον πίνακα 2 παρουσιάζονται κάποια από τα πιο διαδεδομένα shields.

Πίνακας 1.2: Arduino Shields

<p>Arduino Ethernet shield</p>		<p>Συνδέει το Arduino στο διαδίκτυο με ένα καλώδιο τύπου RJ45.</p>
<p>Arduino WiFi shield</p>		<p>Συνδέει το Arduino στο διαδίκτυο ασύρματα.</p>
<p>Arduino Motor shield</p>		<p>Επιτρέπει την οδήγηση δύο DC κινητήρων από την ίδια συσκευή, ελέγχοντας την ταχύτητα και την κατεύθυνση του καθενός ξεχωριστά.</p>
<p>Arduino GPRS shield</p>		<p>Επιτρέπει τη σύνδεση του Arduino στο δίκτυο κινητής τηλεφωνίας GSM.</p>
<p>Arduino GPS shield</p>		<p>Επιτρέπει τη λήψη πληροφοριών γεωγραφικού μήκους και πλάτους.</p>

1.3) Δυνατότητες και πλεονεκτήματα Arduino

1) Το Χαμηλό κόστος. Το κόστος μιας πλακέτας ανέρχεται σε μερικά ευρώ. Πιο συγκεκριμένα το Arduino υπο που είναι η ναυαρχίδα της Arduino μπορεί να αγοραστεί μέσω ίντερνετ για 10-15€. Επίσης υπάρχουν πακέτα ξεκινήματος τα οποία ξεκινούν από 30 € και τα οποία εκτός από την πλακέτα προμηθεύουν μια γκάμα απο ηλεκτρονικά(καλώδια, αισθητήρες, ηλεκτροκινητήρες, led..) για τις πρώτες δόκιμες κατασκευές.

2) Είναι Ανεξαρτήτου πλατφόρμας (cross-platform). Το πρόγραμμα του arduino εκτελείται και στα τρία λειτουργικά συστήματα (windows,macintosh,linux) αγκαλιάζοντας όλο το εύρος των χρηστών προσωπικών υπολογιστών

3) Η Απλότητα. Ίσως το πιο σημαντικό πλεονέκτημα του arduino είναι η απλότητα του. Μέσα σε λίγες ώρες ο άπειρος χρήστης μπορεί να δημιουργήσει την πρώτη του κατασκευή. Αποτελεί ιδανικό δημιουργικό εργαλείο για την απόκτηση ηλεκτρονικών και μηχανικών δεξιοτήτων καθώς επίσης και δημιουργική απασχόληση για τους εφήβους.

4) Οι εκδόσεις του Arduino. Η arduino είναι μια οικογένεια από πλακέτες μικροελεγχτών που σκοπό έχουν να κάνουν ευκολότερη την κατασκευή διαδραστικών αντικείμενων. Κάθε έκδοση καλύπτει διαφορετικές ανάγκες και έχει διαφορετικές δυνατότητες. Έκτος από τις διαφορετικές εκδόσεις υπάρχουν και οι πλακέτες επέκτασης (shield's) που έρχονται να ενισχύσουν και να δώσουν νέες δυνατότητες στις πλατφόρμες του arduino. Όπως γίνεται κατανοητό πάντα θα υπάρχουν διάφορες εκδόσεις του arduino που θα καλύπτουν την τρέχουσα τεχνολογία.

5) Η οικογενεια του Arduino. Ένα ακόμα από τα σημαντικότερα του πλεονεκτήματα είναι το πλήθος των ανθρώπων που ασχολούνται με κάθε τομέα του arduino (υλικό και λογισμικό). Έτσι υπάρχουν αμέτρητα forum και ιστοσελίδες που μπορούν να καθοδηγήσουν, βοηθήσουν,διδάξουν και εμπνεύσουν των κάθε χρήστη, έμπειρο ή όχι.

6) Το Ανοικτού κώδικα και επέκτασης λογισμικό. Ο καθένας μπορεί να βρει τον πηγαίο κωδικά, να τον μελετήσει και να τον τροποποιήσει σύμφωνα με τις ανάγκες του. Έμπειροι χρήστες μπορούν μέσα από τις βιβλιοθήκες τις C++ άλλα και μέσα από τις βιβλιοθήκες του arduino να γράψουν τον δικό τους κωδικά και να τον μοιραστούν. Ενώ αρχάριοι χρήστες μπορούν να χρησιμοποιήσουν αυτόν το κωδικά χωρίς να γνωρίζουν προγραμματισμό χαμηλού επίπεδου.

7) Το Ανοικτού κώδικα και επέκτασης υλικό. Τα σχέδια των πλατφορμών είναι ανοικτού κωδικά πράγμα που σημαίνει ότι έμπειροι χρήστες στην ηλεκτρονική μπορούν να επεκτείνουν και να αναβαθμίσουν τις πλατφόρμες. Ακόμα και άπειροι χρήστες μπορούν να κατασκευάσουν την breadboard εκδοσή εάν θέλουν να καταλάβουν πως λειτουργεί και να γλιτώσουν και λίγα χρήματα.

1.4) Περιγραφή του hardware που χρησιμοποιείται για την υλοποίηση του συστήματος.

1.4.1) Τεχνικά χαρακτηριστικά του Arduino Mega 2560 R3

Παρακάτω [Πίνακας 1.3] παρουσιάζονται τα χαρακτηριστικά του Arduino Mega 2560, με το οποίο υλοποιήθηκε η παρούσα διπλωματική εργασία.

Πίνακας 1.3: Χαρακτηριστικά Arduino Mega 2560

Μικροελεγκτής	ATmega2560
Τάση λειτουργίας	5V
Τάση εισόδου	7-12V
Όρια τάσης εισόδου	6-20V
Ψηφιακοί αποδέκτες I/O	54(15 PWM output)
Αναλογικοί ακροδέκτες εισόδου	16
Ισχύς συνεχόμενου ρεύματος ανά ακροδέκτη	20mA
Ισχύς συνεχόμενου ρεύματος για ακροδέκτη τάσης 3.3V	50mA
Μνήμη flash	256KB(ATmega2560)
Μνήμη EEPROM	4KB(ATmega328)
Μνήμη SRAM	8KB(ATmega328)
Ταχύτητα ρολογιού	16MHz

1.4.2) Μνήμη

Ο μικροελεγκτής ATmega2560 έχει τρεις ομάδες μνήμης. Διαθέτει flash memory, SRAM και EEPROM.

- **256KB μνήμης Flash**

8KB χρησιμοποιούνται από το firmware του Arduino που έχει εγκαταστήσει ο κατασκευαστής του. Το firmware είναι αναγκαίο για την εγκατάσταση προγραμμάτων (sketches) στο μικροελεγκτή μέσω της θύρας USB. Τα υπόλοιπα 248KB της μνήμης Flash χρησιμοποιούνται για την αποθήκευση αυτών ακριβώς των προγραμμάτων, αφού πρώτα μεταγλωττιστούν στον υπολογιστή. Η μνήμη Flash, δε χάνει τα περιεχόμενά της με απώλεια τροφοδοσίας ή επανεκκίνηση.

- **8KB μνήμης SRAM (static random access memory)**

Η ωφέλιμη μνήμη που μπορούν να χρησιμοποιήσουν τα προγράμματα για να αποθηκεύουν μεταβλητές, πίνακες και άλλα. Η μνήμη SRAM χάνει τα δεδομένα της όταν η τροφοδοσία στο Arduino διακοπεί ή πατηθεί το κουμπί επανεκκίνησης.

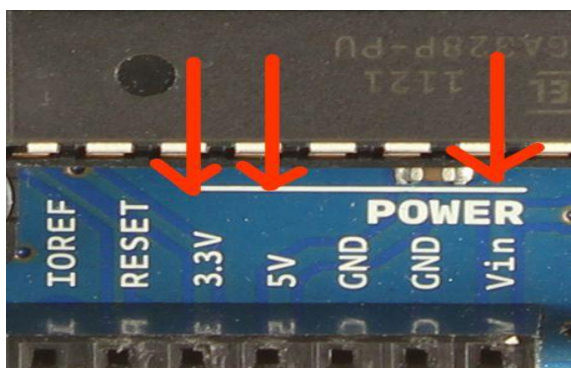
- **4KB μνήμης EEPROM**

Χρησιμοποιείται από τους προγραμματιστές για την αποθήκευση μακροχρόνιων πληροφοριών και για εγγραφή ή ανάγνωση δεδομένων από τα προγράμματα. Σε αντίθεση με την SRAM, δεν χάνει τα περιεχόμενά της με απώλεια τροφοδοσίας ή επανεκκίνησης.

1.4.3) Τροφοδοσία

Η τροφοδοσία του Arduino γίνεται είτε από τον υπολογιστή με ένα καλώδιο usb είτε από εξωτερική τάση που παρέχεται μέσω μιας υποδοχής φισ των 2.1mm. Για την αποφυγή προβλημάτων, η εξωτερική τροφοδοσία πρέπει να

είναι από 7 μέχρι 12V. Στην εικόνα 2 παρουσιάζονται οι είσοδοι και οι έξοδοι τροφοδοσίας του Arduino UNO.

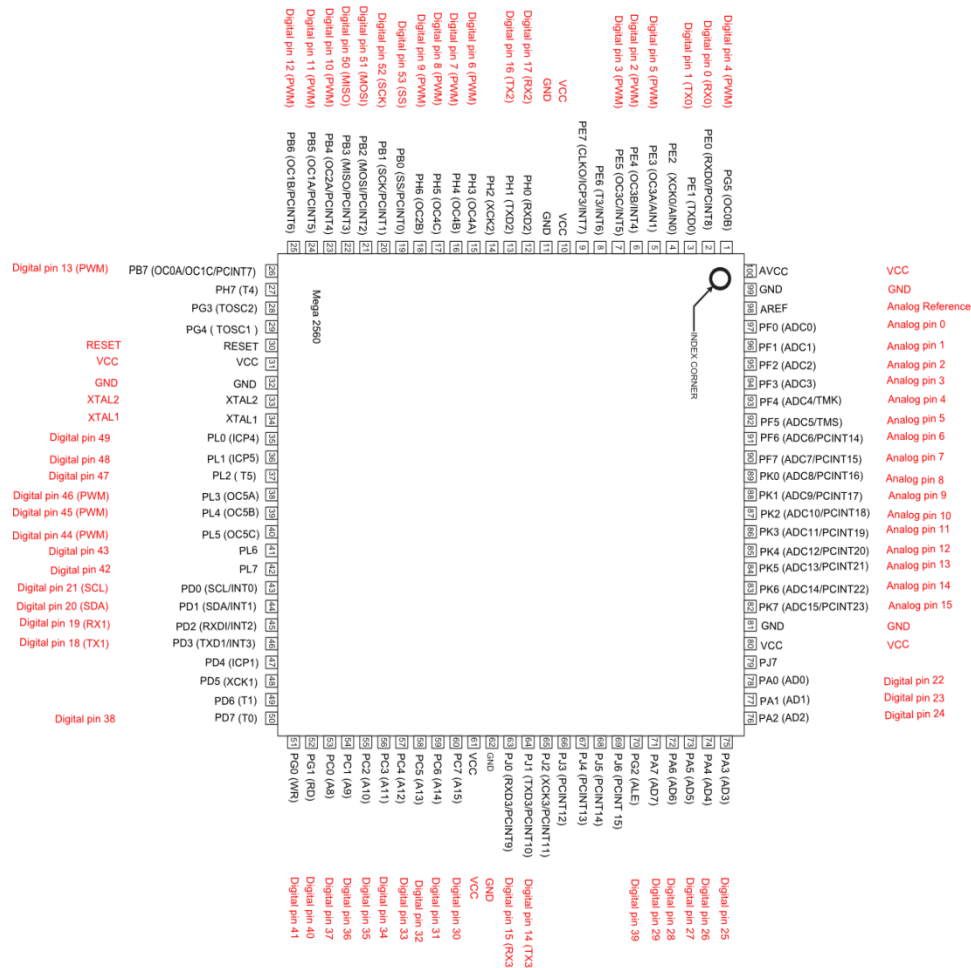


Εικόνα 1.2: Είσοδοι/έξοδοι τροφοδοσίας μικροελεγκτή arduino

Οι ακροδέκτες τροφοδοσίας είναι οι:

- **Vin** : Τάση εισόδου της πλακέτας όταν χρησιμοποιεί εξωτερική τροφοδοσία. Η τροφοδοσία της πλακέτας γίνεται μέσω αυτού του ακροδέκτη.
- **5V** : Τάση που χρησιμοποιείται από το μικροελεγκτή και από διάφορα μέρη της πλακέτας. Η τάση η οποία δίνει αυτός ο ακροδέκτης είναι είτε 5V από τη σύνδεση USB είτε η ρυθμισμένη τάση που δίνεται μέσω του Vin.
- **3.3** : Η τάση αυτή παράγεται από το ολοκληρωμένο FTDI. Το όριο άντλησης ρεύματος είναι 50mA.
- **GND** : Ακροδέκτης γείωσης.
- **IOREF** : Ο ακροδέκτης αυτός παρέχει στο board το κατάλληλο voltage reference για τη λειτουργία του microcontroller.
- **Reset**: όταν γειωθεί (με οποιοδήποτε GND) έχει ως αποτέλεσμα την επανεκκίνηση της υπολογιστικής πλατφόρμας Arduino.

1.4.4) Ακροδέκτες



Εικόνα 1.3: Digital and analog pins

1.4.4.1) Ακροδέκτες DIGITAL IN/OUT

- Τα **digital pin 0** και **1**: λειτουργούν ως RX και TX της σειριακής όταν το πρόγραμμά σας ενεργοποιεί την σειριακή θύρα. Έτσι, όταν λόγω χάρη το πρόγραμμά σας στέλνει δεδομένα στην σειριακή, αυτά προωθούνται και στην θύρα USB μέσω του ελεγκτή Serial-Over-USB αλλά και στο pin 0 για να τα διαβάσει ενδεχομένως μια άλλη συσκευή (π.χ. ένα δεύτερο Arduino στο δικό του pin 1). Αυτό

φυσικά σημαίνει ότι αν στο πρόγραμμά σας ενεργοποιήσετε το σειριακό interface, χάνετε 2 ψηφιακές εισόδους/εξόδους.

- **Τα digital pin 2 και 3:** λειτουργούν και ως εξωτερικά interrupt (interrupt 0 και 1 αντίστοιχα). Με άλλα λόγια, μπορείτε να τα ρυθμίσετε μέσα από το πρόγραμμά σας ώστε να λειτουργούν αποκλειστικά ως ψηφιακές εισοδοι στις οποίες όταν συμβαίνουν συγκεκριμένες αλλαγές, η κανονική ροή του προγράμματος σταματάει *άμεσα* και εκτελείται μια συγκεκριμένη συνάρτηση. Τα εξωτερικά interrupt είναι ιδιαίτερα χρήσιμα σε εφαρμογές που απαιτούν συγχρονισμό μεγάλης ακρίβειας.
- **Τα digital pin 2 έως 13 και 44 έως 46:** μπορούν να λειτουργήσουν και ως ψευδο-αναλογικές έξοδοι με το σύστημα PWM(Pulse Width Modulation). Έτσι έχουμε 8bit PWM output μέσω της analogWrite() function και μπορούμε αντί να έχουμε απλά την δυνατότητα HIGH-LOW, που παρέχουν οι υπόλοιπες ψηφιακές έξοδοι, να έχουμε ενδιάμεσες τιμές, μεταξύ 0 και 5 volt. Αυτό επιτυγχάνεται μέσω του PWM, το οποίο παρέχει ένα παλμό που εναλλάσσεται με κάποια συχνότητα και για ίσους χρόνους μεταξύ των τιμών 0 και 5V.
- **Τα digital pin 50(MISO), 51(MOSI), 52(SCK), 53(SS):** παρέχουν SPI communication, με τη χρήση της SPI library.
- **Στο digital pin 13:** υπάρχει ένα ενσωματωμένο LED, το οποίο μας δείχνει την κατάσταση του pin, γηλαδή ότα αυτό βρίσκεται σε τιμή HIGH το ενσωματωμένο LED είναι αναμμένο, ενώ όταν το pin είναι σε LOW το LED είναι σβηστό.
- **AREF PIN:** Το Arduino Mega έχει 16 αναλογικές εισόδους(A0 έως A15), όπου είναι προεπιλεγμένες να μετράνε από 0 έως 5 Volts. Μέσω του pin αυτού, έχουμε τη δυνατότητα να αλλάζουμε το

Reference Voltage των αναλογικών εισόδων, μέσω της εντολής `analogReference()`.

- **12C:** Τα digital pin 20(SDA) και 21 (SCL) έχουν προστεθεί για υποστήριξη TWI(Two-Wire-Interface) communication χρησιμοποιώντας την Wire library. Στο Arduino Mega ADK rev3 board η βιβλιοθήκη αυτή επιτρέπει την επικοινωνία με 12C / TWI devices μέσω των pins SDA (data line), SCL(clock line) και GND.
- **USB Host MAX3421E:** επικοινωνεί με το arduino μέσω του SPI bus χρησιμοποιώντας τα digital pins 7(RST), 50(MOSI), 52(SCK). Επειδή το digital pin 7 χρησιμοποιείται για την επικοινωνία με το MAX3421E, δεν πρέπει να προγραμματιστεί ως είσοδος ή έξοδος.

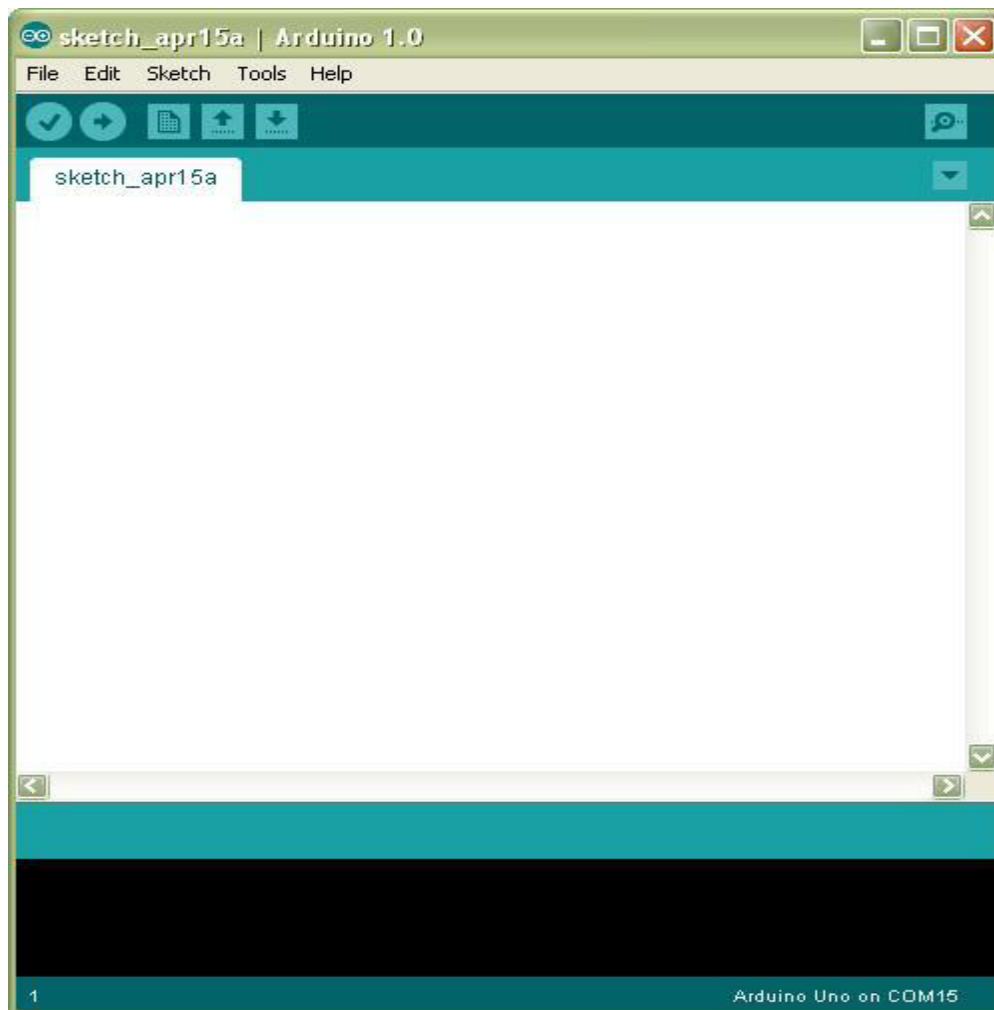
1.4.4.2) Ακροδέκτες ANALOG IN

Δίπλα από τους ακροδέκτες Power του Arduino, υπάρχουν οι ακροδέκτες A0 έως A15 με τη σήμανση ANALOG IN, όπως φαίνεται και στην εικόνα 3. Η τάση αναφοράς(Reference Voltage) για τις αναλογικές εισόδους μπορεί να ρυθμιστεί με τη χρήση εντολών, αφού πρώτα τροφοδοτήσουμε εξωτερικά με τάση το pin AREF, που βρίσκεται στην απέναντι πλευρά της πλακέτας.

1.5) Ολοκληρωμένο Περιβάλλον Ανάπτυξης του Arduino (Λογισμικό IDE)

Το περιβάλλον ανάπτυξης Arduino περιέχει μια περιοχή επεξεργασίας κειμένου για τη συγγραφή κώδικα, μια περιοχή μηνυμάτων, ένα μενού, μια γραμμή εργαλείων με κουμπιά για κοινές λειτουργίες, καθώς και μια σειρά από μενού. Συνδέεται με το υλικό Arduino για τη φόρτωση προγραμμάτων και για να επικοινωνούν μεταξύ τους.

Ένα ολοκληρωμένο πρόγραμμα συνήθως ονομάζεται sketch. Αυτό το sketch είναι γραμμένο με το πρόγραμμα επεξεργασίας κειμένου. Έχει δυνατότητες για την αντιγραφή/επικόλληση και για την αναζήτηση/αντικατάσταση κειμένου. Η κονσόλα απεικονίζει την έξοδο του κειμένου από το περιβάλλον Arduino συμπεριλαμβάνοντας πλήρη μηνύματα λάθους και άλλες πληροφορίες. Τα κουμπιά της γραμμής εργαλείων επιτρέπουν τον έλεγχο και το ανέβασμα των προγραμμάτων, τη δημιουργία νέου sketch, το άνοιγμα και την αποθήκευση των sketch και άνοιγμα της σειριακής οθόνης.



Εικόνα 1.4: Ολοκληρωμένο περιβάλλον ανάπτυξης(IDE) Arduino.

Ένα τυπικό πρόγραμμα Arduino έχει την παρακάτω δομή:

```
//δήλωση μεταβλητών  
void setup ()  
{  
  //αρχικοποιήσεις
```









```

}
void loop ()
{
//Κώδικας
}

```

Υπάρχουν δυο ειδικές συναρτήσεις που είναι μέρος του κάθε sketch του Arduino οι οποίες είναι η setup() και η loop(). Η setup() καλείται μια φορά, όταν το sketch ξεκινά ή όποτε κάνει επαναφορά (reset) η πλατφόρμα Arduino. Κυρίως, σε αυτήν γίνονται οι αρχικοποιήσεις των μεταβλητών, η ρύθμιση της κατάστασης των ακίδων (pins) και η προετοιμασία των βιβλιοθηκών. Αντιθέτως, η συνάρτηση loop() καλείται ξανά και ξανά επιτρέποντας έτσι στο πρόγραμμα να ανταποκριθεί σε εξωτερικά ερεθίσματα. Και οι δυο συναρτήσεις πρέπει να περιλαμβάνονται στο sketch, ακόμα και αν δεν περιέχουν κάτι και να είναι κενές.

Πίνακας 5: Εργαλεία του IDE υπο μορφή κουμπιών.

 Verify	Ελέγχει για συντακτικά λάθη τον κώδικα
 Upload	Μεταγλωττίζει τον κώδικα και τον φορτώνει στο arduino. Αν δεν είναι συντακτικά σωστός δε μπορεί να γίνει η φόρτωση.
 New	Δημιουργεί ένα νέο sketch.
 Open	Παραθέτε ένα μενού με όλα τα sketch. Ενεργοποιώντας ένα από αυτά, θα ανοίξει αυτόματα στο τρέχον παράθυρο.
 Save	Αποθηκεύει ένα sketch.
 Serial Monitor	Ανοίγει τη σειριακή οθόνη, και μέσω αυτής παρακολουθείται η ανταλλαγή δεδομένων που γίνεται στη σειριακή θύρα.

1.5.1) Βασικές δομές και λειτουργίες προγραμματισμού

Παρακάτω, ακολουθούν μερικές από τις πιο βασικές δομές και λειτουργίες που μπορεί να αξιοποιηθεί ως εργαλεία κατά την συγγραφή ενός προγράμματος Arduino :

Δομές ελέγχου ροής

if (δομή ελέγχου μίας συνθήκης)
if ... else (δομή ελέγχου πολλαπλών συνθηκών)
for (δομή επαναληπτικού ελέγχου συνθήκης)
while (δομή επαναληπτικού ελέγχου συνθήκης)
do ... while (δομή επαναληπτικού ελέγχου συνθήκης)
switch ... case (δομή ελέγχου περιπτώσεων)
break (εντολή διακοπής μιας επαναληπτικής δομής)
continue (εντολή παράλειψης της τρέχουσας επανάληψης)
return (εντολή επιστροφής από μία συνάρτηση)
goto (εντολή μετάβασης σε κάποιο σημείο του κώδικα)

Αριθμητικοί τελεστές

= (τελεστής εκχώρησης)
+ (τελεστής πρόσθεσης)
- (τελεστής αφαίρεσης)
* (τελεστής πολλαπλασιασμού)
/ (τελεστής διαίρεσης)
% (τελεστής υπόλοιπου ακεραίας διαίρεσης)

Λογικοί τελεστές

&& (λογική σύζευξη)
|| (λογική διάζευξη)
! (λογική άρνηση)

Δυαδικοί τελεστές

& (δυαδική σύζευξη)
| (δυαδική διάζευξη)
^ (δυαδική αποκλειστική διάζευξη)
~ (δυαδική άρνηση)
<< (δυαδική αριστερή ολίσθηση)
>> (δυαδική δεξιά ολίσθηση)

Τελεστές αύξησης και μείωσης

++ (αύξηση κατά μία ακέραιη μονάδα)

-- (μείωση κατά μία ακέραιη μονάδα)

Σύνθετοι τελεστές

+=, -=, *=, /=, %= (σύνθετοι αριθμητικοί τελεστές)

&=, |=, ^=, ~=, <<=, >>= (σύνθετοι δυαδικοί τελεστές)

Τελεστές σύγκρισης

== (ισότητα)

!= (ανισότητα)

< (μικρότερο)

> (μεγαλύτερο)

<= (μικρότερο ή ίσο)

>= (μεγαλύτερο ή ίσο)

Τελεστές δεικτών

* (τελεστής απόκτησης περιεχομένου)

& (τελεστής απόκτησης διεύθυνσης)

Σταθερές

HIGH (τιμή υψηλής στάθμης για μία επαφή εισόδου ή εξόδου)

LOW (τιμή χαμηλής στάθμης για μία επαφή εισόδου ή εξόδου)

false (λογικό επίπεδο ψεύδους σε μία συνθήκη)

true (λογικό επίπεδο αλήθειας σε μία συνθήκη)

INPUT (χρησιμοποιείται για τον ορισμό μίας επαφής ως είσοδο)

OUTPUT (χρησιμοποιείται για τον ορισμό μίας επαφής ως έξοδο)

A0, ..., A5 (συμβολοσταθερές για τις αναλογικές επαφές εισόδου)

Τύποι δεδομένων

boolean (λογική δυαδική τιμή)
char (προσημασμένος χαρακτήρας 8 ψηφίων)
unsigned char (μη προσημασμένος χαρακτήρας 8 ψηφίων)
byte (μη προσημασμένος χαρακτήρας 8 ψηφίων)
int (προσημασμένος ακέραιος αριθμός 16 ψηφίων)
unsigned int (μη προσημασμένος ακέραιος αριθμός 16 ψηφίων)
word (μη προσημασμένος ακέραιος αριθμός 16 ψηφίων)
long (προσημασμένος ακέραιος αριθμός 32 ψηφίων)
unsigned long (μη προσημασμένος ακέραιος αριθμός 32 ψηφίων)
float, double (αριθμός κινητής υποδιαστολής απλής ακρίβειας)
String (αντικείμενο αλφαριθμητικού με χρήσιμες μεθόδους)
Ως αλφαριθμητικό μπορεί να θεωρηθεί και ο πίνακας χαρακτήρων

Συναρτήσεις μετατροπής τύπων

char(), byte()
int(), word(), long()
float(), double()

Συναρτήσεις εισόδου και εξόδου

pinMode() (ορίζει μια επαφή ως είσοδο ή έξοδο)

Συναρτήσεις ψηφιακής εισόδου και εξόδου

digitalWrite() (γράφει σε μία ψηφιακή επαφή εξόδου)
digitalRead() (διαβάζει από μία ψηφιακή επαφή εισόδου)

Συναρτήσεις αναλογικής εισόδου και εξόδου

analogReference() (ορίζει την τάση αναλογικής αναφοράς)
analogWrite() (γράφει PWM σήματα σε μία επαφή εξόδου)
analogRead() (διαβάζει από μία αναλογική επαφή εισόδου)

Προηγμένες συναρτήσεις εισόδου και εξόδου

tone() (παράγει ένα τετραγωνικό σήμα ορισμένης συχνότητας)
noTone() (διακόπτει την παραγωγή τετραγωνικών σημάτων)
shiftOut() (ολισθαίνει τα ψηφία μιας τιμής σε μία επαφή εξόδου)
pulseIn() (επιστρέφει την διάρκεια σε μς ενός παλμού HIGH ή LOW)

Συναρτήσεις χρόνου

millis() (διάρκεια εκτέλεσης του προγράμματος σε ms)
micros() (διάρκεια εκτέλεσης του προγράμματος σε μς)
delay() (παύση προγράμματος - η διάρκεια δίδεται σε ms)
delayMicroseconds() (παύση προγράμματος - η διάρκεια δίδεται σε μς)

Μαθηματικές και Τριγωνομετρικές συναρτήσεις

max() (βρίσκει τον μεγαλύτερο ανάμεσα σε δύο αριθμούς)
min() (βρίσκει τον μικρότερο ανάμεσα σε δύο αριθμούς)
abs() (επιστρέφει την απόλυτη τιμή ενός αριθμού)
constrain() (ελέγχει για υπερχείλιση ή υποχείλιση ορίων)
map() (πραγματοποιεί γραμμικό μετασχηματισμό ορίων)
pow() (επιστρέφει το αποτέλεσμα μίας δύναμης)

sqrt() (επιστρέφει την ρίζα ενός αριθμού)
sin() (υπολογίζει το ημίτονο ενός αριθμού)
cos() (υπολογίζει το συνημίτονο ενός αριθμού)
tan() (υπολογίζει την εφαπτομένη ενός αριθμού)

Συναρτήσεις γεννήτριας ψευδοτυχαίων αριθμών

random() (δίδεται ένας νέος αριθμός από την γεννήτρια)
randomSeed() (θέτει τον σπόρο της γεννήτριας παραγωγής)

Συναρτήσεις επεξεργασίας δυαδικών αριθμών

lowByte() (επιστρέφει το δεξιότερο byte μίας μεταβλητής)
highByte() (επιστρέφει το αριστερότερο byte μίας μεταβλητής)
bitRead() (διαβάζει ένα συγκεκριμένο ψηφίο μίας μεταβλητής)
bitWrite() (γράφει σε ένα συγκεκριμένο ψηφίο μιας μεταβλητής)
bitSet() (γράφει την τιμή 1 σε κάποιο ψηφίο μίας μεταβλητής)
bitClear() (γράφει την τιμή 0 σε κάποιο ψηφίο μιας μεταβλητής)
bit() (υπολογίζει μία συγκεκριμένη δύναμη με βάση το 2)

Συναρτήσεις χρήσης ρουτινών εξυπηρέτησης διακοπών

attachInterrupt() (ενεργοποιεί μία ρουτίνα εξυπηρέτησης διακοπής)
detachInterrupt() (απενεργοποιεί μία ρουτίνα εξυπηρέτησης διακοπής)

Συναρτήσεις ενεργοποίησης και απενεργοποίησης διακοπών

interrupts() (ενεργοποιεί τα σήματα διακοπής)
noInterrupts() (απενεργοποιεί τα σήματα διακοπής)

Υποστήριξη σειριακής επικοινωνίας

Serial (αντικείμενο σειριακής επικοινωνίας με χρήσιμες μεθόδους)

Κεφάλαιο 2- Υλικό μέρος του συστήματος

Η υλοποίηση της κατασκευής βασίστηκε στη πλακέτα ανάπτυξης **Arduino Mega 2560 R3**. Η αρχιτεκτονική arduino επιτρέπει τη σύνδεση διαφόρων εξαρτημάτων και αισθητήρων στη πλακέτα και τον έλεγχο τους μέσω κατάλληλου προγραμματισμού.

Για την παρούσα εργασία, χρησιμοποιήθηκαν :

- Η βασική πλακέτα arduino mega 2560 r3
- Ένα mp3 module(BY8001-16P)
- Ένα real time clock module(DS3231)
- Ένα TFT Display Mega Shield και μια οθόνη TFT 3.2" touch screen

- 2 ηχεία 0.5w ισχύος
- Μια κάρτα sd που περιέχει όλα τα αρχεία ήχου(σε μορφή mp3 ή wav)
- Μια αντίσταση 1kohm και καλώδια για τις επιμέρους συνδέσεις των εξαρτημάτων
- Arduino Mega Proto Shield Rev3(Διάτρητη πλακέτα συμβατή με το Arduino Mega 2560 R3)
- Ένα κολλητήρι(Soldering Iron) και το καλάι (solder wire), με το οποίο πραγματοποιήθηκαν οι κολλήσεις στη διάτρητη πλακέτα.

2.1 – TFT Display Mega Shield



Εικόνα 7: TFT Display Mega Shield

ΠΕΡΙΓΡΑΦΗ

Το **TFT Display Mega Shield** χρησιμοποιείται για να μπορέσουμε να συνδέσουμε την οθόνη touch screen πάνω στη πλακέτα **Arduino Mega** (είναι συμβατή με όλα τα μοντέλα του Arduino Mega). Τοποθετείται ακριβώς πάνω στη πλακέτα και όπως φαίνεται και στην εικόνα(7) πάνω του εφαρμόζει η οθόνη touch screen.

ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ

- Συμβατό με τάση λειτουργίας 3.3V/5V
- Συμβατό με τη βιβλιοθήκη UTFT
- Δέχεται sd card

ΠΡΟΔΙΑΓΡΑΦΕΣ

PCB size	79.3mm X 57.5mm X 1.6mm
Indicators	-
Power supply	Compatible with Arduino Mega
Communication Protocol	-
RoHS	Yes

Πίνακας 6: Προδιαγραφές TFT Display Mega Shield

ΗΛΕΚΤΡΙΚΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ

Specification	Min	Type	Max	Unit
Power Voltage	4.5	5	5.5	VDC
Input Voltage VH:	3	-	5.5	V
Input Voltage VL:	-0.3	0	0.5	V

Πίνακας 7: Ηλεκτρικά χαρακτηριστικά TFT Display Mega Shield

HARDWARE

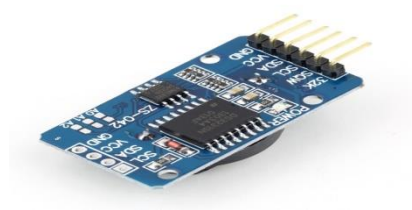
Pin map of Arduino TFT Display Mega Shield :

Arduino Mega PIN	Description
D2	T_IRQ
D3	T_DOUT
D4	T_DIN
D5	T_CS
D6	T_CLK
D22	DB8
D23	DB9
D24	DB10
D25	DB11
D26	DB12
D27	DB13
D28	DB14
D29	DB15

D30	DB7
D31	DB6
D32	DB5
D33	DB4
D34	DB3
D35	DB2
D36	DB1
D37	DB0
D38	RS
D39	WR
D40	CS
D41	RST
D50	SD_MISO
D51	SD_MOSI
D52	SD_SCK
D53	SD_NSS

Το Arduino TFT Display Mega Shield χρησιμοποιεί τον controller SSD1289, και υποστηρίζει 16bit data interface.

2.2 - Real Time Clock DS3231



Εικόνα 8: DS3231 Real Time Clock

ΠΕΡΙΓΡΑΦΗ

Το DS3231 είναι ένα χαμηλού κόστους, εξαιρετικά ακριβές I2C ρολόι πραγματικού χρόνου (RTC) με ενσωματωμένη θερμοκρασία κρυσταλλικός ταλαντωτής (TCXO) και κρύσταλλο.

Η συσκευή ενσωματώνει μια είσοδο μπαταρίας και διατηρεί ακριβή χρονομέτρηση όταν η κύρια τροφοδοσία της συσκευής διακόπτεται. Η ενσωμάτωση του κρυσταλλικού συντονιστή βελτιώνει επίσης τη μακροπρόθεσμη ακρίβεια της συσκευής καθώς μειώνει τον αριθμό των τεμαχίων σε μια γραμμή παραγωγής. Το DS3231

προσφέρεται σε ένα 16-pin, 300-mil SO πακέτο. Το RTC διατηρεί δευτερόλεπτα,

λεπτά, ώρες, ημέρα, ημερομηνία, μηνών και έτους. Η ημερομηνία στο τέλος του μήνα

ρυθμίζεται αυτόματα για μήνες με λιγότερους από τις 31 ημέρες,

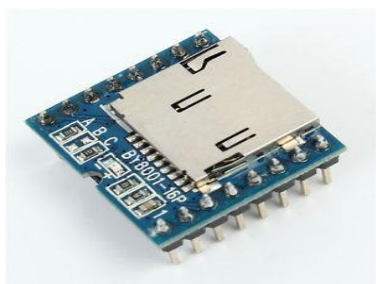
συμπεριλαμβανομένων των διορθώσεων για το έτος άλμα. οΤο ρολόι λειτουργεί είτε

σε 24ωρη είτε σε 12ωρη μορφή με δείκτη AM / PM.

ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ

- Εξαιρετικά ακριβές RTC τα διαχειρίζεται πλήρως όλα
- **Λειτουργίες χρονομέτρησης:**
 1. Το ρολόι πραγματικού χρόνου μετρά δευτερόλεπτα, λεπτά, ώρες, ημερομηνία του μήνα, μήνας, ημέρα της εβδομάδας και έτος, με την αποζημίωση Leap-Year Valid μέχρι 2100.
 2. Ακρίβεια $\pm 2\text{ppm}$ από 0°C έως $+40^\circ\text{C}$
 3. Ακρίβεια $\pm 3,5\text{ppm}$ από -40°C έως $+85^\circ\text{C}$
 4. Έξοδος ψηφιακού αισθητήρα θερμοκρασίας: $\pm 3^\circ\text{C}$ Ακρίβεια
 5. Εγγραφείτε για την επιλογή Trimming Aging
 6. Έξοδος RST / Πλήκτρο επανεκκίνησης εισόδου Debounce
 7. Δύο συναγερμούς ώρας της ημέρας
 8. Προγραμματιζόμενο σήμα εξόδου τετραγώνου κύματος
- Η απλή σειριακή διεπαφή συνδέεται με τους περισσότερους μικροελεγκτές:
 - Γρήγορη διεπαφή I2C (400kHz)
- Είσοδος αντιγράφων ασφαλείας μπαταρίας για συνεχή χρονική διάρκεια:
 1. Λειτουργία χαμηλής κατανάλωσης ρεύματος επεκτείνει την επαναφορά της μπαταρίας, Ωρα εκτέλεσης.
 2. Λειτουργία 3.3V
- Εύρος θερμοκρασίας λειτουργίας:
 - Εμπορική: (0°C έως $+70^\circ\text{C}$) και Βιομηχανική (-40°C έως $+85^\circ\text{C}$)

2.3 - Mp3 Module(BY8001-16P)



Εικόνα 9 : mp3 module(BY8001-16p)

ΠΕΡΙΓΡΑΦΗ

Το BY8001-16P είναι μια συμπαγής, υψηλής ποιότητας μονάδα MP3. Το βασικό τσιπ του BY8001 (SSOP24) υποστηρίζει αποκωδικοποίηση μορφών MP3 και WAV. Το εξάρτημα διαθέτει επίσης ενσωματωμένη θήκη κάρτας TF και υποστηρίζει αντικατάσταση του περιεχομένου ήχου από εξωτερικό USB stick χρησιμοποιώντας καλώδιο δεδομένων USB. Το εξάρτημα διαθέτει επίσης ενσωματωμένο 3w ενισχυτή ισχύος που μπορεί να οδηγήσει απευθείας ένα απλό ηχείο 3W.

ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ

- Υποστηρίζει αρχεία μορφής ήχου υψηλής ποιότητας MP3 και WAV.
- 24 DAC Δυναμικό εύρος εξόδου 90dB και SNR άνω των 85dB.
- Υποστηρίζει UART Ασύγχρονη ρύθμιση σειριακής θύρας για αναπαραγωγή, παύση, επόμενο κομμάτι, επίπεδα έντασης ενισχυτή, επιλογές κομματιών, διακοπές για διαφημιστικές ανακοινώσεις κλπ.
- Ενσωματωμένη ένταση ήχου, κομμάτι, Ισοσταθμιστής φάσματος, απενεργοποίηση αποθηκευμένων λειτουργιών μνήμης.
- Υποδοχή κάρτας TF (Micro SD) υποστηρίζει μνήμη μέχρι 32GB, αφαιρούμενη κάρτα TF για την αλλαγή ηχητικού περιεχόμενου.
- Υποστηρίζει ανάγνωση USB flash drives μέχρι 32GB ως αντικατάσταση για περιεχόμενο TF μέσω δεδομένων USB καλώδιου.

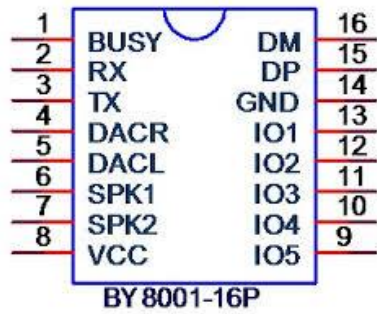
- Ενσωματωμένος ενισχυτής ισχύος 3W κατευθύνει άμεσα εξωτερικό μονοφωνικό ηχείο. οι πελάτες μπορούν επίσης να προσθέσουν 2-καναλιών ενισχυτή ισχύος για στερεοφωνική αναπαραγωγή.
- Τυπική απόσταση 2,54 χιλιοστών εντός πλαισίου, συμπαγές πακέτο DIP16 .

ΤΕΧΝΙΚΕΣ ΠΡΟΔΙΑΓΡΑΦΕΣ

Design Specification	Parameters
MP3 And WAV file formats	Supported sampling rates 8 - 48K and Bit rates 8 - 320Kbps audio files
UART Interface	Standard serial port, 3.3V, TTL-level, baud rate 9600
Input voltage	3.6V-5V (Typical 4.2V)
Quiescent current	16MA (in Standby mode)
Amplifier power	3W/4ohm or 2W/8ohm speaker (module's amplifier chip gets rather hot driving 4 ohm speaker over very short period, 8 ohm speaker is recommended)
Dimensions	22mm*21mm
Working temperature	-40°C - 70°C
Humidity	5% - 95%

Πίνακας 8: Προδιαγραφές Mp3 Module(DS3231)

Module Pin Mapping



Εικόνα 10: Mp3 Module(DS3231)

Pin Number	Pin Name	Functional Description	Notes
1	BUSY	Playback output high, stopping for low	Busy signal
2	RX	UART Asynchronous serial data input	3.3V TTL leve
3	TX	UART Asynchronous serial data output	3.3V TTL leve
4	DACR	DAC Right channel output	External amplifier, headphone
5	DACL	DAC The left channel output	External amplifier, headphone
6	SPK1	External monaural speaker	3W/4 ω 2W/8 ω passive loudspeaker
7	SPK2	External monaural speaker	3W/4 ω 2W/8 ω passive loudspeaker
8	VCC	Positive power supply	3.6-5V
9	IO5	INPUT 5	Key/Button function trigger
10	IO4	INPUT 4	Key/Button function

			trigger
11	IO3	INPUT 3	Key/Button function trigger
12	IO2	INPUT 2	Key/Button function trigger
13	IO1	INPUT 1	Key/Button function trigger
14	GND	Power common ground	System Ground
15	DP	USB Data cable	USB Flash Drive
16	DM	USB Data cable	USB Flash Drive

I/O ΠΕΡΙΓΡΑΦΗ ΘΥΡΩΝ

I/O Pin	Function	I/O Pin	Function
IO1	Play track 1	IO1-IO5	Play track 9
IO2	Play track 2	IO2-IO3	Play track 10
IO3	Play track 3	IO2-IO4	Play track 11
IO4	Play track 4	IO2-IO5	Play track 12
IO5	Play track 5	IO3-IO4	Play track 13
IO1-IO2	Play track 6	IO3-IO5	Play track 14
IO1-IO3	Play track 7	IO4-IO5	Play track 15
IO1-IO4	Play track 8		

2.4 – TFT 3.2” TOUCH SCREEN

ΠΡΟΔΙΑΓΡΑΦΕΣ

- 3.2" έγχρωμη TFT LCD
- Ανάλυση 320 x 240
- 65,536 χρώματα (16 Bit)
- Ισχυρός μικροελεγκτής 16 Bit (SSD1289)
- Built-in video RAM buffer
- Integrated Resistive Touchscreen
- Integrated SD card
- 5V τάση λειτουργίας
- LED Backlight

LCD1		TFT_LCD	
LCD CS	1	LCD_CS	RS
LCD WR	3	WR/CLK	RD
LCD RST	5	RST	DB1
DB2	7	DB2	DB3
DB4	9	DB4	DB5
DB6	11	DB6	DB7
DB8	13	DB8	DB10
DB11	15	DB11	DB12
DB13	17	DB13	DB14
DB15	19	DB15	DB16
DB17	21	DB17	GND
BL_CTR	23	BL	VDD3.3
VCC	25	VDD3.3	GND
GND	27	GND	BL_VDD
SPI MISO	29	MISO	MOSI
T_PEN	31	T_PEN	MO
T_CS	33	T_CS	CLK
			2
			LCD RS
			4
			LCD RD
			6
			DB1
			8
			DB3
			10
			DB5
			12
			DB7
			14
			DB10
			16
			DB12
			18
			DB14
			20
			DB16
			22
			SD_CS
			24
			VCC
			26
			GND
			28
			BL_VDD
			30
			SPI MOSI
			32
			FLASH_CS
			34
			SPI_CLK

Εικόνα 11: Επεξήγηση των Pin

2.5-Arduino Mega Proto Shield Rev3

ΠΕΡΙΓΡΑΦΗ

Το Arduino Prototyping Shield μας διευκολύνει να σχεδιάσουμε προσαρμοσμένα κυκλώματα. Μπορούμε να κολλήσουμε εξαρτήματα στην περιοχή του πρωτοτύπου για να δημιουργήσουμε το έργο μας ή να το χρησιμοποιήσουμε με ένα μικρό συγκολλητικό breadboard(που δεν περιλαμβάνεται) για να δοκιμάσουμε γρήγορα τις ιδέες του κυκλώματος χωρίς να χρειάζεται να τις κολλήσουμε. Έχει επιπλέον συνδέσεις για όλους τους ακροδέκτες I / O Arduino MEGA και έχει χώρο για να τοποθετήσει ολοκληρωμένα κυκλώματα διαμπερούς και επιφανειακής στήριξης. Είναι ένας βολικός τρόπος να φτιάξουμε το κύκλωμά του Arduino σε μια ενιαία μονάδα.

Τεχνικά Χαρακτηριστικά

PCB Size	101.5 x 53.3 mm
Weight	0.013 Kg

Τα χαρακτηριστικά του board έχουν ως εξής:

- 1.0 Arduino pinout
- 1 Reset button
- 1 ICSP connector
- 14 pins SMD footprint (50 mils pitch)
- 32 double row through Hole pads, standard Arduino breakout layout
- Proto area with multiple THT pads, 100 mils pitch

Τροφοδοσία

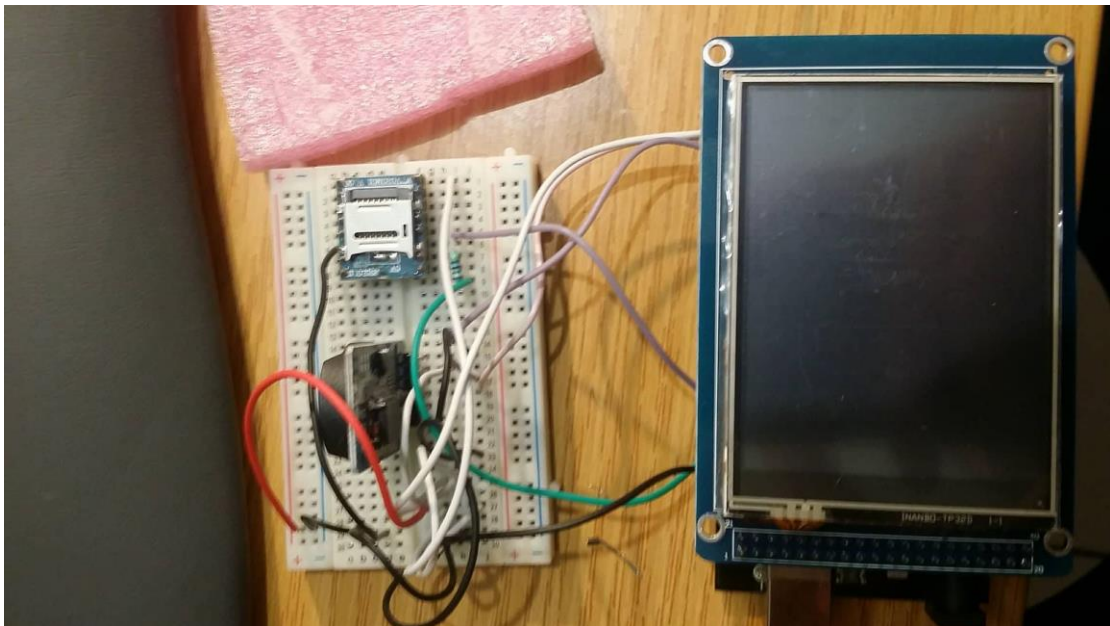
Το Proto Shield φέρνει την ισχύ από τις καρφίτσες Arduino 5V και GND στις δύο σειρές διαύλων ισχύος που τοποθετούνται ανάμεσα στο αποτύπωμα του πακέτου Through Hole, το οποίο μπορεί να χρησιμοποιηθεί για την τροφοδοσία των υποδοχών DIP ή για τις γραμμές ισχύος και γείωσης.

Φυσικά Χαρακτηριστικά

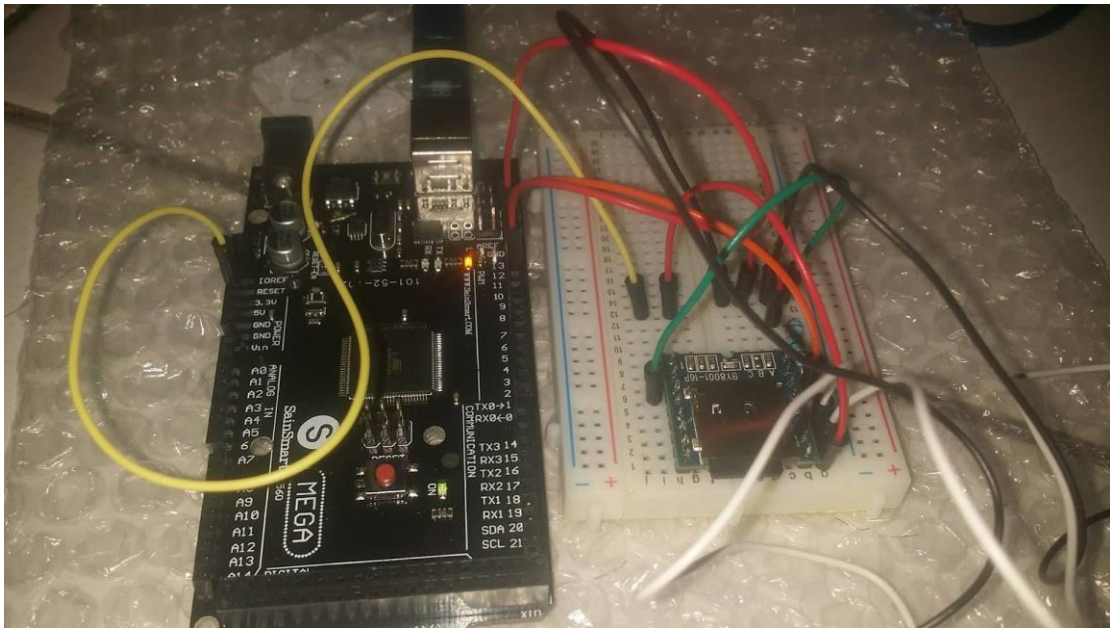
Το μέγιστο μήκος και το πλάτος του PCI Proto Shield είναι 2.7 και 2.1 ίντσες αντίστοιχα. Οι τρεις οπές των βιδών επιτρέπουν την προσάρτηση της θωράκισης σε επιφάνεια ή θήκη. Σημειώστε ότι η απόσταση μεταξύ των ψηφιακών ακίδων 7 και 8 είναι 160 χιλιοστά(0.16"), όχι ένα ομοιόμορφο πολλαπλάσιο της απόστασης των 100 μιλίων των άλλων ακίδων.

Κεφάλαιο 3- Κώδικας του πειράματος και κάποιες φωτογραφίες(πειραματικό στάδιο)

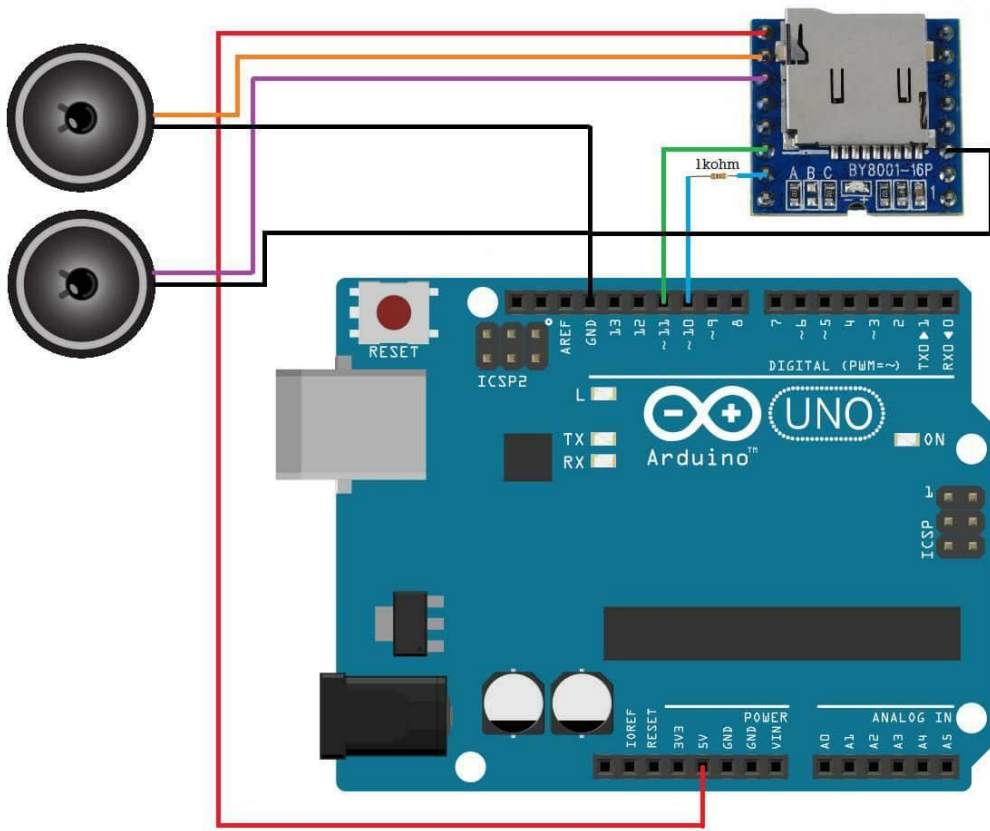
Φωτογραφίες (πειραματικού σταδίου)



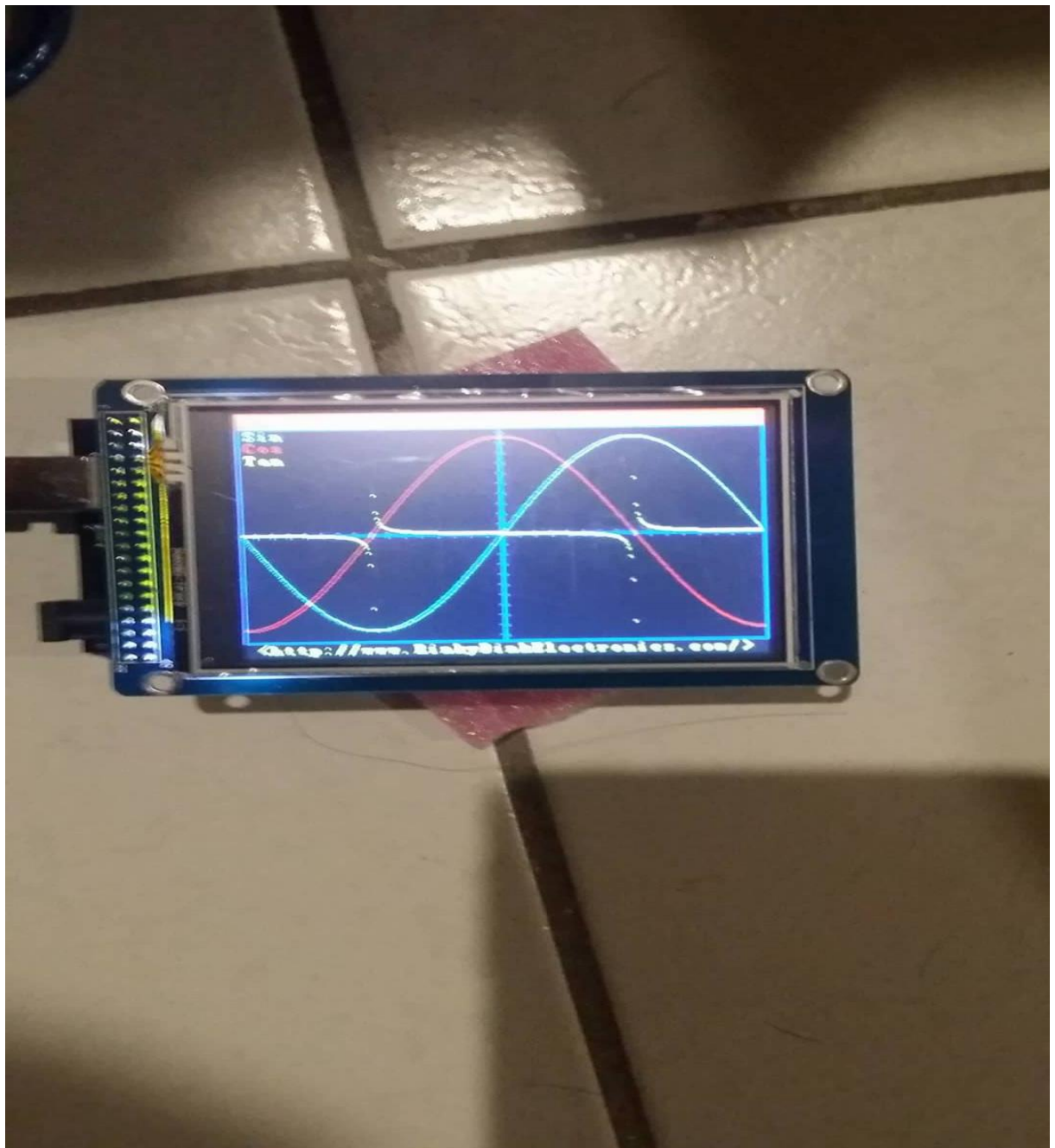
Εικόνα 12: Δοκιμή κατασκευής σε breadboard



Εικόνα 11: Δοκιμή Mp3 module



Εικόνα 13: Δοκιμή Mp3 module



Εικόνα 14: Δοκιμή λειτουργίας οθόνης TFT

Εικόνες μετά την ολοκλήρωση της κατασκευής



Εικόνα 15:Λειτουργία mp3



Εικόνα 16: Αρχική οθόνη Mp3 and alarm clock



Εικόνα 17: Ρυθμίσεις για το ξυπνητήρι



Εικόνα 18: Οθόνη όταν χτυπάει το ξυπνητήρι

Παρακάτω ακολουθεί ο κώδικας που χρησιμοποιήθηκε για την περαίωση του πειράματος, βήμα-βήμα, από την αρχή μέχρι το τέλος.

```
1. #include <UTFT.h>
2. #include <URTouch.h>
3. #include <SoftwareSerial.h>
4. #include <BY8001.h>
5. #include <DS3231.h>
6.
7. //==== Δημιουργία Αντικειμένων
8. UTFT myGLCD(SSD1289, 38, 39, 40, 41); //Παράμετροι για τη ορθή
   λειτουργία των εξαρτημάτων. (Δίδονται από τον κατασκευαστή)
9. URTouch myTouch( 6, 5, 4, 3, 2);
10.
11.
12. SoftwareSerial mp3Serial(11, 10); // RX, TX
13. BY8001 mp3; // Δημιουργία κλάσης BY8001 που την ονομάζουμε 'mp3'
14.
15. DS3231 rtc(SDA, SCL);
16.
17. //==== Ορισμός Γραματοσειρών
18. extern uint8_t SmallFont[];
19. extern uint8_t BigFont[];
20. extern uint8_t SevenSegNumFont[];
21.
22. extern unsigned int MusicPlayerButton[0x1040]; ; // Δήλωση κουμπιών σαν
   παράμετροι.
23. extern unsigned int AlarmButton[0x1040];
24. extern unsigned int ButtonPlay[0x1AE9];
25. extern unsigned int ButtonPause[0x1AE9];
26. extern unsigned int PreviousButton[0x9C4];
27. extern unsigned int NextButton[0x9C4];
28. extern unsigned int VolumeDown[0x170];
29. extern unsigned int VolumeUp[0x3B8];
30.
31. int x, y; // Παράμετροι για τον «εντοπισμό» του αγγίγματος στην οθόνη.
32. char currentPage, playStatus; //Δήλωση παραμέτρων για την αρίθμηση των
   τραγουδιών, τα μέρη της ώρας, τη θερμοκρασία κλπ.
33.
34. int iV = 15;
35. int trackNum = 1;
36. int b = 16;
37. int aHours = 0;
38. int aMinutes = 0;
39. boolean alarmNotSet = true;
40. String alarmString = "";
```

```

41. float currentTemperature, temperature;
42. static word totalTime, elapsedTime, playback, minutes, seconds, lastSeconds,
    minutesR, secondsR;
43. String currentClock, currentHours, currentMinutes, currentSeconds,
    currentDate;
44. String timeString, hoursString, minutesString, secondsString, hoursS,
    minutesS, secondsS, dateS;

```

```

1. void setup() {
2. // Απεικόνιση.
3. myGLCD.InitLCD();
4. myGLCD.clrScr();
5. myTouch.InitTouch();
6. myTouch.setPrecision(PREC_MEDIUM);
7.
8. // Αρχικοποίηση αντικειμένου rtc.
9. rtc.begin();
10. rtc.setTime(00, 00, 00); // Ρύθμιση ώρας
11. rtc.setDate(00, 00, 0000); // Ρύθμιση ημερομηνίας
12. // Μουσική.
13. Serial.begin(9600); // Ρύθμιση αντικειμένων σχετικών με την αναπαραγωγή
    μουσικής.
14. mp3Serial.begin(9600);
15. mp3.setup(mp3Serial);
16. delay(800); //
17.
18. drawHomeScreen(); // Κλήση διαδικασίας υπευθυνης για την απεικόνιση
    όλων των γραφικών στην αρχική οθόνη.
19. currentPage = '0'; // Δηλώνει ότι βρισκόμαστε στην αρχική οθόνη.
20. playStatus = '0';
21.
22. mp3.setVolume(15);
23. delay(100);
24. currentTemperature = rtc.getTemp(); // Κλήση διαδικασιών για
    ώρα, ημερομηνία, θερμοκρασία.
25. currentDate = rtc.getDateStr();
26. currentClock = rtc.getTimeStr();
27. timeString = rtc.getTimeStr();
28. currentHours = timeString.substring(0, 2);
29. currentMinutes = timeString.substring(3, 5);
30. currentSeconds = timeString.substring(6, 8);
31. }

```

```

1. void loop() {

```



```

2. // Αρχική Οθόνη.
3. if (currentPage == '0') {
4. // Έλεγχος για αλλαγές στην ώρα.
5. if ( currentClock != rtc.getTimeStr() ) {
6. timeString = rtc.getTimeStr();
7. hoursS = timeString.substring(0, 2);
8. minutesS = timeString.substring(3, 5);
9. secondsS = timeString.substring(6, 8);
10.
11. myGLCD.setFont(SevenSegNumFont);
12. myGLCD.setColor(0, 255, 0);
13.
14. myGLCD.print(secondsS, 224, 50);
15.
16. if ( currentMinutes != minutesS ) {
17. myGLCD.print(minutesS, 128, 50);
18. currentMinutes = minutesS;
19. }
20. if ( currentHours != hoursS ) {
21. myGLCD.print(hoursS, 32, 50);
22. currentHours = hoursS;
23. }
24. // Έλεγχος για αλλαγές στην ημερομηνία.
25. dateS = rtc.getDateStr();
26. delay(10);
27. if ( currentDate != dateS){
28. myGLCD.setColor(255, 255, 255); // Θέτει λευκό χρώμα.
29. myGLCD.setFont(BigFont); // Θέτει μεγάλο μέγεθος γραμματοσειράς.
30. myGLCD.print(rtc.getDateStr(), 153, 7);
31. }
32. // Έλεγχος για αλλαγές στη θερμοκρασία.
33. temperature = rtc.getTemp();
34. delay(10);
35. if ( currentTemperature != temperature ){
36. myGLCD.setColor(255, 255, 255); // Θέτει λευκό χρώμα.
37. myGLCD.setFont(BigFont); // Θέτει μεγάλο μέγεθος γραμματοσειράς.
38. myGLCD.printNumI(temperature, 39, 7);
39. currentTemperature = temperature;
40. }
41. delay(10);
42. currentClock = rtc.getTimeStr();

```

```

1. // Έλεγχος για το αν έχει «αγγιχθεί» η οθόνη.
2. if (myTouch.dataAvailable()) {
3. myTouch.read();
4. x = myTouch.getX(); // Συντεταγμένη X όπου έχει «αγγιχθεί η οθόνη».
5. y = myTouch.getY(); // Συντεταγμένη Y όπου έχει «αγγιχθεί η οθόνη»
6. // Έλεγχος για πάτημα κουμπιού αναπαραγωγής μουσικής.
7. if ((x >= 55) && (x <= 120) && (y >= 125) && (y <= 190)) {
8. drawFrame(87, 157, 33);
9. currentPage = '1';
10. myGLCD.clrScr();
11. delay(100);
12. drawMusicPlayerScreen();
13. delay(100);
14. }
15. // Έλεγχος για πάτημα κουμπιού για το ξυπνητήρι.
16. if ((x >= 195) && (x <= 260) && (y >= 125) && (y <= 190)) {
17. drawFrame(227, 160, 29);
18. currentPage = '2';
19. myGLCD.clrScr();
20. }
21. }

```

```

1. // Οθόνη αναπαραγωγής μουσικής.
2. if (currentPage == '1') {
3. if (myTouch.dataAvailable()) {
4. myTouch.read();
5. x = myTouch.getX(); // Συντεταγμένη X όπου έχει «αγγιχθεί» η οθόνη.
6. y = myTouch.getY(); // Συντεταγμένη Y όπου έχει «αγγιχθεί» η οθόνη.
7. // Έλεγχος για πάτημα κουμπιού αναπαραγωγής μουσικής.
8. if ((x >= 116) && (x <= 204) && (y >= 77) && (y <= 165)) {
9. if (playStatus == '0') { // Συνθήκες που καθορίζουν, ανάλογα την
    περίπτωση, αν θα βλέπουμε το κουμπί αναπαραγωγής ή το κουμπί
    παύσης.
10. drawFrame(159, 121, 42);
11. drawPauseButton();
12. mp3.playTrackFromFolder(00, 001);
13. delay(100);
14. playStatus = '2';
15. return;
16. }
17. if (playStatus == '1') {
18. drawFrame(159, 121, 42);
19. drawPauseButton();

```

```

20. mp3.play();
21. delay(100);
22. playStatus = '2';
23. return;
24. }
25. if (playStatus == '2') {
26. drawFrame(159, 121, 42);
27. drawPlayButton();
28. mp3.pause();
29. delay(100);
30. playStatus = '1';
31. return;
32. }
33. }

```

```

1. // Περίπτωση πατήματος κουμπιού «προηγούμενο».
2. if ((x >= 45) && (x <= 95) && (y >= 97) && (y <= 147)) {
3. drawFrame(70, 121, 26);
4. mp3.previousTrack();
5. delay(100);
6. drawTrackBar();
7. }

8. // Περίπτωση πατήματος κουμπιού «επόμενο».
9. if ((x >= 227) && (x <= 277) && (y >= 97) && (y <= 147)) {
10. drawFrame(252, 122, 26);
11. mp3.nextTrack();
12. delay(100);
13. drawTrackBar();
14. }

15. // Περίπτωση πατήματος κουμπιού μείωσης έντασης.
16. if ((x >= 35) && (x <= 75) && (y >= 165) && (y <= 209)) {
17. drawUnderline(45, 205, 65, 205);
18. if (iV >= 0 & iV <= 30) {
19. iV--;
20. drawVolume(iV);
21. }
22. mp3.decreaseVolume();
23. delay(100);
24. }

25. // Περίπτωση πατήματος κουμπιού αύξησης έντασης.

```

```

26. if ((x >= 230) && (x <= 280) && (y >= 165) && (y <= 209)) {
27. drawUnderline(235, 205, 275, 205);
28. if (iV >= 0 & iV <= 30) {
29. iV++;
30. drawVolume(iV);
31. }
32. mp3.increaseVolume();
33. delay(100);
34. }
35. // Περίπτωση πατήματος κουμπιού MENU.
36. if ((x >= 0) && (x <= 75) && (y >= 0) && (y <= 30)) {
37. myGLCD.clrScr();
38. drawHomeScreen();
39. currentPage = '0';
40. return;
41. }
42.
1. // Κλήση διαδικασίας trackPlayTime() για ενημέρωση μπάρας αναπαραγωγής
   τραγουδιού.
2. if (playStatus == '1' || playStatus == '2') {
3. trackPlayTime();
4. }
5. void trackPlayTime() {
6. totalTime = mp3.getTotalTrackPlaybackTime(); // Επιστρέφει το συνολικό
   χρόνο αναπαραγωγής ενός τραγουδιού.
7. delay(10);
8. elapsedTime = mp3.getElapsedTrackPlaybackTime();
9. // Επιστρέφει το χρόνο που έχει περάσει από την έναρξη της αναπαραγωγής
   ενός τραγουδιού.
10. delay(10);
11. minutes = (int)elapsedTime / 60;
12. seconds = (((float)elapsedTime / 60) - minutes) * 60;
13. playback = totalTime - elapsedTime;
14. minutesR = (int)playback / 60;
15. secondsR = (((float)playback / 60) - minutesR) * 60;
16.
17. myGLCD.setFont(SmallFont); // Απεικόνιση μπάρας αναπαραγωγής
   ανάλογα με τον χρόνο.
18. myGLCD.setColor(255, 255, 255);
19. myGLCD.printNumI(minutes, 8, 48);
20.
21. myGLCD.print(":", 16, 48);
22. myGLCD.printNumI((int)seconds, 24, 48, 2, '0');
23.
24. myGLCD.print("-", 276, 48);

```

```

25.     myGLCD.printNumI(minutesR, 284, 48);
26.     myGLCD.print(":", 292, 48);
27.     myGLCD.printNumI((int)secondsR, 300, 48, 2, '0');
28.
29.     int trackBarX = map(elapsedTime, 0, totalTime, 0, 224);
30.     myGLCD.setColor(255, 0, 0);
31.     myGLCD.fillRect (48, 50, 48 + trackBarX, 50 + 8);
32.
33.     if (totalTime == elapsedTime) {
34.         mp3.nextTrack();
35.         delay(30);
36.         myGLCD.setColor(255, 255, 255);
37.         myGLCD.fillRect (48, 50, 48 + 224, 50 + 8);
38.     }
39. }
40.

```

```

1. // Οθόνη αφύπνησης.
2. if (currentPage == '2') {
3. myGLCD.setFont(BigFont);
4. myGLCD.setColor(255, 255, 255);
5. myGLCD.print("MENU", 5, 5);
6. myGLCD.print("Set Alarm", CENTER, 20);
7.
8. // Δημιουργία συμβόλου «:» που διαχωρίζει ώρες και λεπτά.
9. myGLCD.setColor(0, 255, 0);
10. myGLCD.fillCircle (112, 65, 4);
11. myGLCD.setColor(0, 255, 0);
12. myGLCD.fillCircle (112, 85, 4);
13.
14. myGLCD.setFont(SevenSegNumFont);
15. myGLCD.setColor(0, 255, 0);
16. myGLCD.printNumI(aHours, 32, 50, 2, '0');
17. myGLCD.printNumI(aMinutes, 128, 50, 2, '0');
18. myGLCD.setColor(255, 255, 255);
19. myGLCD.drawRoundRect (42, 115, 82, 145);
20. myGLCD.drawRoundRect (138, 115, 178, 145);
21. myGLCD.setFont(BigFont);
22. myGLCD.print("H", 54, 122);
23. myGLCD.print("M", 150, 122);
24.
25. myGLCD.drawRoundRect (215, 60, 303, 90);
26. myGLCD.print("SET", 236, 67);
27. myGLCD.drawRoundRect (215, 115, 303, 145);
28. myGLCD.print("CLEAR", 220, 122);
29.
30. alarmNotSet = true;
31.
32. while (alarmNotSet){

```

```

33. if (myTouch.dataAvailable()) {
34. myTouch.read();
35. x = myTouch.getX(); // Συντεταγμένη X όπου έχει «αγγιχθεί» η οθόνη.
36. y = myTouch.getY(); // Συντεταγμένη Y όπου έχει «αγγιχθεί» η οθόνη.
37. // Ρύθμιση κουμπιού για ώρες.
38. if ((x >= 42) && (x <= 82) && (y >= 115) && (y <= 145)) {
39. drawRectFrame(42, 115, 82, 145);
40. aHours++;
41. if(aHours >=24){
42. aHours = 0;
43. }
44. myGLCD.setFont(SevenSegNumFont);
45. myGLCD.setColor(0, 255, 0);
46. myGLCD.printNumI(aHours, 32, 50, 2, '0');
47. }
48. // Ρύθμιση κουμπιού για λεπτά.
49. if ((x >= 138) && (x <= 178) && (y >= 115) && (y <= 145)) {
50. drawRectFrame(138, 115, 178, 145);
51. aMinutes++;
52. if(aMinutes >=60){
53. aMinutes = 0;
54. }
55. myGLCD.setFont(SevenSegNumFont);
56. myGLCD.setColor(0, 255, 0);
57. myGLCD.printNumI(aMinutes, 128, 50, 2, '0');
58. }
59. // Ρύθμιση κουμπιού αφύπνισης.
60. if ((x >= 215) && (x <= 303) && (y >= 60) && (y <= 80)) {
61. drawRectFrame(215, 60, 303, 90);
62. if (aHours < 10 && aMinutes < 10){
63. alarmString = "0" + (String)aHours + ":" + "0" + (String)aMinutes + ":" + "00";
64. }
65. else if (aHours < 10 && aMinutes > 9){
66. alarmString = "0" + (String)aHours + ":" + (String)aMinutes + ":" + "00";
67. }
68. else if (aHours > 9 && aMinutes < 10){
69. alarmString = (String)aHours + ":" + "0" + (String)aMinutes + ":" + "00";
70. }
71. else {
72. alarmString = (String)aHours + ":" + (String)aMinutes + ":" + "00";
73. }
74. myGLCD.setFont(BigFont);
75. myGLCD.print("Alarm set for:", CENTER, 165);
76. myGLCD.print(alarmString, CENTER, 191);
77.
78. }
79. // Καθαρισμός κουμπιού αφύπνισης.
80. if ((x >= 215) && (x <= 303) && (y >= 115) && (y <= 145)) {
81. drawRectFrame(215, 115, 303, 145);
82. alarmString="";

```

```

83. myGLCD.setColor(0, 0, 0);
84. myGLCD.fillRect(45, 165, 275, 210);
85. }
86. // Περίπτωση που πατήσουμε το κουμπί του MENU.
87. if ((x >= 0) && (x <= 75) && (y >= 0) && (y <= 30)) {
88. alarmNotSet = false;
89. currentPage = '0';
90. myGLCD.clrScr();
91. drawHomeScreen(); // Μας πηγαίνει στην αρχική οθόνη.
92. }
93. }
94. }
95. }

```

```

1. // Ενεργοποίηση αφύπνισης.
2. if (alarmNotSet == false) {
3. if (alarmString == rtc.getTimeStr()){
4. myGLCD.clrScr();
5. mp3.setVolume(25);
6. mp3.playTrackByIndexNumber(1);
7. delay(100);
8. myGLCD.setFont(BigFont);
9. myGLCD.setColor(255, 255, 255);
10. myGLCD.print("ALARM", CENTER, 90);
11. myGLCD.drawBitmap (127, 10, 65, 64, AlarmButton);
12. myGLCD.print(alarmString, CENTER, 114);
13. myGLCD.drawRoundRect (94, 146, 226, 170);
14. myGLCD.print("DISMISS", CENTER, 150);
15. boolean alarmOn = true;
16. while (alarmOn){
17. if (myTouch.dataAvailable()) {
18. myTouch.read();
19. x = myTouch.getX(); // Συντεταγμένη X όπου έχει «αγγιχθεί» η οθόνη.
20. y = myTouch.getY(); // Συντεταγμένη Y όπου έχει «αγγιχθεί» η οθόνη.
21.
22. // Παύση αφύπνισης.
23. if ((x >= 94) && (x <= 226) && (y >= 146) && (y <= 170)) {
24. drawRectFrame(94, 146, 226, 170);
25. alarmOn = false;
26. alarmString="";
27. myGLCD.clrScr();

```

```
28. mp3.stopPlayback();
29. delay(100);
30. currentPage = '0';
31. playStatus = '0';
32. mp3.setVolume(15);
33. drawHomeScreen();
34. }
35.
36. }
37. }
38. }
39. }
```


ΒΙΒΛΙΟΓΡΑΦΙΑ

1.)Arduino UNO, <https://store.arduino.cc/arduino-uno-rev3>

2.)Arduino Mega 2560 R3, <https://store.arduino.cc/arduino-mega-2560-rev3>

3.)Arduino, <https://el.wikipedia.org/wiki/Arduino>

4.)Arduino IDE, <https://www.arduino.cc/en/Guide/Environment>

5.)<http://howtomechatronics.com/projects/arduino-touch-screen-music-player-alarm-clock-project/>

6.)Libraries BY8001-16p module, <https://github.com/r0ndL/BY8001>
(Βιβλιοθήκες)

7.)Libraries Tft touch screen-Utouch,
<http://www.rinkydinkelectronics.com/library.php?id=51> (Βιβλιοθήκες)