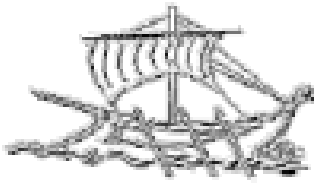


ΑΕΙ ΠΕΙΡΑΙΑ Τ.Τ.
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ Τ.Ε.

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Ασύρματη Ενημέρωση Οθόνης Κειμένου Με LED

Ανδρέας Κωτσάκης

Ανδρέας Λυμπέρης

Εισηγητής: Δρ. Ιωάννης Έλληνας, Καθηγητής

ΑΘΗΝΑ
ΟΚΤΩΒΡΙΟΣ 2015

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Ασύρματη Ενημέρωση Οθόνης Κειμένου Με LED

Ανδρέας Κωτσάκης
Α.Μ. 36966

Ανδρέας Λυμπέρης
Α.Μ. 36688

Εισηγητής:

Δρ. Ιωάννης Έλληνας, Καθηγητής

Εξεταστική Επιτροπή:

Ημερομηνία εξέτασης:

ΠΕΡΙΛΗΨΗ

Σκοπός της πτυχιακής εργασίας είναι η ανάπτυξη υπολογιστικού συστήματος με δυνατότητα διαχείρισης οθόνης από συστοιχίες φωτοδιόδων (LED Dot-Matrix). Η διαχείριση της οθόνης και η αποστολή του μηνύματος προς απεικόνιση γίνεται είτε απομακρυσμένα μέσω δικτύου κινητής τηλεφωνάς είτε τοπικά από το ίδιο το σύστημα μέσω οθόνης αφής.

Συγκεκριμένα για την εργασία γίνεται χρήση της πλατφόρμας Arduino τόσο για το σύστημα ελέγχου διαχείρισης όσο για την ιδιά την οθόνη. Το πάνελ έχει υλοποιηθεί αυτόνομα και επικοινωνεί ασύρματα με το σύστημα ελέγχου το οποίο αποτελείται από τα Arduino, GSM Modem, την οθόνη αφής, κύκλωμα υπερύθρων για την ασύρματη επικοινωνία και διεπαφή για σύνδεση με κάρτα μνήμης τύπου SD.

Η σύνδεση με το δίκτυο κινητής τηλεφωνίας GSM γίνεται μέσω της μονάδας διαμορφωτή - αποδιαμορφωτή (modem) SIM900 της εταιρίας SIMCOM και με την χρήση του πρωτοκόλλου SMS.

Στο πλαίσιο της πτυχιακής εργασίας, επιλέχθηκε το σενάριο στο οποίο θα γίνεται χρήση του συστήματος από καθηγητή τεχνολογικού ιδρύματος για την αποστολή ανακοινώσεων προς τους φοιτητές. Το σύστημα θα μπορεί να χρησιμοποιείται και από τους φοιτητές για περαιτέρω ενημέρωση μέσω της οθόνης αφής.

ΕΠΙΣΤΗΜΟΝΙΚΗ ΠΕΡΙΟΧΗ: Ανάπτυξη Εφαρμοσμένων Συστημάτων
ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: GSM, Dot-Matrix, Πάνελ, Arduino, TFT LCD

ABSTRACT

The aim of this final paper is the development of an integrated computer system, circuitry and software, designed to control a Dot-Matrix LED panel. The system can be operated not only locally, by accessing its touch screen control panel but also remotely using an available mobile network.

For the purpose of the development of this paper's electronics the Arduino platform is used. Both the touch screen control panel and the Dot-Matrix LED display are developed using Arduino modules. The Dot-Matrix LED display panel is a standalone component and communicates wirelessly with the main control panel. The main control panel, apart from the aforementioned Arduino and touch screen modules, also consists of a GSM modem module, circuitry for the wireless infrared (I.R) communication between the devices and an interface making possible the connection of SD memory cards.

Connection to the mobile network is achieved using the GSM modem SIM900 by SIMCOM and the messages are sent using the widely available sort message system (S.M.S).

For the presentation of this final paper, was decided that the system will be used in a scenario in which a professor of an educational institute will be making use of system to issue announcements to the students remotely. Students on the other hand will be able to access the touch screen interface for additional personalized information concerning the announcement.

SCIENTIFIC FIELD: Embedded Systems Development
KEYWORDS: GSM, Dot-Matrix, Panel, TFT LCD, Arduino

ΠΕΡΙΕΧΟΜΕΝΑ

ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

ΚΕΦΑΛΑΙΟ 1

ΕΙΣΑΓΩΓΗ

1.1 Αντικείμενο της Πτυχιακής Εργασίας

Αντικείμενο της παρούσας πτυχιακής εργασίας είναι η κατασκευή ενός συστήματος συσκευών που έχουν ως στόχο την προβολή μηνύματος σε πάνελ με συστοιχίες φωτοδιόδων (LED Dot-Matrix Panel).

Στόχος είναι να δημιουργηθεί ένα ολοκληρωμένο σύστημα από το οποίο να μπορεί να γίνεται ο ορισμός του μηνύματος προς προβολή τόσο τοπικά όσο και απομακρυσμένα. Για αυτόν τον λόγο το σύστημα θα διαθέτει οθόνη αφής για την τοπική εισαγωγή του μηνύματος και περαιτέρω ρυθμίσεων αλλά και συσκευή για πρόσβαση σε δίκτυο κινητής τηλεφωνίας για απομακρυσμένη αποστολή του μηνύματος.

Στο πλαίσιο της πτυχιακής εργασίας θα γίνει τόσο η κατασκευή του πάνελ και της συσκευής ελέγχου αλλά και η συγγραφή του απαραίτητου λογισμικού για να επιτευχθούν οι παραπάνω στόχοι. Για το ηλεκτρονικό - κατασκευαστικό κομμάτι της εργασίας έγινε χρήση της πλατφόρμας Arduino ενώ το λογισμικό αναπτύχθηκε κάνοντας χρήση της γλώσσας προγραμματισμού C++, που είναι και η υποστηριζόμενη γλώσσα προγραμματισμού για την πλατφόρμα Arduino.

1.2 Εισαγωγή - Πρόλογος

Η παρούσα πτυχιακή εργασία αποσκοπεί στην κατασκευή ενός συστήματος που θα επιτρέπει στον χρήστη την αποστολή μηνύματος με ασύρματο τρόπο και την εμφάνιση του σε οθόνη Dot-Matrix.

Με τον όρο οθόνη Dot-Matrix εννοούμε μια μονάδα προβολής εικόνας και κειμένου που είναι κατασκευασμένη από συστοιχίες διόδων LED. Παράδειγμα τέτοιων

μονάδων αποτελούν οι οθόνες που βρίσκονται στα μετρό και ενημερώνουν τους επιβάτες για την ώρα διέλευσης του συρμού.

Εικόνα 1.1 Πάνελ δρομολογίων λεωφορείων

Η υλοποίηση αποτελείται από δύο αυτόνομες μονάδες. Η πρώτη μονάδα είναι η οθόνη Dot-Matrix όπου θα γίνεται η προβολή του μηνύματος, η δεύτερη μονάδα θα είναι η μονάδα ελέγχου στην οποία μεταξύ άλλων θα γίνεται και η λήψη και επεξεργασία του μηνύματος προς προβολή. Οι μονάδες θα επικοινωνούν μεταξύ τους ασύρματα.

Γίνεται αντιληπτό ότι ένα τέτοιο σύστημα μπορεί να έχει πολλές εφαρμογές καθώς επιτρέπει την ασύρματη ανάρτηση μηνυμάτων, ενημερώσεων ή και διαφημίσεων. Για την εν λόγο πτυχιακή εργασία θεωρούμε το σενάριο στο οποίο θα γίνεται χρήση του συστήματος από καθηγητή για να ενημερώνει ανά πάσα στιγμή τους φοιτητές που βρίσκονται στον χώρο που τεχνολογικού ιδρύματος μέσω ενός πίνακα ανακοινώσεων κατασκευασμένο από συστοιχίες Dot-Matrix.

Με γνώμονα τα παραπάνω θα ακολουθήσει μια περαιτέρω περιγραφή των δύο μονάδων και των γενικών λειτουργιών του συστήματος.

Η πρώτη μονάδα, δηλαδή η οθόνη Dot-Matrix είναι κατασκευασμένη εξ' ολοκλήρου στο πλαίσιο της πτυχιακής εργασίας. Αποτελείται από έξι μονάδες Dot-Matrix 8 x 8 και υλοποιείται με την πλατφόρμα Arduino και συγκεκριμένα με το Arduino Pro Mini αλλά και ένα σύστημα ανάγνωσης υπέρυθρων για να μπορεί να επικοινωνεί με την μονάδα ελέγχου. Σκοπός της μονάδας είναι η ανάγνωση του μηνύματος από την μονάδα ελέγχου και έπειτα η προβολή του στις συστοιχίες Dot-Matrix.

Εκτός από το ίδιο το μήνυμα η μονάδα Dot-Matrix λαμβάνει από την κεντρική μονάδα και διάφορες παραμέτρους για τον τρόπο προβολής του μηνύματος όπως για παράδειγμα φορά και ταχύτητα κύλισης.

Η κεντρική μονάδα, που είναι η δεύτερη μονάδα του συστήματος, έχει πιο σύνθετο έργο. Υλοποιείται και αυτή με την πλατφόρμα Arduino κάνοντας συγκεκριμένα χρήση του Arduino Mega. Επιλέχθηκε το συγκείμενο μοντέλο διότι προσφέρει τον μεγαλύτερο αποθηκευτικό χώρο κώδικα και διαθέσιμης μνήμης, καθώς είναι απαραίτητα για την εκτέλεση των πολλαπλών και συνθετών λειτουργιών που προορίζεται. Πέρα από το Arduino την μονάδα συνθέτουν, μονάδα GSM για την λειψή μηνυμάτων SMS, οθόνη αφής για διεπαφή με τον χρήστη και μονάδα εισαγωγής κάρτας μνήμης SD για την αποθήκευση και μεταφορά ρυθμίσεων.

Με τον όρο μονάδα GSM εννοούμε μονάδα που επιτρέπει την σύνδεση σε δίκτυο κινητής τηλεφωνίας δεύτερης γενιάς (2G) και πάνω. Συγκεκριμένα χρησιμοποιείται η μονάδα διαμορφωτή - αποδιαμορφωτή (modem) SIM900 της εταιρείας SIMCOM. Η μονάδα χρειάζεται κάρτα SIM με έγκυρο τηλεφωνικό αριθμό για να μπορεί να συνδεθεί στο δίκτυο κινητής τηλεφωνίας.

Η οθόνη αφής είναι τύπου TFT LCD και ανάλυσης 320 x 240 πίξελ και περιέχει και πρόσθετη διεπαφή για την εισαγωγή κάρτας μνήμης τύπου SD. Για τις ανάγκες της πτυχιακής εργασίας προστέθηκε και δεύτερη κάρτα μνήμης SD για την αποθήκευση των ρυθμίσεων και παραμέτρων του συστήματος.

Για τον σχεδιασμό του γραφικού περιβάλλοντος της οθόνης αφής δημιουργήθηκε βιβλιοθήκη γραφικών που επιτρέπει μεταξύ άλλων τον σχεδιασμό διαφόρων γραφικών αντικειμένων όπως κουμπιά και περιοχές εισαγωγής και εμφάνισης

κειμένου. Η βιβλιοθήκη είναι γραμμένη με τέτοιο τρόπο ώστε να χρησιμοποιεί όσο το δυνατόν λιγότερους πόρους, κάτι εύλογο μιας και η πλατφόρμα Arduino χρησιμοποιείται συνήθως για απλούστερες εφαρμογές λόγο των περιορισμένων πόρων που διαθέτει.

Τόσο η κεντρική μονάδα όσο και η μονάδα Dot-Matrix υποστηρίζουν την εμφάνιση αγγλικού και ελληνικού κειμένου. Η υποστήριξη της ελληνικής γλώσσας προστέθηκε στο πλαίσιο της πτυχιακής εργασίας και δεν υποστηρίζονταν αρχικώς από τις βιβλιοθήκες που χρησιμοποιήθηκαν.

Προτού προχωρήσουμε στην λεπτομερή ανάλυση των προαναφερθέντων, θα γίνει μια μικρή περιγραφή στις λειτουργίες που επιτρέπει το σύστημα σύμφωνα με το σενάριο χρήσης του συστήματος από τον καθηγητή και τους φοιτητές.

Η κεντρική μονάδα θα βρίσκεται προσβάσιμη στον χώρο του τεχνολογικού ιδρύματος όπου θα μπορεί να γίνεται χρήση της τόσο από τους φοιτητές όσο και από τον καθηγητή. Στον ίδιο χώρο θα βρίσκεται και αναρτημένη η μονάδα Dot-Matrix η οποία θα επικοινωνεί με την κεντρική μονάδα ασύρματα μέσω υπέρυθρων.

Ο καθηγητής θα μπορεί να αποστείλει μήνυμα SMS για προβολή μέσω του κινητού του τηλεφώνου. Το μήνυμα θα λαμβάνεται από την κεντρική μονάδα, στην οποία βρίσκεται και η μονάδα GSM, και στην συνέχεια θα αποστέλλεται ασύρματα στην μονάδα Dot-Matrix για προβολή.

Εκτός από τον απομακρυσμένο τρόπο, ο καθηγητής θα μπορεί να γράψει και να αποστείλει το μήνυμα στην μονάδα Dot-Matrix και τοπικά μέσω της κεντρικής μονάδας αποκτώντας πρόσβαση στο μενού διαχείρισης της κεντρικής μονάδας με την εισαγωγή των προσωπικών του στοιχείων. Εκτός από την αποστολή του μηνύματος το μενού διαχείρισης προσφέρει και άλλες δυνατότητες όπως, ρύθμιση διαφόρων παραμέτρων για τον τρόπο απεικόνισης του μηνύματος στην μονάδα Dot-Matrix, αντιγραφή των ρυθμίσεων του συστήματος και της μονάδας Dot-Matrix από την κάρτα μνήμης SD και ορισμό μέχρι και τριών αρχείων CSV, που βρίσκονται στην κάρτα μνήμης από τα οποία μπορούν να γίνεται ανάγνωση βαθμολογιών σε μέχρι και τρία μαθήματα.

Από την πλευρά τους οι φοιτητές μπορούν να χρησιμοποιούν την κεντρική μονάδα και με την εισαγωγή του αριθμού μητρώου τους να ενημερώνονται για τις βαθμολογίες τους στα τρία προαναφερθέντα μαθήματα. Ο καθηγητής μπορεί να ενημερώνει τα αρχεία κάνοντας χρήση του γραφικού περιβάλλοντος πλοήγησης στα αρχεία που βρίσκονται στην συνδεδεμένη κάρτα μνήμης.

ΚΕΦΑΛΑΙΟ 2

ΥΛΙΚΟ ΚΑΙ ΚΑΤΑΣΚΕΥΗ

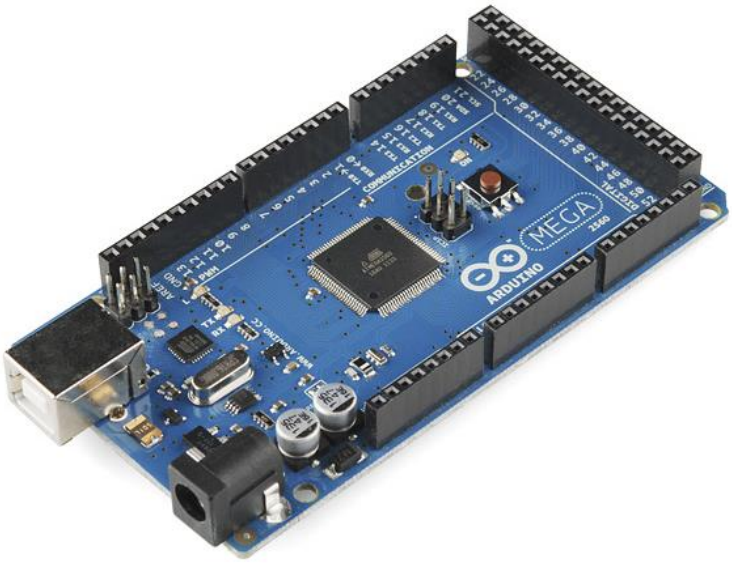
Στο κεφάλαιο που ακολουθεί θα γίνει αναφορά στο υλικό της πτυχιακής εργασίας. Αρχικώς θα αναλυθούν τα εξαρτήματα και οι έτοιμες μονάδες (modules) που χρησιμοποιήθηκαν και στην συνέχεια θα γίνει ανάλυση του τρόπου συνδεσμολογίας και του ηλεκτρονικού και κατασκευαστικού έργου ώστε να ολοκληρωθούν και οι δύο μονάδες από τις οποίες αποτελείται το εφαρμοσμένο σύστημα που αναπτύχθηκε στο πλαίσιο της πτυχιακής εργασίας.

2.1 Εξαρτήματα και Μονάδες

Σε αυτό το υποκεφάλαιο θα γίνει η παρουσίαση όλων των εξαρτημάτων και των μονάδων που χρειάστηκαν για την κατασκευή και της κεντρικής μονάδας ελέγχου και του Dot-Matrix πάνελ.

2.1.1 Arduino Mega - Arduino Pro Mini

Το Arduino είναι μια πλατφόρμα ελεύθερου λογισμικού βασισμένη στην ευκολία χρήσης των υλικών και του λογισμικού. Οι πλακέτες του Arduino μπορούν να δεχτούν ως είσοδο ένα κουμπί, ένα μήνυμα από το Twitter ή έναν αισθητήρα φωτός και να ενεργοποιήσει ένα LED, έναν κινητήρα ή να δημοσιεύσει κάτι στο διαδίκτυο και όλα αυτά με την χρήση κάποιων ορισμένων προγραμματιστικών εντολών του λογισμικού Arduino (Arduino Software (IDE)).



Εικόνα 2.1: Arduino Mega

Ο λόγος που η πλατφόρμα Arduino είναι τόσο διαδεδομένη είναι γιατί είναι :

- Αρκετά οικονομική σε αντίθεση με άλλες πλατφόρμες με μικροελεγκτές
- Μπορεί να χρησιμοποιηθεί σε λειτουργικά Windows, Macintosh OSX και Linux. Τα περισσότερα συστήματα με μικροελεγκτές υποστηρίζονται μόνο από τη λειτουργικά Windows.
- Εύκολο προγραμματιστικό περιβάλλον, ιδανικό για αρχάριους αλλά και έμπειρους χρήστες.
- Ελεύθερο λογισμικό με δυνατότητες επέκτασής του από έμπειρους προγραμματιστές.
- Ελεύθερο λογισμικό με δυνατότητα επέκτασης του κατασκευαστικού τομέα

Οι μονάδες Arduino που επιλέχθηκαν για την εκπόνηση της πτυχιακής εργασίας είναι το Arduino Mega και το Arduino Pro Mini. Το πρώτο χρησιμοποιήθηκε για την κατασκευή της κεντρικής μονάδας ενώ το δεύτερο για το Dot-Matrix πάνελ. Η επιλογή του Arduino Mega ήταν απαραίτητη καθώς, όπως θα δούμε και παρακάτω,

είναι το μοναδικό από τα Arduino με οχτάμπιτο μικροελεγκτή που διαθέτει την απαιτούμενη μνήμη RAM και ένα μεγάλο αριθμό ακροδεκτών, απαραίτητα για την συνδεσμολογία όλων των μονάδων και των εξαρτημάτων που χρειάζονται για την κατασκευή της κεντρικής μονάδας. Αντίθετα, το Arduino Pro Mini επιλέχθηκε γιατί ήταν το φθηνότερο αφού οι λειτουργίες που χρειάζεται να κάνει είναι πολύ συγκεκριμένες. Παρακάτω ακολουθούν τα χαρακτηριστικά των δύο μονάδων και διαφορές τους.

Εικόνα 2.2: Arduino Pro Mini

Microcontroller	**ATmega2560**	**ATmega328**
Operating Voltage	5V	3.3V or 5V (depending on model)
Input Voltage	7-12V	5 - 12 V (5V model)
Digital I/O Pins	54 (15 PWM)	14 (6 PWM)
Analog Input Pins	16	6
DC Current per I/O Pin	20 mA	40 mA
Flash Memory	256 KB - 8 KB bootloader	32 kB - 0.5 kB bootloader
SRAM	8 KB	2 kB
EEPROM	4 KB	1 kB
Clock Speed	16 MHz	16 MHz (5V model)

Πίνακας 2.1: Τεχνικά χαρακτηριστικά Arduino Mega και Pro Mini

2.1.2 GSM Module – AT Commands

Για την σύνδεση της κεντρικής μονάδας στο δίκτυο τηλεφωνίας και κατά συνέπεια την δυνατότητα λήψης μηνυμάτων SMS χρησιμοποιήσαμε την μονάδα GSM της παρακάτω εικόνας.

Εικόνα 2.3: SIM900 GSM Shield

Η μονάδα GSM βασίζεται στον μικροελεγκτή SIM900 της εταιρίας SIMCOM. Ο μικροελεγκτής επιτρέπει την σύνδεση σε δίκτια τηλεφωνίας 2.5G, δηλαδή επιτρέπει εκτός από τις κλήσεις και τις αποστολές SMS και MMS μέσω του πρωτοκόλλου GSM και την ασύρματη πρόσβαση στο διαδίκτυο με ταχύτητες έως 85.6 mbps μέσω GPRS. Η υλοποίηση που χρησιμοποιήσαμε είναι της εταιρίας SaintSmart και υποστηρίζει εκτός των άλλων και την σύνδεση μικροφώνου και ακουστικών. Όπως και με όλες τις μονάδες GSM για να γίνει εφικτή η σύνδεση στο δίκτυο κινητής τηλεφωνίας είναι απαραίτητη η εισαγωγή κάρτας SIM στην αντίστοιχη διεπαφή στο κάτω μέρος της μονάδας.

```
if (gsm_flg[7] == 1) {
    sendATCommand(Serial2, "AT\r", "OK");
}
if (gsm_flg[7] == 2) {
    sendATCommand(Serial2, "AT+CMGF=1\r", "OK");
}
if (gsm_flg[7] == 3) {
    sendATCommand(Serial2, "AT+CMGD=1\r", "OK");
}
if (gsm_flg[7] == 4) {
    sendSMS(Serial2, "AT+CMGS=1269\r", 5);
}
if (gsm_flg[7] == 5) {
    waitGSMEvent("+CMTI");
}
if (gsm_flg[7] == 6) {
    sendATCommand(Serial2, "AT+CMGR=1\r", "OK");
}
```

Εικόνα 2.4: Παράδειγμα κώδικα με AT Commands

Η συσκευή GSM επικοινωνεί με το Arduino μέσω σειριακής σύνδεσης και ο τρόπος με τον οποίο μπορούμε να αναθέσουμε κάποια λειτουργία στην μονάδα είναι μέσω εντολών που ονομάζονται AT (AT Commands).

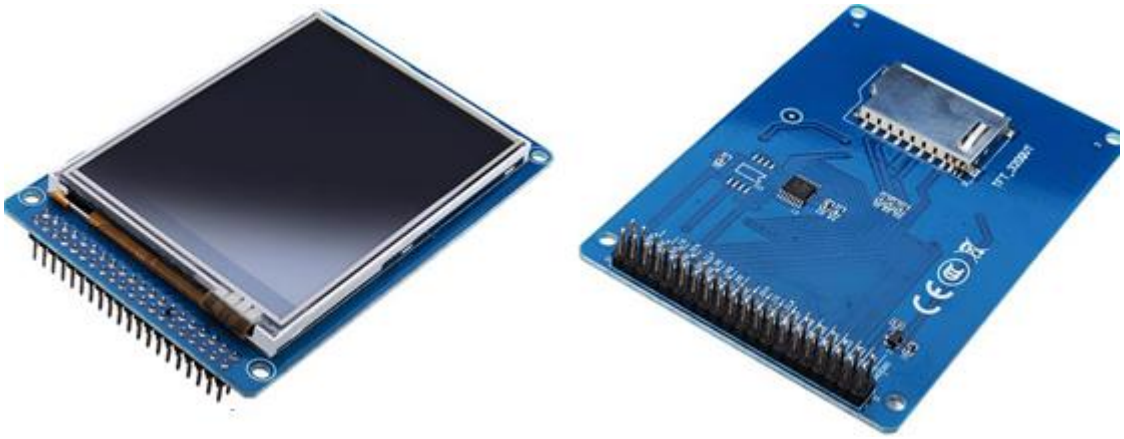
Αυτές είναι ένα σετ από εντολές που ξεκινάνε πάντα με τα γράμματα AT, εξού και η ονομασία τους, και μετά είτε έχουν άλλο ένα γράμμα που υποδηλώνει την ενέργεια που ζητάμε (π.χ. ATD είναι η εντολή για την κλήση ή αλλιώς Dial) είτε έχουν το σύμβολο + και μετά την εντολή που αντιστοιχεί στην ενέργεια που θέλουμε (π.χ. AT+CMGS και AT+CMGR είναι οι εντολές για την αποστολή και λήψη γραπτών μηνυμάτων ή αλλιώς Send και Receive). Παρακάτω ακολουθεί ένας πίνακας με τις εντολές AT που χρησιμοποιήθηκαν στο λογισμικό της πτυχιακής εργασίας.

AT+CPIN="SIMPIN”	Εκχωρεί το pin στην κάρτα SIM
AT+CMGF=1	Ορίζει τον τρόπο ανάγνωσης του εισερχόμενου SMS σε txt mode.
AT+CMGD=1,4	Διαγράφει όλα τα αποθηκευμένα μηνύματα SMS που υπάρχουν στην κάρτα SIM.
+CMGL	Μήνυμα ενημέρωσης του συστήματος για εισερχόμενο SMS.
+CMGR	Ανάγνωση SMS μηνύματος αμέσως μετά τη λήψη.
+CMTI	Μήνυμα ενημέρωσης εισερχόμενου SMS συμπεριλαμβανομένου και του δείκτη θέσης στην μνήμη.

Πίνακας 2.2: AT Commands που χρησιμοποιήθηκαν

2.1.3 TFT LCD Touch Screen

Για την προβολή του γραφικού περιβάλλοντος ελέγχου και την διεπαφή του χρήστη με την κεντρική μονάδα χρησιμοποιήθηκε οθόνη αφής τύπου TFT LCD. Η οθόνη αφής που επιλέχθηκε είναι στις 3.2 ίντσες, βασίζεται στο ολοκληρωμένο HX8347-A και έχει ανάλυση εικόνας 320 x 240 πίξελ και βάθος χρώματος 262 χιλιάδων χρωμάτων. Η επιλογή της οθόνης έγινε τόσο με γνώμονα το κόστος όσο και την διαθέσιμη επεξεργαστική ισχύ του Arduino.



Εικόνα 2.5: 3.2 TFT LCD touch screen

Παρακάτω ακολουθεί ένας πίνακας με τα τεχνικά χαρακτηριστικά και τις απαιτήσεις της οθόνης. Εδώ πρέπει να σημειωθεί ότι η οθόνη λειτουργεί στα 3.3V εν αντιθέσει με το Arduino που έχει ως τάση λειτουργίας τα 5V. Αυτό σημαίνει ότι η δύο μονάδες δεν είναι συμβατές και χρειάζεται ενδιάμεσο κύκλωμα για την σωστή λειτουργία του συστήματος.

Φυσικές Διαστάσεις	57.54 (W) x 79.2 (H) x 4.40(T) mm
Τύπος Οθόνης	TFT LCD
Ανάλυση Οθόνης	320 x 240
Βάθος Χρωμάτων	16 bit
Τύπος LED φωτισμού	WHITE LED
Επεξεργαστής	HX8347-A

Πίνακας 2.3: Τεχνικά χαρακτηριστικά οθόνης αφής

Ένας τρόπος για να αποφευχθεί το πρόβλημα είναι η χρήση διαιρέτη τάσης σε κάθε ακροδέκτη που συνδέει τις δύο μονάδες. Αυτό όμως σημαίνει ότι πρέπει να κάνουμε διαιρέτη για πάνω από τριάντα ακροδέκτες, δηλαδή να χρησιμοποιήσουμε πάνω από εξήντα αντιστάτες. Αυτό θα καθιστούσε το κύκλωμα πολύ μεγάλο και ουσιαστικά δεν θα έλυνε και το πρόβλημα της συνδεσμολογίας καθώς η οθόνη δεν ακολουθεί τα πρότυπα σχεδίασης του Arduino που είναι σε μορφή modules.

Ευτυχώς υπάρχουν ήδη έτοιμοι αντάπτορες στην αγορά όπου απλοποιούν την διαδικασία της σύνδεσης της οθόνης με το Arduino Mega. Ο αντάπτορας που επιλέχθηκε φαίνεται στη παρακάτω εικόνα.

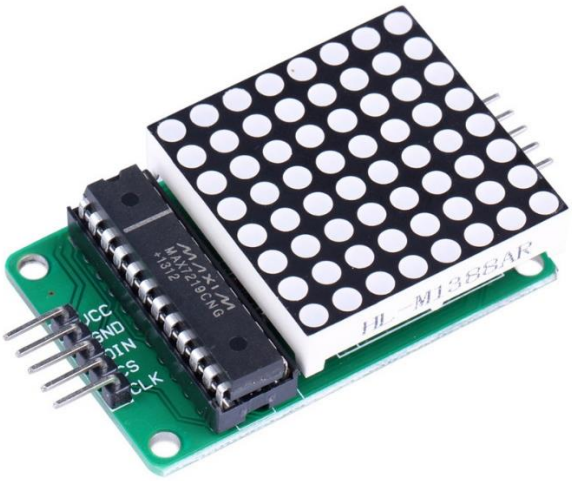
Εικόνα 2.6: Αντάπτορας TFT LCD σε Arduino Mega

Τέλος, αναφέρουμε ότι η μονάδα της οθόνης προσφέρει και διεπαφή για εισαγωγή κάρτα μνήμης τύπου SD. Η κάρτα δεν αποτελεί μέρος της οθόνης απλώς υπάρχει ως διεπαφή διότι αποτελεί τον καλύτερο τρόπο αποθήκευσης εικόνων και γενικά γραφικών για προβολή στην οθόνη.

2.1.4 Dot-Matrix Module

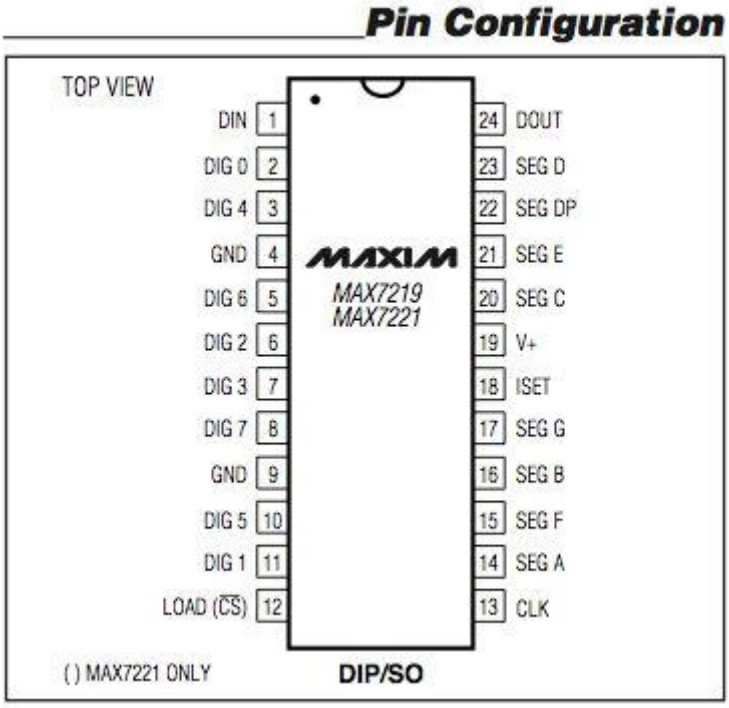
Για την υλοποίηση του Dot-Matrix πάνελ επιλέχθηκαν έτοιμες μονάδες που αποτελούνται από μια συστοιχία από LED διόδους σε διάταξη 8 X 8 και το ολοκληρωμένο MAX7219. Το MAX7219 είναι ολοκληρωμένο ειδικά κατασκευασμένο για την οδήγηση συστοιχιών από LED και μπορεί να οδηγήσει μέχρι και 64 LED ταυτόχρονα, ενώ παράλληλα μπορεί να ελεγχθεί πολύ εύκολα από το Arduino μέσω του πρωτοκόλλου σύγχρονης σειριακής επικοινωνίας SPI. Περισσότερα από ένα ολοκληρωμένα MAX7219 μπορούν να συνδεθούν σε σειρά

για να δημιουργηθεί μεγαλύτερη επιφάνεια απεικόνισης χωρίς να χρειάζεται αλλαγή στην συνδεσμολογία με το Arduino ούτε η χρήση περισσοτέρων ακροδεκτών.



Εικόνα 2.7: Dot-Matrix 8x8 module

Αυτό επιτυγχάνεται συνδέοντας την έξοδο της μιας μονάδας στην είσοδο της επόμενης. Η μονάδα επίσης υποστηρίζει τόσο το σβήσιμο των LED για εξοικονόμηση ενέργειας όσο και τον έλεγχο της φωτεινότητας τους.



Εικόνα 2.8 MAX7219 pinout

Όπως μπορούμε να δούμε και στην παραπάνω εικόνα το ολοκληρωμένο MAX7219 διαθέτει μόνο 16 ακροδέκτες για την σύνδεση σε LED ωστόσο όμως μπορεί να διαχειριστεί μέχρι και 64 LED αυτό συμβαίνει γιατί χρησιμοποιεί μια τεχνική κατά την οποία φωτίζεται μόνο μια στήλη από LED κάθε χρονική στιγμή και αμέσως μετά η επόμενη της. Η όλη διαδικασία γίνεται σε χρόνο μικρότερο από αυτόν που μπορεί να αντιληφθεί το ανθρώπινο μάτι με αποτέλεσμα να δίνεται η ψευδαίσθηση ότι δυνητικά και τα 64 LED φωτίζουν ταυτόχρονα. Κάνοντας χρήση αυτής της τεχνικής γίνεται και μεγάλη εξοικονόμηση ενέργειας αφού στην χειρότερη των περιπτώσεων θα φωτίζουν ταυτόχρονα 8 από τα 64 LED.

2.1.5 TSOP4840 – IR LED

Τα υπέρυθρα LED χρησιμοποιούνται σε πολλές περιπτώσεις που θέλουμε να μεταφέρουμε δεδομένα ανάμεσα σε ένα πομπό και ένα δέκτη με την προϋπόθεση να υπάρχει ελεύθερο πεδίο για να μεταφερθεί το υπέρυθρο φως ανάμεσα τους. Μερικά κλασικά παραδείγματα είναι τα τηλεκοντρόλ των τηλεοράσεων και των κλιματιστικών.

Εικόνα 2.9: Υπέρυθρα LED

Στην κατασκευή μας επιλέχθηκε η χρήση υπέρυθρων για την μεταφορά των μηνυμάτων από την κεντρική μονάδα στην μονάδα Dot-Matrix. Εδώ πρέπει να αναφέρουμε ότι θα μπορούσε να γίνει χρήση κάποιας άλλης, πιο ασφαλής, τεχνολογίας όπως Bluetooth ή ραδιοκύματα αλλά προτιμήθηκε η χρήση των υπερύθρων καθώς εκτός του ότι είναι η πιο φθηνή λύση, έχει και αρκετό ενδιαφέρον

η κατασκευή του κυκλώματος και η μελέτη του τρόπου κωδικοποίησης και αποστολής των μηνυμάτων με τον εν λόγο τρόπο.

Εικόνα 2.10: Σένσορας υπερύθρων TSOP4840

Όπως αναφέραμε και παραπάνω για την αποστολή υπέρυθρου σήματος χρειαζόμαστε ένα απλό υπέρυθρο LED το οποίο θα εκπέμπει σε μια συγκεκριμένη συχνότητα. Για την ανάγνωση του απεσταλμένου υπέρυθρου μηνύματος χρειαζόμαστε τον ανάλογο αισθητήρα υπερύθρων. Στην περίπτωση μας επιλέχθηκε ο αισθητήρας TSOP4840 που μπορεί να αποκωδικοποιήσει μήνυμα σε συχνότητα 40KHz, συχνότητα στην οποία θα αποστέλνεται και το μήνυμα από την κεντρική μονάδα.

2.2 Συνδεσμολογία και Ολοκλήρωση Κατασκευής

Συνεχίζοντας από το προηγούμενο κεφάλαιο, πρέπει πλέον να χρησιμοποιηθούν τα προαναφερθέντα εξαρτήματα και συσκευές για την κατασκευή των δύο μονάδων από τις οποίες θα αποτελείται το σύστημα μας.

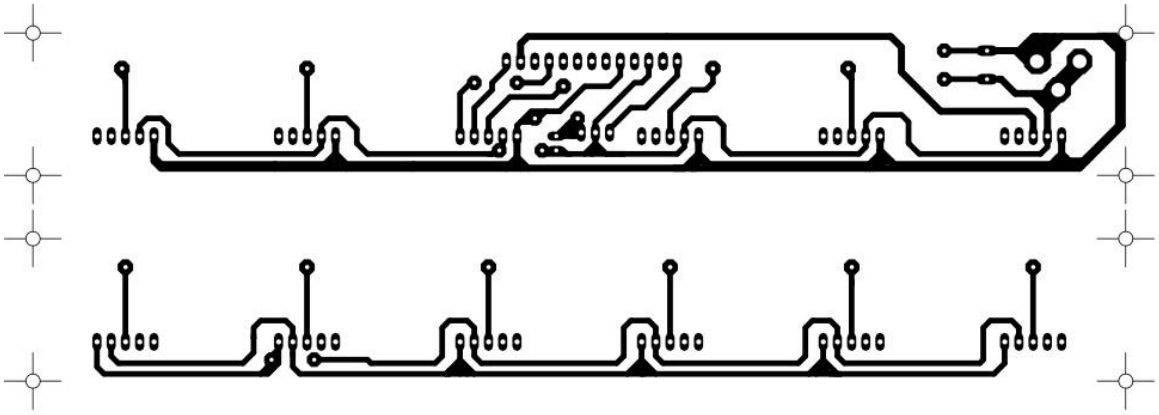
Αρχικά θα ξεκινήσουμε με την δημιουργία του Dot-Matrix πάνελ. Το πάνελ αποφασίστηκε ότι θα αποτελείται από έξι μονάδες Dot-Matrix 8 x 8 σε οριζόντια διάταξη. Η συνολική ανάλυση του πάνελ μας λοιπόν θα είναι 48 x 8. Το μέγεθος του πάνελ θα μπορούσε να είναι πολύ μεγαλύτερο και ο μόνος λόγος που δεν έγινε μεγαλύτερο είναι για να παραμείνει το κόστος της κατασκευής χαμηλό μιας και οι έξι μονάδες είναι αρκετές για την παρουσίαση στο πλαίσιο της πτυχιακής εργασίας.

Εκτός από τα Dot-Matrix την μονάδα αποτελούν το Arduino Pro Mini και ο αισθητήρας υπερύθρων TSOP4840. Παρακάτω ακολουθεί ο πίνακας με όλες τις συνδέσεις για να επιτευχθεί η λειτουργία της μονάδας Dot-Matrix.

Arduino Pro Mini	**Max7219 Dot-Matrix Module**	**TSOP**
5V	VCC	VCC
GND	GND	GND
D11 (MOSI)	DIN	
D10 (SS)	CS	
D13 (SCK)	CLK	
RX0 (RX)		OUT

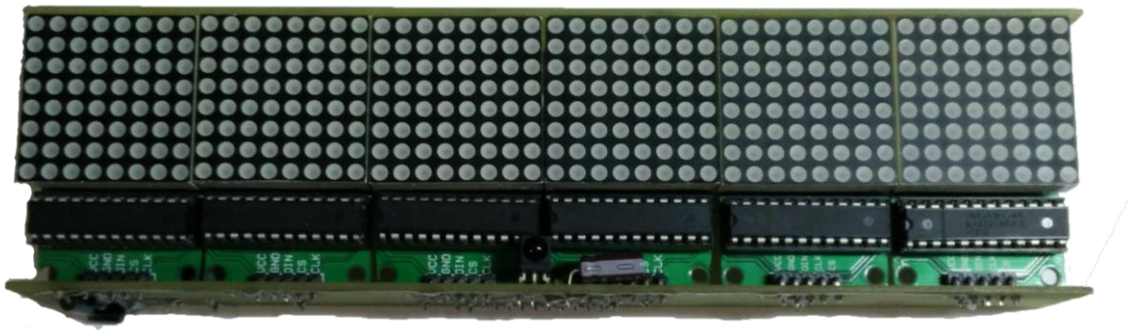
Πίνακας 2.4: Συνδεσμολογία μονάδας Dot-Matrix

Αρχικά δοκιμάστηκε η συνδεσμολογία σε breadboard και αφού βεβαιωθήκαμε ότι η λειτουργία του συστήματος είναι η ζητούμενη συνεχίσαμε στην σχεδίαση πλακέτας για το κύκλωμα μας. Για τον σχεδιασμό της πλακέτας χρησιμοποιήθηκε το πολύ γνωστό πρόγραμμα σχεδίασης ηλεκτρονικών κυκλωμάτων της εταιρίας CadSoft, EAGLE. Προτιμήθηκε η σχεδίαση με το EAGLE και η εκτύπωση του κυκλώματος εν αντιθέσει με την υλοποίηση του σε διατρητή πλακέτα για να γίνει το κύκλωμα όσο το δυνατόν μικρότερο, συμβάλλοντας στο να φαίνεται πιο επαγγελματική η όλη κατασκευή του.



Εικόνα 2.11: Ηλεκτρονικό σχέδιο μονάδας Dot-Matrix

Το τελικό αποτέλεσμα φαίνεται στην παρακάτω εικόνα. Όπως αναφέραμε και πιο πάνω το κύκλωμα είναι πλέον έτοιμο να τεθεί σε λειτουργία. Βλέπουμε ότι το κύκλωμα έχει το ελάχιστο δυνατό μέγεθος και μπορεί πολύ εύκολα κλειστεί μέσα σε κάποιο κουτί κάνοντας το οπτικά πιο ωραίο, προστατεύοντας όμως παράλληλα και το ίδιο το κύκλωμα αφού δεν ενδείκνυται η παρατεταμένη χρήση απροστάτευτων ηλεκτρονικών εξαρτημάτων.



Εικόνα 2.12: Μονάδα Dot-Matrix

Συνεχίζουμε με την υλοποίηση της κύριας μονάδας, εδώ τα περισσότερα εξαρτήματα είναι της μορφής «module», δηλαδή μονάδες ειδικά φτιαγμένες να κουμπώνουν πάνω στο Arduino χωρίς να χρειάζεται επιπλέον κόλληση. Αυτό είναι ιδιαίτερα επιθυμητό στην περίπτωση μας καθώς μας επιτρέπει να κάνουμε την μονάδα όσο το δυνατόν μικρότερη αποφεύγοντας παράλληλα και την ανάγκη δημιουργίας και άλλης πλακέτας. Ωστόσο μένει ακόμα η κατασκευή του κυκλώματος υπερύθρων. Εδώ επιλέχθηκε το εν λόγο κύκλωμα να υλοποιηθεί σε διάτρητη πλακέτα καθώς, όπως θα δούμε παρακάτω, μπορεί και αυτό να κουμπώσει στον απομένοντα χώρο δίπλα από τον GSM module. Τα υλικά που χρειάζονται για την δημιουργία του κυκλώματος υπερύθρων ακολουθούν παρακάτω.

NOR Gate	MC74HC02N
R1	22Ω
R2	4.7KΩ
NPN Transistor	C33725

Πίνακας 2.5: Υλικά που χρησιμοποιήθηκαν για την κατασκευή του κυκλώματος υπέρυθρων

Το τελικό κύκλωμα υπερύθρων καθώς και επίσης ο τρόπος με τον οποίο μπορέσαμε συνδυάσουμε με τα άλλα modules φαίνεται στην εικόνα που ακολουθεί. Όπως μπορούμε να δούμε και στη εικόνα με την χρήση μακριάς «χτένας» μπορούμε να χωρέσουμε το κύκλωμα υπερύθρων ενώ παράλληλα επιτρέπουμε και την πρόσβαση στο module της οθόνης που πρέπει να κουμπώσει από πάνω.

Εικόνα 2.13: Συνδυασμός GSM και κυκλώματος υπέρυθρων

Έχοντας λοιπόν ολοκληρώσει το κύκλωμα υπέρυθρων μπορούμε να συνεχίσουμε με την συναρμολόγηση της μονάδας. Στους παρακάτω πίνακες βλέπουμε το σύνολο με τις συνδεσμολογίες για να επιτευχθεί η λειτουργία του κυκλώματος.

Arduino Mega	**TFT LCD Adapter**
D2	DPENRQ
D3	D_OUT
D4	D_IN
D5	D_CS
D6	D_CLK
D37	DB0
D36	DB1
D35	DB2
D34	DB3
D33	DB4
D32	DB5
D31	DB6
D30	DB7
D22	DB8
D23	DB9
D24	DB10
D25	DB11
D26	DB12
D27	DB13
D28	DB14
D29	DB15
D38	RS
D39	WR
D40	CS
D41	RST

Πίνακας 2.6: Συνδεσμολογία Arduino Mega και οθόνης αφής

Arduino Mega	**SD Socket 1**	**SD Socket 2**
3.3v	VCC	VCC
GND	GND	GND
50 (MISO)	DO	DO
51 (MOSI)	DI	DI
52 (SCK)	SCK	SCK
42	CS	
53 (SS)		CS

Πίνακας 2.7: Συνδεσμολογία Arduino Mega και διεπαφών κάρτας μνήμης SD

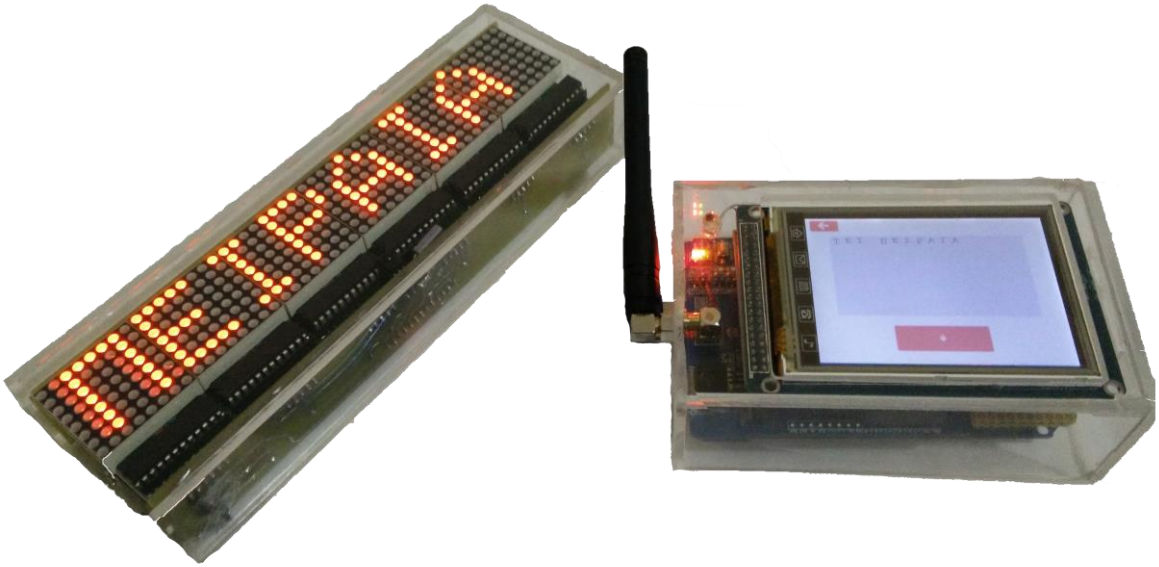
Arduino Mega	**SIM900 Module**
5V	VCC
GND	GND
RX1	TX
TX1	RX
D9	PWR

Πίνακας 2.8: Συνδεσμολογία Arduino Mega και μονάδας GSM

Παρακάτω βλέπουμε και την ολοκληρωμένη κατασκευή της κύριας μονάδας

Εικόνα 2.14: Ολοκληρωμένη κεντρική μονάδα

Τέλος, απομένει ένα ακόμα βήμα για την πλήρης ολοκλήρωση του κατασκευαστικού μέρους και αυτό δεν είναι άλλο από την επιλογή δύο κουτιών για να βάλουμε τις κατασκευές μας. Επιλέχθηκε η κατασκευή κουτιών από πλεξιγκλάς διότι μας επιτρέπει να δημιουργήσουμε κουτιά ακριβώς στο μέγεθος και στις διαστάσεις που θέλουμε χωρίς ιδιαίτερα μεγάλο κόστος. Επίσης η χρήση διάφανου πλεξιγκλάς αφήνει ορατό το εσωτερικό κύκλωμα και τις συνδεσμολογίες της κάθε μονάδας.



Εικόνα 2.15: Ολοκληρωμένη κατασκευή σε κουτιά από πλεξιγκλάς

ΚΕΦΑΛΑΙΟ 3

ΛΟΓΙΣΜΙΚΟ ΚΑΙ ΒΙΒΛΙΟΘΗΚΕΣ

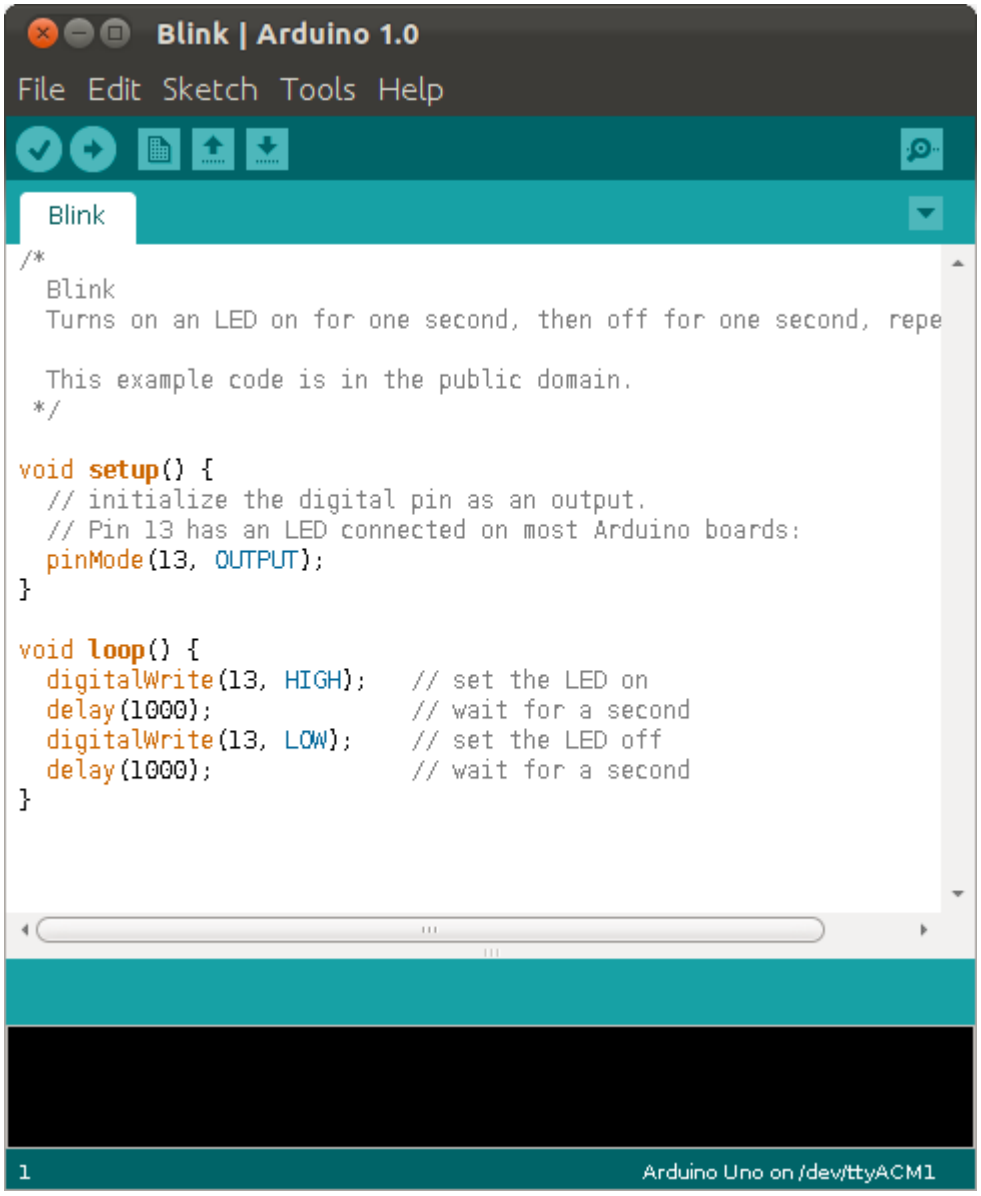
Σε αυτό το κεφάλαιο θα αναφερθούμε στο λογισμικό κομμάτι της πτυχιακής εργασίας. Θα γίνει μια συνοπτική αναφορά στις εφαρμογές που χρειαστήκαν για την συγγραφή του κώδικα της εφαρμογής και στην συνέχεια μια παρουσίαση όλων των βιβλιοθηκών που χρησιμοποιηθήκαν στην εφαρμογή. Τέλος θα γίνει η παρουσίαση της ίδιας της εφαρμογής και θα αναπτυχθούν όλα τα σημαντικά κομμάτια κώδικα ώστε αρχικώς να γίνει αντιληπτός ο τρόπος λειτουργείας της εφαρμογής αλλά και να επιτραπεί στον αναγνώστη η επαναχρησιμοποίηση μέρους του κώδικα σε άλλη, παρόμοια εφαρμογή.

3.1 Εφαρμογές και Προγράμματα

Σε αυτό το υποκεφάλαιο γίνεται μια αναφορά στις εφαρμογές που χρησιμοποιήθηκαν για την ανάπτυξη του λογισμικού της πτυχιακής εργασίας και στον τρόπο που η κάθε μια από αυτές συνέβαλε στην εκπόνηση τής.

3.1.1 Arduino IDE

Το ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) του Arduino είναι μία εφαρμογή γραμμένη σε Java, που λειτουργεί σε πολλές πλατφόρμες, και προέρχεται από το IDE για τη γλώσσα προγραμματισμού Processing και το σχέδιο Wiring. Έχει σχεδιαστεί για να εισαγάγει τον προγραμματισμό στους καλλιτέχνες και τους νέους που δεν είναι εξοικειωμένοι με την ανάπτυξη λογισμικού. Περιλαμβάνει ένα πρόγραμμα επεξεργασίας κώδικα με χαρακτηριστικά όπως είναι η επισήμανση σύνταξης και ο συνδυασμός αγκύλων και είναι επίσης σε θέση να μεταγλωττίζει και να φορτώνει προγράμματα στην πλακέτα με ένα μόνο κλικ. Δεν υπάρχει συνήθως καμία ανάγκη να επεξεργαστούν αρχεία make ή να τρέξουν προγράμματα σε ένα περιβάλλον γραμμής εντολών. Ένα πρόγραμμα ή κώδικας που γράφτηκε για Arduino ονομάζεται σκίτσο (sketch)



Εικόνα 3.1: Προγραμματιστικό περιβάλλον Arduino

Τα Arduino προγράμματα είναι γραμμένα σε C ή C++. Το Arduino IDE έρχεται με μια βιβλιοθήκη λογισμικού που ονομάζεται «Wiring» από το πρωτότυπο σχέδιο Wiring γεγονός που καθιστά πολλές κοινές λειτουργίες εισόδου/εξόδου πολύ πιο εύκολες. Οι χρήστες πρέπει μόνο να ορίσουν δύο λειτουργίες για να κάνουν ένα πρόγραμμα κυκλικής εκτέλεσης:

- setup(): Μία συνάρτηση που τρέχει μία φορά στην αρχή του προγράμματος η οποία αρχικοποιεί τις ρυθμίσεις

- loop(): Μία συνάρτηση η οποία καλείται συνέχεια μέχρι η πλακέτα να απενεργοποιηθεί

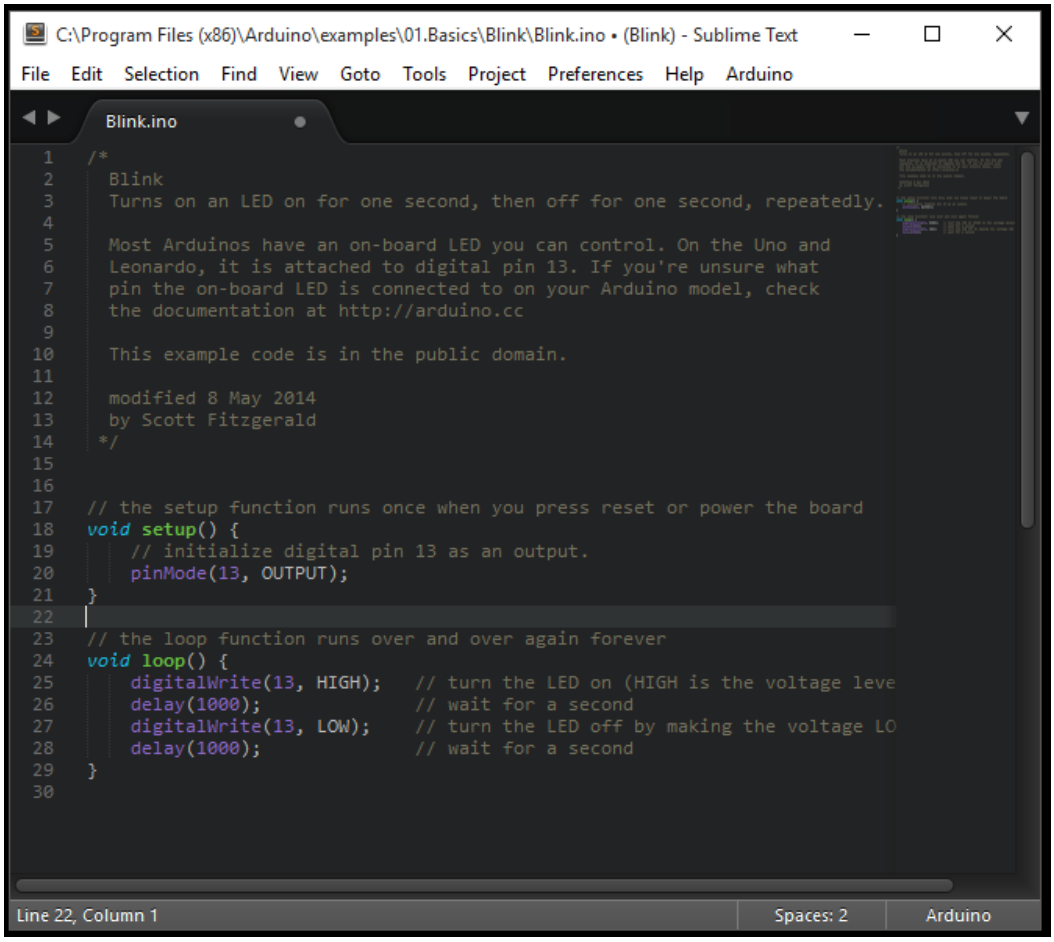
Το IDE του Arduino χρησιμοποιεί το GNU toolchain και το AVR Libc για να μεταγλωττίζει προγράμματα και το avrdude για να φορτώνει προγράμματα στην πλακέτα.

Δεδομένου ότι η πλατφόρμα Arduino χρησιμοποιεί Atmel μικροελεγκτές, το περιβάλλον ανάπτυξης της Atmel, το AVR Studio ή το νεότερη έκδοση του Atmel Studio, μπορεί επίσης να χρησιμοποιηθεί για την ανάπτυξη λογισμικού για το Arduino.

3.1.2 Sublime Text

Το IDE του Arduino είναι αρκετό για ανάπτυξη μικρών προγραμμάτων αλλά γίνεται αμέσως αντιληπτό ότι για πιο μεγάλα και σύνθετα προγράμματα είναι αρκετά ελλιπές.

Για την ανάπτυξη του κώδικα τις πτυχιακής εργασίας χρησιμοποιήθηκε το Sublime Text. Πρόκειται για ένα εξελιγμένο πρόγραμμα επεξεργασίας κειμένου που χρησιμοποιείται συνήθως για ανάπτυξη ιστοσελίδων (PHP, JavaScript, CSS κτλ.).



Εικόνα 3.2: Προγραμματιστικό περιβάλλον Sublime Text

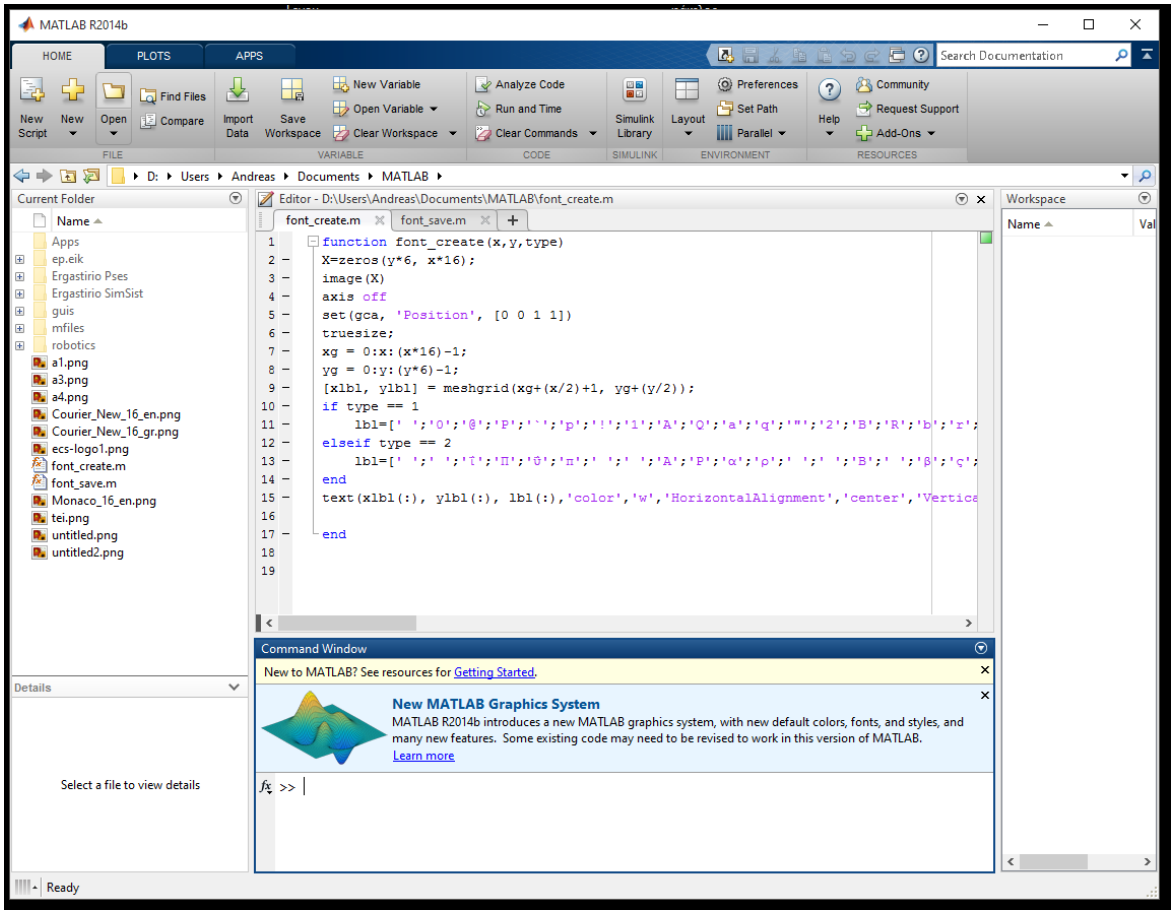
Ένα από το μεγαλύτερα πλεονεκτήματα του είναι η πληθώρα των επεκτάσεων (plugins) που υποστηρίζει το πρόγραμμα, που σε συνδυασμό με την τεράστια παραμετροποιησιμότητα του, επιτρέπει στον εκάστοτε χρήστη να το παραμετροποιεί ακριβώς πάνω στις ανάγκες του.

Μια τέτοια επέκταση είναι και το Stino που επιτρέπει στον χρήστη του Sublime Text να προγραμματίσει το Arduino μέσα από την εφαρμογή, συνδυάζοντας έτσι την απλότητα με την οποία επιτρέπει την μεταφόρτωση ενός προγράμματος το Arduino IDE με τις τεράστιες δυνατότητες του Sublime Text.

Το Sublime Text είναι το βασικό πρόγραμμα που χρησιμοποιήθηκε τόσο για την ανάπτυξη του κώδικα της εφαρμογής όσο και για τον προγραμματισμό των Arduino μας, κάνοντας χρήση της επέκτασης Stino που αναφέρεται παραπάνω.

3.1.3 Matlab

Το MATLAB® είναι ένα διαδραστικό προγραμματιστικό περιβάλλον για υπολογισμούς, ανάλυση, επεξεργασία και απεικόνιση δεδομένων, ανάπτυξη αλγορίθμων και μοντελοποίηση συστημάτων. Επιπλέον παρέχει ένα μεγάλο αριθμό εξειδικευμένων εργαλειοθηκών (toolboxes) που Matlab απευθύνονται σε συγκεκριμένες επιστημονικές εφαρμογές.



Εικόνα 3.3: Περιβάλλον εργασίας Matlab

Οι έτοιμες συναρτήσεις που εμπεριέχονται στο βασικό πακέτο του Matlab και στις εργαλειοθήκες, παρέχουν τη δυνατότητα στον τελικό χρήστη, να τις χρησιμοποιήσει χωρίς ιδιαίτερες γνώσεις προγραμματισμού.

Ένα από το πεδία στα οποία χρησιμοποιείται το Matlab είναι η επεξεργασία εικόνας. Το Matlab μπορεί να μετατρέπει ένα αρχείο εικόνας σε πίνακα όπου στην συνέχεια υπάρχει η δυνατότητα να επεξεργαστεί ο πίνακας με διάφορους τρόπους, από το να μεταβάλουμε την φωτεινότητα ή την αντίθεση μέχρι στο να εφαρμοστούν διάφορα φίλτρα, από πολύ απλά έως πολύπλοκα.

Για την εργασία δημιουργήσαμε μια συνάρτηση (m-function) που μας αφήνει να διαλέξουμε κάποια από τις διαθέσιμες γραμματοσειρές (fonts) και στην συνέχεια αφού την μετατρέψει σε δυαδική εικόνα - εικόνα με μόνο δυο χρώματα , άσπρο και μαύρο - , μας επιτρέπει να κατασκευάσουμε αντίστοιχο .c αρχείο που μπορούμε έπειτα να το χρησιμοποιήσουμε στον κώδικα μας, δίνοντας στην ουσία στον προγραμματιστή την δυνατότητα να χρησιμοποιεί στην LCD οθόνη μια πληθώρα από γραμματοσειρές.

Εικόνα 3.4: Γραφική απεικόνιση γραμματοσειράς για εισαγωγή στην εφαρμογή

Ένα μειονέκτημα ωστόσο είναι το ότι λόγο τις μικρής επεξεργαστικής ισχύς του Arduino δεν μπορούμε να εφαρμόσουμε τεχνικές εξομάλυνσης (antialising) στην απεικόνιση τον γραφικών της οθόνης με αποτέλεσμα κάποιες γραμματοσειρές που έχουν κατασκευαστεί για να είναι ευανάγνωστες με την χρήση antialising να μην είναι άνετες στο διάβασμα στην οθόνη τις εργασίας. Παρόλα αυτά όμως υπάρχουν αρκετές γραμματοσειρές που είναι πολύ ευανάγνωστες και οπτικά ευχάριστες στην κατασκευή μας.

3.2 Βιβλιοθήκες

Ένα από τα κύρια προτερήματα της χρήσης της πλατφόρμας Arduino αντί για κάποιου μεμονωμένου μικροελεγκτή είναι η πληθώρα των βιβλιοθηκών που υπάρχουν διαθέσιμες για την εν λόγω πλατφόρμα.

Η χρήση βιβλιοθηκών μας επιτρέπει να χρησιμοποιήσουμε κάποιες από τις διαθέσιμες λειτουργίες του μικροελεγκτή του Arduino με πιο εύκολο τρόπο η να επιτύχουμε συνδεσμολογία με διάφορα περιφερειακά όπως οθόνες LCD , κάρτες μνήμης κτλ., για τα οποία υπάρχουν διαθέσιμες βιβλιοθήκες, χωρίς να χρειαστεί να ξαναγραφτεί ο κώδικας. Ένα ακόμη προτέρημα αποτελεί και το γεγονός ότι, οι βιβλιοθήκες είναι συνήθως δοκιμασμένες από αρκετά άτομα καθώς διατίθενται δωρεάν στο διαδίκτυο λόγο της ανοικτού λογισμικού (open source) φύσης της πλατφόρμας. Επιπλέον, δεδομένου ότι ο πηγαίος κώδικας διατίθεται όπως αναφέρθηκε στο διαδίκτυο, ο προγραμματιστής μπορεί να τροποποιήσει η να βελτιώσει μια βιβλιοθήκη πάνω στις ανάγκες του ή ακόμα να δημιουργήσει μια τελείως καινούρια βιβλιοθήκη που βασίζεται πάνω σε κάποια άλλη.

Σε αυτό το υποκεφάλαιο θα γίνει μια σύντομη ανάλυση τον βιβλιοθηκών που χρησιμοποιήθηκαν κατά την εκπόνηση της εν λόγω πτυχιακής εργασίας.

3.2.1 Max72xxPanel

Η Max72xxPanel είναι μια βιβλιοθήκη για τον έλεγχο των 8x8 dot-matrix modules. Μέσω του πρωτοκόλλου SPI μας επιτρέπει να οδηγήσουμε τους οδηγούς MAX7219 ή MAX7221 που υπάρχουν πάνω στα dot-matrix modules επιτρέποντας μας έτσι να απεικονίσουμε το ζητούμενο κείμενο στην οθόνη μας.

Τα βασικά χαρακτηριστικά της βιβλιοθήκης είναι:

- Διπλό buffering για να μην τρεμοπαίζει η οθόνη κατά την εναλλαγή η κύλιση του κειμένου.

- Υποστήριξη για πολλαπλές οθόνες Dot-Matrix, τοποθετημένες σε μια αυθαίρετη ορθογώνια διάταξη.
- Με χρήση της βιβλιοθήκης SPI, υποστηρίζει οθόνες Dot-Matrix συνδεμένες σε σειρά.
- Μεγάλη ταχύτητα εκτέλεσης και ελάχιστη χρήση μνήμης.

Από τα παραπάνω γίνετε αντιληπτό ότι η βιβλιοθήκη είναι η βασική βιβλιοθήκη που χρησιμοποιείτε στο Arduino Pro Mini για την οδήγηση της μονάδας Dot-Matrix. Η βιβλιοθήκη δεν μας επιτρέπει μόνο να γράψουμε κείμενο αλλά επίσης τα ορίσουμε διάφορες παραμέτρους όπως πχ την φωτεινότητα της οθόνης.

```
int pinCS = 10; // Attach CS to this pin, DIN to MOSI and CLK to SCK (cf http://arduino.cc/en/Re
int numberOfHorizontalDisplays = 2;
int numberOfVerticalDisplays = 1;

Max72xxPanel matrix = Max72xxPanel(pinCS, numberOfHorizontalDisplays, numberOfVerticalDisplays);

String tape = "Arduino";
int wait = 100; // In milliseconds

int spacer = 1;
int width = 5 + spacer; // The font width is 5 pixels

void setup() {

    matrix.setIntensity(1); // Use a value between 0 and 15 for brightness

}

void loop() {

    for ( int i = 0 ; i < width * tape.length() + matrix.width() - 1 - spacer; i++ ) {

        matrix.fillScreen(LOW);

        int letter = i / width;
        int x = (matrix.width() - 1) - i % width;
        int y = (matrix.height() - 8) / 2; // center the text vertically

        while ( x + width - spacer >= 0 && letter >= 0 ) {
            if ( letter < tape.length() ) {
                matrix.drawChar(x, y, tape[letter], HIGH, LOW, 1);
            }

            letter--;
            x -= width;
        }

        matrix.write(); // Send bitmap to display

        delay(wait);
    }
}
```

Εικόνα 3.5: Παράδειγμα προγραμματισμού με την βιβλιοθήκη Max72xxPanel

Οι συναρτήσεις που περιλαμβάνει η βιβλιοθήκη είναι εξής:

- Max72xxPanel()

Ο constructor της βιβλιοθήκης μας. Δημιουργεί ένα αντικείμενο τύπου Max72xxPanel. Τα ορίσματα που παίρνει είναι το πόδι στο οποίο έχουμε συνδέσει το pin CS (chip select) από τα Dot-Matrix modules και άλλα δυο προαιρετικά ορίσματα που μας επιτρέπουν να ορίσουμε αν έχουμε συνδέσει πάνω από ένα dot-matrix module σε σειρά δίνοντας μας έτσι την δυνατότητα να φτιάξουμε μια συστοιχία από modules. Τα δυο προαιρετικά ορίσματα, ορίζουν τον αριθμό τον dot-matrix modules που είναι συνδεμένα σε οριζόντια και κάθετα αντίστοιχα. Αν δεν τους δώσουμε τιμή περνούν την τιμή 1 οπότε Η συστοιχία μας είναι 1x1 άρα υπάρχει μόνο ένα module.

- setIntensity()

  Η συνάρτηση μας επιτρέπει να ορίσουμε την φωτεινότητα τον dot-matrix module δίνοντας στην συνάρτηση μια τιμή από το 0 -10 για το αντίστοιχο επίπεδο φωτεινότητας που θέλουμε.

- drawChar()

  Η βασική συνάρτηση τις βιβλιοθήκης. Μέσο αυτής μπορούμε να γράψουμε στο dot-matrix module ένα από τους αποθηκευμένους χαρακτήρες της βιβλιοθήκης. Στην βιβλιοθήκη περιέχεται ήδη το αγγλικό αλφάβητο το οποίο χρησιμοποιήσαμε μαζί με το ελληνικό το οποίο δημιουργήθηκε και μπήκε ως επέκταση στην βιβλιοθήκη κατά την διάρκεια τις εκπόνησης της πτυχιακής εργασίας.

3.2.2 Metro

Πρόκειται για μια απλή αλλά σημαντική βιβλιοθήκη. Η βιβλιοθήκη Metro επιτρέπει στον χρήστη να ορίσει διάφορους μετρητές ώστε να μπορεί να επαναλάβει κάποια κομμάτια κώδικα ανά μια χρονική σταθερά.

Με αυτόν τον τρόπο αποφεύγεται η χρήση της συνάρτησης delay() και έτσι το πρόγραμμα μας μπορεί να εκτελείται συνέχεια χωρίς να σταματά την λειτουργία του. Αυτό είναι ιδιαίτερα σημαντικό τόσο στη μονάδα Dot-Matrix όπου πρέπει να γίνετε κύλιση του τρέχοντος μηνύματος με μια συγκεκριμένη ταχύτητα αλλά παράλληλα το πρόγραμμα να μπορεί να λαμβάνει το επόμενο μήνυμα από τον χρήστη χωρίς να σταματάει η ομαλή κύλιση του μηνύματος. Ομοίως και η οθόνη αφής πρέπει να δέχεται τα πατήματα από τον χρήστη ενώ παράλληλα θα πρέπει να λαμβάνει το εισερχόμενο SMS ή να αναβαθμίσει τα γραφικά που απεικονίζονται στην οθόνη.

Εδώ πρέπει να τονίσουμε ότι οι μικροελεγκτές της Atmel που χρησιμοποιούνται στα Arduino δεν υποστηρίζουν νήματα (threads), εκτός από το Arduino Due που βασίζεται σε μικροελεγκτή τύπου ARM, ωστόσο όμως με την χρήση της βιβλιοθήκης Metro, αν ο προγραμματιστής φροντίσει τα κομμάτια του κώδικα που εκτελούνται να χρειάζονται μικρό χρόνο εκτέλεσης, μπορούμε να δώσουμε στον χρήστη την ψευδαίσθηση ότι οι λειτουργίες γίνονται παράλληλα αφού λόγο της μεγάλης ταχύτητας εκτέλεσης δεν θα γίνετε αντιληπτή κάποιου είδους καθυστέρηση.

```
#include <Metro.h> //Include Metro library
#define LED 13 // Define the led's pin

//Create a variable to hold theled's current state
int state = HIGH;

Metro ledMetro = Metro(250);

void setup()
{
  pinMode(LED,OUTPUT);
  digitalWrite(LED,state);
}

void loop()
{

  if (ledMetro.check() == 1) {
    if (state==HIGH)  state=LOW;
    else state=HIGH;

    digitalWrite(LED,state);
  }
}
```

Εικόνα 3.6: Παράδειγμα προγραμματισμού με την βιβλιοθήκη Metro

Οι συναρτήσεις που περιλαμβάνει η βιβλιοθήκη είναι εξής:

- Metro()

  Ο constructor της βιβλιοθήκης Metro. Παίρνει μόνο ένα όρισμα που είναι ο χρόνος τον οποίο θέλει να μετρήσουμε σε millisecond

- check()

  Η συνάρτηση check() επιστρέφει 1 όταν ο χρόνος που βάλαμε να μετρήσει έχει περάσει και 0 όταν δεν έχει περάσει ακόμα.
  Από τον τρόπο με τον οποίο λειτουργεί η συνάρτηση μπορούμε να καταλάβουμε ότι ο προγραμματιστής πρέπει να εκτελεί την check() ανά τακτά χρονικά διαστήματα ώστε να φροντίσει να ενημερωθεί ότι έφτασε ο χρόνος που περίμενε (μέθοδος polling). Εδώ βλέπουμε και το μειονέκτημα της metro σε σχέση με κάποια υλοποίηση με διακοπές (interrupts), ο προγραμματιστής ενημερώνεται αν πέρασε η όχι ο χρόνος όταν εκτελεστεί η check() και όχι ακριβώς την στιγμή που το χρονικό διάστημα που ζητείται πέρασε. Σε περιπτώσεις λοιπόν που απαιτείται μεγάλη ακρίβεια ενδείκνυται η χρήση κάποιου hardware timer που θα είναι ακριβής. Σε αντίθετη περίπτωση και με δεδομένο ότι ο κώδικας είναι σωστά γραμμένος και η συνάρτηση check() εκτελείται ανά τακτά διαστήματα η βιβλιοθήκη Metro μπορεί να αποτελέσει ένα πολύ σημαντικό εργαλείο.

3.2.3 UTFT

Η UTFT είναι μια βιβλιοθήκη γραφικών που προσφέρει στον προγραμματιστή την δυνατότητα να χειριστεί μια πληθώρα από 8-bit , 16-bit και σειριακές TFT οθόνες χωρίς να χρειάζεται να γράψει συγκεκριμένο κώδικα για κάθε μια από αυτές. Το μεγαλύτερο μέρος τις δουλειάς που γλιτώνει ο προγραμματιστής με την χρήση της βιβλιοθήκης είναι η ανάγκη να γράψει άλλο κώδικα κάθε φορά που θα θέλει να χρησιμοποιήσει το ίδιο πρόγραμμα σε ένα διαφορετικό μοντέλο οθόνης.

Η βιβλιοθήκη προσφέρει μεθόδους για το σχεδιασμό γραμμών και απλών γεωμετρικών σχημάτων όπως ορθογώνια και κύκλους, όπως επίσης το γράψιμο κειμένου σε μορφή δυαδικού bitmap αλλά και έγχρωμες εικόνες σε μορφή bitmap. Τόσο τα γράμματα όσο και οι εικόνες πρέπει να μετατραπούν σε πίνακες για να τα χρησιμοποιήσει η βιβλιοθήκη. Το αγγλικό αλφάβητο υπάρχει ήδη στην βιβλιοθήκη και σε δυο μεγέθη γραμματοσειράς, 16x16 pixel και 8x12 pixel ανά γράμμα. Ο χρήστης μπορεί να δημιουργήσει ο ίδιος άλλα μεγέθη γραμματοσειράς η τελείως καινούριες γραμματοσειρές.

Για αυτόν ακριβώς τον λόγο, όπως θα αναφερθούμε και παρακάτω πιο αναλυτικά, δημιουργήσαμε μια συνάρτηση με το πρόγραμμα Matlab που μπορεί να μετατρέψει οποιαδήποτε γραμματοσειρά υπάρχει στο εν λόγω πρόγραμμα σε πίνακα για να γίνει χρήση στην εφαρμογή μας.

```
#include <UTFT.h>
#include <avr/pgmspace.h>

// Declare which fonts we will be using
extern uint8_t SmallFont[];

UTFT myGLCD(ITDB32, 38, 39, 40, 41);

extern unsigned int info[0x400];
extern unsigned int icon[0x400];
extern unsigned int tux[0x400];

void setup() {
    myGLCD.InitLCD();
    myGLCD.setFont(SmallFont);
}

void loop() {
    myGLCD.fillScr(255, 255, 255);
    myGLCD.setColor(255, 255, 255);
    myGLCD.print(" *** A 10 by 7 grid of a 32x32 icon *** ", CENTER, 228);
    for (int x = 0; x < 10; x++)
        for (int y = 0; y < 7; y++)
            myGLCD.drawBitmap (x * 32, y * 32, 32, 32, info);

    delay(5000);

    myGLCD.fillScr(255, 255, 255);
    myGLCD.setColor(255, 255, 255);
    myGLCD.print("   Two different icons in scale 1 to 4  ", CENTER, 228);
    int x = 0;
    for (int s = 0; s < 4; s++) {
        x += (s * 32);
        myGLCD.drawBitmap (x, 0, 32, 32, tux, s + 1);
    }
    x = 0;
    for (int s = 4; s > 0; s--) {
        myGLCD.drawBitmap (x, 224 - (s * 32), 32, 32, icon, s);
        x += (s * 32);
    }

    delay(5000);
}
```

Εικόνα 3.7: Παράδειγμα προγραμματισμού με την βιβλιοθήκη UTFT

Παρακάτω αναφέρονται μερικές από τις συναρτήσεις τις βιβλιοθήκης UTFT που χρησιμοποιήσαμε εκτεταμένα κατά την συγγραφή του προγράμματος μας.

- UTFT()

  Δημιουργεί ένα αντικείμενο τύπου UTFT. Τα ορίσματα που πρέπει να εισάγουμε στην συνάρτηση είναι το μοντέλο της οθόνης όπως επίσης και έναν αριθμό από Arduino pins στα οποία έχει συνδεθεί η οθόνη. Όπως αναφέρθηκε και παραπάνω η βιβλιοθήκη υποστηρίζει μια πληθώρα από

οθόνες. Στον φάκελο tft_driver τις βιβλιοθήκης μπορούμε να δούμε τον αριθμό τον οθονών που υποστηρίζονται αυτήν την στιγμή. Εδώ πρέπει να τονίσουμε ότι ο δημιουργός της βιβλιοθήκης συνεχίζει να αναβαθμίζει την βιβλιοθήκη, προσφέροντας συμβατότητα με καινούριες οθόνες που κυκλοφορούν.

- InitLCD()

  Η συνάρτηση που ξεκινάει την οθόνη μας. Η συνάρτηση είναι απαραίτητη για να την λειτουργία τις οθόνης. Πρέπει απλά να εκτελεστεί κατά την εκκίνηση του προγράμματος.

- getDisplayXSize() / getDisplayXSize()

  Οι δυο συναρτήσεις επιστρέφουν το μέγεθος της αντίστοιχης πλευράς της οθόνης σε pixel.

- clrScr()

  Καθαρίζει ότι είναι σχεδιασμένο στην οθόνη. Το φόντο της οθόνης (background) γίνετε μαύρο.

- fillScr()

  Γεμίζει την οθόνη με ένα χρώμα που του ορίζει ο χρήστης. Για να δηλωθεί το χρώμα που ζητείται πρέπει να πάρουν τιμές τα τρία ορίσματα που αντιπροσωπεύουν τρία βασικά χρώματα RGB, Κόκκινο (Red), Πράσινο (Green) και Μπλε (Blue). Οι τιμές που μπορούν να πάρουν τα χρώματα είναι από 0 έως 255. Επιλέγοντας διαφορετικό συνδυασμό των τριών βασικών χρωμάτων μπορούμε να σχεδιάσουμε όλη την χρωματική παλέτα.

- setColor()

Η συνάρτηση μας επιτρέπει να ορίσουμε το χρώμα που θα χρησιμοποιηθεί από τις επόμενες συναρτήσεις για να σχεδιάσουν διάφορα γεωμετρικά σχήματα η να αποτυπωθεί κάποιο κείμενο στην οθόνη. Όπως και παραπάνω τα ορίσματα είναι οι τιμές των τριών βασικών χρωμάτων RGB.

- drawLine()

  Σχεδιάζει μια ευθεία γραμμή από τις συντεταγμένες που μας δίνουν τα δυο πρώτα ορίσματα έως τις συντεταγμένες που μας δίνουν τα επόμενα δυο ορίσματα. Το χρώμα της ευθείας ορίζεται από την συνάρτηση setColor() που αναφέραμε παραπάνω.

- drawRect() / drawRoundRect()

  Σχεδιάζει ένα ορθογώνιο χρησιμοποιώντας τα δυο ζεύγη συντεταγμένων που παίρνει ως ορίσματα. Ομοίως με παραπάνω, η συνάρτηση setColor() ορίζει το χρώμα που θα χρησιμοποιηθεί για τον σχεδιασμό του ορθογώνιου. Υπάρχει επίσης και μια παραλλαγή της drawRect() η drawRoundRect() που ομοίως σχεδιάζει ένα ορθογώνιο αλλά με μοναδική διαφορά ότι πλέον το ορθογώνιο σχεδιάζεται με στρογγυλεμένες γωνιές.

- fillRect() / fillRoundRect()

  Παρόμοια με την drawRect() οι συναρτήσεις σχεδιάζουν ορθογώνια είτε με στρογγυλεμένες γωνιές είτε με μη αλλά επιπλέον γεμίζουν το εμβαδό του ορθογώνιου με το χρώμα που έχει οριστεί από την setColor()

- setFont()

  Επιτρέπει την επιλογή αρχείου – πίνακα με γραμματοσειρά (font) για να χρησιμοποιηθεί από την συνάρτηση print() για την σχεδίαση κειμένου στην οθόνη. Η βιβλιοθήκη έχει εξ ορισμού δυο πίνακες με γραμματοσειρές 8 x 12 και 16 x 16 pixel αντίστοιχα αλλά, όπως αναφέραμε και στην παράγραφο για

το Matlab μπορούμε με το αναφερθέν εργαλείο να δημιουργήσουμε και άλλους πίνακες με γραμματοσειρές της επιλογής μας.

- print()

  Η συνάρτηση μα επιτρέπει την σχεδίαση κειμένου στην οθόνη. Σαν ορίσματα παίρνει το κείμενο που θέλουμε να αποτυπώσουμε και ένα ζεύγος συντεταγμένων από το οποίο θα ξεκινήσει να σχεδιάζεται το εν λόγω κείμενο. Εδώ πρέπει να αναφέρουμε ότι είναι στην αρμοδιότητα του προγραμματιστεί να φροντίσει το κείμενο που θέλει να αποτυπώσει, να χωρέσει στην οθόνη. Όπως και με τις προηγούμενες συναρτήσεις το χρώμα που θα έχουν τα γράμματα του κειμένου ορίζεται από την συνάρτηση setColor().

- setBackColor()

  Αντίθετα τον σχεδιασμό των γεωμετρικών σχημάτων η αποτύπωση κειμένου με την εντολή print() χρειάζεται παραπάνω από ένα χρώμα. Το χρώμα των γραμμάτων που το ορίζουμε με την χρήση της setColor() αλλά και το χρώμα που θα έχει το φόντο (background) πίσω από κάθε γράμμα. Αυτό το χρώμα μας επιτρέπει να ορίσουμε η συνάρτηση setBackColor().

- getFontXsize() / getFontYsize()

  Μας επιστρέφουν το μέγεθος σε pixel, για την κάθε διάσταση αντίστοιχα, της γραμματοσειράς που χρησιμοποιείται. Χρήσιμα στο να κεντράρουμε το κείμενο ή για να δούμε αν το κείμενο που θέλουμε να αποτυπώσουμε χωράει στην οθόνη.

- drawBitmap()

  Επιτρέπει των σχεδιασμό εικόνας που έχει αποθηκευτεί σε πίνακα. Τα ορίσματα που πρέπει να εισαχθούν είναι το ζεύγος συντεταγμένων από το οποίο θα ξεκινήσει να σχεδιάζεται η εικόνα το μέγεθος της εικόνας σε pixel

αλλά και φυσικά ο πίνακας χρωμάτων που αντιπροσωπεύει την εικόνα που θέλουμε να σχεδιάσουμε. Ο δημιουργός της βιβλιοθήκης UTFT παρέχει στην ιστοσελίδα της βιβλιοθήκης online εργαλείο που μετατρέπει εικόνες από διαφόρους τύπους αρχείων εικόνας (png, jpeg κτλ.) σε πίνακα για να χρησιμοποιηθεί από την βιβλιοθήκη.

3.2.4 UTouch

Αν και αυτοτελής βιβλιοθήκη αποτελεί συμπλήρωμα της βιβλιοθήκης UTFT αφού και οι δυο βιβλιοθήκες χρησιμοποιούνται πάνω στο ίδιο υλικό (hardware) δηλαδή σε διάφορα μοντέλα οθονών TFT μικρού μεγέθους. Αντίθετα με την UTFT που έχει ως σκοπό την σχεδίαση γραφικών σε οθόνες TFT μικρού μεγέθους, η UTouch προσφέρει την δυνατότητα να διαβάζει τα αγγίγματα σε οθόνη αφής.

```
#include <UTFT.h>
#include <UTouch.h>

#define D_CLK 42
#define D_CS 43
#define D_DIN 44
#define D_OUT 45
#define DPENRQ 46

UTFT        myGLCD(ITDB32, 38, 39, 40, 41);

UTouch myTouch(D_CLK, D_CS, D_DIN, D_OUT, DPENRQ);

void setup() {
    myGLCD.InitLCD();
    myGLCD.clrScr();

    myTouch.InitTouch();
    myTouch.setPrecision(PREC_MEDIUM);
}

void loop() {
    long x, y;

    while (myTouch.dataAvailable() == true) {
        myTouch.read();
        x = myTouch.getX();
        y = myTouch.getY();
        if ((x != -1) and (y != -1)) {
            myGLCD.drawPixel (x, y);
        }
    }
}
```

Εικόνα 3.8: Παράδειγμα προγραμματισμού με την βιβλιοθήκη UTouch

Οι συναρτήσεις για να επιτευχθεί η ανάγνωση των αγγιγμάτων είναι οι παρακάτω:

- UTouch()

  Ο constructor της βιβλιοθήκης. Δημιουργεί ένα αντικείμενο UTouch. Τα ορίσματα που απαιτούνται είναι οι ακροδέκτες της οθόνης στους οποίους έχει συνδεθεί το Arduino.

- InitTouch()

  Η συνάρτηση που πρέπει να εκτελεστεί κατά την εκκίνηση του προγράμματος για να ξεκινήσει η βιβλιοθήκη να καταγράφει τα αγγίγματα στην οθόνη αφής.

- dataAvailable()

  Επιστρέφει τιμή 1 όταν ο χρήστης έχει κάνει κάποιο άγγιγμα στην οθόνη αφής, σε διαφορετική περίπτωση επιστρέφει 0. Η συνάρτηση πρέπει να εκτελείται ανά τακτά χρονικά διαστήματα κατά την λειτουργία του προγράμματος για να μπορέσει η εφαρμογή να ενημερωθεί για κάποιο τυχόν άγγιγμα (τεχνική polling).

- read()

  Διαβάζει και αποθηκεύει το ζεύγος συντεταγμένων στο οποίο έγινε το τελευταίο άγγιγμα στην οθόνη αφής. Πρέπει να εκτελεστεί εφόσον η dataAvailable() επιστρέψει 1.

- getX() / getY()

  Επιστρέφουν την αντίστοιχη συντεταγμένη το καθένα που έχει αποθηκευτεί από την read()

- setPrecision()

  Ορίζει την ακρίβεια με την οποία θα γίνεται η μέτρηση των συντεταγμένων από την βιβλιοθήκη. Όσο πιο μεγάλη ακρίβεια τόσο περισσότερος χρόνος χρειάζεται για να γίνει η μέτρηση και το αντίστροφο.

3.2.5 Sdfat

Η βιβλιοθήκη Sdfat δίνει την δυνατότητα στο Arduino να συνδέεται και αναγνωρίζει κάρτες μνήμης τύπου SD / MicroSD οι οποίες είναι διαμορφωμένες (format) σε FAT και FAT32 συστήματα αρχείων (file format). Η βιβλιοθήκη είναι ίσος η πιο σύνθετη από αυτές που χρησιμοποιούνται κατά την εκπόνηση αυτής της πτυχιακής εργασίας. Οι συναρτήσεις που παρέχονται με την βιβλιοθήκη επιτρέπουν στον χρήστη να εκτελέσει σχεδόν όλες τις αντίστοιχες λειτουργίες που μας προσφέρουν τα λειτουργικά τύπου UNIX (Ubuntu , Centos, Debian κτλ.) όπως για παράδειγμα ls, mkdir κτλ.

Εδώ πρέπει να τονίσουμε ότι Η υλοποίηση κάποιων από των συναρτήσεων δεν είναι τόσο σύνθετη όσο η αντίστοιχη του λειτουργικού αλλά κάτι τέτοιο είναι απόλυτα λογικό καθώς η βιβλιοθήκη προορίζεται για χρήση με την πλατφόρμα Arduino και οι διαθέσιμοι πόροι (επεξεργαστική ισχύς, μέγεθος μνήμης) είναι πολύ πιο περιορισμένοι.

```
#include <SdFat.h>

const uint8_t chipSelect = SS;

SdFat sd;

SdFile file;

ArduinoOutStream cout(Serial);

void setup() {
  Serial.begin(9600);
  while (!Serial) {} // wait for Leonardo
  delay(1000);

  if (!sd.begin(chipSelect, SPI_HALF_SPEED)) sd.initErrorHalt();

  while (file.openNext(sd.vwd(), O_READ)) {
    file.printName(&Serial);
    cout << ' ';
    file.printModifyDateTime(&Serial);
    cout << endl;
    file.close();
  }
  cout << "\nDone!" << endl;
}
```

Εικόνα 3.9: Παράδειγμα προγραμματισμού με την βιβλιοθήκη SdFat

Οι τρεις βασικές κλάσεις της βιβλιοθήκης που χρησιμοποιήθηκαν στον κώδικα της πτυχιακής εργασίας ίνα οι SdFat, SdFile και ifstream. Η βιβλιοθήκη διαθέτει πολλές παραπάνω κλάσεις που απλώς δεν αναφέρονται γιατί δεν χρησιμοποιήθηκαν. Ακόμα και οι κλάσεις που χρησιμοποιήθηκαν έχουν πολλές παραπάνω μεθόδους οι οποίες δεν χρειάστηκαν κατά την συγγραφή του κώδικα.

- SdFat

  Αποτελεί το βασικό αντικείμενο της βιβλιοθήκης. Είναι απαραίτητο για την αρχικοποίηση της κάρτας μνήμης που έχει συνδεθεί με το Arduino. Μπορούν να δημιουργηθούν πάνω από ένα αντικείμενα που το καθένα αντιπροσωπεύει και από μια κάρτα μνήμης που είναι συνδεμένη πάνω στον μικροελεγκτή μας.

  Οι επόμενες μέθοδοι που αναλύονται παρακάτω ανήκουν στην κλάση SdFat.

- begin()

Ξεκινά την επικοινωνία με την κάρτα μνήμης. Καθώς οι κάρτες συνδέονται στο Arduino μέσο του πρωτοκόλλου SPI, το πρώτο όρισμα της συνάρτησης είναι ο ακροδέκτης του Arduino που έχουμε συνδέσει το CS (chip select) της κάρτας μας. Το δεύτερο όρισμα είναι η ταχύτητα επικοινωνίας ανάμεσα στο Arduino και την κάρτα. Συνήθως είναι προτιμότερο να χρησιμοποιηθεί η τιμή που αντιστοιχεί στην μέγιστη ταχύτητα, αλλά υπάρχουν και περιπτώσεις όπου λόγο διαφόρων εξωτερικών παραγόντων, όπως θόρυβος από κάποιο διπλανό υλικό, κακή κόλληση, υλοποίηση σε breadboard κτλ., όπου δεν εφικτή η επικοινωνία σε μέγιστη ταχύτητα. Σε τέτοιες περιπτώσεις μπορούμε να ορίσουμε μια πιο μικρή ταχύτητα ανάγνωσης / εγγραφής για να επιτευχθεί ομαλή επικοινωνία.

- vwd()

  Επιστρέφει έναν δείκτη (pointer) στο φάκελο στον οποίο βρισκόμαστε.

- chvol()

  Αλλάζει τον τωρινό φάκελο. Με αυτήν την μέθοδο μπορούμε να αλλάξουμε από την μια κάρτα μνήμης στην άλλη αν έχουμε πάνω από μια κάρτες μνήμης συνδεμένες με το Arduino.

- rewind()

  Πηγαίνει στην αρχή του φακέλου στον οποίο βρισκόμαστε.

- curPosition()

  Το τωρινό σημείο στο οποίο βρισκόμαστε σε ένα φάκελο η ένα αρχείο. Αυτό είναι απαραίτητο καθώς η βιβλιοθήκη τεχνολογεί ένα φάκελο σειριακά.

- seekSet()

Ορίζει που βρίσκετε ένα αρχείο. Το όρισμα είναι η “απόσταση” σε byte από την αρχή του φακέλου.

- SdFile

  Είναι το αντικείμενο που αντιπροσωπεύει το αρχείο που θέλουμε να προσπελάσουμε στην μνήμη. Με τις παρακάτω μεθόδους μπορούμε να κάνουμε μεταξύ άλλων τις βασικές λειτουργίες που επιτρέπει ένα αρχείο όπως άνοιγμα, ανάγνωση, εγγραφή κτλ.

- open()

  Ανοίγει το αρχείο που του δίνουμε στο πρώτο όρισμα. Στο δεύτερο όρισμα ορίζουμε τον τρόπο με τον οποίο θα ανοιχτεί το αρχείο. Για παράδειγμα άνοιγμα για ανάγνωση, άνοιγμα για εγγραφή, για ανάγνωση και εγγραφή κτλ.

- openNext()

  Ανοίγει το άμεσος επόμενο αρχείο στον φάκελο από εκείνο που ήταν προηγουμένως ανοιχτό

- isFile()

  Μας ενημερώνει αν το «αρχείο» που ανοίξαμε είναι πραγματικά αρχείο η όχι. Οι φάκελοι θεωρούνται και αυτοί αρχεία τα οποία είναι δείκτες σε άλλα αρχεία.

- isDir()

  Ενημερώνει αν το αρχείο που έχει ανοιχτεί είναι φάκελος και όχι μεμονωμένο αρχείο.

- write()

Γραφεί το όρισμα που του περνάμε στο αρχείο.

- print()

  Παρόμοια με την write() αλλά χρησιμοποιεί την ίδια δομή με την print() που χρησιμοποιείται από το Arduino

- fgets()

  Διαβάζει από το αρχείο όσα byte του ορίζουμε. Σαν τρίτο όρισμα μπορούμε να του περάσουμε ένα χαρακτήρα στον οποίο θα σταματάει να διαβάζει. Για παράδειγμα να του δώσουμε τον χαρακτήρα του τέλους μια γραμμής \n (line terminator), τότε θα διαβάσει μόνο μια γραμμή κειμένου από το αρχείο.

- close()

  Κλείνει το αρχείο που έχουμε ανοίξει. Είναι πολύ σημαντική συνάρτηση καθώς όλες οι αλλαγές που γίνονται στο αρχείο με τις προηγούμενες μεθόδους επιφέρουν τις αλλαγές τους στο προσπελασμένο αντίγραφο του αρχείου που αποθηκεύεται προσωρινά στην μνήμη. Για να γίνει η εγγραφή και στο πραγματικό αρχείο που βρίσκεται στην κάρτα μνήμης πρέπει να καλέσουμε την close().

- ifstream

  Αντικείμενο τύπου stream (από τα streams της γλώσσας προγραμματισμού C++)

- getline()

  Αποθηκεύει στον πίνακα που έχουμε ορίσει (buffer) την γραμμή μου αναγνώστηκε από το ifstream. Ουσιαστικά είναι το ίδιο με το fgets αλλά σε υλοποίηση C++ με streams.

ΚΕΦΑΛΑΙΟ 4

ΕΦΑΡΜΟΓΗ - ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ

Στο κεφάλαιο που ακολουθεί γίνετε η ανάλυση του κυρίως προγράμματος της πτυχιακής εργασίας. Θα μελετηθεί τόσο ο κώδικας που γράφτηκε για το Dot-matrix πάνελ (Arduino Pro Mini) όσο και ο κώδικας για την κύρια μονάδα της κατασκευής που υλοποιείται με το Arduino Mega.

Ο κώδικας για το Dot-Matrix πάνελ γράφτηκε με τον κλασικό τρόπο συγγραφής κώδικα για την πλατφόρμα Arduino, δηλαδή σε ένα αρχείο .ino. Για την συγγραφή του κώδικα για την κύρια μονάδα προτιμήθηκε, λόγο του αρκετά μεγάλου μεγέθους του, ο κώδικας να γραφτεί σε πολλαπλά αρχεία/ κλάσεις. Με αυτόν το τρόπο είναι πιο εύκολη η διαχείριση και η παραμετροποίηση του προγράμματος όπως και επίσης η αποσφαλμάτωση του κατά την διάρκεια της συγγραφής. Επιπλέον, κάποιες από τις κλάσεις μπορούν πολύ εύκολα να χρησιμοποιηθούν και σε άλλες εφαρμογές με παρεμφερείς λειτουργίες. Παραδείγματος χάρι η κλάση για την επικοινωνία του GSM θα μπορούσε πολύ εύκολα να αναπτυχθεί σε ολοκληρωμένη βιβλιοθήκη.

Θα γίνει επίσης αναφορά στην βιβλιοθήκη UTFTGui που αναπτύχθηκε εξολοκλήρου στο πλαίσιο της εν λόγω πτυχιακής εργασίας και δίνει την δυνατότητα δημιουργίας πιο σύνθετων γραφικών για την οθόνη αφής. Όπως αφήνει να εννοηθεί και η ονομασία της η βιβλιοθήκη είναι προέκταση των βιβλιοθηκών UTFT/ UTouch που αναφέρθηκαν στο προηγούμενο κεφάλαιο.

Εδώ θα πρέπει να σημειωθεί στο κεφάλαιο που ακολουθεί δεν θα γίνετε ανάλυση του κώδικα γραμμή-γραμμή αλλά, όπως και στο προηγούμενο κεφάλαιο που αναφερόταν στις βιβλιοθήκες θα γίνετε μια γενική ανάλυση του κώδικα. Σκοπός του κεφαλαίου είναι ο αναγνώστης να κατανοήσει τον τρόπο λειτουργίας της κατασκευής ώστε αν θελήσει να μπορέσει να χρησιμοποιήσει κάποιο μέρος του προγράμματος στην συγγραφή περαιτέρω κώδικα, καθώς όπως θα γίνει αντιληπτό και στο τέλος του κεφαλαίου ο κώδικας έχει γραφτεί με τέτοιον τρόπο ώστε να είναι

εύκολα παραμετροποιήσημος. Επίσης ένα μεγάλο μέρος του είναι στην μορφή βιβλιοθήκης και μπορεί πολύ εύκολα να χρησιμοποιηθεί ανεξάρτητα σε άλλο πρόγραμμα.

4.1 Arduino Pro Mini

Παρακάτω θα γίνει μελέτη του κώδικα της μονάδας Dot-Matrix. Ο κώδικας τόσο ποσοτικά όσο και ποιοτικά είναι πολύ πιο απλός από εκείνον της κεντρικής μονάδας και για αυτόν τον λόγο επιλέχθηκε να γραφτεί όλος σε ένα αρχείο.

Ουσιαστικά οι δυο λειτουργίες που πρέπει να εκτελεί η μονάδα είναι η παρουσίαση του κειμένου στο Dot-Matrix πάνελ και η ανάγνωση και επεξεργασία του μηνύματος που λαμβάνει από την κεντρική μονάδα.

Για την παρουσίαση του κειμένου γίνεται χρήση της συνάρτησης dotMatrix() η οποία χρησιμοποιώντας τη βιβλιοθήκη Max72xxPanel είναι υπεύθυνη για τον σχεδιασμό και την κύλιση του μηνύματος στο πάνελ. Η συνάρτηση παίρνει ως όρισμα εκτός από το ίδιο το μήνυμα και άλλες παραμέτρους που ορίζουν, την κατεύθυνση της κύλισης, τον τρόπο σχεδιασμού του κειμένου, την ταχύτητα της κύλισης και την φωτεινότητα των φωτοδιόδων του πάνελ. Όλες οι παράμετροι ξεκινάνε με κάποιες αρχικές τιμές, παραδείγματος χάρη η φωτεινότητα του κειμένου είναι ορισμένη στην ελάχιστη τιμή, και οι τιμές τους μπορούν να αλλάξουν κάνοντας χρήση του κατάλληλου μενού της κεντρικής μονάδα, για τις παραμέτρους, και με SMS η με χρήση της κεντρικής μονάδας για το κείμενο.

```
void dotMatrix(Max72xxPanel matrix, String message, uint8_t
    fontWidth, uint8_t space, direction dir, uint8_t color) {

    static int i = 0;

    matrix.fillScreen(color);
    int letter = i / fontWidth;
    int x = matrix.width() - (i % fontWidth);
    int y = (matrix.height() - 8) / 2; // center the text vertically

    while ( x + fontWidth - space >= 0 && letter >= 0 ) {
        if ( letter < message.length() )
            matrix.drawChar(x, y, message[letter], !color, color, 1);
        letter--;
        x -= fontWidth;
    }

    matrix.write(); // Send bitmap to display

    if (dir == SCRLEFT)
        i = (i < fontWidth * message.length() + matrix.width() - 1 -
            space) ? i += 1 : i = 0;
    else
        i = (i > 0) ? i -= 1 : i = fontWidth * message.length() +
            matrix.width() - space;
    dm_done_flag = (i == 0) ? true : false;
}
```

Εικόνα 4.1: Κλάση dotMatrix

Η ανάγνωση του μηνύματος από την κεντρική μονάδα γίνεται μέσο της συριακής θήρας στην οποία είναι συνδεμένο το κύκλωμα ανάγνωσης υπέρυθρων. Την στιγμή που γίνεται λήψη ενός υπέρυθρου σήματος η μονάδα διαβάζει τα πρώτα τέσσερα ψηφία του μηνύματος που αντιστοιχούν στην κεφαλίδα. Αφού γίνεται επιβεβαίωση μέσο της κεφαλίδας ότι το μήνυμα προέρχεται από την κεντρική μονάδα και όχι από κάποια άλλη συσκευή που παρεμβάλλεται, παραδείγματος χάρη κάποιο τηλεκοντρόλ τηλεόρασης η μονάδας κλιματισμού, το μήνυμα χωρίζεται σε δυο κατηγορίες ανάλογα την κεφαλίδα.

Η πρώτη κατηγορία είναι το μήνυμα κειμένου, σε αυτήν την περίπτωση γίνεται αποθήκευση του μηνύματος και στην συνέχεια αρχίζει η επαναλαμβανόμενη προβολή του στο πάνελ της μονάδας. Η δεύτερη κατηγορία είναι το μήνυμα ρυθμίσεων, εδώ το μήνυμα αποκωδικοποιείται και στην συνέχεια εκχωρούνται οι τιμές που περιχέει στις μεταβλητές που αντιπροσωπεύουν την κάθε ρύθμιση και έπειτα οι αλλαγές εφαρμόζονται στην λειτουργία της μονάδας.

```
if (Serial.available() > 0 ) {

    gsm_char = Serial.readBytesUntil('\0', responce_buffer, sizeof(
        responce_buffer) - 1);
    responce_buffer[gsm_char] = '\0';

    if (strncmp(responce_buffer, HEADER1, 4) == 0) {
        dotmatrix_string = &responce_buffer[4];
        dotmatrix_string = " " + dotmatrix_string + " ";
    }
    if (strncmp(responce_buffer, HEADER2, 4) == 0) {
        matrix.setIntensity(atoi(&responce_buffer[9]));
        if (responce_buffer[5] == 'N') {
            color = LOW;
        } else if (responce_buffer[5] == 'R') {
            color = HIGH;
        }
        if (responce_buffer[4] == 'B') {
            left_and_right = true;
            step = SCRLEFT;
        } else if (responce_buffer[4] == 'L') {
            left_and_right = false;
            step = SCRLEFT;
        } else if (responce_buffer[4] == 'R') {
            left_and_right = false;
            step = SCRRIGHT;
        }
        convertStr();
    }
}
```

Εικόνα 4.2: Κώδικας διαχείρισης μηνύματος σειριακής επικοινωνίας

Το κορδόνι κειμένου (string) που αντιπροσωπεύει έχει την μορφή BR1000. Ο πρώτος χαρακτήρας αποτελεί τον συμβολισμό για την κατεύθυνση της κύλισης του κειμένου. Μπορεί να πάρει τις τιμές R για κύλιση προς τα δεξιά, L για κύλιση προς τα αριστερά και B για εναλλασσόμενη κύλιση μια φορά προς τα δεξιά και μια προς τα αριστερά. Ο δεύτερος χαρακτήρας αποτελεί τον τρόπο που θα σχεδιάζεται το κείμενο, με τον χαρακτήρα H φωτίζονται τα γράμματα και το φόντο παραμένει μαύρο και με τον χαρακτήρα R φωτίζεται το φόντο και οι χαρακτήρες εμφανίζονται με μαύρο χρώμα. Τα επόμενα τρία ψηφία αποτελούν έναν κωδικοποιημένο αριθμό που αντιστοιχεί στην ταχύτητα κύλισης του μηνύματος, ενώ το τελευταίο ψηφίο συμβολίζει την ένταση της φωτεινότητας με την οποία θα φωτίζουν οι φωτοδίοδοι του πάνελ της μονάδας Dot-Matrix.

4.2 Arduino Mega

Σε αυτήν την ενότητα θα μελετηθούν οι βιβλιοθήκες , κλάσεις και δομές που γράφτηκαν για την κύρια μονάδα της κατασκευής. Λόγο τον πολλαπλών λειτουργιών που πρέπει να εκτελεί η μονάδα, προτιμήθηκε ο κώδικας να γραφτεί σε πολλαπλά αρχεία/ κλάσεις ώστε κάθε μια από αυτές να υλοποιεί και κάθε μια από τις εν λόγω λειτουργίες.

Για την καλύτερη κατανόηση αλλά και παρουσίαση ο κώδικας του προγράμματος έχει χωριστεί σε τρεις κατηγορίες που θα αναλυθούν περαιτέρω στις επόμενες υποενότητες που ακολουθούν.

4.2.1 Δομές και Βιβλιοθήκες

Προτού προχωρήσουμε στην ανάλυση του βασικού κώδικα της εφαρμογής θα πρέπει να αναφερθούμε στις δομές και της βιβλιοθήκες που χρησιμοποιήθηκαν σε αυτήν. Αντίθετα με τις βιβλιοθήκες που αναφέρθηκαν στο προηγούμενο κεφάλαιο, η βιβλιοθήκη UTFTGui γράφτηκε συγκεκριμένα για την υλοποίηση της πτυχιακής εργασίας.

4.2.1.1 GfxStruct

Στο συγκεκριμένο αρχείο έχουν οριστεί όλες οι δομές που θα χρησιμοποιηθούν από την βιβλιοθήκη UTFTGui αλλά και μερικές Global μεταβλητές.

```
extern uint8_t tahoma_8[];
extern uint8_t dejavu_16[];

extern uint8_t* _Largefont;
extern uint8_t* _Smallfont;

enum ScreenSelect {USER, RESULT, LOGIN,
    KEYBOARD, SMALLKEYBOARD, OPTIONS,
    MESSAGE, DOTMATRIX_CONTROL, SD_CONTROL,
    SD_EXPLORER, REFRESH};

extern ScreenSelect current_screen;
extern ScreenSelect previous_screen;
extern boolean update_screen;
```

Εικόνα 4.3: Global μεταβλητές

Με τον όρο Global μεταβλητή εννοούμε μια μεταβλητή που θα έχουν πρόσβαση όλες οι κλάσεις του προγράμματος. Κάνουμε χρήση κάποιον global μεταβλητών διότι χρειαζόμαστε ένα τρόπο να ενημερώνουμε το κύριος πρόγραμμα ότι ο χρήστης έκανε κάποια ενέργεια που οδηγεί το πρόγραμμα να μεταφερθεί από την μια κλάση στην άλλη. Με αυτόν τον τρόπο το πρόγραμμα μπορεί να εναλλάσσεται ανάμεσα στα γραφικά μενού της οθόνης αφής όταν ο χρήστης προβεί σε κάποια ενέργεια που το καθιστά εφικτό.

Πέρα από τα παραπάνω, υπάρχουν και οι δυο δομές GuiText και Color. Η δομή Color όπως προδίδει και η ονομασία της μας επιτρέπει να ορίσουμε ένα χρώμα. Το χρώμα δηλώνεται με την μορφή RGB. Δηλαδή η κάθε μεταβλητή της δομής συμβολίζει το επίπεδο φωτεινότητας σε κάθε ένα από τα τρία βασικά χρώματα, κόκκινο (Red) , πράσινο (Green) και μπλε (Blue). Οι τιμές φωτεινότητας που μπορεί να πάρει το κάθε χρώμα είναι από το 0 έως το 255.

```
struct Color {

    byte r;
    byte g;
    byte b;
};

struct GuiText {

    int x1;
    int y1;
    int x2;
    int y2;
    char txt[15];
};
```

Εικόνα 4.4: Δομές που χρησιμοποιήθηκαν

Η δομή GuiText μας επιτρέπει να δηλώσουμε τις συντεταγμένες στην οθόνη που θέλουμε να βρίσκετε ένα από γραφικά αντικείμενα που μπορούμε να δημιουργήσουμε με την βιβλιοθήκη UTFTGui. Μπορούμε επίσης να δηλώσουμε τειχών κείμενο που θέλουμε να εμφανίζεται πάνω στο αντικείμενο μας. Για παράδειγμα, το κείμενο που θέλουμε να εμφανίζεται πάνω σε ένα Button που ορίσαμε. Περαιτέρω αναφορά για το πως χρησιμοποιείται η δομή GuiText με την βιβλιοθήκη UTFTGui θα γίνει στην επόμενη ενότητα στην οποία θα γίνει ανάλυση της εν λόγω βιβλιοθήκης.

4.2.1.2 UTFTGui

Η βιβλιοθήκη UTFTGui δημιουργήθηκε με σκοπό την δημιουργία γραφικών αντικειμένων όπως κουμπιά (Buttons) και περιοχές προβολής και επεξεργασίας κειμένου (Textbox και TextArea). Είναι προέκταση της βιβλιοθήκης UTFT/ UTouch καθώς χρησιμοποίει τις βασικές γεωμετρικές δομές που προσφέρονται από την βιβλιοθήκη, για να σχεδιάσει τα πιο σύνθετα αντικείμενα που αναφέρονται παραπάνω.

Γνώμονας για την συγγραφή της βιβλιοθήκης ήταν η μικρή επεξεργαστική ισχύς που προσφέρει η πλατφόρμα Arduino, για αυτόν τον λόγο η βιβλιοθήκη γράφτηκε με τέτοιον τρόπο ώστε να μπορεί να εμφανίσει αντικείμενα που θα είναι όσο το δυνατόν πιο ωραία σε εμφάνιση χωρίς όμως να χρειάζονται μεγάλη επεξεργαστική ισχύ για τον σχεδιασμό τους. Κάτι τέτοιο είναι απόλυτος λογικό αν σκεφτεί κανείς

ότι ένα σχετικά φθηνό smartphone σήμερα έχει περίπου εκατό φορές την επεξεργαστική ισχύ ενός Arduino Mega. Το πρόβλημα γίνετε ακόμα μεγαλύτερο αν αναλογιστεί κανείς το πολύ μικρό μέγεθος μνήμης RAM που έχει το Arduino.

Παρόλα αυτά όπως θα φαίνεται και από το αποτέλεσμα του τελικού προγράμματος, η βιβλιοθήκη μπορεί να σχεδιάσει γραφικά τα οποία μπορούν να σταθούν στο ύψος τον περιστάσεων και προσφέρουν ένα λειτουργικό και οπτικά ευχάριστο γραφικό περιβάλλον.

Βέβαια για να μπορέσει η βιβλιοθήκη να είναι όσο το δυνατόν πιο «ελαφριά» ήταν αναγκαίο να γραφτεί όσο το δυνατόν λιγότερος κώδικας. Αυτό κάνει τον την χρήση της βιβλιοθήκης από τον προγραμματιστεί δυσκολότερη απ’ ότι για παράδειγμα την δημιουργία γραφικών στην πλατφόρμα Android όπου το ίδιο το λειτουργικό σύστημα προσφέρει μια πληθώρα από κλάσεις για τον σχεδιασμό γραφικών.

Εδώ επίσης πρέπει να σημειωθεί ότι η βιβλιοθήκη χρήζει αρκετής βελτίωσης ακόμα και με την περαιτέρω ανάπτυξη της θα μπορούσε να προσφέρει ακόμα περισσότερες δυνατότητες που ξεφεύγουν όμως από τα πλαίσια της εν λόγω πτυχιακής καθώς ο χρόνος που θα χρειάζονταν για την ανάπτυξη τους θα ήταν αρκετά μεγαλύτερος από τα πλαίσια μιας πτυχιακής εργασίας. Περισσότερα για της μελλοντικές βελτιστοποιήσεις θα αναφερθούν σε επόμενο κεφάλαιο.

Προτού προχωρήσουμε στην ανάλυση τον μεθόδων που προσφέρει η UTFTGui να αναφέρουμε ότι η βασική δομή που χρησιμοποιεί η βιβλιοθήκη είναι η GuiText. Με την εν λόγω δομή ο προγραμματιστής μπορεί να ορίσει τις συντεταγμένες στις οποίες θα βρίσκετε το γραφικό του αντικείμενο και το μέγεθος του. Έπειτα με την χρήση των επόμενων μεθόδων μπορεί πλέον να σχεδιάσει και να μεταβάλει το αντικείμενο που δημιούργησε. Επίσης η δομή Color χρησιμοποιείται για να μπορέσει ο προγραμματιστής να ορίσει τα χρώματα τα οποία θέλει να έχει το γραφικό αντικείμενο.

- UTFTGui()

Ο constructor της βιβλιοθήκης. Εδώ πρέπει να ορίσουμε την βιβλιοθήκη UTFT της οποίας της μεθόδους θα χρησιμοποιήσει η βιβλιοθήκη για το σχεδιασμό και την απεικόνιση των γραφικών αντικειμένων που θα δημιουργήσει ο προγραμματιστής.

- drawButton()

Η πιο πολυχρησιμοποιημένη μέθοδος της βιβλιοθήκης. Μας επιτρέπει να σχεδιάσουμε ένα κουμπί (Button) η μια περιοχή κειμένου (TextBox). Εκτός από της συντεταγμένες την οθόνη που θέλουμε να σχεδιαστεί το αντικείμενο, μπορούμε επίσης να ορίσουμε τα χρώματα που θα έχει το Button/ TextBox όπως επίσης και το μέγεθος του κειμένου που μπορεί να θέλουμε να απεικονίσουμε πάνω στο αντικείμενο. Το κείμενο θα είναι κεντραρισμένο στο κέντρο του αντικειμένου.

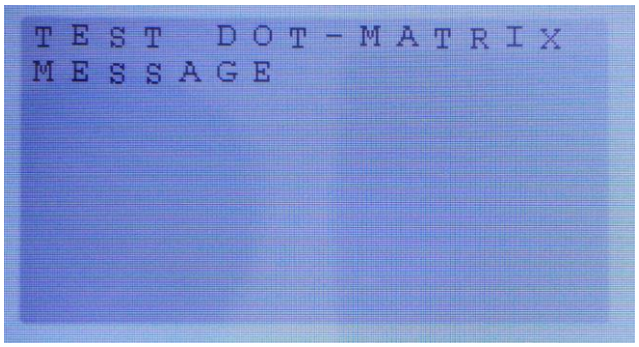
Εικόνα 4.5: Παράδειγμα Button και TextBox

Εδώ πρέπει να τονίσουμε ότι ο προγραμματίσεις πρέπει να ελέγξει αν το κείμενο που θέλει να γραφτεί χωράει στο πλαίσιο του αντικειμένου καθώς ο αυτομάτως έλεγχος κρίθηκε σωστό να μην υπάρχει για την αποφυγή της αύξησης του μεγέθους της βιβλιοθήκης.

- drawTextArea()

Η μέθοδος drawTextArea() είναι παρόμοια με την μέθοδο drawButton() αλλά επιτρέπει την συγγραφή κειμένου το οποίο μπορεί να είναι παραπάνω από μια γραμμές. Η δομή προσφέρει και την δυνατότητα αναδίπλωσης κειμένου

(Word Wrap), δηλαδή την ικανότητα να σπάει τις γραμμές του κειμένου αυτόματα ώστε να παραμένουν εντός των περιθωρίων της δομής.



Εικόνα 4.6: Γραφική απεικόνιση TextArea

- pressButton()

  Η συνάρτηση μας δίνει την δυνατότητα να κάνουμε ένα υποτυπώδες animation όταν ο χρήστης πατήσει κάποια από της γραφικές δομές που έχουμε σχεδιάσει. Η συνάρτησης όπως και όλη η βιβλιοθήκη σχεδιάστηκε με γνώμονα την γρήγορη ταχύτητα εκτέλεσης για αυτό και γίνεται προσπάθεια το οπτικό αποτέλεσμα να είναι όσο το δυνατόν πιο εύκολο στην σχεδίαση αλλά και ταυτόχρονα να είναι αρκετά αρεστό αισθητικά.

  Ο προγραμματιστής μπορεί να διαλέξει το χρώμα το οποίο θα αλλάξει το γραφικό αντικείμενο κατά την διάρκεια που ο χρήστης το ενεργοποιεί πατώντας στην οθόνη αφής.

- changeButtonText()

  Η συγκεκριμένη συνάρτηση χρησιμοποιείται για να αλλάξει το κείμενο που βρίσκετε πάνω σε ένα κουμπί (Button). Η συνάρτηση είναι πολύ γρήγορη και απαιτεί ελάχιστη επεξεργαστική ισχύ σε σχέση με την εκτέλεση της drawButton(). Το μειονέκτημα της είναι ότι το κείμενο πρέπει να είναι σε μήκος γραμμάτων μικρότερο η ίσο με το μήκος του κειμένου που ήταν προηγούμενος σχεδιασμένο στην οθόνη. Για αυτόν τον λόγο η συνάρτηση πρέπει να χρησιμοποιείται με μεγάλη προσοχή καθώς μπορεί πολύ εύκολα

να οδηγήσει σε γραφικά λάθη στην οθόνη. Αντ' αυτού, λόγου του πολύ μικρού χρόνου εκτέλεσης της μπορεί να χρησιμοποιηθεί και χρησιμοποιείται στην περίπτωση που θέλουμε για παράδειγμα, να αλλάξουμε τα γράμματα από κεφαλαία σε πεζά σε ένα πληκτρολόγιο που έχουμε σχεδιάσει. Για την περίπτωση που θέλουμε να σχεδιάσουμε σε μια γραφική δομή οποιοδήποτε κείμενο ανεξάρτητα μεγέθους μπορούμε να χρησιμοποιήσουμε την συνάρτηση printTextBoxMessage(), την οποία θα αναλύσουμε παρακάτω, που είναι πιο γενικής χρήσης αλλά και πιο αργή στην εκτέλεση της.

- writeTextBoxChar()

  Η συνάρτηση μας επιτρέπει να γράψουμε έναν χαρακτήρα σε πραγματικό χρόνο σε ένα TextBox η TextArea. Η συνάρτηση είναι ιδιαίτερα σημαντική γιατί μας επιτρέπει να γράψουμε κείμενο σε πραγματικό χρόνο χρησιμοποιώντας το πληκτρολόγιο που σχεδιάσαμε για την εφαρμογή μας.

- deleteTextBoxChar()

  Συμπληρώνοντας την προηγούμενη συνάρτηση που μας επέτρεπε την συγγραφή χαρακτήρα σε πραγματικό χρόνο. Η εν λόγω συνάρτηση μας επιτρέπει την διαγραφή του τελευταίου χαρακτήρα που σχεδιάστηκε με την συνάρτηση writeTextBoxChar().

- printTextBoxMessage()

  Η συνάρτηση δίνει στον προγραμματιστεί την δυνατότητα να γράψει ένα μήνυμα σε ένα γραφικό αντικείμενο είτε αυτό είναι ένα κουμπί η κάποια περιοχή εκχώρησης κειμένου. Είναι πιο γενικής χρήσης από την συνάρτηση changeButtonText() και πιο μεγάλη ευελιξία καθιστώντας την όμως πιο αργή. Προσφέρει την δυνατότητα επιλογής το κείμενο να είναι γραμμένο στο κέντρο του γραφικού αντικειμένου η ευθυγραμμισμένο στην αριστερή του πλευρά. Η ευθυγράμμιση στην δεξιά πλευρά δεν κρίθηκε αναγκαία καθώς στης περισσότερες γλώσσες δεν χρησιμοποιείται και τόσο συχνά.

- getTextBox()

  Επιστρέφει το κείμενο που είναι γραμμένο σε να TextBox ή TextArea.

- setTextBoxLength()

  Επιτρέπει στον προγραμματιστεί να ορίσει το μέγιστο αριθμό χαρακτήρων που μπορούν να γραφτούν σε μια δομή γραφικών. Πολύ σημαντική συνάρτησης καθώς επιτρέπει τον έλεγχο του μεγέθους του κειμένου ώστε να μην βγει έξω από τα όρια της δομής γραφικών.

4.2.2 Κλάσεις Λειτουργίας

Οι κλάσεις που θα αναλυθούν σε αυτό το υποκεφάλαιο ασχολούνται με το λειτουργικό κομμάτι της κεντρικής μονάδας δηλαδή δεν αφορούν το γραφικό περιβάλλον και οποιαδήποτε άλλη κλάση από την οποία ο χρήστης μπορεί να διαχειριστή την συσκευή. Αντίθετα αφορούν αυτόματες λειτουργίες του συστήματος η λειτουργίες που αφορούν το πρόσθετο υλικό που είναι συνδεμένο πάνω στο Arduino, όπως για παράδειγμα την μονάδα GSM η το σύστημα υπέρυθρων για την αποστολή των ρυθμίσεων η του μηνύματος στην μονάδα Dot-Matrix.

4.2.2.1 Gsm

Η κλάση GSM όπως μπορούμε να καταλάβουμε και από την ονομασία της έχει ως σκοπό την διαχείριση της μονάδας GSM που είναι συνδεμένο πάνω στο Arduino Mega και βασίζεται στον διαμορφωτή – αποδιαμορφωτή SIM900 της εταιρίας SIMCOM. Συγκεκριμένα πρέπει να λαμβάνει τα εισερχόμενα μηνύματα SMS, να τα διαχωρίζει ανάλογα με την γλώσσα την οποία είναι γραμμένα, αγγλικά ή ελληνικά και στην συνέχεια να κάνει τις απαραίτητες ενέργειες για να είναι έτοιμα για την αποστολή τους στην μονάδα Dot-Matrix. Πέρα από αυτά περιέχει και κλάσεις για την ενεργοποίηση της μονάδας SIM900 κατά την εκκίνηση της κατασκευής. Η κλάση επιτρέπει τον έλεγχο του αποστολέα του μηνύματος SMS.

- Gsm()

  Ο constructor της κλάσης παίρνει ως ορίσματα την σειριακή πόρτα στην οποία έχουμε συνδέσει την μονάδα GSM και την ταχύτητα συριακής επικοινωνίας μεταξύ του Arduino και της μονάδας SIM900.

- setPanel()

  Βοηθητική συνάρτηση στην οποία ορίζουμε τις κλάσεις στις οποίες θα μεταφερθεί το κείμενο του μηνύματος SMS αφού πρώτα έχει υποστεί την κατάλληλη επεξεργασία μέσο τον μεθόδων read() και convert().

- powerUp()

  Η μέθοδος powerUp() επιτρέπει στο Arduino να ενεργοποιεί την μονάδα GSM χωρίς να χρειάζεται ο χρήστης να πατάει το πλήκτρο ενεργοποίησης κάθε φορά που ενεργοποιεί την συσκευή.

```
void Gsm::powerUp(uint8_t pin) {

    pinMode(pin, OUTPUT);
    digitalWrite(pin, LOW);
    delay(1000);
    digitalWrite(pin, HIGH);
    delay(2000);
    digitalWrite(pin, LOW);
    delay(3000);
}
```

Εικόνα 4.7: Κώδικας ενεργοποίησης μονάδας GSM

- init()

  Πρέπει να εκτελεστεί μετά την method powerUp() για να εκτελέσει όλες τις απαραίτητες εντολές για να γίνει σωστή εκκίνηση της μονάδας GSM. Όπως την εκκίνηση της επικοινωνίας μεταξύ τον δυο συσκευών, την εκχώρηση του

κωδικού PIN της κάρτας SIM της συσκευής GSM και την ρύθμιση της μεθόδου ανάγνωσης των εισερχόμενων SMS μηνυμάτων.

```
void Gsm::init() {

    _Serial->begin(_baud_rate);

    byte i = 0;
    char gsm_char;
    _Serial->print("AT+CPIN="SIMPIN"\r");
    delay(8000);
    while (_Serial->available() > 0 ) {
        i++;
        gsm_char = _Serial->read();
    }
    if (i == 0) {
        delay(5000);
        asm volatile ("  jmp 0");
    }
    _Serial->print("AT+CMGF=1\r");
    delay(2000);
    _Serial->print("ATE0\r");
    delay(2000);
}
```

Εικόνα 4.8: Κώδικας εκκίνησης ομαλής λειτουργείας μονάδας GSM

- read()

Η πιο σημαντική μέθοδος της κλάσης. Σκοπός της είναι να λαμβάνει τα εισερχόμενα SMS. Όταν γίνει η λήψη ενός SMS γίνετε έλεγχος αν ο αριθμός του αποστολέα ανήκει σε έναν από τους αποθηκευμένους αριθμούς που βρίσκονται στο αρχείο ρυθμίσεων στην κάρτα μνήμης τύπου SD και στην συνέχεια ακολουθεί η ανάγνωση και επεξεργασία του εισερχόμενου μηνύματος. Το εισερχόμενο μήνυμα περνάει μέσα από την μέθοδο convert(), που θα αναλυθεί παρακάτω, και αφού γίνει η κατάλληλη επεξεργασία αποθηκεύεται και αποστέλλεται μέσο του κυκλώματος υπέρυθρων στην μονάδα Dot-Matrix. Για την αποστολή του μηνύματος υπεύθυνη είναι η κλάση DotMatrix που αναλύεται επίσης παρακάτω.

```
if (_Serial->available() > 0) {
    _gsm_char_num = _Serial->readBytesUntil('\n', _gsm_buffer, sizeof(
        _gsm_buffer) - 1);
    _gsm_buffer[_gsm_char_num - 1] = '\0';
    if (_gsm_char_num > 1) {
        if (_message_buffer_flag == 1) {
            if (_gsm_buffer[0] == '0' && (_gsm_buffer[1] == '0' ||
                _gsm_buffer[1] == '3')) {
                convert(_gsm_buffer, strlen(_gsm_buffer), 1);
            } else {
                convert(_gsm_buffer, strlen(_gsm_buffer), 2);
            }
            _DotMatrix->send(_gsm_buffer, 1);
            _DotMatrixMessage->updateMessage(_gsm_buffer);
            _DotMatrixMessage->save();
            _message_buffer_flag = 0;
            _delete_sms_flag = 1;
        }
```

Εικόνα 4.9: Κώδικας διαχείρισης εισερχόμενου μηνύματος από μονάδα GSM

- convert()

  Ένα από μειονεκτήματα του προγραμματιστικού περιβάλλοντος της πλατφόρμας Arduino είναι ότι εκτός από την κλασική κωδικοποίηση χαρακτήρων ASCII δεν υποστηρίζει κάποια άλλη όπως για παράδειγμα την πλέον διαδεδομένη UTF8.

  Αποτέλεσμα αυτής της αδυναμίας του Arduino είναι να μην υποστηρίζεται ανάγνωση αλλά και εγγραφή των περισσοτέρων χαρακτήρων εκτός από τους αγγλικούς. Αυτό είναι αρκετά μεγάλο πρόβλημα στην εφαρμογή μας καθώς ουσιαστικά σημαίνει ότι δεν μπορεί να γίνει ανάγνωση μηνύματος γραμμένο με ελληνικούς χαρακτήρες. Για να ξεπεράσουμε αυτό το πρόβλημα δημιουργήθηκε η μέθοδος convert()

  Η μέθοδος διαβάζει το εισερχόμενο μήνυμα χαρακτήρα χαρακτήρα και μπορεί από τα δεκαεξαδικά ψηφία με τα οποία είναι κωδικοποιημένος ο κάθε χαρακτήρας να καταλάβει αν το κείμενο είναι του μηνύματος SMS είναι γραμμένο σε κωδικοποίηση ASCII η σε κωδικοποίηση UTF8 η ακόμα και σε συνδυασμό των δυο. Έπειτα η κλάση αποκωδικοποιεί το μήνυμα και το μετατρέπει σε μορφή τέτοια ώστε να μπορεί να διαβαστεί και να αποτυπωθεί στην μονάδα Dot-Matrix.

4.2.2.2 DotMatrix

Η κλάση DotMatrix είναι υπεύθυνη για την υπέρυθρη επικοινωνία της κεντρικής μονάδας και της μονάδας απεικόνισης του κειμένου, Dot-Matrix. Σε αυτήν την κλάση γίνονται όλες οι απαραίτητες energies τόσο για την εκκίνηση του κυκλώματος αποστολής υπέρυθρων όσο και για την κωδικοποίηση και αποστολή του μηνύματος που θέλουμε να αποστείλουμε στην μονάδα Dot-Matrix.

- DotMatrix()

  Ο constructor της κλάσης. Εδώ πρέπει να οριστεί η συριακή πόρτα στην οποία έχει συνδεθεί το κύκλωμα υπέρυθρων, η ταχύτητα με την οποία θα γίνεται η συριακή επικοινωνία όπως επίσης και η συχνότητα του αισθητήρα υπέρυθρων που βρίσκετε στην μονάδα Dot-Matrix. Αυτό συμβαίνει διότι η συχνότητα του κύματος υπέρυθρων που τα σταλθούν πρέπει να είναι ίδια με την συχνότητα του αισθητήρα για να μην υπάρξει απώλεια δεδομένων. Επίσης πρέπει να τονίσουμε ότι η ταχύτητα συριακής επικοινωνίας δεν πρέπει να ξεπερνά τα 2400bps.

- init()

  Σε αυτή την μέθοδο γίνεται η ενεργοποίηση του κυκλώματος υπέρυθρων. Γίνεται η εκκίνηση της συριακής επικοινωνίας και ενεργοποιείται η εσωτερική γεννήτρια τετραγωνικού παλμού που εκπέμπει από το Pin 10 του Arduino Mega.

```
void DotMatrix::init() {

    _Serial->begin(_baud_rate);

    pinMode (IROUT, OUTPUT);
    TCCR2A = _BV (COM2A0) | _BV(WGM21);
    TCCR2B = _BV (CS20);
    OCR2A =  (16000000 / (2 * _herz)) - 1;
}
```

Εικόνα 4.10: Κώδικας εκκίνησης κυκλώματος υπέρυθρων

Η πλατφόρμα Arduino δεν προσφέρει κάποια μέθοδο για την ενεργοποίηση της γεννήτριας και γενικά για την ρύθμιση των χρονιστών του μικροελεγκτή της Atmel ATmega2560 και για αυτόν τον λόγο χρειάστηκε ο απ' ευθείας προγραμματισμός των καταχωρητών του μικροελεγκτή.

- send()

Σε αυτήν την μέθοδο γίνεται η αποστολή του μηνύματος. Πριν το μήνυμα αποσταλεί κωδικοποιείται με μια κεφαλίδα. Με αυτόν τον τρόπο μπορούμε να ενημερώσουμε την μονάδα Dot-Matrix ότι το μήνυμα προέρχεται από την κεντρική μονάδα και όχι από κάποια άλλη μονάδα που μπορεί να κάνει παρεμβολές.

```
void DotMatrix::send(char* buffer, uint8_t mode) {

    if (mode == 1) {
        sprintf(_dotmatrix_message, HEADER1"%s", buffer);
    } else if (mode == 2) {
        sprintf(_dotmatrix_message, HEADER2"%s", buffer);
    }
    _Serial->print(_dotmatrix_message);
}
```

Εικόνα 4.11: Κλάση για την αποστολή μηνύματος μέσω της μονάδας υπέρυθρων

4.2.2.3 SDSettingsImport

Ένα από τα μειονεκτήματα της πλατφόρμας Arduino είναι ότι δεν διαθέτει κάποιου είδους μόνιμης μνήμης, δηλαδή κάποιο χορό αποθήκευσης δεδομένων όπου τα δεδομένα μπορούν να μείνουν αναλλοίωτα μετά την διακοπή της τροφοδοσίας του κυκλώματος. Το παραπάνω δεν είναι τελείως αληθές καθώς το Arduino διαθέτει μνήμη EPROM η οποία διατηρεί τα δεδομένα και μετά την επανεκκίνηση του μικροελεγκτή αλλά δυστυχώς η μνήμη EPROM είναι πολύ αργή στην προσπέλαση, πολύ μικρή σε μέγεθος και το κυριότερο προσφέρει μόνο ένα μικρό αριθμό περιορισμένων εγγραφών.

Από τα παραπάνω προκύπτει το πρόβλημα της αποθήκευσης δεδομένων στην κεντρική μας μονάδα όσο και στη μονάδα Dot-Matrix. Για να ξεπεράσουμε το εν

λόγω πρόβλημα χρησιμοποιήσαμε την κάρτα μνήμης micro SD που συνδέσαμε με το Arduino Mega. Μερικά από τα προτερήματα της χρήσης μνήμης τύπου SD είναι το φθηνό της κόστος, η υποστήριξη συστήματος αχρείων FAT / FAT32 που μας επιτρέπει να μπορούμε να μεταφέρουμε τα αρχεία μας σε πρακτικά οποιονδήποτε ηλεκτρονικό υπολογιστή, η μεγάλη διάρκεια ζωής και η μεγάλη ταχύτητα μεταφοράς δεδομένων.

Σκοπός, λοιπόν, της κλάσης SDSettingsImport είναι να ανακτά από την κάρτα μνήμης micro SD όλες τις απαραίτητες ρυθμίσεις που θέλουμε να αποθηκεύονται κατά την λειτουργία όπως για παράδειγμα, το τελευταίο μήνυμα που στάλθηκε στην μονάδα Dot-Matrix, τις διάφορες ρυθμίσεις της μονάδας Dot-Matrix, η λίστα με τους αριθμούς τηλεφώνων που επιτρέπεται να στέλνουν μήνυμα στην μονάδα Dot-Matrix αλλά και διάφορες ρυθμίσεις για τα μενού γραφικών ώστε να επιστρέψουν στην προηγούμενη κατάσταση τους μετά από τυχόν απώλεια τροφοδοσία η κάποια μη προγραμματισμένη επανεκκίνηση.

Για να αποφύγουμε και χρήσης δεύτερης κάρτας μνήμης τύπου SD στην μονάδα Dot-Matrix αποφασίστηκε οι ρυθμίσεις της εν λόγω μονάδας να αποθηκεύονται και αυτές στην κεντρική μονάδα και σε περίπτωση επανεκκίνησης της μονάδας Dot-Matrix να γίνετε επαναποστολή τους.

Οι ρυθμίσεις αποθηκεύονται μέσα στην κάρτα μνήμης micro SD σε αρχείο με ονομασία CONF.CFG. Μερικές από τις ρυθμίσεις είναι, οι ονομασίες των μαθημάτων που θέλουμε να παρουσιάζονται, το αντίστοιχο αρχείο CSV με τις βαθμολογίες για κάθε μάθημα, οι ρυθμίσεις και το μήνυμα της μονάδας Dot-Matrix.

```
##Settings File
##Below must be = 1
##To update
//Settings = 1
[LOGIN]
USERNAME
PASSWORD

[LESSONS]
Αρχιτεκτονική
Αρχ. ΙΙ
ΕΠ. ΕΙΚΟΝΑΣ

[FILES]
TEXT.CSV
TEXT.CSV
A.CSV

[DTM_CONTROL]
BR1000

[DTM_MESSAGE]
Δοκιμαστικό Μήνυμα 2!!

[PHONES]
6992671234
6989080123
```

Εικόνα 4.12: Δομή αρχείου CONF.CFG

- SDSettingsImport()

  Κατά την αρχικοποίηση της κλάσης πρέπει να δηλωθούν όλες οι κλάσεις οι οποίες θέλουν να ανακτήσουν η να αποθηκεύσουν δεδομένα και ρυθμίσεις από την κάρτα μνήμης.

- setSDCard()

  Μέθοδος για την δήλωση της κάρτας μνήμης την οποία θα προσπελάσει η κλάση ώστε να αποθηκεύσει τις ρυθμίσεις και τα δεδομένα αλλά και να προσπελάσει τις ρυθμίσεις που αποθηκεύτηκαν σε προηγούμενη εκτέλεση.

- import()

Η μέθοδος import() εκτελείται κατά την εκκίνηση της μονάδας και επιτρέπει την ανάκτηση των ρυθμίσεων και των δεδομένων που έχουν αποθηκευτεί στο αρχείο CONF.CFG

```
switch (_buffer[0]) {
    case '/':
        if (CMP_STR(&_buffer[1], "/Settings = 1")) {
            update_sd = true;
        }
        if (CMP_STR(&_buffer[1], "/Settings = 0")) {
            update_sd = false;
        }
        break;
    case '[':
        if (CMP_STR(&_buffer[1], "LOGIN]")) {
            setting_type = 1;
            settings_count = 0;
        }
        if (CMP_STR(&_buffer[1], "LESSONS]")) {
            setting_type = 2;
            settings_count = 0;
        }
```

Εικόνα 4.13: Απόσπασμα κώδικα της μεθόδου import

- checkPhone()

Στο προαναφερθέν αρχείο CONF.CFG υπάρχει εκτός από ρυθμίσεις και η λίστα με τους τηλεφωνικούς αριθμούς από τους οποίους θα γίνονται δεκτά τα μηνύματα προς εμφάνιση στην μονάδα Dot-Matrix. Η μέθοδος checkPhone() επιτρέπει τον έλεγχο τηλεφωνικού αριθμού με τους παρόντες στην παραπάνω λίστα. Χρήση της μεθόδου checkPhone() γίνετε από την κλάση GSM κάθε φορά που γίνετε λήψη κάποιου εισερχόμενου μηνύματος SMS.

```
boolean SDSettingsImport::checkPhone(char* phone_number) {

    boolean phones_section = false;

    _SdFat->begin(_pin, SPI_HALF_SPEED);
    ifstream sdin("CONF.CFG");

    while (sdin.getline(_buffer, sizeof(_buffer), '\n') ||
        sdin.gcount()) {
        if (phones_section) {
            if (CMP_STR(_buffer, phone_number)) return true;
        }
        if (CMP_STR(_buffer, "[PHONES]")) {
            phones_section = true;
        }
    }

    return false;
}
```

Εικόνα 4.14: Κώδικας μεθόδου checkPhone

- updateSettings()

  Όπως έχει αναφερθεί σε προηγούμενο κεφάλαιο η κεντρική μονάδα διαθέτει και είσοδο για εξωτερική κάρτα μνήμης SD. Μια από της δυνατότητες της εφαρμογής, είναι η αντιγραφή του αρχείου CONF.CFG από την εξωτερική SD κάρτα στην εσωτερική κάρτα μνήμης micro SD. Σε περίπτωση που το αρχείο έχει αντιγράφει η μέθοδος είναι υπεύθυνη για την ενημέρωση, σε πραγματικό χρόνο, των ρυθμίσεων στις κλάσεις που αντιστοιχούν.

- convert_UTF8()

  Μετατρέπει το κείμενο που βρίσκετε στο αρχείο CONF.CFG από κωδικοποίηση UTF8 στην προσαρμοσμένη κωδικοποίηση ASCII που χρησιμοποιείται από την μονάδα για την εμφάνιση ελληνικών χαρακτήρων.

4.2.3 Κλάσεις Γραφικών

Στο υποκεφάλαιο που ακολουθεί θα αναλυθούν οι κλάσεις γραφικών που δημιουργήθηκαν με την βιβλιοθήκη UTFUGui που κατασκευάστηκε για να διευκολύνει την συγγραφή τους.

Οι κλάσεις έχουν χωριστεί σε δυο κατηγορίες τις γενικές κλάσεις γραφικών, δηλαδή τις κλάσεις που μπορούν να χρησιμοποιηθούν και σε άλλες εφαρμογές αυτούσιες χωρίς σχεδόν καμία αλλαγή, και τις ειδικές κλάσεις γραφικών, δηλαδή κάποια γραφικά μενού τα οποία δημιουργήθηκαν συγκεκριμένα για τις ανάγκες της συγκεκριμένης πτυχιακής εργασίας.

Προτού προχωρήσουμε στην ανάλυση της κάθε κλάσης ξεχωριστά πρέπει να γίνει μια αναφορά στην γενική δομή κάθε κλάσης γραφικών. Κάθε κλάση περιέχει τις μεθόδους draw() και readTouch() παρακάτω γίνετε μια μικρή ανάλυση για την κάθε μια.

- draw()

  Η μέθοδος σχεδιασμού της κλάσης γραφικών. Κάθε κλάση γραφικών αποτελείται από μια δομή η έναν πίνακα δομών GuiText. Η μέθοδος draw() κατά την εκτέλεση της διαβάζει τις συγκεκριμένες δομές και σχεδιάζει τα γραφικά προς απεικόνιση κάνοντας χρήση των μεθόδων της βιβλιοθήκης UTFTGui.

```
void OptionScreen::draw() {
    byte i;

    MCP(_temp_Color[0], _background);
    _UTFT->fillScr(_background.r, _background.g, _background.b
        );

    MCP(_temp_Color, _buttons);

    MCP(_temp_GuiText_1, data[0]);
    _UTFTGui->drawButton(_temp_GuiText_1, _temp_Color[0],
        _temp_Color[1], _temp_Color[2]);

    for (i = 1; i < 4; i++) {
        MCP(_temp_GuiText_1, data[i]);
        _UTFTGui->drawButton(_temp_GuiText_1, _temp_Color[0],
            _temp_Color[1], _temp_Color[2], 2);
    }

}
```

Εικόνα 4.15: Παράδειγμα προγραμματισμού μεθόδου draw

- redraw()

Σε μερικές περιπτώσεις ειδικά όταν το γραφικό περιβάλλον σε μια κλάση είναι αρκετά σύνθετο η συνάρτηση draw() μπορεί να χρειαστεί αρκετό χρόνο για να σχεδιάσει τα γραφικά. Για αυτές τις περιπτώσεις κάποιες κλάσεις γραφικών έχουν και την εσωτερική μέθοδο redraw() η οποία σχεδιάζει μόνο κάποιο μέρος του γραφικού περιβάλλοντος, δηλαδή μόνο τα μέρη που θα υποστούν αλλαγή. Με αυτόν τον τρόπο οι γραφικές εναλλαγές κατά την διάρκεια της απεικόνισης μιας κλάσης γραφικών γίνονται σε μικρότερο χρονικό διάστημα με αποτέλεσμα ο χρήστης να βλέπει ότι οι εναλλαγές γίνονται άμεσα χωρίς καθυστερήσεις.

- readTouch()

Η δεύτερη μέθοδος που έχουν σχεδόν όλες οι κλάσεις γραφικών. Η μέθοδος αυτή πρέπει να εκτελείται ανά σύντομα χρονικά διαστήματα και είναι υπεύθυνη για την καταγραφή τον πατημάτων στην οθόνη αφής. Μόλις γίνει η καταγραφή του πατήματος η μέθοδος βρίσκει πιο γραφικό αντικείμενο την οθόνη πατήθηκε και εκτελεί το κατάλληλο animation και οποιαδήποτε άλλη λειτουργία πρέπει να εκτελεστεί κατά το συγκεκριμένο πάτημα.

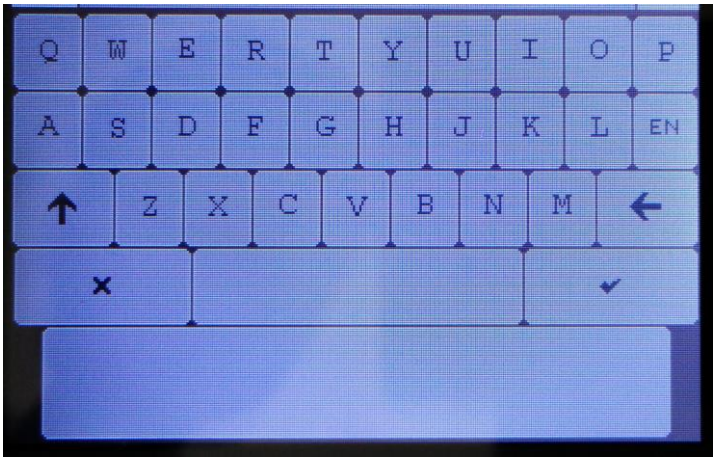
4.2.3.1 Γενικές Κλάσεις Γραφικών

Οι επόμενες κλάσεις αποτελούν αυτόνομα γραφικά αντικείμενα που μπορούν να χρησιμοποιηθούν αυτούσιες η με ελάχιστες αλλαγές και σε άλλες εφαρμογές που χρησιμοποιούν την βιβλιοθήκη UTFTGui.

Οι κλάσεις που δημιουργήθηκαν είναι δυο πληκτρολόγια, ένα πλήρες και ένα αριθμητικό, επίσης ένα γραφικό περιβάλλον για την περιήγηση στα αρχεία της κάρτας μνήμης τύπου SD. Παρακάτω θα γίνει μια ανάλυση των κλάσεων και των δυνατοτήτων που προσφέρουν.

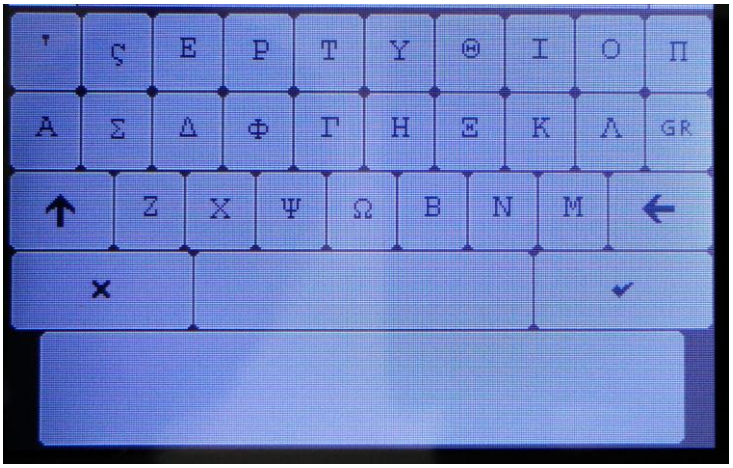
4.2.3.1.1 BigKeyboard

Η κλάση BigKeyboard είναι υπεύθυνη για τον σχεδιασμό ενός πλήρους πληκτρολογίου για την συγγραφή κειμένου στις TextBox και TextArea δομές που σχεδιάζονται με την βιβλιοθήκη UTFTGui.



Εικόνα 4.16: Γραφική απεικόνιση αγγλικού πληκτρολογίου με την χρήση της κλάσης BigKeyboard

Το πληκτρολόγιο επιτρέπει την συγγραφή αγγλικού και ελληνικού κειμένου αλλά και πρόσθετων συμβόλων και αριθμών. Το ελληνικό κείμενο υποστηρίζει και τον τονισμό γραμμάτων.



Εικόνα 4.17: Γραφική απεικόνιση ελληνικού πληκτρολογίου με την χρήση της κλάσης BigKeyboard

- setTextBox()

Με αυτήν την μέθοδο ορίζεται το γραφικό αντικείμενο στο οποίο θέλουμε να γραφτεί το κείμενο που θα πληκτρολογηθεί. Μπορούμε επίσης να ορίσουμε το μέγιστο μήκος σε χαρακτήρες του κειμένου που θέλουμε να γραφτεί αλλά και αν επιτρέπεται η συγγραφή ελληνικών χαρακτήρων στο εν λόγω γραφικό αντικείμενο.

- updateText()

  Μέθοδος που εκτελείται κατά το πάτημα του κουμπιού επιβεβαίωσης και είναι υπεύθυνη για την μεταφορά του κειμένου που πληκτρολογήθηκε στο αντίστοιχο TextBox η TextArea.

- intonateChar()

  Εσωτερική συνάρτηση του πληκτρολογίου με την οποία γίνετε ο τονισμός των ελληνικών χαρακτήρων, που μπορούν να τονιστούν, όταν έχει πατηθεί το πλήκτρο του τόνου. Ο τονισμός γίνεται με τον ίδιο τρόπο που γίνεται και στους επιτραπέζιους υπολογιστές, δηλαδή προηγείται το πάτημα του τόνου και στην συνέχεια αν ο χαρακτήρας που πατηθεί μπορεί να τονιστεί θα λάβει και τον τόνο.

4.2.3.1.2 SmallKeyboard

Εκτός από το πλήρες πληκτρολόγιο για τις ανάγκες τις εφαρμογής δημιουργήθηκε και ένα αριθμητικό πληκτρολόγιο όπου οι χρήστες της κεντρικής μονάδας μπορούν να εισάγουν τον αριθμό μητρώου τους για να ενημερωθούν για τις βαθμολογίες των μαθημάτων.

Εικόνα 4.18: Γραφική απεικόνιση αριθμητικού πληκτρολογίου με την κλάση SmallKeyboard

- setTextBox()

  Ομοίως με το πλήρες πληκτρολόγιο η συγκεκριμένη μέθοδος επιτρέπει την δήλωση του γραφικού αντικειμένου στο οποίο θα αποτυπωθεί ο αριθμός που έχει πληκτρολογηθεί.

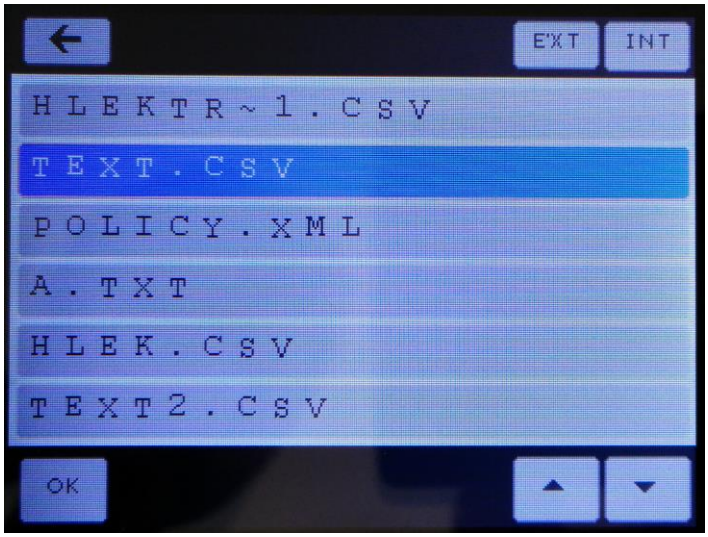
- updateText()

  Ενημερώνει το γραφικό αντικείμενο που ορίστηκε με της μέθοδο setTextBox() με τον αριθμό που πληκτρολογήθηκε.

4.2.3.1.3 SDFileExplorer

Η κλάση SDFileExplorer προσφέρει ένα γραφικό περιβάλλον για την απεικόνιση του συστήματος αρχείων στης συνδεόμενες κάρτες μνήμης SD.

Εκτός από την δυνατότητα απεικόνισης τον αρχείων που βρίσκονται στις κάρτες μνήμης που είναι συνδεμένες με την κεντρική μονάδα της κατασκευής η κλάση μας δίνει και την δυνατότητα αντιγραφής ενός αρχείου από την εξωτερική κάρτα μνήμης SD στην εσωτερική micro SD κάρτα μνήμης.



Εικόνα 4.19: Γραφικό περιβάλλον για την απεικόνιση του συστήματος αρχείων της κάρτας SD με την χρήση της κλάσης SDFileExplorer

Η κλάση κάνει εκτενής χρήση της βιβλιοθήκης SdFat για την πρόσβαση στις κάρτες μνήμης.

- SDFileExplorer()

  Εκτός από τις βιβλιοθήκες γραφικών που είναι απαραίτητες για τον σχεδιασμό του γραφικού περιβάλλοντος πλοήγησης στην δομή των καρτών μνήμης SD, πρέπει να δηλωθεί και η κλάση SDSettingsImport για να γίνεται η μεταφορά των ρυθμίσεων σε περίπτωση που γίνει αντιγραφή του αρχείου CONF.CFG.

- setSDCard()

  Μέθοδος για των ορισμό μιας η δυο καρτών τύπου SD.

- setTextBox()

  Σε περίπτωση που χρειαστεί να επιλεγεί κάποιο αρχείο για να εμφανιστεί η ονομασία του σε κάποιο γραφικό αντικείμενο, μπορούμε να το ορίσουμε με την χρήση αυτής της μεθόδου.

- getFileNames()

  Εσωτερική μέθοδος της κλάσεις. Επιστρέφει τα ονόματα των έξι ή λιγότερων αρχείων που απεικονίζονται στην οθόνη την στιγμή που εκτελείται. Το γραφικό περιβάλλον διαθέτει κουμπιά για την εμφάνιση των έξι προηγούμενων η των έξι επόμενων αρχείων. Με την χρήση τους γίνεται περιήγηση σε όλα τα αρχεία της κάρτας μνήμης εάν αυτά είναι πάνω από έξι και δεν μπορούν να απεικονιστούν όλα ταυτόχρονα στην οθόνη.

- copyFile()

  Η μέθοδος ξεκινά την διαδικασία αντιγραφής του επιλεγμένου αρχείου από την εξωτερική κάρτα μνήμης SD στην εσωτερική micro SD μνήμη. Η μέθοδος εκτελείται όταν πατηθεί το κουμπί Copy που υπάρχει στο γραφικό περιβάλλον.

```
int SDFileExplorer::copyFile() {

    pinMode(_pin2, OUTPUT);
    digitalWrite(_pin2, HIGH);

    if (!_sd_card1->begin(_pin1, SPI_HALF_SPEED)) return 0;
    if (!_sd_card2->begin(_pin2, SPI_HALF_SPEED)) return 0;

    if (selected_file >= 0) {
        _sd_card2->chvol();
        if (!_file2->open(_filename[selected_file], O_READ))
            return 0;
        _sd_card1->chvol();
        if (!_file1->open(_filename[selected_file], O_WRITE |
            O_CREAT | O_TRUNC)) return 0;
        _file2->rewind();
        while (1) {
            int n = _file2->read(buf, sizeof(buf));
            if (n < 0) return 0;
            if (n == 0) break;
            if (_file1->write(buf, n) != n) return 0;
        }
        _file1->close();
        _file2->close();
        return 1;
    } else {
        return 0;
    }
}
```

Εικόνα 4.20: Μέθοδος για την αντιγραφή αρχείων από την εξωτερική στην εσωτερική κάρτα μνήμης του συστήματος

- update()

  Η μέθοδος είναι υπεύθυνη για την εμφάνιση των έξι επόμενων η προηγούμενων αρχείων στην λίστα που διαθέτει το γραφικό περιβάλλον της κλάσης.

- updateText()

  Επιστρέφει το όνομα του επιλεγμένου αρχείου στην κλάση από την οποία έγινε η κλήση της κλάσης SDFileExplorer.

4.2.3.2 Ειδικές Κλάσεις Γραφικών

Οι κλάσεις γραφικών που θα αναλυθούν στο ακόλουθο υποκεφάλαιο γράφτηκαν ειδικά για την εφαρμογή της κεντρικής μονάδας και δύσκολα μπορούν να χρησιμοποιηθούν σε κάποια άλλη εφαρμογή χωρίς να προηγηθούν αρκετές αλλαγές στον πηγαίο κώδικα τους.

Ουσιαστικά η κάθε κλάση αποτελεί από ένα αυτοτελές menu το οποίο προσφέρει την δυνατότητα για κάποιες συγκεκριμένες ενέργειες. Συνδυάζοντας όλες αυτές τις κλάσεις μαζί με τα πληκτρολόγια και το γραφικό περιβάλλον περιήγησης στα περιεχόμενα των καρτών SD δημιουργήθηκε το γραφικό περιβάλλον της κεντρικής μονάδας που επιτρέπει όλες τις λειτουργίες που αναφέρθηκαν στην αρχή του κεφαλαίου.

Πριν προχωρήσουμε στην ανάλυση κάθε κλάσης χωριστά θα αναφέρουμε μερικές μεθόδους που περιέχονται σε όλες η στης περισσότερες κλάσεις.

- actionBack()

  Σχεδόν όλα τα γραφικά περιβάλλοντα περιέχουν την μέθοδο actionBack(), είναι η μέθοδος που καλείται όταν πατηθεί το κουμπί “Back” και επιτρέπει την πλοήγηση στο προηγούμενο menu.

```
void SDFileExplorer::actionBack() {

    update_screen = true;
    current_screen = SD_CONTROL;
    previous_screen = SD_EXPLORER;
}
```

Εικόνα 4.21: Παράδειγμα κώδικα για την πλοήγηση σε προηγούμενο μενού

- setSDCard()

  Πολλές από τις κλάσεις χρειάζονται πρόσβαση στην κάρτα μνήμης για να επαναφέρουν υποθηκευμένες ρυθμίσεις η να αποθηκεύσουν τις αλλαγές που έκανε ο χρήστης. Επίσης κάποιες κλάσεις διαβάζουν αρχεία από την κάρτα μνήμης κτλ. Για να μπορούν όλα τα παραπάνω να είναι εφικτά η εν λόγω κλάση πρέπει να έχει πρόσβαση στην κάρτα μνήμης, αυτό επιτυγχάνεται με την χρήση της μεθόδου setSDCard().

- setKeyboard()

  Σε περίπτωση που κάποια κλάση έχει ένα η παραπάνω TextBox ή TextArea και χρειάζεται να κάνει χρήση ενός από τα διαθέσιμα πληκτρολόγια για εισαγωγή κειμένου πρέπει να καλέσει την μέθοδο setKeyboard() για να πάρει πρόσβαση στην κλάση του πληκτρολογίου.

- copySettings()

  Η μέθοδος αυτή καλείται κατά την ανάγνωση του αρχείου CONF.CFG για να αντιγράψει της ρυθμίσεις του αρχείου στην αντίστοιχη κλάση.

- save()

  Μέθοδος που καλείται για να αποθηκεύσει τις κλάσεις στην κάρτα μνήμης.

```
void DotMatrixControl::save() {
    _SdFat->begin(_pin, SPI_HALF_SPEED);
    _file->open("DTMCTRL.CFG", O_WRITE | O_CREAT | O_TRUNC);
    _file->print(_dotmatrix_settings);
    _file->close();
}
```

Εικόνα 4.22: Κώδικας για την αποθήκευση ρυθμίσεων στην εσωτερική μνήμη SD

- load()

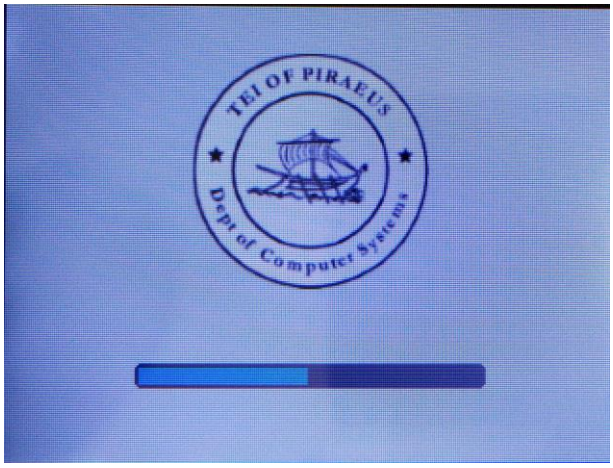
  Μέθοδος που καλείται για να επαναφέρει τις αποθηκευμένες ρυθμίσεις μιας κλάσης από την κάρτα μνήμης.

```
void DotMatrixControl::load() {
    _SdFat->begin(_pin, SPI_HALF_SPEED);
    _file->open("DTMCTRL.CFG", O_READ);
    _file->fgets(_dotmatrix_settings, sizeof(_dotmatrix_settings));
    _file->close();
}
```

Εικόνα 4.23: Κώδικας για την ανάκτηση ρυθμίσεων από την εσωτερική κάρτα μνήμης SD

4.2.3.2.1 BootScreen

Η κλάση εκτελείται κατά την εκκίνηση της συσκευής και σχεδιάζει ένα menu εκκίνησης με μπάρα που φορτώνει για όσο χρονικό διάστημα χρειάζεται η συσκευή να εκτελέσει όλες τις ενέργειες για ομαλή ενεργοποίηση του συστήματος



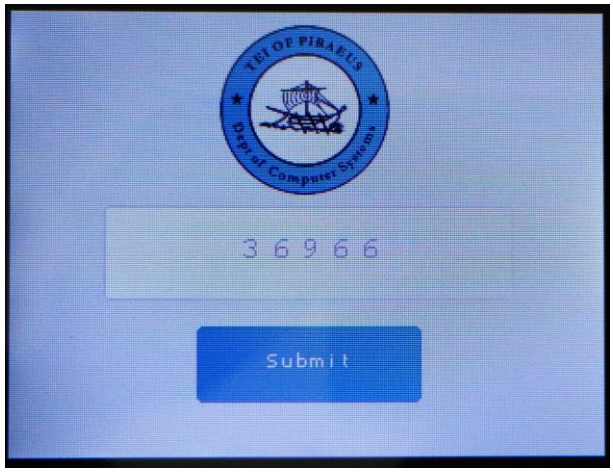
Εικόνα 4.24: Γραφική απεικόνιση της κλάσης BootScreen

- loadBar()

  Καλείται κάθε φορά που ολοκληρώνεται μια από τις λειτουργίες κατά την εκκίνηση για φορτώνει κατά ένα ποσοστό την μπάρα φόρτισης.

4.2.3.2.2 SubmitScreen

Menu στο οποίο ο χρήσης μπορεί να εκχωρήσει τον αριθμό μητρώου του για να αναζητήσει τα βαθμολογικά αποτελέσματα στα μαθήματα που τον ενδιαφέρουν.



Εικόνα 4.25: Γραφική απεικόνιση της κλάσης SubmitScreen

- actionAM()

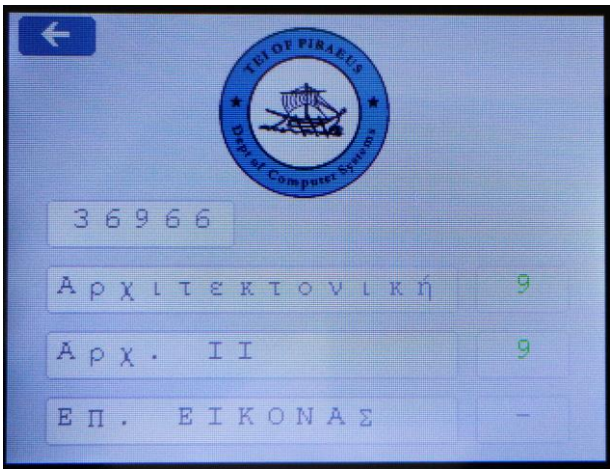
Καλείται όταν ο χριστείς πατάει το TetxBox και εμφανίζει το αριθμητικό πληκτρολόγιο για την εισαγωγή του αριθμού μητρώου του μαθητή.

- actionSubmit()

Καλείται όταν ο χρήστης πατήσει το κουμπί επιβεβαίωσης. Αν έχει καταχωρηθεί αριθμός μητρώου εμφανίζεται το menu με τα αποτελέσματα της αναζήτησης.

4.2.3.2.3 ResultScreen

Menu που παρουσιάζει τις βαθμολογίες που αναζήτησε ο χρήστης από την κλάση SubmitScreen.



Εικόνα 4.26: Γραφική απεικόνιση της κλάσης ResultScreen

- setAM()

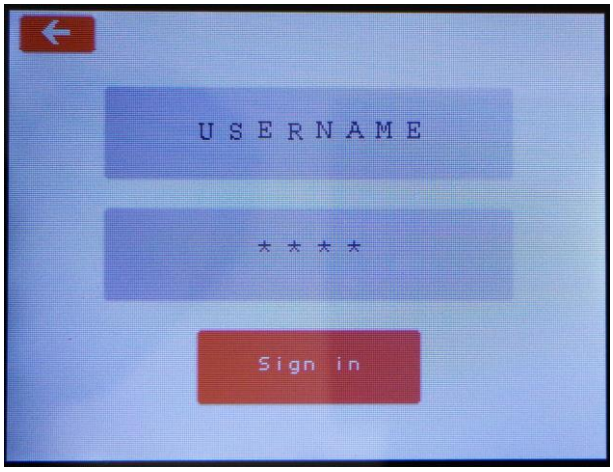
Ορίζει τον αριθμό μητρώου για τον οποίο θα γίνει η αναζήτηση. Καλείται από την κλάση SubmitScreen.

- getGrade()

Βρίσκει την βαθμολογία για κάθε μάθημα από το κατάλληλο αρχείο. Τα αρχεία με τις βαθμολογίες ορίζονται από την κλάση SDControl και αυτόματα κατά την εισαγωγή ρυθμίσεων από το αρχείο CONF.CFG.

4.2.3.2.4 LoginScreen

Σε περίπτωση που ο χρήστης της μονάδας δεν είναι κάποιος μαθητής αλλά ο διαχειριστής του συστήματος μπορεί να μπει στο menu εισόδου του συστήματος όπου εκχωρώντας το κατάλληλο όνομα χρηστη και κωδικό μπορεί να πάρει πρόσβαση στην διαχείριση τόσο της κεντρικής μονάδας όσο και της μονάδας Dot-Matrix.



Εικόνα 4.27: Γραφική απεικόνιση της κλάσης LoginScreen

- actionUser()

  Καλείται κατά το πάτημα του πλαισίου εισαγωγής όνομα χρήστη και καλεί την κλάση του πληκτρολογίου για εισάγει του ονόματος χρήστη.

- actionPass()

  Αντίστοιχα με την παραπάνω μεθώ καλείται όταν ενεργοποιηθεί το πλαίσιο εισαγωγής του κωδικού χρήστη.

- actionSubmit()

  Εκτελείται όταν πατηθεί το πλήκτρο επιβεβαίωσης. Αν το όνομα και ο κωδικός χρήση είναι σωστά τότε δίνει πρόσβαση στις ρυθμίσεις του συστήματος καλώντας την κλάση OptionScreen.

4.2.3.2.5 OptionScreen

Αποτελεί ένα ενδιάμεσο menu με τρεις επιλογές. Ο χρήστης μπορεί να επιλέξει πιο από τα τρία menu θέλει να εμφανιστεί. Υπάρχει το menu από το οποίο ο χρήστης μπορεί να αλλάξει το μήνυμα που εμφανίζεται στην μονάδα Dot-Matrix, το menu που μπορεί να αλλάξει τις ρυθμίσεις τις μονάδας Dot-Matrix και το menu από το οποίο μπορεί να διαχειριστεί την κάρτα μνήμης micro SD.

Εικόνα 4.28: Γραφική απεικόνιση της κλάσης OptionScreen

- actionMessage()

  Καλείται με το πάτημα του προτού κουμπιού. Καλεί την κλάση DotMatrixMessage όπου ο χρήστης μπορεί να αλλάξει το μήνυμα που εμφανίζεται στην μονάδα Dot-Matrix.

- actionDotMatrix()

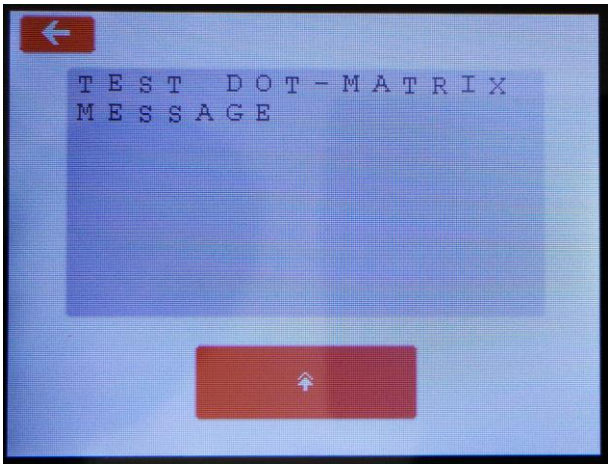
Η μέθοδος που καλείται με το πάτημα του δευτέρου κουμπιού. Καλεί την κλάση DotMatrixControl που επιτρέπει την διαχείριση της μονάδας Dot-Matrix.

- actionSD()

Το τρίτο κουμπί της κλάσης δίνει πρόσβαση στην κλάση SDControl μέσα από την οποία μπορούμε να διαχειριστούμε την κάρτα μνήμης SD.

4.2.3.2.6 DotMatrixMessage

Η κλάση σχεδιάζει ένα γραφικό περιβάλλον που αποτελείται από ένα πλαίσιο εισαγωγής κειμένου τύπου TextArea και ένα κουμπί αποστολής. Σκοπός της κλάσης είναι η συγγραφή κειμένου και η αποστολή του στην μονάδα Dot-Matrix όπου στην συνέχεια θα εμφανίζεται.



Εικόνα 4.29: Γραφική απεικόνιση της κλάσης DotMatrixMessage

- actionMessage()

Καλείται κατά το πάτημα του πλαισίου εισαγωγής κειμένου και εμφανίζει το πληκτρολόγιο για την εισαγωγή του εν λόγω κειμένου.

- setPanel()

Για να μπορέσει να αποσταλεί το μήνυμα στην μονάδα Dot-Matrix πρέπει να γίνει χρήση της κλάσης DotMatrix, με αυτήν την μέθοδο μπορούμε να πάρουμε πρόσβαση στην εν λόγω κλάση.

- actionSend()

  Αποστέλλει το μήνυμα στην μονάδα Dot-Matrix χρησιμοποιώντας την κλάση DotMatrix και την μέθοδο send() που ανήκει σε αυτήν.

- updateMessage()

  Στην σπάνια περίπτωση που την στιγμή που γράφουμε ένα μήνυμα προς αποστολή κάποιο άλλο μήνυμα λαμβάνεται από την μονάδα GSM σε μορφή SMS, η μέθοδος updateMessage() αναλαμβάνει να αναβαθμίσει σε πραγματικό χρόνο το κείμενο που βλέπουμε στο πλαίσιο κειμένου με αυτό που λήφθηκε από το SMS.

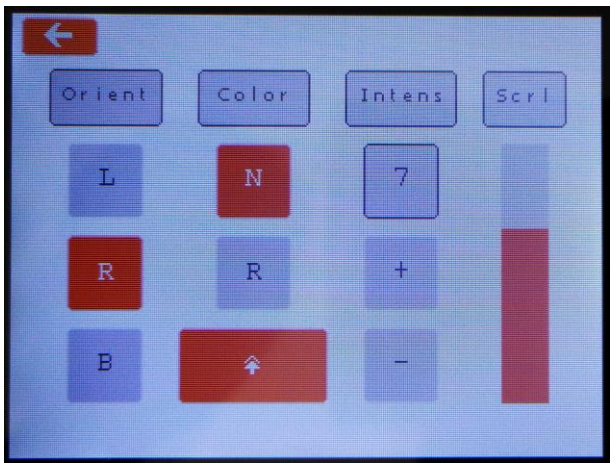
- convert()

  Μετατρέπει το μήνυμα προς αποστολή σε μορφή που μπορεί να διαβαστεί σωστά από την μονάδα Dot-Matrix. Εκτελείται πριν από την αποστολή του μηνύματος.

4.2.3.2.7 DotMatrixControl

Η κλάση DotMatrixControl δίνει την δυνατότητα στον χρήστη να αλλάξει διάφορες παραμέτρους λειτουργίας στην μονάδα Dot-Matrix. Αρχικώς ο χρήστης μπορεί να ορίσει την κατεύθυνση στην οποία θα κυλάει το κείμενο προς προβολή. Οι πιθανές επιλογές σε αυτή την περίπτωση είναι κύλιση προς τα δεξιά (κουμπί R), κύλιση προς τα αριστερά (κουμπί L) και εναλλαγή κύλισης μια προς τα δεξιά μια προς τα αριστερά (κουμπί B).

Επιπλέον υπάρχει η επιλογή για το πιο κομμάτι του κειμένου θα φωτίζεται. Οι επιλογές είναι να φωτίζονται με κόκκινο χρώμα τα γράμματα του κειμένου και το φόντο να παραμείνει μαύρο (κουμπί N) η αντίθετα τα γράμματα να παραμένουν μαύρα και το φόντο να είναι κόκκινο (κουμπί R).



Εικόνα 4.30: Γραφική απεικόνιση της κλάσης DotMatrixControl

Πρόσθετα, υπάρχουν κουμπιά για την ένταση της φωτεινότητας των LED στην μονάδα Dot-Matrix και ένα slider όπου μπορεί να οριστεί η ταχύτητα κύλισης του κειμένου.

- setPanel()

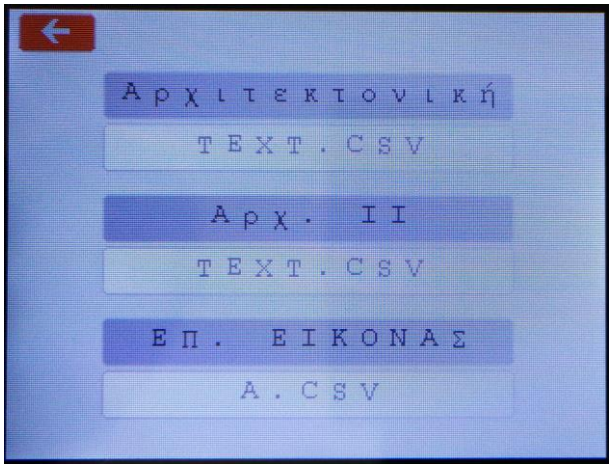
  Για να μπορέσουν να αποσταλθούν οι ρυθμίσεις στην μονάδα Dot-Matrix πρέπει να γίνει χρήση της κλάσης DotMatrix, με αυτήν την μέθοδο μπορούμε να πάρουμε πρόσβαση στην εν λόγω κλάση.

- actionSend()

  Καλείται όταν πατηθεί το κουμπί αποστολής, κάνοντας χρήση της κλάσης DotMatrix στέλνει μέσο υπερύθρων τις ρυθμίσεις στην μονάδα Dot-Matrix.

4.2.3.2.8 SDControl

Η κλάση SDControl επιτρέπει στον χρήστη να ορίσει τα αρχεία από τα οποία θα διαβάζονται οι βαθμολογίες για τα μαθήματα. Επίσης επιτρέπει την πρόσβαση στην κλάση SDFileExplorer.



Εικόνα 4.31: Γραφική απεικόνιση της κλάσης SDControl

- setFileExplorer()

  Για να μπορέσει η κλάση να κάνει χρήση της κλάσης SDFileExplorer πρέπει να εκτελεστεί η συγκεκριμένη μέθοδος.

- getFilename()

  Διαβάζει το όνομα των αρχείων που έχουν επιλεχθεί. Από αυτά τα αρχεία γίνεται η ανάγνωση των βαθμολογιών από την κλάση ResultScreen.

- actionFileSelect()

  Καλείται κατά το πάτημα ενός από τα τρία πλαίσια εισαγωγής αρχείου και επιτρέπει την επιλογή του αρχείου που θα εμφανιστεί κάνοντας χρήση της κλάσης SDFileExplorer.

4.2.3.2.9 DotMatrixFull.ino

Αποτελεί το κύριος αρχείο της εφαρμογής της κεντρικής μονάδας. Εδώ γίνεται η αρχικοποίηση όλων των προαναφερθέντων κλάσεων και βιβλιοθηκών και στην συνέχεια η εκκίνηση της συσκευής και των περιφερειακών μονάδων, GSM, οθόνη αφής, κύκλωμα υπέρυθρων κτλ. Στην συνέχεια η εφαρμογή αναμένει την έλευση μηνύματος SMS μέσο της μονάδας GSM η την ενεργοποίηση της οθόνης αφής από κάποιον χρήστη και ανάλογα προβαίνει στην κλήση κάθε μιας από τις παραπάνω κλάσεις.

```
ScreenSelect current_screen;
ScreenSelect previous_screen;
boolean update_screen;
boolean update_sd;
boolean send_order;

uint8_t* _Largefont = dejavu_16;
uint8_t* _Smallfont = tahoma_8;

SdFat internal_sd;
SdFat external_sd;

SdFile a_file;
SdFile b_file;

UTFT myGLCD(ITDB32, RS, WR, CS, RST);
UTouch myTouch(D_CLK, D_CS, D_DIN, D_OUT, DPENRQ);

Metro touch_timer = Metro(40);
Metro resend_timer = Metro(60000L);

BigKeyboard keyboard(&myGLCD, &myTouch);
SmallKeyboard smallkeyboard(&myGLCD, &myTouch);

LoginScreen loginscreen(&myGLCD, &myTouch);
SubmitScreen submitscreen(&myGLCD, &myTouch);
OptionScreen optionscreen(&myGLCD, &myTouch);
DotMatrixControl dotmatrixcontrol(&myGLCD, &myTouch);
DotMatrixMessage dotmatrixmessage(&myGLCD, &myTouch);
ResultScreen resultscreen(&myGLCD, &myTouch);
SDControl sdcontrol(&myGLCD, &myTouch);

DotMatrix dotmatrix(&Serial2, 2400, 40000);
SDSettingsImport sdsettingsimport(&sdcontrol, &resultscreen,
    &dotmatrixcontrol, &loginscreen, &dotmatrixmessage);
SDFileExplorer sdfileexplorer(&myGLCD, &myTouch, &sdsettingsimport);
Gsm gsm(&Serial1, 115200, &sdsettingsimport);
BootScreen bootscreen(&myGLCD, &myTouch, 8);
```

Εικόνα 4.32: Αρχικοποίηση των κλάσεων για την εκτέλεση της εφαρμογής

ΚΕΦΑΛΑΙΟ 5

ΕΦΑΡΜΟΓΕΣ - ΠΡΟΒΛΗΜΑΤΑ - ΒΕΛΤΙΣΤΟΠΟΙΗΣΕΙΣ

Στο κεφάλαιο που ακολουθεί θα γίνει αναφορά στις εφαρμογές που μπορεί να έχει το σύστημα που αναπτύχθηκε στο πλαίσιο στης πτυχιακής εργασίας. Στην συνέχεια θα αναλυθούν τα προβλήματα και οι δυσκολίες που παρουσιάστηκαν κατά την εκπόνηση της. Τέλος, θα αναφερθούμε σε περαιτέρω βελτιστοποιήσεις και μελλοντικές αλλαγές που μπορούν να γίνουν τόσο στο υλικό όσο και το λογισμικό με σκοπό την βελτίωση της λειτουργίας των τωρινών λειτουργιών ή και την προσθήκη νέων.

5.1 Εφαρμογές της Πτυχιακής Εργασίας

Στο πλαίσιο της πτυχιακής εργασίας όπως είδαμε και στα προηγούμενα κεφάλαια το σύστημα χρησιμοποιήθηκε για την προβολή ανακοινώσεων σε φοιτητές. Σαν σύνολο το σύστημα μπορεί να χρησιμοποιηθεί σε μια πληθώρα από σενάρια στα οποία χρειάζεται να χρησιμοποιηθούν πάνελ με συστοιχίες φωτοδιόδων (Dot-Matrix Panel) για την προβολή ανακοινώσεων και λοιπών μηνυμάτων. Στα περισσότερά πάνελ της αγοράς ο χρήστης πρέπει να συνδέσει κάποιον ηλεκτρονικό υπολογιστή σε ειδική θύρα στο πάνελ για να προβεί στην αλλαγή του μηνύματος που προβάλλεται και μόνο αφού έχει εγκαταστήσει το κατάλληλο λογισμικό . Το σύστημα μας, επιτρέπει από στον χρήστη την τοπική αλλαγή του μηνύματος και των πρόσθετων ρυθμίσεων χωρίς καμία παραπάνω συσκευή κάνοντας χρήση της κεντρική μονάδας. Ωστόσο όμως το μεγαλύτερο πλεονέκτημα της μονάδας είναι η δυνατότητα αλλαγής του μηνύματος προς προβολή απομακρυσμένα με ένα απλό SMS.

Πέρα από το σύνολο του συστήματος της πτυχιακής και μεμονωμένα κομμάτια της μπορούν να χρησιμοποιηθούν και σε άλλες κατασκευές και εφαρμογές. Το σημαντικότερο από αυτά είναι η βιβλιοθήκη γραφικών UTFTGui που αναπτύχθηκε κατά την εκπόνηση της πτυχιακής εργασίας. Με την χρήση ενός Arduino, κατά

προτίμηση Arduino Mega ή κάποιο καλύτερο λόγο της παραπάνω μνήμης που διαθέτει, και μια φθηνής οθόνης αφής μπορούμε πλέον να δημιουργήσουμε συσκευές εισόδου με ουσιαστικά ελάχιστο κόστος. Μία τέτοιά συσκευή μπορεί να έχει πάρα πολλές εφαρμογές καθώς μπορεί να ενσωματωθεί σε σχεδόν οποιαδήποτε κατασκευή στην οποία χρειάζεται κάποια διεπαφή με τον χρήστη. Με αυτόν τον τρόπο αποφεύγεται η χρήση κουμπιών και διακοπτών αφού πλέον θα μπορεί να γίνεται διαχείριση της συσκευής από ένα προσιτό και εύκολο στην χρήση γραφικό περιβάλλον.

Ένα ακόμα κομμάτι κώδικα της πτυχιακής εργασίας που θα μπορούσε να χρησιμοποιηθεί αυτόνομα είναι το κύκλωμα υπερύθρων. Ουσιαστικά με μόνο ένα υπέρυθρο LED και τον αντίστοιχο στοιχείο ανάγνωσης μπορούμε να μετατρέψουμε την ενσύρματη σειριακή επικοινωνία μεταξύ δυο μονάδων σε ασύρματη.

Τέλος η κλάση που είναι υπεύθυνη για την λήψη του μηνύματος SMS μπορεί στην ουσία να χρησιμοποιηθεί για οποιαδήποτε εφαρμογή που θέλει να κάνει χρήση της μονάδας GSM.

5.2 Προβλήματα που Παρουσιάστηκαν

Κατά την διάρκεια της εκπόνηση της πτυχιακής εργασίας παρουσιάστηκαν διάφορα προβλήματα, κάποια αφορούσαν το υλικό κομμάτι της εργασίας και κάποια άλλα το λογισμικό. Μερικά από τα σημαντικότερα ακολουθούν παρακάτω.

Το βασικότερο πρόβλημα που παρουσιάστηκε ήταν η περιορισμένη ποσότητα διαθέσιμης μνήμης και υπολογιστικής ισχύς στις μονάδες Arduino. Αυτός είναι και ο λόγος που προτιμήθηκε η χρήση δύο μονάδων Arduino καθώς μία μονάδα δεν μπορούσε να εμφανίζει ομαλά το κινούμενο κείμενο στο Dot-Matrix πάνελ και παράλληλα να επεξεργάζεται τα δεδομένα του τηλεφωνικού δικτύου και της διεπαφής του χρήστη με την οθόνη αφής.

Συγκεκριμένα το πρόβλημα της ελλιπής διαθέσιμης μνήμης έγινε εμφανές κατά την ανάπτυξη της βιβλιοθήκης γραφικών UTFTGui η οποία καταναλώνει μεγάλη ποσότητα μνήμης με για την αποθήκευση των γραφικών που θα απεικονίζονται στην οθόνη αφής. Για να ξεπεράσουμε αυτό το πρόβλημα κάναμε χρήση της μνήμης flash που χρησιμοποιείται κανονικά για την αποθήκευση του εκτελέσιμου κώδικα και όχι των μεταβλητών. Το Arduino Mega ιδιαίτερα, διαθέτει μια αρκετά μεγάλη χωρητικότητας μνήμη flash. Το μεγάλο μειονέκτημα της μνήμης flash είναι ότι μπορεί να γίνει εγγραφή σε αυτήν μόνο κατά την διάρκεια του προγραμματισμού ενώ ανάγνωση μπορεί να γίνει κανονικά και κατά την εκτέλεση του προγράμματος. Για να ξεπεράσουμε το συγκεκριμένο πρόβλημα, αποθηκεύουμε όλο το γραφικό περιβάλλον στην μνήμη flash και φορτώνουμε σε πραγματικό χρόνο την στιγμή της σχεδίασης το γραφικό κομμάτι που πρέπει να σχεδιαστεί. Με αυτόν τον τρόπο μπορούμε να έχουμε ένα μεγάλο όγκο από γραφικά περιβάλλοντα, από απλά έως πιο σύνθετα χωρίς να χρειάζεται περεταίρω μνήμη RAM.

Ένα επιπλέον πρόβλημα που παρουσιάστηκε ήταν η έλλειψη ελληνικών χαρακτήρων στην βιβλιοθήκη UTFT. Το πρόβλημα αυτό ξεπεράστηκε με την χρήση του προγράμματος Matlab, όπου καταφέραμε μέσω συνάρτησης που δημιουργήσαμε, να μετατρέψουμε οποιαδήποτε γραμματοσειρά του λειτουργικού συστήματος σε πίνακα συμβατό με την βιβλιοθήκη μας. Καταφέραμε με αυτόν τον τρόπο όχι μόνο να προσθέσουμε υποστήριξη ελληνικών χαρακτήρων αλλά και να δώσουμε την επιλογή στον προγραμματιστή να χρησιμοποιήσει από μια πληθώρα από γραμματοσειρές κάνοντας μόνο μερικά απλά βήματα.

Όσον αφορά το υλικό κομμάτι της εφαρμογής, ένα από τα προβλήματα που παρουσιάστηκαν είναι η δυσκολία της σύνδεση της οθόνης αφής με το Arduino Mega καθώς η οθόνη αφής χρειάζεται να συνδεθεί σε πολλούς ακροδέκτες και η τάση λειτουργίας της είναι τα 3.3V εν αντιθέσει με το Arduino το οποίο λειτουργεί στα 5V. Αυτό ουσιαστικά σημαίνει ότι χρειαζόμαστε έναν μεγάλο αριθμό από διαιρέτες τάση για την απροβλημάτιστη λειτουργία της μονάδας. Ευτυχώς, όπως είδαμε και στο κεφάλαιο με τα υλικά που χρησιμοποιήσαμε, υπάρχει ειδικός αντάπτορας που μας επιτρέπει να συνδέσουμε το Arduino Mega με την οθόνη αφής χωρίς να χρειάζεται κάποια άλλη συνδεσμολογία ή τροποποίηση.

5.3 Βελτιστοποιήσεις

Το σύστημα που αναπτύχθηκε στο πλαίσιο της πτυχιακής εργασίας, αν εξαιρέσουμε το μέγεθος του Dot-Matrix πάνελ, είναι ήδη πλήρως λειτουργικό και μπορεί άμεσα να χρησιμοποιηθεί σε πραγματικές συνθήκες, ωστόσο αυτό δεν σημαίνει ότι τόσο το υλικό όσο και το υλικό κομμάτι της εφαρμογής δεν επιδέχονται περαιτέρω βελτιστοποιήσεις.

Θα ξεκινήσουμε από το υλικό κομμάτι το οποίο όπως αναφέραμε και παραπάνω επιδέχεται λιγότερες βελτιστοποιήσεις από το λογισμικό κομμάτι. Ουσιαστικά οι αλλαγές αφορούν δημιουργία ειδικής πλακέτας για τα ολοκληρωμένα ώστε να αποφύγουμε την χρήση των μονάδων τύπου module που προτείνει η πλατφόρμα Arduino. Τα modules είναι εξαιρετικά για την σχεδίαση και ανάπτυξη του προτύπου μιας ηλεκτρονικής κατασκευής καθώς κάνουν την συνδεσμολογία και τις δοκιμές πολύ πιο εύκολες και σύντομες, ωστόσο από την στιγμή που έχουμε κατασταλάξει στην σύνθεση και την συνδεσμολογία μας, η δημιουργία πλακέτας ειδικά για την κατασκευή μας κάνει το κύκλωμα πολύ μικρότερο, ειδικά αν χρησιμοποιήσουμε υλικά τύπου SMD (Surface Mounted Device), και επίσης κάνει την δημιουργία μας να φαίνεται πολύ πιο επαγγελματική. Με αυτόν τον τρόπο τόσο το Dot-Matrix πάνελ όσο και η κεντρική μονάδα μπορούν να γίνουν πολύ μικρότερες έχοντας επίσης και μικρότερη κατανάλωση ρεύματος καθώς το κύκλωμα θα ήταν σχεδιασμένο και κατασκευασμένο ειδικά για της ανάγκες μας.

Επίσης, μια ακόμα βελτιστοποίηση επιτύχουμε είναι να χρησιμοποιήσουμε τρίχρωμα LED για το Dot-Matrix πάνελ. Τα τρίχρωμα LED είναι πιο ακριβά και χρειάζονται άλλο ελεγκτή για την οδήγηση τους αλλά μας επιτρέπουν, με τον συνδυασμό των τριών βασικών χρωμάτων, να έχουμε ένα έγχρωμο πάνελ στο οποίο, ειδικά αν γίνει μεγαλύτερο μπορούμε πλέον να εμφανίζουμε εκτός από πολύχρωμα μηνύματα και έγχρωμες εικόνες.

Συνεχίζοντας με το λογισμικό κομμάτι, εδώ, οι βελτιστοποιήσεις που μπορούν να γίνουν είναι αρκετές. Αρχικώς η βιβλιοθήκη γραφικών UTFTGui που δημιουργήσαμε μπορεί να εξελιχθεί παραπάνω ώστε να μπορεί να υποστηρίζει και άλλα γραφικά αντικείμενα όπως για παράδειγμα check boxes, radio buttons κτλ.,

όπως επίσης και να βελτιωθούν τα ήδη υπάρχοντα γραφικά αντικείμενα αλλά και γενικά η δομή της βιβλιοθήκης, έχοντας πάντα ως γνώμονα την όσο το δυνατόν μικρότερη χρήση της μνήμης του συστήματος. Κάτι τέτοιο είναι όμως πολύ χρονοβόρο και δεν μπορεί να γίνει στα πλαίσια της εν λόγο πτυχιακής άσκησης καθώς από μόνο του θα μπορούσε να αποτελεί θέμα για μια μελλοντική πτυχιακή εργασία.

Μια συγκεκριμένη αλλαγή που πρέπει να τονίσουμε είναι ο ορισμός των γραφικών δομών σε κάποιο αρχείο από το οποίο η εφαρμογή θα μπορεί να τα προσπελάσει. Κάτι παρόμοιο γίνεται και στο ευρέως διαδεδομένο λογισμικό για κινητές συσκευές τηλεφώνου Android. Εκεί όλες οι παράμετροι των γραφικών δομών γράφονται σε αρχεία .xml τα οποία το λειτουργικό προσπελάζει κατά την σχεδίαση του γραφικού περιβάλλοντος. Κάτι τέτοιο είναι πιο δύσκολο με την πλατφόρμα Arduino καθώς δεν υπάρχει αποθηκευτικός χώρος αντίστοιχος με την εσωτερική μνήμη SD των κινητών τηλεφώνων. Ωστόσο, στην περίπτωση μας υπάρχει διαθέσιμη μνήμη SD και κάτι τέτοιο θα ήταν εφικτό, αλλά θα καθιστούσε την βιβλιοθήκη άχρηστη σε εφαρμογές που δεν διαθέτουν ανάλογη μνήμη. Μία ιδέα θα ήταν η βιβλιοθήκη να υποστηρίζει και τους δύο τρόπους και ανάλογα με το αν ο προγραμματιστής ορίσει η όχι την χρήση της μνήμης SD αυτή να χρησιμοποιείται.

Πέραν της βιβλιοθήκης άλλες αλλαγές που θα μπορούσαν να γίνουν είναι ο προγραμματισμός άλλων τρόπων προβολής και κύλισης του μηνύματος αλλά και η προβολή όχι μόνο κειμένου αλλά και εικόνων. Τα περισσότερα από αυτά δεν εφαρμόστηκαν μόνο και μόνο διότι το πάνελ ήταν πολύ μικρό. Ωστόσο, για την προβολή εικόνων συγκεκριμένα θα χρειαζόταν κάποιο άλλο Arduino με περισσότερη διαθέσιμη μνήμη. Προφανώς και θα έπρεπε να γίνουν ακόμα πιο πολλές αλλαγές αν θέλαμε να κάνουμε το πάνελ μας πολύχρωμο όπως αναφέρθηκε παραπάνω.

Μία τελευταία πολύ σημαντική βελτιστοποίηση θα μπορούσε να είναι η χρήση της μονάδας GSM για σύνδεση στο διαδίκτυο μέσω GPRS. Από την στιγμή που θα γίνει η σύνδεση στο διαδίκτυο οι επιλογές μας πλέον είναι πάρα πολλές, από την σύνδεση σε κάποιον mail server και την ανάγνωση των αποστελλόμενων μηνυμάτων και εικόνων μέσω e-mail έως την δημιουργία ιστοσελίδας ή εφαρμογής

σε κινητή συσκευή για την αποστολή μηνυμάτων και διαχείριση του Dot-Matrix πάνελ.

ΒΙΒΛΙΟΓΡΑΦΙΑ

[1] Arduino , Learning + Reference , http://www.arduino.cc

[2] Harold Timmis , “Practical Arduino Engineering” , ISBN: 978-1-430-23885-0

[3] Ιωάννης Έλληνας , “Ψηφιακή Επεξεργασία Εικόνας & Βίντεο” , ISBN: 978-9-609-31836-5

[4] SdFat Library , SdFat Library Documentation Page , https://github.com/greiman/SdFat

[5] Metro Library , Metro Library Documentation Page , https://github.com/thomasfredericks/Metro-Arduino-Wiring/wiki

[6] UTFT Library , UTFT Library Documentation Page , http://www.rinkydinkelectronics.com

[7] Max72xxPanel Library , Max72xxPanel Library Documentation Page , https://github.com/markruys/arduino-Max72xxPanel

[8] Maxim , Serially Interfaced, 8-Digit LED Display Drivers , MAX7219 Datasheet

[9] Vishay , IR Receiver Modules for Remote Control Systems , TSOP4840 Datasheet

[10] SIMCom , SIM900 AT Command Manual V1.03 , SIM900 Manual

[11] Fairchild , NPN Epitaxial Silicon Transistor , C33725 Datasheet

[12] Motorola Semiconductor , Quad 2-Input NOR Gate , MC74C02N Datasheet