



ΙΟΝΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Πρόγραμμα Μεταπτυχιακών Σπουδών:
«Ψηφιακές Εφαρμογές και Καινοτομία»

Διπλωματική Εργασία

Ανάπτυξη εφαρμογής για έξυπνα κινητά η
οποία μετατρέπει κείμενο σε κώδικα
Braille

Κοντίνης Ιωάννης

Επιβλέπον καθηγητής: Δημήτριος Ρίγγας

Κέρκυρα, Φεβρουάριος 2023

Επιβλέπων καθηγητής

Δημήτριος Ρίγγας

Μέλος του Εργαστηριακού Διδακτικού Προσωπικού του Τμήματος Πληροφορικής του Ιονίου Πανεπιστημίου

Τριμελής Επιτροπή

Σπυρίδων Δουκάκης

Επίκουρος Καθηγητής του Τμήματος Πληροφορικής του Ιονίου Πανεπιστημίου

Δημήτριος Ρίγγας

Μέλος του Εργαστηριακού Διδακτικού Προσωπικού του Τμήματος Πληροφορικής του Ιονίου Πανεπιστημίου

Ελένη Χριστοπούλου

Μέλος του Εργαστηριακού Διδακτικού Προσωπικού του Τμήματος Πληροφορικής του Ιονίου Πανεπιστημίου

Ευχαριστίες

Με την ολοκλήρωση της παρούσας Διπλωματικής Εργασίας θα ήθελα να εκφράσω τις ευχαριστίες μου σε όλους τους καθηγητές του Μεταπτυχιακού Προγράμματος Σπουδών που με την πολύτιμη καθοδήγηση τους συνέβαλαν εποικοδομητικά στην μεγαλύτερη εμβάθυνσή μου στο αντικείμενο της επιστήμης των υπολογιστών, των ψηφιακών εφαρμογών και της καινοτομίας.

Τις πιο θερμές μου ευχαριστίες εκφράζω στον επιβλέποντα καθηγητή, κύριο Δημήτριο Ρίγγα, πρωτίστως για την εμπιστοσύνη που μου έδειξε με την ανάθεση του συγκεκριμένου θέματος της εργασίας, ιδιαιτέρως όμως για την επιστημονική καθοδήγηση, τις υποδείξεις, την συμπαράσταση και την υποστήριξη που μου παρείχε από την αρχή μέχρι το τέλος της εκπόνησης της παρούσας Διπλωματικής Εργασίας.

Επίσης θα ήθελα να ευχαριστήσω όλους του συμφοιτητές μου στο Πρόγραμμα Μεταπτυχιακών Σπουδών για την άψογη συνεργασία που είχαμε κατά την διάρκεια της εκπόνησης ομαδικών εργασιών αλλά και για τις συζητήσεις που κάναμε μέσα από τις οποίες μοιραζόμασταν τις ιδέες και τους προβληματισμούς μας. Από την επαφή μαζί τους είχα την ευκαιρία να γνωρίσω υπέροχους ανθρώπους και να δημιουργήσω ισχυρές φιλίες.

Θέλω ακόμα να ευχαριστήσω θερμά τον ξάδερφό μου Αριστείδη, απόφοιτο του Τμήματος Πληροφορικής του Ιονίου Πανεπιστημίου, για την παροχή της πλακέτας Arduino και πολλών συνοδευτικών υλικών και εξαρτημάτων τα οποία βοήθησαν σημαντικά στην κατασκευή του ηλεκτρονικού κυκλώματος που χρησιμοποιήθηκε για τις ανάγκες της Διπλωματικής Εργασίας.

Τέλος, θα ήθελα να εκφράσω τις ευχαριστίες μου και την ευγνωμοσύνη μου προς την σύντροφο μου Δώρα, για την υπομονή, την συμπαράσταση, την κατανόηση, και τη βοήθεια που μου προσέφερε σε όλη την διάρκεια της εκπόνησης της παρούσας Διπλωματικής Εργασίας.

Πίνακας Περιεχομένων

1	Εισαγωγή	10
1.1	Σχετικά με την Διπλωματική Εργασία	10
1.2	Δομή του κειμένου	11
2	Ο κώδικας Braille	12
2.1	Γενικά	12
2.2	Μορφές του κώδικα Braille	13
3	Βιβλιογραφική επισκόπηση	15
3.1	Εργασίες στις οποίες γίνεται χρήση smartphone	15
3.2	Εργασίες που υλοποιούν desktop ή web εφαρμογές	17
3.3	Εργασίες στις οποίες γίνεται χρήση μόνο μικροελεγκτών και κάμερας	20
4	Ανάλυση σχεδιασμός και χρήση εργαλείων	22
4.1	Καθορισμός των απαιτήσεων	22
4.2	Ανάλυση και σχεδιασμός	22
4.2.1	Γενικά	22
4.2.2	Πληκτρολόγηση κειμένου και μετατροπή του σε κώδικα Braille	23
4.2.3	Εξαγωγή κειμένου από εικόνα και μετατροπή του σε κώδικα Braille	24
4.2.4	Μετατροπή ομιλίας σε κείμενο και στην συνέχεια σε κώδικα Braille	25
4.2.5	Το εργαλείο ανάπτυξης της εφαρμογής	26
4.2.6	Το κύκλωμα της προσομοίωσης	27
4.3	Προγραμματιστικά εργαλεία και τεχνολογίες	28
4.3.1	MIT APP Inventor	28
4.3.2	Ο αλγόριθμος Base64	28
4.3.3	Google Apps Script	29
4.3.4	OCR	29
4.3.5	Google Drive	30
4.3.6	Αναγνώριση ομιλίας	30
4.3.7	Fun Translations Braille API	31
4.3.8	Η οικογένεια μικροελεγκτών Arduino	31
4.3.9	Arduino IDE	32
4.3.10	Η διασύνδεση Bluetooth	33
5	Παρουσίαση της εφαρμογής	34
5.1	Η αρχική οθόνη	34
5.2	Η οθόνη TypeTextScreen	35
5.3	Η οθόνη Screen2	37
5.3.1	Λήψη εικόνας	37
5.3.2	Οπτική αναγνώριση χαρακτήρων	40

5.3.3	Η απόκριση του διακομιστή	43
5.3.4	Η μετατροπή σε κώδικα Braille	45
5.3.5	Ο προγραμματισμός της προσομοίωσης του κώδικα	46
5.3.6	Ο προγραμματισμός της πλακέτας Arduino	50
5.4	Η οθόνη Voice To Braille	51
5.4.1	Φωνητική πληκτρολόγηση	51
5.4.2	Εμφάνιση μηνυμάτων	54
6	Αξιολόγηση, μελλοντικές επεκτάσεις και συμπεράσματα	57
6.1	Αξιολόγηση της εφαρμογής	57
6.1.1	Προγραμματιστική αξιολόγηση	57
6.1.2	Αξιολόγηση ευχρηστίας	58
6.2	Μελλοντικές επεκτάσεις	59
6.3	Συμπεράσματα και τελικός απολογισμός	60
	Βιβλιογραφία	62
	Παράρτημα	64

1 Εισαγωγή

1.1 Σχετικά με την Διπλωματική Εργασία

Σύμφωνα με την πλέον πρόσφατη έκθεση του Παγκόσμιου Οργανισμού Υγείας (ΠΟΥ) (2019) 284 εκατομμύρια συνάνθρωποί μας, αντιμετωπίζουν προβλήματα όρασης. Ένα μεγάλο μάλιστα ποσοστό από τους συγκεκριμένους ανθρώπους έχει απωλέσει περισσότερο από το 50% της όρασής του. Επίσης στην ίδια έκθεση αναφέρεται ότι 39 εκατομμύρια άνθρωποι έχουν χάσει πλήρως την όραση τους.

Ο αριθμός των ανθρώπων που αντιμετωπίζουν προβλήματα όρασης αυξάνει διαρκώς ένεκα κυρίως της γήρανσης του πληθυσμού αλλά και για λόγους υγείας, ενδεικτικά αναφερόμενων του διαβήτη, του καταρράκτη, του γλαυκώματος και της υποδιόρθωσης διαθλαστικού σφάλματος. Πλέον πρόσφατα στοιχεία καταδεικνύουν ότι τα τελευταία χρόνια, παρά την τεράστια πρόοδο που σημειώνεται στην ιατρική επιστήμη μεταξύ άλλων και στον συγκεκριμένο τομέα για την πρόληψη, την θεραπεία και την αποκατάσταση προβλημάτων όρασης, και παρά το γεγονός ότι η συχνότητα των ατόμων που έχουν απωλέσει πλήρως ή περιορισμένα την όρασή τους καταγράφεται μειωμένη, ο απόλυτος αριθμός τους αυξάνει συνεχώς εξαιτίας κυρίως της διαρκώς αυξανόμενης γήρανσης του πληθυσμού με την οποία η ιατρική επιστήμη αδυνατεί πλέον να συμβαδίσει [1], [2].

Οι άνθρωποι που αντιμετωπίζουν αντίστοιχα με τα ως άνω προβλήματα έχουν ανάγκη από χρήσιμα βοηθήματα που να υποστηρίζουν τις καθημερινές τους δραστηριότητες και την συμμετοχή τους στην κοινωνική ζωή.

Το Capstone Project για το οποίο γίνεται λόγος στο παρόν κείμενο εκπονήθηκε ως τελική Διπλωματική Εργασία στα πλαίσια του Προγράμματος Μεταπτυχιακών Σπουδών «Ψηφιακές Εφαρμογές και Καινοτομία» που προσφέρεται από το Τμήμα Πληροφορικής του Ιονίου Πανεπιστημίου. Το Project έχει ως στόχο να αξιοποιήσει τεχνολογίες ανοιχτού υλικού (Open Source Hardware) και λογισμικού (Open Source Software) με σκοπό την υλοποίηση μιας εφαρμογής για κινητές συσκευές η οποία θα βοηθάει τα άτομα με οπτική αναπηρία και όσους τους υποστηρίζουν εκπαιδευτικά. Το ερέθισμα που οδήγησε στην συγκεκριμένη επιλογή προέρχεται από την εκπαιδευτική υποστήριξη που παρείχα στο μάθημα των Μαθηματικών σε μαθήτρια με οπτική αναπηρία κατά το χρονικό διάστημα της φοίτησης της στη δευτεροβάθμια εκπαίδευση. Στην διάρκεια της δραστηριότητας αυτής δημιουργήθηκε σε πολλές περιπτώσεις η ανάγκη άμεσης μετατροπής σε κώδικα Braille ενός κείμενου που ήταν γραμμένο σε φυσική γλώσσα. Επιπλέον από την εμπειρία που συλλέχθηκε κατά την διαδικασία της εκπαιδευτικής υποστήριξης έγινε αντιληπτό ότι η χρήση ενός υποστηρικτικού λογισμικού θα βελτιώνει με ουσιαστικό τρόπο την παρεχόμενη βοήθεια. Επίσης από ερευνά που πραγματοποιήθηκε διαπιστώθηκε το γεγονός ότι υπάρχουν ελάχιστα λογισμικά για την υποστήριξη εκπαιδευτικών διαδικασιών σε άτομα με οπτική αναπηρία. Έτσι η δημιουργία ενός λογισμικού που θα υποστηρίζει και θα βελτιώνει την εκπαιδευτική διαδικασία ατόμων με αναπηρία αποτέλεσε μια πρόκληση. Συνέπεια όλων αυτών ήταν η σύλληψη της ιδέας για τον σχεδιασμό και την ανάπτυξη μιας εφαρμογής για έξυπνα κινητά τηλέφωνα η οποία θα έχει την δυνατότητα να μετατρέψει στην γραφή Braille κείμενο, το οποίο ο χρήστης έχει εισάγει στην εφαρμογή είτε με πληκτρολόγηση, είτε με εκφώνηση, είτε με την λήψη φωτογραφίας μέσω της κάμερας της κινητής συσκευής.

Στο σημείο αυτό είναι απαραίτητο να τονιστεί ότι ο πραγματικός σκοπός ανάπτυξης αυτής της εφαρμογής ως αντικείμενο εκπόνησης Διπλωματικής Εργασίας με την μορφή Capstone Project, τρίμηνης περίπου διάρκειας, δεν είναι ασφαλώς η δημιουργία ενός συστήματος πλούσιων δυνατοτήτων και απαλλαγμένο από λάθη, αλλά η ευκαιρία που παρουσιάζεται για ενασχόληση σε βάθος με τις βασικότερες πτυχές μιας διαδικασίας δημιουργίας ολοκληρωμένης εφαρμογής που θα προσφέρει σημαντική εμπειρία σε μελλοντικές προσπάθειες. Έτσι, μέσα από την προσπάθεια αυτή δόθηκε ευκαιρία για βαθύτερη κατανόηση, πολλών από τα αντικείμενα των μαθημάτων που περιείχε το Πρόγραμμα Μεταπτυχιακών Σπουδών αλλά και απόκτηση νέων γνώσεων πάνω σε πρακτικά ζητήματα του κλάδου της Πληροφορικής όπως τα επόμενα:

- Ο σχεδιασμός και η ανάλυση εφαρμογών για έξυπνα κινητά καθώς και η υλοποίηση τους στο περιβάλλον οπτικού προγραμματισμού MIT App Inventor.
- Η ενασχόληση με την τεχνολογία οπτικής αναγνώρισης χαρακτήρων (Optical Character Recognition - OCR) και η ανάπτυξη εφαρμογών που χρησιμοποιούν την τεχνολογία αυτή.
- Η χρήση της πλατφόρμας Google Apps Script και η δημιουργία σε αυτήν εφαρμογών που χρησιμοποιούν γλώσσα προγραμματισμού JavaScript καθώς και την ενσωμάτωση των εφαρμογών αυτών στο περιβάλλον εργασίας της Google.
- Τον προγραμματισμό εφαρμογών που επικοινωνούν μεταξύ τους και ανταλλάσσουν δεδομένα με την χρήση ενός Application Programming Interface (API).
- Τον σχεδιασμό, την ανάπτυξη και τον προγραμματισμό ηλεκτρονικών κυκλωμάτων, στο περιβάλλον του μικροελεγκτή Arduino.

1.2 Δομή του κειμένου

Στο κεφάλαιο 2 γίνεται αναφορά στον κώδικα Braille και στον τρόπο με τον οποίο αυτός κωδικοποιεί τα γράμματα, τις λέξεις, τους αριθμούς και τα σημεία στίξης μιας φυσικής γλώσσας. Επιπλέον παρουσιάζονται οι τρεις διαφορετικές εκδόσεις που έχει ο κώδικας Braille στην αγγλική γλώσσα και δίνονται παραδείγματα μέσα από εικόνες για την αντιστοιχία κάθε αριθμού, γράμματος του αγγλικού αλφάβητου ή φράσης στις δύο πιο διαδεδομένες μορφές του κώδικα.

Στο κεφάλαιο 3 παρουσιάζεται ένα σύνολο από επιστημονικές εργασίες και άρθρα τα οποία ξεχώρισαν κατά την βιβλιογραφική επισκόπηση που πραγματοποιήθηκε με έρευνα στο διαδίκτυο και η οποία είχε σκοπό να γίνει γνωστό αν υπάρχουν κινητές εφαρμογές που επιτελούν τον ίδιο στόχο, ώστε να αναλυθεί ο τρόπος λειτουργίας τους, να εξεταστεί η διεπαφή τους με τον χρήστη αλλά και να αντληθούν οι ιδέες οποίες θα χρησιμοποιηθούν για την ανάπτυξη της.

Στο κεφάλαιο 4 γίνεται ο καθορισμός των απαιτήσεων της εφαρμογής και ο τρόπος με τον οποίο πραγματοποιήθηκε ώστε να είναι συνυφασμένος με τον σκοπό που θα επιτελεί η εφαρμογή. Μετά τον καθορισμό των απαιτήσεων πραγματοποιείται αναφορά στο στάδιο της ανάλυσης και του σχεδιασμού της εφαρμογής το οποίο περιλαμβάνει μια λεπτομερή εξέταση των στόχων της εφαρμογής, ακολουθούμενη από την ανάπτυξη ενός σχεδίου για το πώς θα υλοποιηθεί προγραμματιστικά. Τέλος περιγράφονται τα βασικά προγραμματιστικά εργαλεία που θα αξιοποιηθούν για την υλοποίηση της.

Στο κεφάλαιο 5 πραγματοποιείται περιγραφή της εφαρμογής που υλοποιήθηκε και παρουσιάζονται αναλυτικά όλες οι βασικές τις λειτουργίες μέσα από εικόνες και επεξηγήσεις. Επίσης αναλύεται διεξοδικά ο κώδικας με τον οποίο υλοποιήθηκαν οι βασικότερες από τις λειτουργίες τις εφαρμογής. Επιπλέον γίνεται λόγος για τα προβλήματα που προέκυψαν κατά την διάρκεια της υλοποίησης του κώδικα αυτού, καθώς και για τους τρόπους με τους οποίους ξεπεράστηκαν τα προβλήματα αυτά.

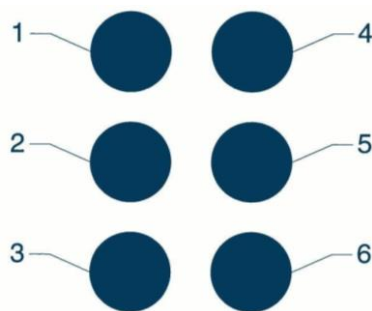
Το κεφάλαιο 6 αναφέρεται στις μελλοντικές επεκτάσεις που προτείνονται για την αναβάθμιση της εφαρμογής και την ποιοτικότερη λειτουργικότητά της. Επίσης πραγματοποιείται η σύνοψη και απολογισμός της προσπάθειας από την εκπόνηση του Capstone Project.

2 Ο κώδικας Braille

Ο κώδικας Braille είναι το σύστημα γραφής και ανάγνωσης των ατόμων με προβλήματα όρασης. Αν ο καθένας μας αναλογιστεί πόσα πράγματα διαβάζει ή γράφει καθημερινά χρησιμοποιώντας τα σπουδαία εργαλεία της γλώσσας και της όρασης θα κατανοήσει πόσα θα στερηθεί αν ζει με την απώλεια τους. Έτσι, όλοι μας μπορούμε να κατανοήσουμε την τεράστια σημασία αυτού του κώδικα. Γιατί ο κώδικας Braille δεν μεταφράζει μόνο γράμματα και λέξεις στην «γλώσσα των τυφλών». Μεταφράζει και την γνώση, την ανάγκη για γραφή και ανάγνωση, την επιθυμία για ανεξαρτησία από τρίτους.

2.1 Γενικά

Η γραφή Braille δίνει τη δυνατότητα σε άτομα με προβλήματα όρασης να διαβάζουν μέσω της αφής. Επειδή η γραφή Braille είναι κώδικας και όχι γλώσσα, μπορεί να μετατραπεί σε πολλές διαλέκτους, όπως αγγλικά, ισπανικά, γαλλικά, γερμανικά και ιταλικά ακόμα και στα κινέζικα. Κάθε σύμβολο σχηματίζεται μέσα σε ένα κελί Braille το οποίο αποτελείται από έξι ανάγλυφες κουκίδες, το λεγόμενο εξάστιγμα, οργανωμένες ανά τρεις, σε δυο κάθετες παραλλήλους. Κάθε κουκίδα μέσα σε ένα κελί έχει έναν αριθμό. Ξεκινώντας από πάνω αριστερά και κινούμενοι προς τα κάτω, οι τελείες αριθμούνται γενικά από το 1 έως το 6 όπως φαίνεται στην παρακάτω εικόνα.



Εικόνα 1

Μία μόνο κουκίδα ή οποιοσδήποτε συνδυασμός κουκίδων μπορεί να βρίσκεται σε οποιαδήποτε από τις 6 θέσεις. Όταν γίνεται αναφορά σε έναν χαρακτήρα Braille, μπορεί κανείς να περιγράψει τις θέσεις όπου βρίσκονται οι κουκίδες. Υπάρχουν 64 διαφορετικοί συνδυασμοί κουκίδων. Από τους συνδυασμούς αυτούς μπορεί κανείς να σχηματίσει το αλφάβητο σε οποιαδήποτε γλώσσα, αφού καθένας από αυτούς αντιπροσωπεύει ένα γράμμα, έναν αριθμό, ένα σημείο στίξης, ένα κομμάτι μιας λέξης ή και μια ολόκληρη λέξη ακόμα και μουσικές νότες.

Επειδή οι 64 διακριτοί χαρακτήρες δεν είναι ποτέ αρκετοί για να καλύψουν όλα τα πιθανά σύμβολα και τις παραλλαγές τους, είναι απαραίτητο να χρησιμοποιούνται ακολουθίες πολλών χαρακτήρων σε αρκετές περιπτώσεις [3]. Συχνά χρησιμοποιούμε ορισμένους χαρακτήρες ως προθέματα ή δείκτες που επηρεάζουν τη σημασία των επόμενων κελιών. Για παράδειγμα, όταν η κουκίδα 6 βρίσκεται πριν από ένα γράμμα, ο αναγνώστης ξέρει ότι το γράμμα είναι κεφαλαίο. Χωρίς την κουκίδα 6, ο αναγνώστης αντιλαμβάνεται ότι είναι πεζό γράμμα. Σε ένα άλλο παράδειγμα, οι κουκίδες 3-4-5-6, που ονομάζονται «αριθμητικός δείκτης», υποδεικνύουν στον αναγνώστη να ερμηνεύσει τα ακόλουθα γράμματα ως αριθμούς. Στη συνέχεια, υπάρχει ένας άλλος χαρακτήρας ή ένα κενό για να υποδείξει στον αναγνώστη ότι ολοκληρώθηκε το αριθμητικό κείμενο και επιστρέφει στα γράμματα.

Άλλα συστήματα σημειογραφίας, εκτός από τις φυσικές γλώσσες, όπως η μουσική, τα μαθηματικά και ο προγραμματισμός υπολογιστών χρησιμοποιούν ξεχωριστούς κώδικες Braille. Για παράδειγμα ο αντίστοιχος Κώδικας Braille για τα μαθηματικά ονομάζεται Κώδικας Nemeth. Εφευρέθηκε το 1946 από τον Abraham Nemeth και αποτελεί ένα σύστημα συμβόλων Braille που βοηθά τους ανθρώπους με προβλήματα όρασης στην ανάγνωση και γραφή των μαθηματικών.

Ο πρώτος κώδικας Braille αναπτύχθηκε για την γαλλική γλώσσα από τον Louis Braille το 1824 και αποτέλεσε εξέλιξη του κώδικα “Ecriture Nocturne” (νυχτερινή γραφή). Αυτή εφευρέθηκε από τον λοχαγό του γαλλικού στρατού Charles Barbier και σκοπός της ήταν η ανάγνωση μηνυμάτων από τους στρατιώτες στο πεδίο της μάχης κατά την διάρκεια της νύχτας χωρίς να γίνονται αντιληπτοί από τον εχθρό [4].

2.2 Μορφές του κώδικα Braille

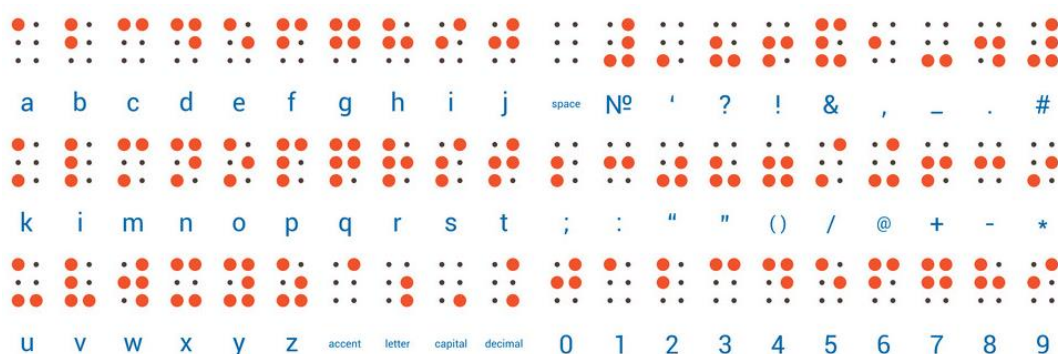
Σε πολλές γλώσσες, συμπεριλαμβανομένων των αγγλικών, υπάρχουν διαφορετικές εκδόσεις της γραφής Braille. Οι εκδόσεις αυτές είναι οι εξής:

Grade 1 Braille (Braille A' τάξης)

Είναι μια γράμμα προς γράμμα αντικατάσταση του κειμένου στους αντίστοιχους χαρακτήρες Braille. Αυτός ο τρόπος είναι προτιμώμενος για αρχάριους, επειδή τους επιτρέπει να εξοικειωθούν και να αναγνωρίσουν διαφορετικές πτυχές του κώδικα ενώ μαθαίνουν πώς να διαβάζουν Braille.

Η αγγλική Grade 1 Braille αποτελείται από τα 26 τυπικά γράμματα του αλφαβήτου καθώς και από σημεία στίξης.

Στην επόμενη εικόνα φαίνεται η αναπαράσταση σε κώδικα Braille των 26 γραμμάτων του αγγλικού αλφαβήτου καθώς και των σημείων στίξης.



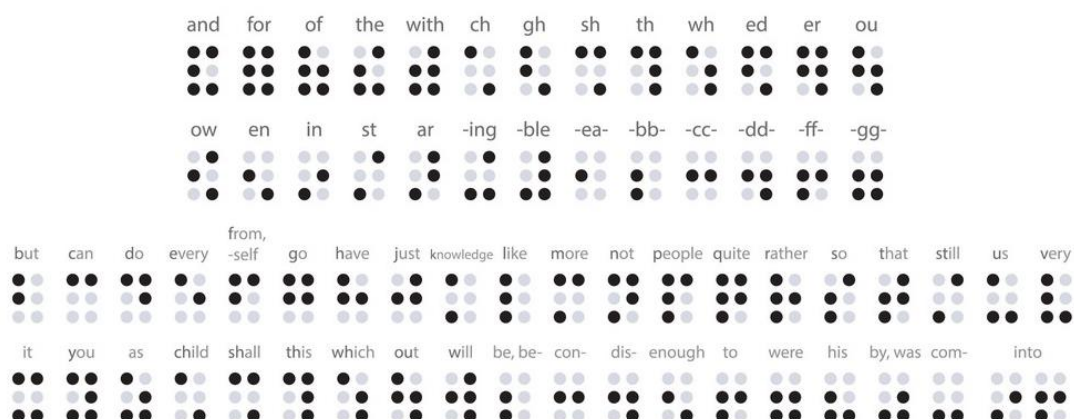
Εικόνα 2

Grade 2 Braille (Braille B' τάξης)

Ο αγγλικός κώδικας Grade 2 Braille, χρησιμοποιεί συντομογραφίες που αντιστοιχούν σε ομάδες γραμμάτων που εμφανίζονται συχνά. Για παράδειγμα, η αγγλική λέξη «the» αναπαρίσταται από ένα μόνο χαρακτήρα στη γραφή Grade 2 Braille.

Ο τύπος κώδικα Grade 2 Braille είναι χρήσιμος γιατί ένα κελί Braille είναι μεγάλο. Έχει μέγεθος περίπου ισοδύναμο της γραμματοσειράς Arial 29 pt. Αυτό σημαίνει ότι μια σελίδα κειμένου σε φυσική γλώσσα αντιστοιχεί περίπου σε τρεις σελίδες στην γραφή Braille. Οι συντομογραφίες συμβάλλουν στη μείωση του αριθμού των χαρακτήρων και ως εκ τούτου στη μείωση του συνολικού μεγέθους ενός εγγράφου. Επιπλέον, η ανάγνωση και η γραφή Braille μπορεί να είναι χρονοβόρα. Εφαρμόζοντας συντομογραφίες, χρειάζεται λιγότερος χρόνος για να γίνουν και τα δύο.

Ο τύπος Grade 2 Braille είναι η πιο συχνά χρησιμοποιούμενη μορφή κώδικα Braille και βρίσκεται σε βιβλία, δημόσια σήμανση και μενού εστιατορίων κλπ. Αποτελείται από τα 26 τυπικά γράμματα του αλφαβήτου και τα σημεία στίξης που χρησιμοποιεί ο τύπος Grade 1 Braille (εικόνα 2) καθώς επίσης και τις συντομογραφίες που φαίνονται στην επόμενη εικόνα:



Εικόνα 3

Grade 3 Braille (Braille Γ' τάξης)

Ο κώδικας Grade 3 Braille είναι η λιγότερο χρησιμοποιούμενη μορφή λογοτεχνικού κώδικα Braille που χρησιμοποιείται στην κοινότητα των ατόμων με προβλήματα όρασης. Θεωρείται συντομογραφία της γραφής Braille. Ο κώδικας αυτός συχνά συμπιέζει ολόκληρες λέξεις σε έναν ή λίγους χαρακτήρες. Δεν υπάρχει επίσημη τυποποίηση για τον κώδικα Grade 3 Braille και ως εκ τούτου οι επίσημες δημοσιεύσεις δεν το χρησιμοποιούν. Αντίθετα, εμφανίζεται πιο συχνά σε προσωπικές επιστολές, ημερολόγια και σημειώσεις.

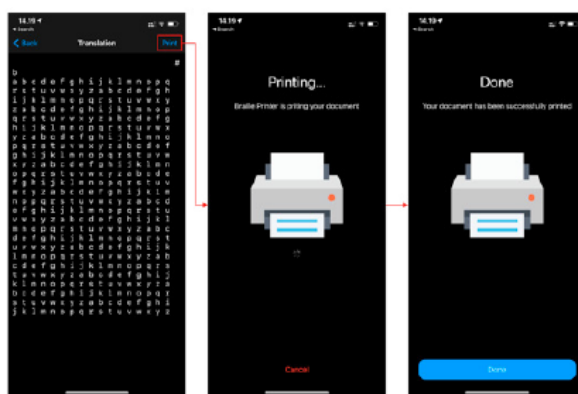
3 Βιβλιογραφική επισκόπηση

Στο κεφάλαιο αυτό γίνεται αναφορά στην βιβλιογραφική επισκόπηση που πραγματοποιήθηκε με έρευνα στο διαδίκτυο και η οποία είχε σκοπό να γίνει γνωστό αν υπάρχουν κινητές εφαρμογές που επιτελούν τον ίδιο στόχο, ώστε να αναλυθεί ο τρόπος λειτουργίας τους, να εξεταστεί η διεπαφή τους με τον χρήστη αλλά και να αντληθούν ιδέες οι οποίες θα χρησιμοποιηθούν για την ανάπτυξη της.

Από την εν λόγω έρευνα προέκυψε ότι υπάρχει πληθώρα εφαρμογών για κινητά τηλέφωνα και ταμπλέτες που έχουν ως κύριο χαρακτηριστικό τους την οπτική αναγνώριση χαρακτήρων σε εικόνες και κείμενα. Λίγες όμως είναι αυτές που χρησιμοποιούν αυτή την τεχνική για να αναπτύξουν ένα σύστημα, μια εφαρμογή ή μια συσκευή η οποία θα μπορεί να υποστηρίξει βοηθητικά τα άτομα με οπτική αναπηρία. Παρακάτω παρουσιάζονται κάποιες από αυτές τις περιπτώσεις.

3.1 Εργασίες στις οποίες γίνεται χρήση smartphone

Το άρθρο [5] εστιάζει στο πρόβλημα ότι σήμερα, υπάρχουν πολλά παλιά έγγραφα Braille που βρίσκονται σε βιβλιοθήκες, τα οποία είναι ξεπερασμένα και χρειάζονται αναπαραγωγή. Τα έγγραφα αυτά, δεν συνοδεύονται από το αρχικό τους αλφαβητικό κείμενο, ώστε να μπορούν να εκτυπωθούν ξανά από οποιονδήποτε εκτυπωτή Braille, οπότε χρειάζεται να μεταφραστούν χειροκίνητα ένα προς ένα. Αυτή η έρευνα αναπτύσσει ένα σύστημα αντιγραφής Braille ως πρόσθετο χαρακτηριστικό του Institut Teknologi Sepuluh Nopember (ITS) του εκτυπωτή Braille. Το σύστημα λειτουργεί φωτογραφίζοντας έγγραφα Braille με ένα smartphone, μεταφράζει το έγγραφο Braille σε φυσικό κείμενο και στέλνει το αποτέλεσμα ασύρματα στην πρόσθετη συσκευή που είναι εγκατεστημένη στη διεπαφή του εκτυπωτή Braille.

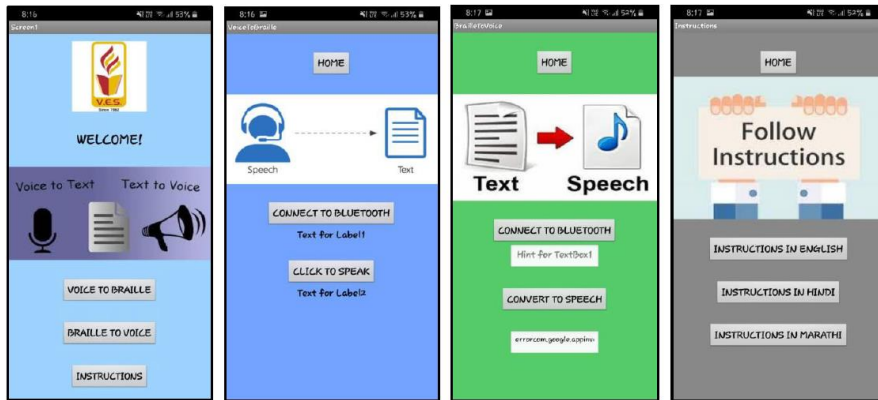


Εικόνα 4

Η μετάφραση γίνεται με επεξεργασία εικόνας. Τα αποτελέσματα των δοκιμών έδειξαν ότι η ακρίβειά του συστήματος φτάνει το 99%. Έτσι, το έγγραφο Braille μπορεί να αντιγραφεί χωρίς να χρειάζεται το αρχικό του κείμενο.

Ο κύριος σκοπός της ερευνητικής εργασίας [6] επικεντρώνεται στο γεγονός ότι λόγω της αισθητηριακής στέρησης, τα άτομα με προβλήματα όρασης αντιμετωπίζουν πολλές δυσκολίες στην αλληλεπίδραση με τον κόσμο, και χωρίς έναν βοηθό δεν μπορούν να επικοινωνήσουν με άλλους.

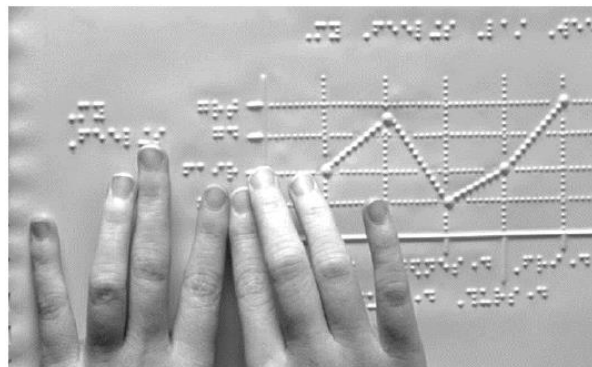
Ως εκ τούτου, προτάθηκε μια ιδέα για το σχεδιασμό και την ανάπτυξη φιλικών προς τον χρήστη και οικονομικά αποδοτικών βοηθημάτων μάθησης για παιδιά με προβλήματα όρασης. Το έργο στοχεύει να σχεδιάσει ένα εκπαιδευτικό βοήθημα για άτομα με προβλήματα όρασης στην αγγλική γλώσσα που εμπνέει την αίσθηση του παιχνιδιού κατά τη μάθηση. Η προτεινόμενη ιδέα υλοποιείται τελικά με την ανάπτυξη μιας εφαρμογής για έξυπνα κινητά, με την χρήση της πλατφόρμας MIT App Inventor, που μετατρέπει ομιλία σε Braille, και αντιστρόφως (Braille σε ομιλία) με έναν μικροελεγκτή Arduino με διασύνδεση με οθόνη LCD, φορητές συσκευές και ηλεκτρομαγνητικούς ενεργοποιητές Solenoids ως συσκευές εξόδου.



Εικόνα 5

Στόχος του έργου ήταν η κατασκευή μιας συσκευής που θα είναι οικονομικά προσιτή σε όλους. Αυτό επιτεύχθηκε με την χρήση του μικροελεγκτή χαμηλού κόστους Arduino ATMEGA 2560. Η συσκευή στοχεύει να προσφέρει αμφίδρομη επικοινωνία μεταξύ των χρηστών. Δεδομένου ότι η γλώσσα Braille είναι γνωστή στα άτομα με προβλήματα όρασης, η χρήση αυτής της συσκευής διευκολύνει την επικοινωνία τους καθώς δεν απαιτείται ιδιαίτερη εκμάθηση. Τα οφέλη αυτού του πρωτοτύπου επιτρέπουν στους ανθρώπους με προβλήματα όρασης να αλληλοεπιδρούν με άλλους χωρίς κανέναν βοηθό.

Το άρθρο [8] παρουσιάζει την πρόταση για το σχεδιασμό ενός χαμηλού κόστους εκτυπωτή Braille που εφαρμόζει την τεχνολογία οπτικής αναγνώρισης χαρακτήρων (Optical Character Recognition) μέσω μιας εφαρμογής για κινητά με έναν αλγόριθμο που επιτρέπει την αναγνώριση κειμένου σε εικόνες και την μετάφραση του ή σε κώδικα Braille ή σε μια ανάγλυφη απτική (ακολουθία που συνδέεται με διατρήσεις σε ένα φύλλο χαρτιού η οποία αναπαράγει σχήματα που σχετίζονται με σημεία και γραμμές γύρω από το κύριο περίγραμμα και δείχνει το σχήμα ή το κείμενο που εντοπίστηκε).



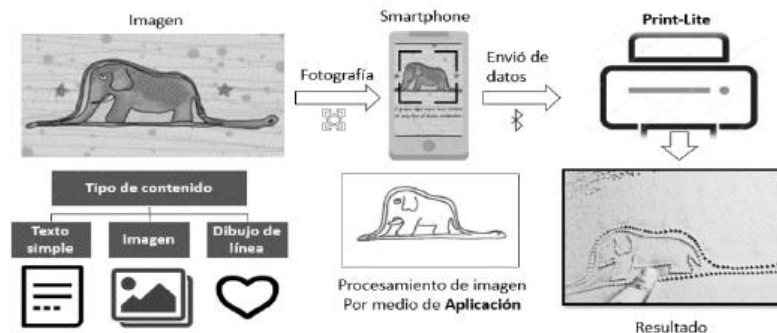
Εικόνα 6

Το πρωτότυπο που δημιουργήθηκε έχει ως στόχο να υποστηρίξει δασκάλους, γονείς ή κηδεμόνες σχετικά με τα παιδιά που χρησιμοποιούν το σύστημα Braille, διευκολύνοντας τη διαδικασία μετατροπής σε Braille βιβλίων, εγγράφων ή οποιουδήποτε άλλου εκπαιδευτικού υλικού.

Η συσκευή που κατασκευάστηκε ως πρωτότυπο ονομάστηκε Print-Lite αποτελείται από έναν μηχανισμό ο οποίος μοιάζει με σφυριά που κινούνται σε δύο άξονες και πιέζουν ένα φύλλο χαρτιού. Τα σφυριά κινούνται σύμφωνα με εντολές που παίρνουν από έναν πίνακα δεδομένων ο οποίος έχει προηγουμένως υποστεί επεξεργασία. Τα δεδομένα αυτά αντιστοιχούν σε τιμές που περιέχουν συντεταγμένες δημιουργώντας ένα ανάγλυφο, προσομοιώνοντας κώδικα Braille ή ακολουθίες σημείων που σχηματίζουν γραμμές (κύριο περίγραμμα μιας εικόνας ή ενός σχεδίου).

Σε αντίθεση με τους εμπορικούς εκτυπωτές Braille, το Print-lite λειτουργεί σε συνεργασία με μια εφαρμογή για κινητά η οποία με την τεχνολογία OCR (Optical Character Recognition)

επιτρέπει την οπτική αναγνώριση των χαρακτήρων που βρίσκονται σε μια εικόνα ή ένα σαρωμένο έγγραφο. Οι χαρακτήρες κειμένου που καταγράφονται και ερμηνεύονται μέσω της κάμερας, όταν υποβληθούν σε επεξεργασία, μεταφράζονται σε πίνακα τιμών, τα δεδομένα του οποίου αποστέλλονται μέσω Bluetooth στη συσκευή για εκτύπωση σε γραφή Braille ή/και ανάγλυφες εικόνες.

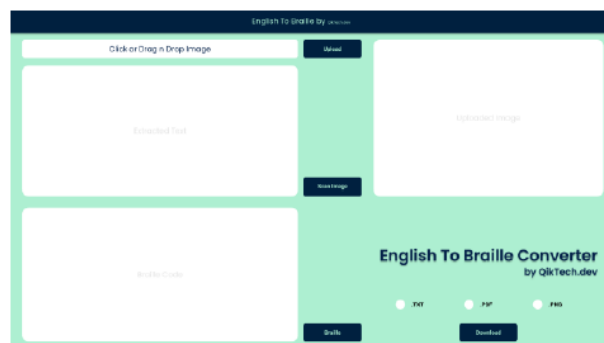


Εικόνα 7

Το σύστημα Print-lite συνδυάζει στοιχεία υλικού και λογισμικού με στόχο τη μετάφραση περιεχομένου σε γλώσσα Braille ή την προσομοίωση μιας εικόνας μέσω ενός αντιγράφου ανάγλυφων που σχεδιάζονται σε χαρτί. Οι εικόνες που λαμβάνονται από την εφαρμογή, αφού μετατραπούν σε κλίμακα του γκρι στη συνέχεια υποβάλλονται σε έναν αλγόριθμο ο οποίος αναλύει τον πίνακα εικονοστοιχείων τους αναζητώντας ξαφνικές αλλαγές και χρωματικές παραλλαγές, με τέτοιο τρόπο ώστε τα περιγράμματα να μπορούν να αναγνωριστούν πιο ως σημαντικά, ενώ παραλείπονται τα τμήματα των εικονοστοιχείων των οποίων η χρωματική διακύμανση είναι ελάχιστη ή μηδενική, αποτέλεσμα αυτού είναι να επιτυγχάνεται η αναγνώριση των αντικειμένων και δημιουργείται η προσομοίωση τους στο σχέδιο.

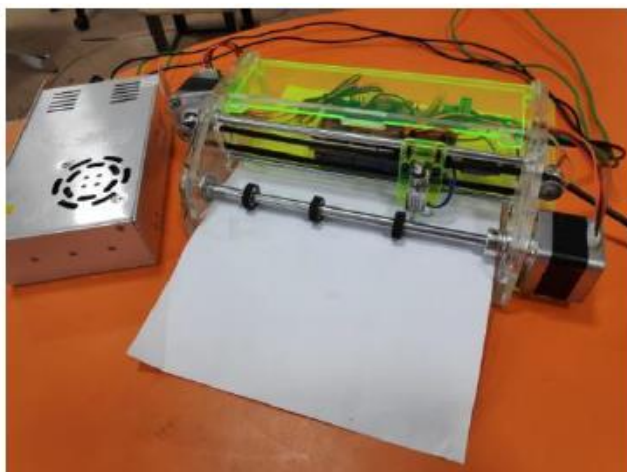
3.2 Εργασίες που υλοποιούν desktop ή web εφαρμογές

Η εργασία [9] αρχικά αναφέρεται στην πλήρη μεθοδολογία σχετικά με τον τρόπο με τον οποίο σαρώνονται οι εικόνες και πώς μεταφράζονται σε μορφή Braille. Σύμφωνα με την μεθοδολογία αυτή οι σαρωμένες εικόνες υποβάλλονται σε προεπεξεργασία και μετατρέπονται σε κλίμακα του γκρι. Στη συνέχεια περνούν από μια προσαρμοστική συνάρτηση κατωφλίου για μετατροπή σε δυαδική εικόνα. Κατόπιν, η δυαδική εικόνα αποστέλλεται για αναγνώριση κειμένου χρησιμοποιώντας την ισχυρή μηχανή οπτικής αναγνώρισης χαρακτήρων Tesseract της Google, η οποία, σύμφωνα με το άρθρο, θεωρείται ότι είναι η καλύτερη Μηχανή OCR ανοιχτού κώδικα που είναι διαθέσιμη αυτήν τη στιγμή. Τελικά, το κείμενο μετατρέπεται σε ψηφιακή μορφή Braille χρησιμοποιώντας ένα σύνολο κανόνων που παρέχονται από το www.iceb.org. Η μετάφραση σε Braille περιλαμβάνει τη μετατροπή αριθμών, αλφαβήτων, συμβόλων και σύνθετων γραμμάτων, το μεταφρασμένο κείμενο μπορεί στη συνέχεια να αποθηκευτεί για εκτύπωση. Μετά από την αναφορά στην μεθοδολογία παρουσιάζεται η υλοποίηση μιας διαδικτυακής εφαρμογής για τη μετάφραση κειμένου σε κώδικα Braille.



Εικόνα 8

Η γραφή Braille δεν είναι προσιτή σε όλους, καθώς οι εκτυπωτές Braille είναι ιδιαίτερα ακριβοί. Η εργασία [11] στοχεύει να δώσει μια λύση στο πρόβλημα αυτό αναπτύσσοντας έναν αποτελεσματικό, και φιλικό προς τον χρήστη εκτυπωτή Braille σε προσιτή τιμή, ο οποίος εκτυπώνει κώδικα Braille σε χαρτί μεγέθους A4. Σε αυτό το έργο, ο εκτυπωτής Braille κατασκευάζεται από την αρχή χρησιμοποιώντας ορισμένα ανταλλακτικά από παλιό εκτυπωτή. Το περίβλημα του εκτυπωτή κατασκευάστηκε με ακρυλικό φύλλο από υλικά προσιτού κόστους. Ο εκτυπωτής αναπτύχθηκε με τέτοιο τρόπο ώστε να μπορεί να εκτυπώσει σε χαρτί αυτό που δίνει ο χρήστης.



Εικόνα 9

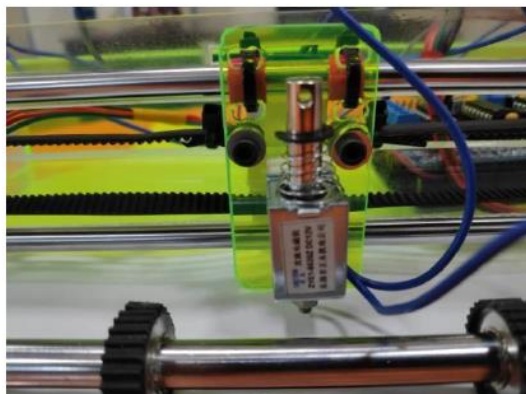
Τα εισαγόμενα κείμενα μετατρέπονται σε μορφή Unicode προκειμένου να μετατραπούν στον αντίστοιχο κώδικα Braille. Στο προτεινόμενο σύστημα δημιουργήθηκε και ένα γραφικό περιβάλλον διεπαφής στο οποίο ο χρήστης μπορεί να πληκτρολογήσει αυτό που θέλει και να το εκτυπώσει σε κώδικα Braille.

Το γραφικό περιβάλλον αναπτύχθηκε στην γλώσσα προγραμματισμού Python και περιέχει ένα πλαίσιο κειμένου όπου ο χρήστης έχει την δυνατότητα να πληκτρολογήσει κείμενό. Στην συνέχεια το εισαγόμενο κείμενο αποθηκεύεται σε αρχείο της μορφής .txt. Κάνοντας κλικ στο κουμπί εκτύπωσης, το κείμενο θα επεξεργαστεί και θα εκτελεστεί από το σενάριο Python. Μετά την ολοκλήρωση της εκτέλεσης δημιουργείται αρχείο με επέκταση .gcode. Το γραφικό περιβάλλον ζητά άδεια αποστολής του αρχείου .gcode στον ελεγκτή Arduino με την βοήθεια του οποίου πραγματοποιείται η διαδικασία της ανάγλυφης εκτύπωσης του κώδικα Braille.

Ο εκτυπωτής περιέχει δύο βηματικούς κινητήρες με τη βοήθεια των οποίων εκτυπώνεται ο κώδικας στο χαρτί. Η γωνιακή θέση κινητήρων ελέγχεται από τον μικροελεγκτή Arduino Uno. Ο ένας από τους δύο βηματικούς κινητήρες χρησιμοποιείται για την τροφοδοσία του χαρτιού. Μια λεία ράβδος προσαρτάται στον άξονα του βηματικού κινητήρα χρησιμοποιώντας τη σύζευξη άξονα. Οι ράβδοι αποτελούνται από τρεις ελαστικούς κυλίνδρους ευθυγραμμισμένους με ίσο χώρο μεταξύ τους. Στη συνέχεια, μια άλλη ράβδος ρολού σφουγγαριού τοποθετείται στην ελεύθερη βάση του συστήματος. Όταν ο κινητήρας Y περιστρέφεται, το χαρτί που ήταν ανάμεσα σε αυτές τις δύο ράβδους τείνει να κινείται προς τα εμπρός. Έτσι γίνεται η τροφοδοσία χαρτιού.

Μια ηλεκτρομαγνητική βαλβίδα χρησιμοποιείται για την παροχή γραμμικής κίνησης μέσω του ενεργοποιητή. Στο άκρο του εμβόλου της ηλεκτρομαγνητικής βαλβίδας έχει προσαρτηθεί ένας πείρος ανάγλυφης εκτύπωσης. Η ηλεκτρομαγνητική βαλβίδα είναι φτιαγμένη να κινείται ελεύθερα σε ράβδους οδήγησης. Ένας οδοντωτός ιμάντας που έχει προσαρτηθεί στην ηλεκτρομαγνητική βαλβίδα μετατρέπει την περιστροφική κίνηση του κινητήρα X σε γραμμική κίνηση. Όταν δοθεί εντολή για εκτύπωση, η ηλεκτρομαγνητική βαλβίδα θα ανοίξει και θα τρυπήσει το φύλλο για να δημιουργήσει μια κουκίδα. Όταν φτάσει στο τέλος της γραμμής, τότε ο κινητήρας ενεργοποιείται για να μετακινηθεί το φύλλο προς τα εμπρός κατά περίπου 2,5 mm. Τη στιγμή

που θα εκτυπωθεί μια γραμμή κειμένου θα μετακινηθεί το φύλλο για 5 mm. Το αποτέλεσμα της εκτύπωσης έχει δοκιμαστεί σε άτομα με προβλήματα όρασης τα οποία αναγνώρισαν με επιτυχία τους χαρακτήρες του κώδικα Braille που περιέχονταν στο εκτυπωμένο κείμενο.

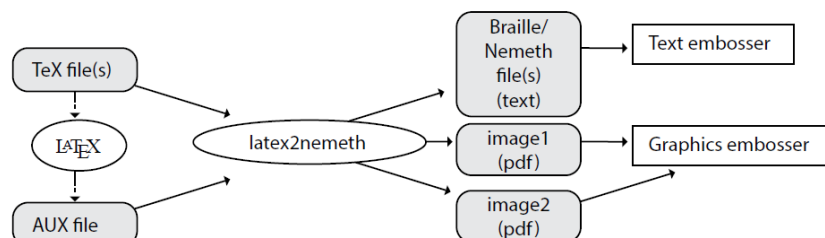


Εικόνα 10

Το άρθρο [12] ασχολείται με το γεγονός ότι δεν υπάρχει αξιόπιστο εργαλείο για την αυτόματη μεταγραφή εγγράφων από το σύστημα στοιχειοθεσίας TeX/LaTeX, που είναι το πιο διαδεδομένο σύστημα για τη δημιουργία εγγράφων στα Μαθηματικά και τις Φυσικές Επιστήμες, σε κώδικα Braille/Nemeth. Έτσι, οι φοιτητές με προβλήματα όρασης συναφών πεδίων δεν έχουν πρόσβαση στο μεγαλύτερο μέρος του εκπαιδευτικού υλικού που διατίθεται σε μορφή LaTeX.

Ως λύση στο πρόβλημα αυτό οι συγγραφείς του άρθρου ανέπτυξαν ένα εργαλείο, που ονομάζεται latex2nemeth, και το οποίο έχει την δυνατότητα να πραγματοποιεί την απευθείας μετάφραση εγγράφων LaTeX σε Nemeth/Braille, διευκολύνοντας έτσι την πρόσβαση των μαθητών και φοιτητών με προβλήματα όραση στην Επιστήμη. Προκειμένου να υποστηρίξουν το εκτεταμένο σύνολο συμβόλων των Μαθηματικών που καλύπτονται από το TeX, προτείνουν μερικά νέα σύμβολα που βασίζονται στους μηχανισμούς επέκτασης του κώδικα Nemeth. Το εργαλείο latex2nemeth, που παρουσιάζεται σε αυτή την εργασία, στοχεύει στην αξιόπιστη μετάφρασή των διαθέσιμων βιβλίων και σημειώσεων μαθημάτων, με σκοπό να υποστηρίξει την ανάπτυξη ενός αποθετηρίου μαθηματικών κειμένων για τυφλούς μαθητές και ερευνητές.

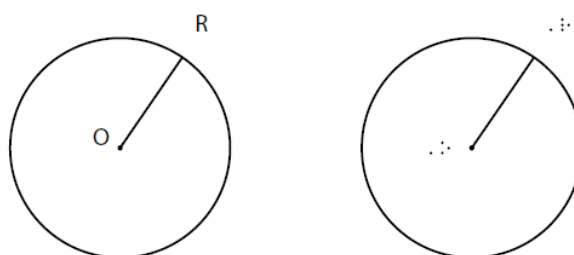
Το πρόγραμμα, εκτελείται ως εργαλείο γραμμής εντολών και λαμβάνει ως είσοδο ένα αρχείο κώδικα LaTeX καθώς και ένα βοηθητικό αρχείο LaTeX, το οποίο παράγεται μετά την επιτυχή εκτέλεση του πηγαίου αρχείου LaTeX. Το βοηθητικό αρχείο περιέχει πληροφορίες σχετικά με την αρίθμηση ορισμένων στοιχείων στο έγγραφο, όπως εξισώσεις, θεωρήματα, ενότητες, κ.λπ. Η είσοδος μπορεί να είναι ένα μεμονωμένο ή το κύριο αρχείο ενός συνόλου πολλών αρχείων LaTeX, τα οποία πρέπει να είναι κωδικοποιημένα σε κωδικοποίηση UTF-8. Το πρόγραμμα παράγει ένα αρχείο κειμένου Braille/Nemeth ως έξοδο, με χαρακτήρες Braille έξι κουκκίδων κωδικοποιημένους σε UTF-16, ή εναλλακτικά, σε μορφή PEF, ώστε να είναι άμεσα διαθέσιμο για ανάγλυφη εκτύπωση σε εξειδικευμένους εκτυπωτές.



Εικόνα 11

Το εργαλείο latex2nemeth παρέχει επίσης υποστήριξη για εικόνες. Δηλαδή, ορισμένες μακροεντολές γραφικών LaTeX που εκφράζονται στη βιβλιοθήκη γραφικών PSTricks μπορούν να φιλτραριστούν έτσι ώστε οι περιγραφές κειμένων και μαθηματικών που απεικονίζονται σε

σχήματα να μετατραπούν σε Braille/Nemeth και έτσι να αποδοθούν σωστά σε μορφή PEF ώστε να είναι ευανάγνωστες από άτομα με προβλήματα όρασης. Επιπλέον, κάθε εικόνα PSTricks έχει την δυνατότητα να αποδοθεί ως ξεχωριστό αρχείο κώδικα LaTeX που μπορεί να μεταγλωττιστεί ξεχωριστά σε PEF, έτσι ώστε να αποτυπωθεί ως ανάγλυφο ξεχωριστά από την έξοδο κειμένου Nemeth/Braille.



Εικόνα 12

Ο μεταφραστής βασίζεται σε έναν πρόγραμμα ανάλυσης (parser) της γλώσσας LaTeX. Ο αναλυτής αναγνωρίζει τις περισσότερες από τις πιο κοινές εντολές και περιβάλλοντα LaTeX σε λειτουργία κειμένου, υποστηρίζοντας αγγλικούς και ελληνικούς χαρακτήρες και καλύπτει τις περισσότερες δομές και βασικά μαθηματικά σύμβολα. Το πρόγραμμα ανάλυσης υλοποιήθηκε στη γλώσσα προγραμματισμού Java χρησιμοποιώντας τον μεταγλωττιστή JavaCC για τη δημιουργία του.

Οι δημιουργοί του λογισμικού πραγματοποίησαν μια μελέτη περίπτωσης η οποία περιελάμβανε τη μεταγραφή ενός ολόκληρου βιβλίου από LaTeX σε μορφή Braille/Nemeth και τη μελέτη αυτού του βιβλίου από φοιτήτρια Μαθηματικών σε Ελληνικό Πανεπιστήμιο ως εγχειρίδιο στο μάθημα της Πραγματικής Ανάλυσης. Η φοιτήτρια ήταν εντελώς τυφλή με άριστες επιδόσεις στα Μαθηματικά και άπταιστη ανάγνωση στον κώδικα Braille/Nemeth. Μετά την μελέτη του μαθήματος υποβλήθηκαν στην φοιτήτρια ένα σύνολο ερωτήσεων σε μορφή δομημένης συνέντευξης για την αξιολόγηση της ικανοποίησης της από την ανάλυση εγγράφων. Συνολικά, η μελέτη περίπτωσης και η συνέντευξη αποκάλυψαν ότι η μετατροπή του βιβλίου σε Braille ήταν κατάλληλη για έναν φοιτητή μαθηματικών με οπτική αναπηρία.

3.3 Εργασίες στις οποίες γίνεται χρήση μόνο μικροελεγκτών και κάμερας

Στην εργασία [7] υλοποιείται και εφαρμόζεται ένα βελτιωμένο σύστημα Braille που βοηθά τα άτομα με οπτική αναπηρία να διαβάζουν κείμενο που είναι γραμμένο σε φυσική γλώσσα. Το σύστημα που αναπτύχθηκε λειτουργεί σαρώνοντας μια εικόνα σε μια ψηφιακή κάμερα, στην συνέχεια επεξεργάζεται την εικόνα αυτή με τεχνικές επεξεργασίας εικόνας και την μετατρέπει σε κείμενο χρησιμοποιώντας τεχνικές οπτικής αναγνώρισης χαρακτήρων. Το κείμενο που εντοπίστηκε στην εικόνα δίνεται στον μικροελεγκτή Raspberry Pi ο οποίος αναγνωρίζει κάθε χαρακτήρα και τον μετατρέπει σε κώδικα Braille. Κατά την αντιστοίχιση κάθε χαρακτήρα, το κατάλληλο σήμα του αποστέλλεται σε ηλεκτρομαγνητικές βαλβίδες που είναι συνδεδεμένες με το Raspberry Pi μέσω ακίδων GPIO. Το σήμα ερμηνεύεται από ηλεκτρομαγνητικές βαλβίδες και οι ηλεκτρομαγνητικές βαλβίδες που λαμβάνουν υψηλό σήμα αναπηδούν για να δημιουργήσουν έναν χαρακτήρα Braille.



Εικόνα 13

Το άρθρο [10] αναφέρεται στην κατασκευή μιας συσκευής που αποτελείται από μια κάμερα η οποία καταγράφει μια εικόνα οποιασδήποτε σελίδας ενός βιβλίου, εγγράφου ή περιοδικού, ανιχνεύει το κείμενο και το μετατρέπει σε ήχο καθώς επίσης και σε ανανεώσιμους χαρακτήρες Braille.

Η ανίχνευση κειμένου επιτυγχάνεται χρησιμοποιώντας το Tesseract, ένα εργαλείο OCR το οποίο λαμβάνεται φορτώνοντας ορισμένα από τα σύνολα βιβλιοθηκών της Python για επεξεργασία φυσικής γλώσσας. Η Python βοηθά επίσης στη μετατροπή των κειμένων που έχουν εντοπιστεί σε αρχεία ήχου παρέχοντας τη βιβλιοθήκη GTTS (Google Text-to-Speech). Επιπλέον χρησιμοποιείται ο μικροελεγκτής Arduino Mega 2560 Rev3 ο οποίος μπορεί να προγραμματιστεί είτε με Embedded C είτε MicroPython. Ο μικροελεγκτής υποστηρίζει παράλληλη διεπαφή κάμερας 8 bit και περισσότερες από 50 ατομικά προγραμματιζόμενες ακίδες gpio. Γίνεται χρήση των εικοσιτεσσάρων από τις διαθέσιμες ακίδες gpio για να ενεργοποιηθούν ηλεκτρομηχανικά 24 ακίδες Braille. Η Κάμερα που χρησιμοποιείται στην συσκευή είναι η μινιατούρα TTL Serial JPEG Camera, λόγω της ικανότητάς της να μεταδίδει δεδομένα με ευκολία μέσω μιας σειριακής σύνδεσης TTL στο Arduino. Το κείμενο που ανιχνεύεται από την κάμερα μετατρέπεται σε ήχο από τον μετατροπέα κειμένου σε ομιλία της Google και στη συνέχεια αποστέλλεται στον μικροελεγκτή Arduino για μετατροπή σε Braille. Οι εικόνες που λαμβάνονται αποθηκεύονται σε διακομιστή ή μεταδίδονται απευθείας στις συσκευές για μετατροπές.

Η συσκευή περιέχει τέσσερις μονάδες Braille που η καθεμία αποτελείται από 6 ακίδες οι οποίες ανυψώνονται ηλεκτρομηχανικά, όπως δηλαδή ανυψώνεται μια ηλεκτρομαγνητική βαλβίδα λόγω του ηλεκτρομαγνητικού φαινομένου που προκαλείται από τη ροή του ρεύματος. Τα άτομα με προβλήματα όρασης μπορούν να ακούσουν ή να περάσουν το χέρι τους στις μονάδες Braille και να διαβάσουν το περιεχόμενο του κειμένου.

4 Ανάλυση σχεδιασμός και χρήση εργαλείων

Σε αυτό το κεφάλαιο γίνεται λόγος για τον καθορισμό των απαιτήσεων της εφαρμογής στον οποίον αξιοποιήθηκαν οι γνώσεις και οι ιδέες που προέκυψαν από την βιβλιογραφική επισκόπηση ώστε αυτός να είναι συνυφασμένος με τον σκοπό που θα επιτελεί η εφαρμογή. Μετά τον καθορισμό των απαιτήσεων πραγματοποιείται αναφορά στο στάδιο της ανάλυσης και του σχεδιασμού της εφαρμογής. Τέλος περιγράφονται τα βασικά προγραμματιστικά εργαλεία και οι τεχνολογίες που θα αξιοποιηθούν για την υλοποίηση της.

4.1 Καθορισμός των απαιτήσεων

Το Capstone Project έχει ως στόχο να αξιοποιήσει τεχνολογίες ανοιχτού υλικού (Open Source Hardware) και λογισμικού (Open Source Software) με σκοπό την υλοποίηση μιας εφαρμογής για έξυπνες κινητές συσκευές η οποία θα βοηθάει τα άτομα με οπτική αναπηρία και όσους τους υποστηρίζουν εκπαιδευτικά. Μέσα από την εφαρμογή αυτή ο χρήστης θα έχει την ευκαιρία να χρησιμοποιεί ένα ψηφιακό πακέτο από βοηθητικά εργαλεία που θα του παρέχουν την δυνατότητα της αυτόματης μετάφρασης σε κώδικα Braille κειμένων που είναι γραμμένα στην αγγλική γλώσσα και έχουν εισαχθεί στην εφαρμογή, με πληκτρολόγηση, με οπτική αναγνώριση χαρακτήρων σε εικόνες η ακόμα και με την εκφώνηση τους.

Πιο αναλυτικά, οι απαιτήσεις που καθορίστηκαν να παρέχονται από την χρήση της εφαρμογής και σύμφωνα με τον σκοπό που αυτή θα επιτελεί, είναι άμεσα συνυφασμένες με τις εξής λειτουργίες:

- Πληκτρολόγηση ενός κειμένου και μετατροπή του σε κώδικα Braille κατόπιν εντολής του χρήστη.
- Εξαγωγή κειμένου από εικόνες σε μορφή jpg, jpeg, png, gif και έγγραφα σε μορφή pdf με την χρήση της τεχνολογίας οπτικής αναγνώρισης χαρακτήρων (OCR).
- Δυνατότητα διόρθωσης από τον χρήστη του εξαγόμενου κειμένου στην περίπτωση που αυτό περιέχει λάθη.
- Μετατροπή του εξαγόμενου κειμένου σε κώδικα Braille κατόπιν εντολής του χρήστη.
- Φωνητική πληκτρολόγηση κειμένου μέσω της τεχνολογίας αναγνώρισης ομιλίας.
- Δυνατότητα διόρθωσης από τον χρήστη του εκφωνημένου κειμένου σε περίπτωση που αυτό περιέχει λάθη.
- Μετατροπή του εκφωνημένου κειμένου σε κώδικα Braille κατόπιν εντολής του χρήστη.
- Σε όλες τις προηγούμενες λειτουργίες το παραγόμενο αποτέλεσμα της μετάφρασης του κειμένου σε κώδικα Braille θα υπάρχει η δυνατότητα να αποσταλεί μέσω ασύρματης διασύνδεσης Bluetooth σε μικροελεγκτή χαμηλού κόστους, ώστε να είναι δυνατόν να προσομοιωθεί η αναπαράσταση κάθε χαρακτήρα του κώδικα από λυχνίες LED. Μελλοντικά οι λυχνίες LED θα αντικατασταθούν από ηλεκτρομαγνητικές βαλβίδες ώστε να κατασκευαστεί μια φορητή (portable) συσκευή ανάγνωσης του κώδικα Braille που θα χρησιμοποιείται από άτομα με οπτική αναπηρία.

Στην επόμενη ενότητα παρουσιάζεται διεξοδικά η ανάλυση και ο σχεδιασμός των λειτουργιών που προέκυψαν από τον καθορισμό των απαιτήσεων της εφαρμογής.

4.2 Ανάλυση και σχεδιασμός

4.2.1 Γενικά

Με την ολοκλήρωση του καθορισμού των απαιτήσεων ακολουθεί το στάδιο της ανάλυσης και του σχεδιασμού της εφαρμογής το οποίο περιλαμβάνει μια λεπτομερή εξέταση των απαιτήσεων και των στόχων της εφαρμογής, ακολουθούμενη από την ανάπτυξη ενός σχεδίου για το πώς θα υλοποιηθεί προγραμματιστικά ώστε να καλύψει αυτές τις απαιτήσεις και τους στόχους. Επιπλέον καθορίζονται τα δεδομένα που θα επεξεργάζεται η εφαρμογή κατά την αλληλεπίδραση της με τον χρήστη αλλά και η δομή η οποία θα χρησιμοποιηθεί για την διαχείριση και την αποθήκευση τους. Ακόμα γίνεται η επιλογή των τεχνολογιών και των εργαλείων που θα χρησιμοποιηθούν, καθώς και η αρχιτεκτονική της εφαρμογής με τελικό σκοπό την δημιουργία ενός

ολοκληρωμένου σχεδίου για την κατασκευή μιας εφαρμογής που θα ανταποκρίνεται στις ανάγκες των χρηστών.

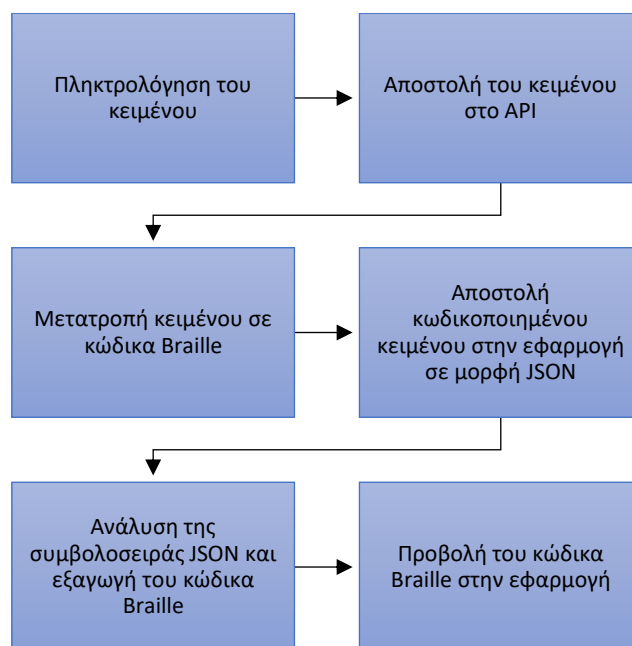
Από την διαδικασία της ανάλυσης προέκυψε ότι είναι ιδιαίτερος πλεονεκτικό η μετάφραση του κειμένου σε κώδικα Braille να γίνεται με την χρήση ενός API. Το προς μετάφραση κείμενο θα αποστέλλεται στο API με HTTP Request. Στην συνέχεια το API θα πραγματοποιεί την μετάφραση και θα διαβιβάζει το αποτέλεσμα της στην εφαρμογή σε μια συμβολοσειρά σε μορφή JSON (JavaScript Object Notation). Η συμβολοσειρά θα αναλύεται από την εφαρμογή ώστε να εξαχθεί το παραγόμενο αποτέλεσμα και να προβληθεί στην εφαρμογή.

Στο σημείο αυτό πρέπει να αναφερθεί ότι ένας εναλλακτικός τρόπος μετάφρασης κειμένων σε κώδικα Braille θα μπορούσε να πραγματοποιηθεί από την εφαρμογή χωρίς την βοήθεια ενός API αλλά με την χρήση μιας δομής δεδομένων τύπου πίνακα ή λεξικού μέσω της οποίας θα οριζόταν η αντιστοιχία σε κώδικα Braille κάθε χαρακτήρα της αγγλικής γλώσσας. Προτιμήθηκε όμως η χρησιμοποίηση ενός API διότι παρουσιάζει σημαντικά πλεονεκτήματα έναντι του τρόπου που προαναφέρθηκε.

Ένα από τα βασικά πλεονεκτήματα του API είναι ότι παρέχει μεγάλη ευκολία στην χρήση του καθώς απαιτούνται λίγες μόνο γραμμές κώδικα για να προγραμματιστεί η λειτουργία της αποστολής αιτημάτων HTTP αλλά και της παραλαβής και ανάλυσης των δεδομένων απόκρισης. Επιπλέον η υλοποίηση της διαδικασίας μετάφρασης εξ ολοκλήρου από την εφαρμογή με την χρήση δομών δεδομένων, εκτός του ότι απαιτεί κάποιο χρόνο για να ολοκληρωθεί, εμπεριέχει και τον κίνδυνο εμφάνισης λογικών λαθών και παραλήψεων. Αντιθέτως το API το οποίο χρησιμοποιήθηκε, συντηρείται και ελέγχεται τακτικά από επαγγελματίες προγραμματιστές ώστε να είναι πλήρως εναρμονισμένο με όλους τους κανόνες και τις απαιτήσεις του αγγλικού κώδικα Braille βαθμού 2, γεγονός που μειώνει στο ελάχιστο την πιθανότητα εμφάνισης λαθών και παραλήψεων. Το API που επιλέχθηκε για την μετάφραση ονομάζεται Fun Translations Braille API.

4.2.2 Πληκτρολόγηση κειμένου και μετατροπή του σε κώδικα Braille

Μια βασική παράμετρος που είναι απαραίτητο να ληφθεί υπόψιν στην φάση της ανάλυσης έχει να κάνει με τη διαδικασία μέσω της οποίας θα γίνεται η μετάφραση σε κώδικα Braille του κειμένου που έχει πληκτρολογήσει ο χρήστης στην κινητή του συσκευή. Από την ανάλυση καθορίστηκε η συγκεκριμένη διαδικασία να γίνεται με την χρήση του API ακολουθώντας τα βήματα που φαίνονται στο επόμενο διάγραμμα.



Εικόνα 14

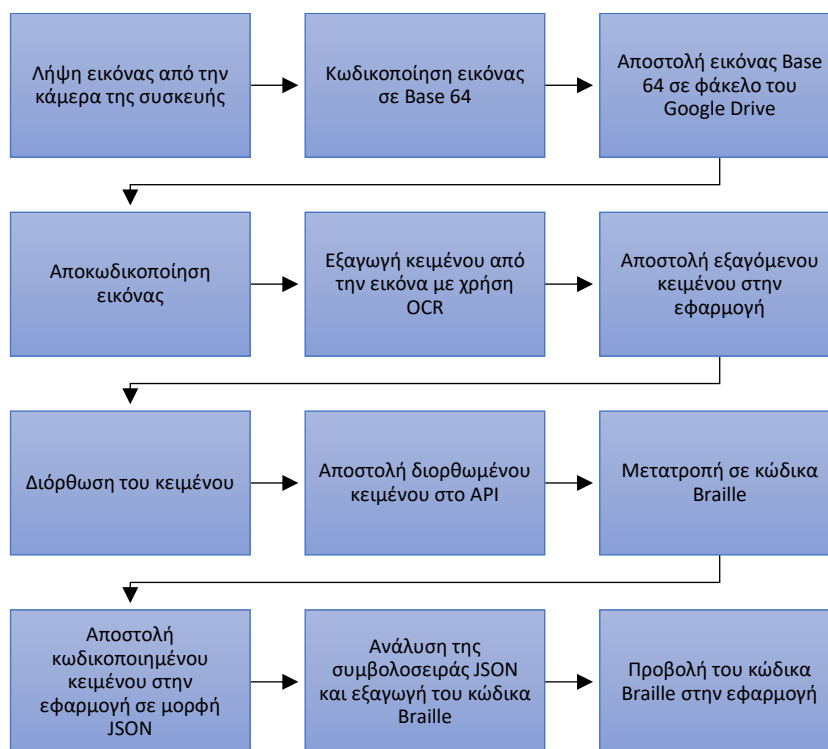
Η διαδικασία ξεκινάει με την πληκτρολόγηση του κειμένου από τον χρήστη. Στην συνέχεια κατόπιν εντολής του το κείμενο αποστέλλεται με HTTP Request στο API το οποίο αναλαμβάνει την διαδικασία της μετάφρασης του σε κώδικα Braille. Με την ολοκλήρωση της μετάφρασης το κείμενο διαβιβάζεται από το API στην εφαρμογή με την μορφή συμβολοσειράς JSON. Η εφαρμογή θα προγραμματιστεί κατάλληλα ώστε να αναλύσει την συμβολοσειρά JSON, να εξάγει τον κώδικα Braille, και να παρουσιάσει το αποτέλεσμα σε κουκίδες Braille.

Όπως αναφέρθηκε στον καθορισμό των απαιτήσεων η εφαρμογή θα έχει την δυνατότητα να εξάγει κείμενο από εικόνες με την χρήση της τεχνολογίας οπτικής αναγνώρισης χαρακτήρων και να το μεταφράζει σε κώδικα Braille. Η διαδικασία αυτή αποτελεί ίσως την κυριότερη και πιο σημαντική λειτουργία της εφαρμογής οπότε είναι μείζονος σημασίας να αναλυθούν εκτενώς τα βήματα που θα ακολουθηθούν για την υλοποίηση της. Επιπλέον είναι ιδιαίτερος σημαντικό να καθοριστεί το κατάλληλο προγραμματιστικό εργαλείο με το οποίο θα πραγματοποιείται η οπτική αναγνώριση χαρακτήρων. Το εργαλείο αυτό θα πρέπει να είναι σε πλήρη συμβατότητα με τις υπόλοιπες τεχνολογίες και εργαλεία που χρησιμοποιούνται στην ανάπτυξη της εφαρμογής ώστε η διαδικασία της οπτικής αναγνώρισης χαρακτήρων να πραγματοποιείται με επιτυχία.

Λαμβάνοντας υπόψιν τις προηγούμενες παραμέτρους καθορίστηκε η τεχνολογία OCR της Google ως το καταλληλότερο προγραμματιστικό εργαλείο με το οποίο θα πραγματοποιείται η οπτική αναγνώριση χαρακτήρων. Η επιλογή του εργαλείου αυτού βασίστηκε στο γεγονός, ότι το Google OCR χρησιμοποιεί προηγμένους αλγόριθμους μηχανικής εκμάθησης για την αναγνώριση κειμένου σε εικόνες και σαρωμένα έγγραφα, με αποτέλεσμα υψηλή ακρίβεια και χαμηλά ποσοστά σφαλμάτων. Επιπλέον είναι διαθέσιμο μέσω του cloud, επομένως δεν υπάρχουν προκαταβολικά έξοδα που σχετίζονται με την αγορά υλικού ή λογισμικού. Η διαθεσιμότητα του δε μέσω του cloud, δίνει την δυνατότητα εύκολης ενσωμάτωσης του σε άλλες εφαρμογές και υπηρεσίες. Επίσης έχει ιδιαίτερα φιλική διεπαφή προς τον χρήστη.

4.2.3 Εξαγωγή κειμένου από εικόνα και μετατροπή του σε κώδικα Braille

Στο επόμενο διάγραμμα παρουσιάζονται τα βήματα της διαδικασίας μέσω της οποίας θα γίνεται η οπτική αναγνώριση χαρακτήρων σε εικόνες, η εξαγωγή του κειμένου, η μετάφραση του σε κώδικα Braille και η προβολή του αποτελέσματος στην εφαρμογή.



Εικόνα 15

Η διαδικασία ξεκινάει με την λήψη εικόνας από την ψηφιακή κάμερα της κινητής συσκευής και στην συνέχεια την κωδικοποίηση της με τον αλγόριθμο Base64. Το Base64 είναι ένα σχήμα κωδικοποίησης δυαδικών δεδομένων σε κείμενο που αναπαριστά τα δυαδικά δεδομένα σε κώδικα ASCII. Χρησιμοποιείται για την κωδικοποίηση δυαδικών δεδομένων όπως εικόνες, αρχεία ήχου και βίντεο, σε μορφή που μπορεί να μεταδοθεί μέσω καναλιών επικοινωνίας που βασίζονται σε κείμενο, όπως email και HTTP.

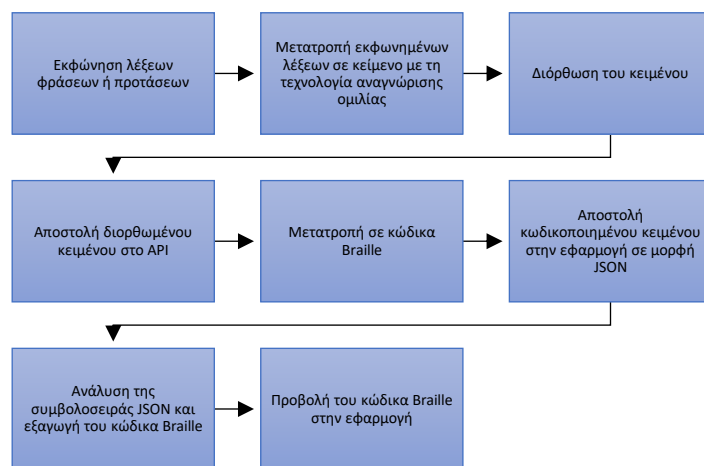
Η κωδικοποιημένη εικόνα αποστέλλεται μέσω HTTP POST σε φάκελο του Google Drive όπου και αποθηκεύεται. Το αίτημα HTTP διαχειρίζεται ένα σενάριο εντολών που θα υλοποιηθεί στην πλατφόρμα Google Apps Script και με το οποίο θα πραγματοποιείται η αποθήκευση της εικόνας στον συγκεκριμένο φάκελο και η αποκωδικοποίησή της, δηλαδή η επαναφορά της στην αρχική της μορφή. Επιπλέον το σενάριο θα περιέχει εντολές οι οποίες θα ενεργοποιούν την χρήση της τεχνολογίας Google OCR στην συγκεκριμένη εικόνα ώστε να γίνει η ανάγνωση και η εξαγωγή του κειμένου που αυτή περιέχει. Η εκτέλεση του σεναρίου ολοκληρώνεται με την αποστολή του εξαγόμενου κειμένου στην εφαρμογή μέσω HTTP POST και την διαγραφή της εικόνας από τον φάκελο του Google Drive για λόγους εξοικονόμησης αποθηκευτικού χώρου.

Στην συνέχεια αναλαμβάνει δράση η εφαρμογή η οποία θα προγραμματιστεί κατάλληλα ώστε να πραγματοποιούνται τα επόμενα βήματα της διαδικασίας. Η εφαρμογή είναι απαραίτητο να δίνει στον χρήστη την δυνατότητα να διορθώσει ή να τροποποιήσει το κείμενο που λήφθηκε από την εικόνα σε περίπτωση που περιέχονται λάθη σε αυτό ή κάποιοι χαρακτήρες δεν έχουν αναγνωριστεί σωστά από την τεχνολογία Google OCR. Κατόπιν το κείμενο διαβιβάζεται με αίτημα HTTP POST στον διακομιστή του API όπου πραγματοποιείται η μετατροπή του σε κώδικα Braille. Με την ολοκλήρωση της μετατροπής το κωδικοποιημένο κείμενο μεταβιβάζεται στην εφαρμογή με την μορφή συμβολοσειράς JSON. Η εφαρμογή θα προγραμματιστεί με τα κατάλληλα σενάρια εντολών τα οποία θα αναλύουν την συμβολοσειρά JSON ώστε να εξαχθεί ο κώδικας Braille, και να παρουσιάσει το αποτέλεσμα της μετάφρασης του κειμένου σε κουκίδες Braille.

Από τον καθορισμό των απαιτήσεων της εφαρμογής ο οποίος παρουσιάστηκε στην προηγούμενη ενότητα, προέκυψε ότι μια ακόμη δυνατότητα η οποία θα προσφέρετε στον χρήστη είναι η εκφώνηση λέξεων, φράσεων και προτάσεων, η μετατροπή τους σε κείμενο και η μετάφραση του κειμένου αυτού σε κώδικα Braille. Η αναγνώριση ομιλίας θα πραγματοποιείται με την χρήση της τεχνολογίας αναγνώρισης ομιλίας η οποία παρέχεται από την Google και μπορεί να χρησιμοποιηθεί σχεδόν σε οποιαδήποτε έξυπνη κινητή συσκευή.

4.2.4 Μετατροπή ομιλίας σε κείμενο και στην συνέχεια σε κώδικα Braille

Στο επόμενο διάγραμμα παρουσιάζονται τα βήματα της διαδικασίας μέσω της οποίας θα πραγματοποιείται η αναγνώριση ομιλίας, η μετατροπή της σε κείμενο, η μετάφραση του κειμένου σε κώδικα Braille και η προβολή του αποτελέσματος στην εφαρμογή.



Εικόνα 16

Η διαδικασία ξεκινάει με την εκφώνηση λέξεων, φράσεων ή προτάσεων από τον χρήστη προς της συσκευή αφού προηγουμένως έχει ενεργοποιήσει την χρήση της τεχνολογίας αναγνώρισης ομιλίας μέσω διεπαφής που θα του προσφέρει η εφαρμογή. Στην συνέχεια η εφαρμογή θα προγραμματιστεί κατά τέτοιο τρόπο ώστε να χρησιμοποιεί την τεχνολογία αναγνώρισης ομιλίας της Google και να μετατρέπει τις εκφωνημένες λέξεις σε κείμενο. Η εφαρμογή είναι απαραίτητο να δίνει στον χρήστη την δυνατότητα να διορθώσει ή να τροποποιήσει το κείμενο που λήφθηκε από την αναγνώριση της ομιλίας σε περίπτωση που περιέχονται λάθη σε αυτό.

Η διαδικασία συνεχίζεται με την διαβίβαση του κειμένου μέσω αιτήματος HTTP GET στον διακομιστή του API όπου πραγματοποιείται η μετατροπή του σε κώδικα Braille. Με την ολοκλήρωση της μετατροπής το κωδικοποιημένο κείμενο μεταβιβάζεται στην εφαρμογή με την μορφή συμβολοσειράς JSON. Η εφαρμογή θα προγραμματιστεί με τα κατάλληλα σενάρια εντολών τα οποία θα αναλύουν την συμβολοσειρά JSON ώστε να εξαχθεί ο κώδικας Braille, και να παρουσιάσει το αποτέλεσμα της μετάφρασης του κειμένου σε κουκίδες Braille.

4.2.5 Το εργαλείο ανάπτυξης της εφαρμογής

Ένας από τους βασικότερους συντελεστές στην ανάπτυξη μιας εφαρμογής είναι το εργαλείο ανάπτυξης και αυτό γιατί ανάλογα με τις δυνατότητες ή τους περιορισμούς του μπορεί να καθορίσει σε μεγάλο βαθμό το τελικό αποτέλεσμα και την ποιότητα του. Με βάση το γεγονός ότι το Capstone Project, για το οποίο γίνεται λόγος σε αυτό το κείμενο, ήταν τρίμηνης διάρκειας αλλά και εξαιτίας της σημαντικότητας του εγχειρήματος ήταν απαραίτητο να επιλεγεί ένα εργαλείο υλοποίησης που να ικανοποιεί στο μέγιστο δυνατό βαθμό όλες τις απαιτήσεις, τις ανάγκες και το χρονοδιάγραμμα του Project.

Λαμβάνοντας υπόψιν αυτές τις παραμέτρους αλλά και τον καθορισμό των απαιτήσεων της εφαρμογής που έχει ως στόχο υλοποίησης της η παρούσα Διπλωματική Εργασία, επιλέχθηκε η ανάπτυξη της εφαρμογής να γίνει με το MIT App Inventor που είναι ένα οπτικό περιβάλλον προγραμματισμού που βασίζεται σε μπλοκ, το οποίο είναι κατάλληλο για την ανάπτυξη εφαρμογών για κινητές συσκευές. Ορισμένοι από τους λόγους για τους οποίους επιλέχθηκε να χρησιμοποιηθεί το MIT App Inventor για την ανάπτυξη της εφαρμογής είναι οι εξής:

1. Ευκολία χρήσης: Το MIT App Inventor διαθέτει μια φιλική προς το χρήστη διεπαφή που επιτρέπει τη δημιουργία εφαρμογών χωρίς να απαιτείται συγγραφή κώδικα. Αυτό είναι ένα σημαντικό πλεονέκτημα έναντι άλλων εργαλείων καθώς διευκολύνει να αναπτυχθούν εφαρμογές, ακόμα και όταν δεν υπάρχει ισχυρό υπόβαθρο και εμπειρία στην συγγραφή κώδικα.
2. Ταχεία ανάπτυξη: Με το MIT App Inventor, υπάρχει η δυνατότητα να δημιουργηθούν σε πολύ σύντομο χρονικό διάστημα γρήγορα πρωτότυπα εφαρμογών και να δοκιμαστούν ιδέες, επιτρέποντάς στις ιδέες αυτές να μετακινηθούν γρήγορα σε υλοποίηση πράγμα που μπορεί να βοηθήσει στην εξοικονόμηση χρόνου.
3. Πρόσβαση σε λειτουργίες Android: Το MIT App Inventor παρέχει πρόσβαση σε πολλές από τις δυνατότητες και τους αισθητήρες που είναι διαθέσιμοι σε συσκευές Android, όπως η κάμερα, το επιταχυνσιόμετρο και το GPS, τα οποία μπορούν να αξιοποιηθούν για τη δημιουργία καινοτόμων εφαρμογών.
4. Υποστήριξη κοινότητας: Το MIT App Inventor έχει μια μεγάλη και ενεργή κοινότητα χρηστών, οι οποίοι μπορούν να παρέχουν υποστήριξη, συμβουλές και να μοιράζονται πόρους γεγονός που σε πολλές περιπτώσεις είναι ιδιαίτερα βοηθητικό στην ανάπτυξη μιας εφαρμογής.
5. Δωρεάν και ανοιχτού κώδικα: Το MIT App Inventor είναι δωρεάν και ανοιχτού κώδικα, πράγμα που σημαίνει ότι μπορεί να υπάρχει πρόσβαση στον πηγαίο κώδικα και να πραγματοποιηθούν τροποποιήσεις ανά πάσα στιγμή χρειαστεί.

Ωστόσο, αξίζει να σημειωθεί ότι το App Inventor έχει και περιορισμούς όσον αφορά τους τύπους εφαρμογών που είναι δυνατόν να δημιουργηθούν με αυτό σε σχέση με πιο προηγμένα και επαγγελματικά περιβάλλοντα ανάπτυξης εφαρμογών για έξυπνες κινητές συσκευές. Επιπλέον πρέπει να γίνει σαφές ότι ο βασικότερος λόγος που οδήγησε στην επιλογή του έχει να κάνει με

το γεγονός ότι υπήρχε ήδη εξοικείωση με το περιβάλλον η οποία προήλθε από την παρακολούθηση του μαθήματος «Κινητό Υπολογίζει και Εφαρμογές» κατά την διάρκεια του Προγράμματος Μεταπτυχιακών Σπουδών. Η εξοικείωση αυτή σε συνδυασμό με τον τρίμηνο χρονικό περιορισμό ολοκλήρωσης της εκπόνησης της Διπλωματικής Εργασίας ο οποίος δεν άφηνε πολλά χρονικά περιθώρια εκμάθησης ενός πιο προηγμένου και επαγγελματικού εργαλείου ανάπτυξης ήταν οι δυο ουσιαστικότεροι λόγοι που οδήγησαν στην επιλογή του MIT App Inventor ως το βασικό περιβάλλον υλοποίησης της εφαρμογής.

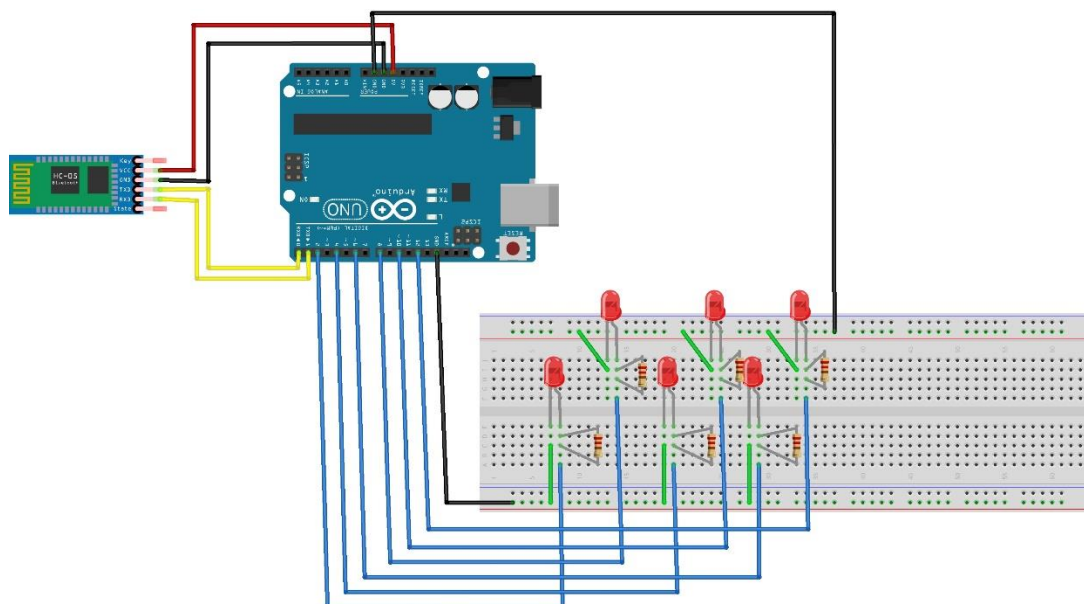
4.2.6 Το κύκλωμα της προσομοίωσης

Από τον καθορισμό των απαιτήσεων της εφαρμογής προέκυψε επίσης η ανάγκη προσομοίωσης του παραγόμενου κώδικα Braille από 6 λυχνίες LED οι οποίες θα διατάσσονται ανά δύο σε τρεις σειρές, αναπαριστώντας έτσι την αντίστοιχη διάταξη που έχουν οι κουκίδες στο εξάστιγμο του κώδικα Braille.

Οι λυχνίες θα βρίσκονται σε ηλεκτρονικό κύκλωμα το οποίο θα ελέγχεται από τον μικροελεγκτή χαμηλού κόστους Arduino. Ο μικροελεγκτής θα προγραμματιστεί κατάλληλα ώστε να επιτρέπει το άναμμα ή το σβήσιμο της κάθε λυχνίας ανάλογα με το αν χρησιμοποιείται η όχι η κουκίδα στην οποία αντιστοιχεί για την αναπαράσταση ενός συγκεκριμένου χαρακτήρα στον κώδικα Braille. Ο κώδικας Braille θα διαβιβάζεται από την εφαρμογή στον Arduino μέσω της διασύνδεσης Bluetooth. Για να πραγματοποιηθεί επιτυχώς η διαδικασία της προσομοίωσης είναι απαραίτητο ο κώδικας Braille στον οποίο μεταφράστηκε το κείμενο να αποσταλεί από την εφαρμογή στον Arduino σε μορφή που να είναι άμεσα επεξεργάσιμη από αυτόν και να επιτρέπει τον εύκολο προγραμματισμό του. Η μονάδα Bluetooth που χρησιμοποιήθηκε στο κύκλωμα είναι η HC-05 ή οποία αποτελεί μια ευρέως χρησιμοποιούμενη μονάδα Bluetooth σε ηλεκτρονικά έργα.

Η προσομοίωση του κώδικα Braille θα χρησιμεύσει μελλοντικά όταν θα αντικατασταθούν οι λυχνίες LED από ηλεκτρομαγνητικές βαλβίδες (Solenoids) ώστε να κατασκευαστεί μια φορητή συσκευή ανάγνωσης κώδικα Braille.

Το ηλεκτρονικό κύκλωμα που κατασκευάστηκε για τις ανάγκες της προσομοίωσης παρουσιάζεται στην επόμενη εικόνα:



Εικόνα 17

4.3 Προγραμματιστικά εργαλεία και τεχνολογίες

Στην ενότητα αυτή παρουσιάζονται τα βασικότερα προγραμματιστικά εργαλεία καθώς και οι τεχνολογίες που χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής και δίνονται πληροφορίες για τον τρόπο χρήσης και την λειτουργία τους.

4.3.1 MIT APP Inventor

Το MIT App Inventor είναι μια δωρεάν διαδικτυακή πλατφόρμα η οποία αποτελεί ένα ολοκληρωμένο περιβάλλον ανάπτυξης εφαρμογών για έξυπνα κινητά και tablet, βασισμένο στον οπτικό προγραμματισμό με την χρήση μπλοκ. Χρησιμοποιεί ένα γραφικό περιβάλλον χρήστη (GUI) πολύ παρόμοιο με τη γλώσσα προγραμματισμού Scratch, το οποίο επιτρέπει στους χρήστες να μεταφέρουν και να αποθέσουν (drag and drop) οπτικά αντικείμενα για να δημιουργήσουν μια εφαρμογή που μπορεί να εκτελεστεί σε συσκευές Android. Για να μπορέσει ένας χρήστης να χρησιμοποιήσει το MIT App Inventor και να δημιουργήσει εφαρμογές απαιτείται να διατηρεί έναν ενεργό λογαριασμό στην Google και σύνδεση στο διαδίκτυο. Η εκτέλεση και ο εντοπισμός σφαλμάτων της υλοποιούμενης εφαρμογής μπορεί να γίνει μέσω ενός emulator (εξομοιωτή), το οποίο είναι ένα λογισμικό που προσομοιώνει στον υπολογιστή ένα κινητό τηλέφωνο. Για να μπορέσει να λειτουργήσει αποτελεσματικά ο emulator απαιτείται ο χρήστης να έχει εγκαταστήσει στον υπολογιστή του το λογισμικό MIT App Inventor Tools το οποίο περιέχει όλα τα αρχεία που είναι απαραίτητα για την λειτουργία του emulator και την προσομοίωση της εφαρμογής σε αυτόν. Εναλλακτικά η εκτέλεση και ο εντοπισμός σφαλμάτων της υλοποιούμενης εφαρμογής είναι δυνατόν να πραγματοποιηθεί με τη βοήθεια της εφαρμογής App Inventor Companion η οποία μπορεί να ληφθεί δωρεάν από το Play Store και επιτρέπει την εκτέλεση και δοκιμή της αναπτυσσόμενης εφαρμογής σε φυσική συσκευή. Η δοκιμή της αναπτυσσόμενης εφαρμογής με την χρήση του APP Inventor Companion είναι αυτή που συνίσταται και από τους κατασκευαστές της πλατφόρμας MIT App Inventor, εξαιτίας του γεγονότος ότι παρουσιάζει πολλά πλεονεκτήματα έναντι της δοκιμής σε emulator. Ενδεικτικά μπορούμε να αναφέρουμε την λήψη εικόνων και βίντεο από την κάμερα της φυσικής συσκευής, τον εντοπισμό της γεωγραφικής θέσης, αλλά και την αλληλεπίδραση με την οθόνη αφής, ενέργειες οι οποίες δεν μπορούν να προσομοιωθούν πλήρως με την χρήση του emulator.



Εικόνα 18

Εξ αιτίας του γεγονότος ότι το εργαλείο βασίζεται σε οπτικό περιβάλλον προγραμματισμού, με την χρήση μπλοκ επιτρέπει σε όλους, ακόμη και στους αρχάριους χρήστες, να δημιουργούν πλήρως λειτουργικές εφαρμογές για έξυπνα κινητά και tablet. Επιπλέον διευκολύνει στη δημιουργία σύνθετων εφαρμογών υψηλής απόδοσης σε σημαντικά λιγότερο χρόνο από τα παραδοσιακά περιβάλλοντα προγραμματισμού.

Το MIT App Inventor αρχικά δημιουργήθηκε από την Google ενώ τώρα συντηρείται από το Ινστιτούτο Τεχνολογίας της Μασαχουσέτης (MIT). Είναι δωρεάν λογισμικό ανοιχτού κώδικα που κυκλοφορεί με διπλή άδεια χρήσης, άδεια Creative Commons και άδεια Apache 2.0 για τον πηγαίο κώδικα, πράγμα το οποίο επιτρέπει σε μια μεγάλη κοινότητα από προγραμματιστές να δημιουργούν επεκτάσεις που βελτιώνουν και προσθέτουν επιπλέον λειτουργικότητα στα στοιχεία του εργαλείου.

4.3.2 Ο αλγόριθμος Base64

Ο Base64 είναι ένας αλγόριθμος κωδικοποίησης που μετατρέπει οποιουσδήποτε χαρακτήρες, δυαδικά δεδομένα, ακόμη και εικόνες ή αρχεία ήχου σε μια ευανάγνωστη συμβολοσειρά, η οποία μπορεί να αποθηκευτεί ή να μεταφερθεί μέσω του δικτύου χωρίς απώλεια δεδομένων. Η

κωδικοποίηση Base64 λειτουργεί διαιρώντας τα δυαδικά δεδομένα σε ομάδες των 6 bit και αναπαριστά κάθε ομάδα ως χαρακτήρα από ένα περιορισμένο σύνολο 64 χαρακτήρων, συμπεριλαμβανομένων γραμμάτων, αριθμών και συμβόλων. Αυτό επιτρέπει την αναπαράσταση δυαδικών δεδομένων ως κείμενο, το οποίο μπορεί να αποθηκευτεί και να μεταδοθεί επίσης ως κείμενο και στη συνέχεια να αποκωδικοποιηθεί ξανά στην αρχική τους δυαδική μορφή για χρήση.



Εικόνα 19

Η κωδικοποίηση Base64 χρησιμοποιείται ευρέως για την κωδικοποίηση δυαδικών δεδομένων για αποθήκευση ή μετάδοση σε συστήματα που βασίζονται σε κείμενο και χρησιμοποιείται συχνά για την κωδικοποίηση δυαδικών δεδομένων σε έγγραφα JSON και XML, καθώς και για την κωδικοποίηση διαπιστευτηρίων χρήστη για βασικό έλεγχο ταυτότητας HTTP. Ο Base64 χρησιμοποιείται επιπλέον και ως κωδικοποίηση μεταφοράς MIME (Επεκτάσεις αλληλογραφίας πολλαπλών χρήσεων Διαδικτύου) για email.

Οι εικόνες Base64 χρησιμοποιούνται κυρίως για την ενσωμάτωση δεδομένων εικόνας σε άλλες μορφές όπως HTML, CSS ή JSON. Συμπεριλαμβάνοντας δεδομένα εικόνας σε ένα έγγραφο HTML, το πρόγραμμα περιήγησης δεν χρειάζεται να υποβάλει πρόσθετο αίτημα στον διακομιστή για την ανάκτηση του αρχείου, καθώς η εικόνα είναι ήδη ενσωματωμένη στο έγγραφο HTML. Η αναπαράσταση Base64 έχει μεγαλύτερο μέγεθος μνήμης από μια εικόνα που δεν έχει κωδικοποιηθεί και η συμβολοσειρά γίνεται πολύ μεγάλη για μεγάλες σε μέγεθος εικόνες, έτσι για τον λόγο αυτό η κωδικοποίηση Base64 έχει βέλτιστα αποτελέσματα για μικρές σε μέγεθος μνήμης εικόνες.

4.3.3 Google Apps Script

Το Google Apps Script είναι μια πλατφόρμα ταχείας ανάπτυξης εφαρμογών η οποία καθιστά γρήγορη και εύκολη τη δημιουργία επαγγελματικών εφαρμογών που ενσωματώνονται στο περιβάλλον εργασίας της Google (Google Workspace). Στην πλατφόρμα αυτή ένας προγραμματιστής μπορεί να γράψει κώδικα στη σύγχρονη μορφή της γλώσσας προγραμματισμού JavaScript και να έχει πρόσβαση σε ενσωματωμένες βιβλιοθήκες εντολών για δημοφιλείς εφαρμογές της Google όπως το Gmail, το ημερολόγιο, το Drive και άλλα. Αυτό γίνεται εφικτό χωρίς να απαιτείται η εγκατάσταση κάποιου λογισμικού στον υπολογιστή του προγραμματιστή, αφού η πλατφόρμα παρέχει μια εφαρμογή επεξεργασίας κώδικα που μπορεί να προσπελαστεί απευθείας από το πρόγραμμα περιήγησης και τα σενάρια εκτελούνται στους διακομιστές της Google.



Εικόνα 20

4.3.4 OCR

Τα αρχικά OCR προέρχονται από τις λέξεις Optical Character Recognition (Οπτική Αναγνώριση Χαρακτήρων) η οποία είναι μια τεχνολογία που αναγνωρίζει κείμενο μέσα σε μια ψηφιακή εικόνα. Είναι δηλαδή μια διαδικασία μετατροπής μιας εικόνας κειμένου σε μορφή κειμένου η οποία είναι αναγνώσιμη από μηχανή. Χρησιμοποιείται συνήθως για την αναγνώριση κειμένου σε σαρωμένα έγγραφα και εικόνες και την μετατροπή του σε προσβάσιμη ηλεκτρονική μορφή. Κατά την σάρωση ενός εγγράφου ή μιας φωτογραφίας που είναι τυπωμένα σε χαρτί, ο σαρωτής δημιουργεί σε ένα αρχείο μια ψηφιακή εικόνα του εντύπου αυτού. Το κείμενο που περιέχεται στο παραγόμενο αρχείο δεν μπορεί να χρησιμοποιηθεί από ένα πρόγραμμα επεξεργασίας κειμένου με το οποίο θα ήταν δυνατή η μορφοποίηση και περεταίρω επεξεργασία του ή η

αναζήτηση και καταμέτρηση των λέξεων που αυτό περιέχει. Για να καταστούν εφικτές αυτές οι διαδικασίες γίνεται χρήση ενός λογισμικού τεχνολογίας OCR με το οποίο μετατρέπεται η ψηφιακή εικόνα σε επεξεργάσιμο αρχείο κειμένου.



Εικόνα 21

Στις μέρες μας η τεχνολογία OCR έχει γίνει διαθέσιμη στο διαδίκτυο ως υπηρεσία σε περιβάλλον υπολογιστικού νέφους και σε εφαρμογές για κινητές συσκευές, όπως η μετάφραση γλωσσών σε πραγματικό χρόνο από έξυπνα κινητά τηλέφωνα. Με την έλευση των έξυπνων τηλεφώνων, η τεχνολογία OCR μπορεί να χρησιμοποιηθεί σε εφαρμογές κινητών συσκευών που είναι συνδεδεμένες στο Διαδίκτυο και εξάγουν κείμενο που καταγράφηκε χρησιμοποιώντας την κάμερα της συσκευής. Αυτές οι συσκευές δεν διαθέτουν τεχνολογία OCR ενσωματωμένη στο λειτουργικό τους σύστημα αλλά χρησιμοποιούν συνήθως ένα OCR API για να εξαγάγουν το κείμενο από το αρχείο εικόνας που καταγράφηκε από την κάμερα της συσκευής. Το OCR API επιστρέφει το εξαγόμενο κείμενο, στην εφαρμογή της κινητής συσκευής για περαιτέρω επεξεργασία, όπως μετατροπή του κειμένου σε ομιλία ή την προβολή αυτού.

4.3.5 Google Drive

Το Google Drive είναι μια υπηρεσία αποθήκευσης cloud που αναπτύχθηκε από την Google. Επιτρέπει στους χρήστες να αποθηκεύουν και να έχουν πρόσβαση στα αρχεία τους, συμπεριλαμβανομένων φωτογραφιών, βίντεο και εγγράφων, από οπουδήποτε. Το Google Drive περιλαμβάνει επίσης έγγραφα, υπολογιστικά φύλλα και παρουσιάσεις Google, τα οποία είναι εργαλεία διαδικτυακής σουίτας γραφείου της Google που επιτρέπει στους χρήστες να δημιουργούν και να επεξεργάζονται έγγραφα, υπολογιστικά φύλλα και παρουσιάσεις. Με το Google Drive, οι χρήστες μπορούν επίσης να συνεργάζονται με άλλους σε πραγματικό χρόνο, να μοιράζονται αρχεία και να αφήνουν σχόλια και να κάνουν ανατροφοδότηση.



Εικόνα 22

Ένα από τα βασικά πλεονεκτήματα της χρήσης του Google Drive είναι η δυνατότητα που δίνεται στους χρήστες του να έχουν πρόσβασης στα αρχεία τους από οποιαδήποτε συσκευή με σύνδεση στο διαδίκτυο, καθώς και η εύκολη κοινή χρήση αρχείων με άλλους. Το Google Drive προσφέρει επίσης ενοποίηση με άλλες υπηρεσίες της Google, όπως το Gmail και το Google Photos, και περιλαμβάνει διάφορες λειτουργίες ασφαλείας, όπως κρυπτογράφηση και έλεγχο ταυτότητας δύο παραγόντων, για την προστασία των δεδομένων χρήστη.

Το Google Drive διαθέτει επίσης τεχνολογία οπτικής αναγνώρισης χαρακτήρων (OCR) που μπορεί να εξάγει αυτόματα κείμενο από εικόνες και σαρωμένα έγγραφα. Αυτή η δυνατότητα επιτρέπει στους χρήστες να αναζητούν κείμενο μέσα σε εικόνες και να μετατρέπουν το κείμενο σε επεξεργάσιμη μορφή. Η δυνατότητα OCR είναι ενσωματωμένη στο Google Drive και είναι διαθέσιμη σε όλους τους χρήστες.

Το Google Drive χρησιμοποιήθηκε σε αυτό το Capstone Project ως διαδικτυακός χώρος προσωρινής αποθήκευσης εικόνων στις οποίες θα εφαρμοστεί η τεχνολογία OCR της Google.

4.3.6 Αναγνώριση ομιλίας

Στην επιστήμη της πληροφορικής, η αναγνώριση ομιλίας αναφέρεται στην μετατροπή προφερόμενων λέξεων σε κείμενο. Είναι επίσης αλλιώς γνωστή και ως «αυτόματη αναγνώριση ομιλίας», («υπολογιστική αναγνώριση ομιλίας» ή «speech-to-text» (STT), δηλαδή (μετατροπή) «από-ομιλία-σε-κείμενο».



Εικόνα 23

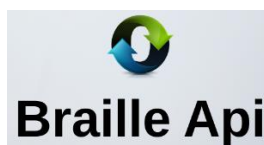
Κάποια συστήματα αναγνώρισης ομιλίας χρησιμοποιούν «αναγνώριση ομιλίας ανεξάρτητη από τον ομιλητή», ενώ άλλα χρησιμοποιούν «εξάσκηση», όπου ένα άτομο διαβάζει κομμάτια κειμένου σε ένα σύστημα αναγνώρισης ομιλίας. Τότε, αυτού του είδους τα συστήματα αναλύουν τη φωνή ενός ομιλητή και την χρησιμοποιούν για να προσαρμόσουν την αναγνώριση της ομιλίας του συγκεκριμένου ατόμου από τον υπολογιστή, με αποτέλεσμα την πιο ακριβή καταγραφή της. Συστήματα που δεν χρησιμοποιούν εξάσκηση ονομάζονται συστήματα «ανεξάρτητα από τον ομιλητή».

Οι εφαρμογές αναγνώρισης ομιλίας συμπεριλαμβάνουν τα Φωνητικά Περιβάλλοντα Χρήστη (Voice User Interfaces) όπως η φωνητική πληκτρολόγηση, ο έλεγχος των οικιακών ηλεκτρονικών συσκευών και συστημάτων, η διαδικτυακή αναζήτηση, η απλή τοποθέτηση δεδομένων, η προετοιμασία δομημένων εγγράφων, η επεξεργασία "από-ομιλία-σε-κείμενο"

Ο όρος αναγνώριση ομιλίας δεν αναφέρεται τόσο στο ποιος μιλάει, αλλά στο τι λέει. Αναγνωρίζοντας επιπλέον όμως και την ταυτότητα του ομιλητή είναι δυνατόν να διευκολυνθεί η διαδικασία της μετάφρασης της ομιλίας του σε συστήματα που έχουν προηγουμένως εξασκηθεί στην φωνή του συγκεκριμένου ατόμου ή ακόμη είναι δυνατόν να πιστοποιηθεί ή αναγνωριστεί η ταυτότητα του ομιλητή σε συστήματα ασφαλείας. (WikiPedia)

4.3.7 Fun Translations Braille API

Το FunTranslations Braille API είναι ένα εργαλείο που επιτρέπει στους προγραμματιστές να μεταφράζουν κείμενο σε χαρακτήρες Braille. Το FunTranslations Braille API παρέχει έναν απλό και εύκολο τρόπο στους προγραμματιστές να ενσωματώσουν τη λειτουργία μετάφρασης Braille στις εφαρμογές τους. Το API επιτρέπει στους προγραμματιστές να υποβάλλουν αιτήματα HTTP στον διακομιστή FunTranslations και να λαμβάνουν μια απάντηση που περιέχει το μεταφρασμένο κείμενο σε χαρακτήρες Braille.



Εικόνα 24

Το Braille API έχει σχεδιαστεί για να είναι εύκολο στη χρήση και υποστηρίζει ένα ευρύ φάσμα γλωσσών εισαγωγής. Η τεκμηρίωση του API παρέχει λεπτομερείς πληροφορίες σχετικά με τον τρόπο χρήσης του API, συμπεριλαμβανομένων παραδειγμάτων για τον τρόπο υποβολής αιτημάτων και τον τρόπο χειρισμού των δεδομένων απόκρισης.

Συνολικά, το FunTranslations Braille API παρέχει ένα ισχυρό και εύχρηστο εργαλείο για προγραμματιστές που θέλουν να προσθέσουν λειτουργικότητα μετάφρασης Braille στις εφαρμογές τους. Με το ευρύ φάσμα χαρακτηριστικών και την ευκολία χρήσης του, το Braille API έχει τη δυνατότητα να δημιουργήσει θετικό αντίκτυπο στη ζωή των ατόμων με προβλήματα όρασης και να συμβάλλει στην πρόωθηση του γραμματισμού Braille.

4.3.8 Η οικογένεια μικροελεγκτών Arduino

Οι μικροελεγκτές της οικογένειας Arduino είναι μια πλατφόρμα ανοιχτού κώδικα που χρησιμοποιείται για την κατασκευή ηλεκτρονικών συσκευών και έργων. Αποτελείται τόσο από φυσικό υλικό, όπως μια πλακέτα μικροελεγκτή, όσο και από ένα περιβάλλον ανάπτυξης

λογισμικού που εκτελείται σε υπολογιστή, το οποίο χρησιμοποιείται για τον προγραμματισμό του μικροελεγκτή για τον έλεγχο διαφόρων συσκευών και αισθητήρων.



Εικόνα 25

Οι πλακέτες Arduino διατίθενται σε διάφορα σχήματα και μεγέθη και μπορούν να προγραμματιστούν για να ελέγχουν ένα ευρύ φάσμα ηλεκτρονικών ειδών, από απλά φώτα LED έως πολύπλοκα ρομποτικά συστήματα. Πρόκειται για υλικό χαμηλού κόστους που παρουσιάζει ευκολία στην χρήση του και χρησιμοποιείται για την υλοποίηση ηλεκτρονικών έργων, αυτοματισμών και πρωτότυπων κατασκευών. Το περιβάλλον ανάπτυξης λογισμικού Arduino είναι εύκολο στη χρήση και υπάρχει μια μεγάλη κοινότητα χρηστών που μοιράζονται τα έργα και την τεχνογνωσία τους, καθιστώντας το προσβάσιμο σε οποιονδήποτε ενδιαφέρεται για τα ηλεκτρονικά και τον προγραμματισμό. Το συγκεκριμένο υλικό συνοδεύεται από πλήθος εγχειριδίων, βιβλιοθηκών ανοιχτού κώδικα και εφαρμογών που έχουν αναπτυχθεί από την κοινότητα χρηστών.

Η ευελιξία και η ευκολία χρήσης του Arduino έχουν ως αποτέλεσμα να είναι μια δημοφιλής πλατφόρμα που χρησιμοποιείται από αρχάριους μέχρι και επαγγελματίες, για την δημιουργία πρωτοτύπων και συσκευών proof-of-concept στη βιομηχανία και την έρευνα.

Όλα τα παραπάνω καθιστούν το Arduino ιδανική επιλογή για την υλοποίηση της κατασκευής προσομοίωσης της οπτικής αναπαράστασης του κώδικα Braille, όπως αυτή ορίστηκε από την ανάλυση του συστήματος.

Συγκεκριμένα στην υλοποίηση του Capstone Project για το οποίο γίνεται λόγος στο παρόν κείμενο, χρησιμοποιήθηκε ο μικροελεγκτής Arduino Uno ο οποίος είναι μια πλακέτα ανοιχτού κώδικα που βασίζεται στον μικροελεγκτή ATmega328P. Είναι μια από τις πιο δημοφιλείς πλακέτες στην οικογένεια μικροελεγκτών Arduino και χρησιμοποιείται ευρέως για μια ποικιλία έργων στους τομείς της ηλεκτρονικής, της ρομποτικής και του αυτοματισμού.

Η πλακέτα Arduino Uno έχει μια ποικιλία από ψηφιακές και αναλογικές ακίδες εισόδου/εξόδου και μπορεί να προγραμματιστεί χρησιμοποιώντας το ολοκληρωμένο περιβάλλον ανάπτυξης Arduino (IDE). Η πλακέτα μπορεί να τροφοδοτηθεί χρησιμοποιώντας ένα καλώδιο USB ή ένα εξωτερικό τροφοδοτικό και μπορεί να επικοινωνεί με άλλες συσκευές χρησιμοποιώντας μια ποικιλία πρωτοκόλλων επικοινωνίας όπως σειριακά, I2C και SPI. Η πλακέτα είναι επίσης συμβατή με ένα ευρύ φάσμα από πλακέτες άλλων κυκλωμάτων που μπορούν να συνδεθούν στο Arduino Uno για να προσθέσουν πρόσθετη λειτουργικότητα.

4.3.9 Arduino IDE

Το Arduino IDE είναι ένα ολοκληρωμένο περιβάλλον ανάπτυξης σεναρίων προγραμματισμού σε Arduino. Αποτελεί μια εφαρμογή λογισμικού ανοιχτού κώδικα που χρησιμοποιείται για τη σύνταξη, τη μεταγλώττιση και τη μεταφόρτωση κώδικα σε πλακέτες Arduino.



Εικόνα 26

Το περιβάλλον παρέχει ένα πρόγραμμα επεξεργασίας κειμένου για τη σύνταξη κώδικα, μια περιοχή μηνυμάτων που εμφανίζει σχόλια σχετικά με τον κώδικα και τη διαδικασία αποστολής, μια σειριακή οθόνη για την επικοινωνία με την πλακέτα και μια γραμμή εργαλείων με μενού και κουμπιά για λειτουργίες που χρησιμοποιούνται συνήθως. Έχει σχεδιαστεί για να απλοποιεί τη διαδικασία ανάπτυξης σεναρίων τόσο από αρχάριους όσο και από προχωρημένους χρήστες,

παρέχοντας μια εύχρηστη διεπαφή που τους επιτρέπει να δημιουργούν και να ανεβάζουν κώδικα σε πλακέτες Arduino. Το λογισμικό υποστηρίζει μια ευρεία γκάμα από πλακέτες Arduino, από τις πιο βασικές έως τις πιο προηγμένες. Το Arduino IDE είναι επίσης επεκτάσιμο μέσω προσθέτων και βιβλιοθηκών, επιτρέποντας στους προγραμματιστές να προσθέσουν νέες δυνατότητες και λειτουργίες στο λογισμικό.

Το περιβάλλον είναι διαθέσιμο για Windows, Mac OS X και Linux και υποστηρίζει πολλές διαφορετικές γλώσσες προγραμματισμού, συμπεριλαμβανομένων των C και C++. Είναι ένα δημοφιλές και ισχυρό εργαλείο τόσο για αρχάριους όσο και για επαγγελματίες χρήστες που τους επιτρέπει να αναπτύσσουν και να ανεβάζουν γρήγορα και εύκολα τον κώδικα στις πλακέτες Arduino με τις οποίες εργάζονται.

4.3.10 Η διασύνδεση Bluetooth

Το Bluetooth είναι μια ασύρματη τεχνολογία χαμηλής κατανάλωσης και κόστους που επιτρέπει στις συσκευές να επικοινωνούν μεταξύ τους σε μικρές αποστάσεις. Η τεχνολογία αυτή αναπτύχθηκε για να απλοποιήσει την επικοινωνία μεταξύ συσκευών και να εξαλείψει την ανάγκη για καλώδια. Χρησιμοποιείται σε ένα ευρύ φάσμα συσκευών, όπως smartphone, φορητούς υπολογιστές, tablet, smartwatches, ακουστικά, ηχεία και άλλα. Με το Bluetooth, οι συσκευές μπορούν να συνδέονται μεταξύ τους και να μεταφέρουν δεδομένα ασύρματα, καθιστώντας το έναν βολικό και αποτελεσματικό τρόπο μεταφοράς πληροφοριών.



Εικόνα 27

Η τεχνολογία Bluetooth χρησιμοποιεί ραδιοκύματα για την επικοινωνία μεταξύ συσκευών. Η χρήση του βασίζεται σε μια τεχνική που ονομάζεται αναπήδηση συχνότητας, όπου η συσκευή Bluetooth αλλάζει τη συχνότητα που χρησιμοποιεί σε τακτά χρονικά διαστήματα. Αυτό βοηθά στην αποφυγή παρεμβολών από άλλες συσκευές και διασφαλίζει ότι το σήμα Bluetooth παραμένει ισχυρό και σταθερό. Το Bluetooth έχει εξελιχθεί με τα χρόνια και περιλαμβάνει πολλές εκδόσεις, με την πιο πρόσφατη έκδοση, το Bluetooth 5.2, να προσφέρει βελτιωμένη ταχύτητα, εμβέλεια και αξιοπιστία σε σύγκριση με προηγούμενες εκδόσεις. Το Bluetooth αποτελεί ουσιαστικό μέρος της σύγχρονης τεχνολογίας και έχει διαδραματίσει βασικό ρόλο στην ανάπτυξη του Διαδικτύου των Πραγμάτων (IoT) και άλλων αναδυόμενων τεχνολογιών.

Για τις ανάγκες του Capstone Project χρησιμοποιήθηκε η μονάδα Bluetooth HC-05 η οποία είναι μια ευρέως χρησιμοποιούμενη μονάδα Bluetooth σε ηλεκτρονικά έργα. Η μονάδα HC-05 είναι μια συσκευή που χρησιμοποιεί πρωτόκολλο σειριακής θύρας (SPP), η οποία επιτρέπει την ασύρματη επικοινωνία μεταξύ ηλεκτρονικών συσκευών. Λειτουργεί στη ζώνη συχνοτήτων 2,4 GHz και έχει εμβέλεια έως και 10 μέτρα σε ανοιχτό χώρο.

Επιπλέον μπορεί να χρησιμοποιήσει και το πρωτόκολλο επικοινωνίας master slave, καθιστώντας την ευέλικτη σε διάφορες εφαρμογές. Διαθέτει τέσσερις ακίδες VCC, γείωση, TX και RX και μπορεί να ελεγχθεί μέσω μικροελεγκτή ή υπολογιστή με μονάδα Bluetooth αποτελώντας ένα χρήσιμο στοιχείο ασύρματης επικοινωνίας σε ηλεκτρονικές εφαρμογές.

5 Παρουσίαση της εφαρμογής

5.1 Η αρχική οθόνη

Η αρχική οθόνη της εφαρμογής, όπως φαίνεται και στην εικόνα 7, αποτελείται από τρία κουμπιά με τη βοήθεια των οποίων πραγματοποιείται η πλοήγησή του χρήστη στις άλλες οθόνες της εφαρμογής. Καθένα από τα κουμπιά περιέχει κείμενο το οποίο υποδεικνύει στον χρήστη την λειτουργία που το συγκεκριμένο κουμπί θα επιτελέσει σε περίπτωση που πατηθεί. Οι λειτουργίες των εν λόγω κουμπιών είναι οι εξής:



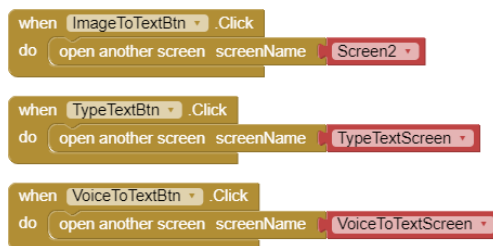
Εικόνα 28

Το πάτημα του κουμπιού με κείμενο «Image To Text» έχει ως αποτέλεσμα να κατευθύνει τον χρήστη σε μια νέα οθόνη στην οποία του παρέχεται η δυνατότητα να φωτογραφίσει με την κάμερα της συσκευής του μια εικόνα κειμένου. Στη συνέχεια, με την χρήση της τεχνολογίας OCR, το κείμενο αυτό μπορεί να επεξεργαστεί από την εφαρμογή και να μετατραπεί σε κώδικα Braille. Αναλυτικότερη περιγραφή της διαδικασίας αυτής γίνεται παρακάτω όταν θα παρουσιαστούν τα συστατικά και οι λειτουργίες της συγκεκριμένης οθόνης.

Με το πάτημα του κουμπιού «Type Text» ο χρήστης πλοηγείται σε οθόνη όπου του παρέχει την δυνατότητα να πληκτρολογήσει ένα κείμενο και στην συνέχεια να δώσει εντολή στην εφαρμογή να μετατρέψει το κείμενο αυτό σε κώδικα Braille.

Τέλος πατώντας το κουμπί που περιέχει το κείμενο «Voice To Text» ο χρήστης της εφαρμογής οδηγείται σε οθόνη η οποία περιέχει διεπαφή που του επιτρέπει να εκφωνήσει λέξεις ή ακόμα και ολόκληρες προτάσεις. Οι προτάσεις αυτές μέσω της λειτουργίας αναγνώρισης ομιλίας που παρέχεται από την κινητή συσκευή, μετατρέπονται σε κείμενο το οποίο κατόπιν, με εντολή του χρήστη μπορεί να μετατραπεί από την εφαρμογή σε γραφή Braille.

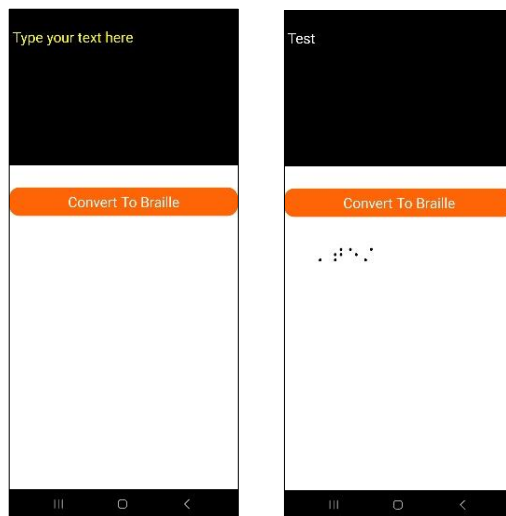
Στην επόμενη εικόνα φαίνεται ο κώδικας, σε μορφή Block στο περιβάλλον App Inventor, που περιέχει τις εντολές με τις οποίες υλοποιούνται οι ενέργειες που προαναφέρθηκαν.



Εικόνα 29

5.2 Η οθόνη TypeTextScreen

Στην εικόνα 9 φαίνεται το γραφικό περιβάλλον της διεπαφής του χρήστη, με την οθόνη TypeTextScreen στην οποία ο χρήστης οδηγείται με το πάτημα του κουμπιού με κείμενο «Type Text» που βρίσκεται στην αρχική οθόνη της εφαρμογής.



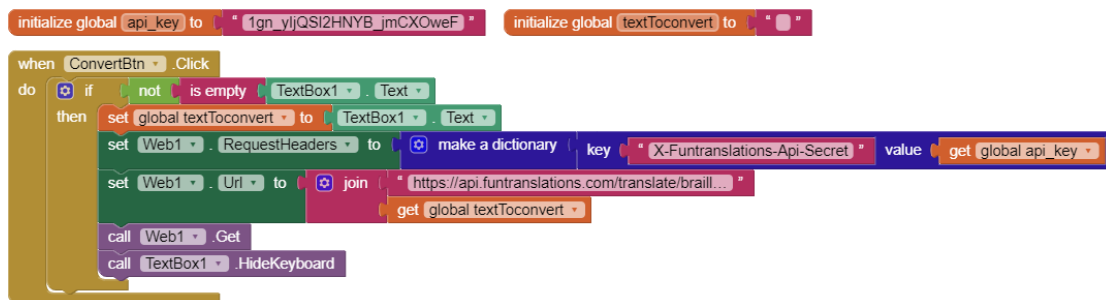
Εικόνα 30

Αρχικά υπάρχει ένα συστατικό πλαίσιου κειμένου (TextBox) στο οποίο ο χρήστης έχει την δυνατότητα να πληκτρολογήσει ένα κείμενο. Το κείμενο αυτό μπορεί στη συνέχεια να μετατραπεί σε κώδικα Braille με το πάτημα του κουμπιού που βρίσκεται ακριβώς κάτω από το πλαίσιο κειμένου και στο οποίο αναγράφεται η λέξη «Convert», ώστε να πληροφορεί τον χρήστη για την λειτουργία που επιτελεί. Ο κώδικας Braille στον οποίο μεταφράστηκε το εισαγόμενο κείμενο εμφανίζεται σε συστατικό ετικέτας (Label) το οποίο βρίσκεται ακριβώς κάτω από το κουμπί με κείμενο «Convert». Στην ετικέτα έχει ενεργοποιηθεί η ιδιότητα HTML Format ώστε να υπάρχει η δυνατότητα αποκωδικοποίησης εντολών της γλώσσας HTML. Η ετικέτα περιέχεται μέσα σε μια κατακόρυφη διάταξη με ράβδο κύλισης (VerticalScrollArrangement) ώστε σε περίπτωση που ο κώδικας Braille καταλαμβάνει έκταση μεγαλύτερη από το υπολειπόμενο ύψος της οθόνης, ο χρήστης να έχει την δυνατότητα να κυλίσει προς τα κάτω το τμήμα της οθόνης που τον περιέχει και να δει ολόκληρο το παραγόμενο αποτέλεσμα.

Η μετατροπή του κειμένου, που πληκτρολογεί ο χρήστης, σε κώδικα Braille γίνεται μέσω ενός API σε μορφή JSON με ονομασία Braille API Braille translator API το οποίο παρέχεται από τον ισότοπο <https://funtranslations.com/api/braille> κατόπιν εγγραφής ώστε να υπάρχει η δυνατότητα λήψης κλειδιού (API Key) το οποίο θα χρησιμοποιηθεί για την αυθεντικοποίηση και την πρόσβαση στο API.

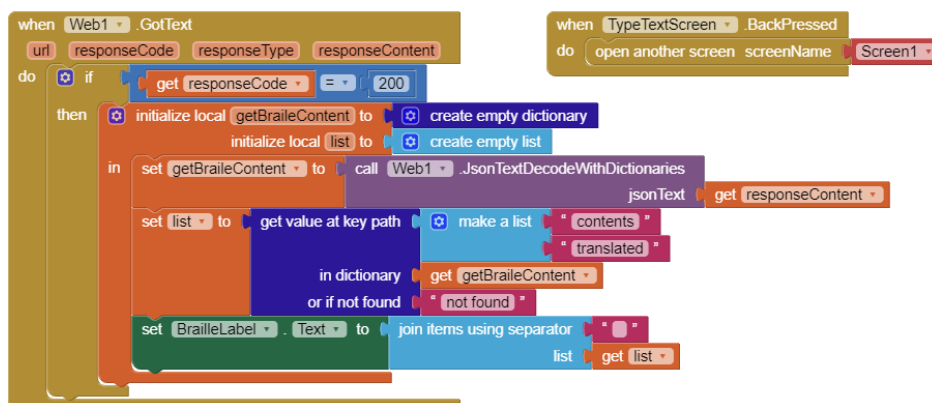
Το προς μετάφραση κείμενο αποστέλλεται στο API με GET Request, μαζί με κλήση Header Request με την οποία αποστέλλεται το κλειδί του API με το οποίο θα γίνει η αυθεντικοποίηση. Στην συνέχεια το API μεταφράζει το εισαγόμενο κείμενο σε κώδικα Braille και στέλνει (Response) το αποτέλεσμα της μετάφρασης στην εφαρμογή με τη μορφή JSON. Η συμβολοσειρά JSON περιέχει το αποτέλεσμα της μετάφρασης σε μια λίστα που κάθε στοιχείο της είναι ένας χαρακτήρας της γραφής Braille κωδικοποιημένος σε γλώσσα HTML. Η εφαρμογή αναλύει το αρχείο JSON και προβάλλει το αποτέλεσμα στην ετικέτα η οποία λόγω της ιδιότητας HTML Format που της ανατέθηκε εμφανίζει το αποτέλεσμα της μετάφρασης σε κουκίδες Braille.

Οι εντολές με τις οποίες το κείμενο που πληκτρολογεί ο χρήστης αποστέλλεται στο API φαίνονται στη επόμενη εικόνα.



Εικόνα 31

Αρχικά ορίζονται δυο καθολικές (global) μεταβλητές με όνομα `api_key` και `textToConvert`. Η μεταβλητή `api_key` χρησιμοποιείται για να αποθηκευτεί η συμβολοσειρά που περιέχει το κλειδί με το οποίο πραγματοποιείται η αυθεντικοποίηση και η επικοινωνία με το API. Η μεταβλητή `textToConvert` χρησιμοποιείται για να αποθηκευτεί το κείμενο που θα πληκτρολογήσει ο χρήστης και το οποίο στην συνέχεια θα σταλεί στο API για να μεταφραστεί σε κώδικα Braille. Η μεταβλητή αυτή έχει αρχικοποιηθεί στην κενή συμβολοσειρά.



Εικόνα 32

Το μπλοκ εντολών `when ConvertBtn.Click` είναι μια μέθοδος χειρισμού συμβάντος που εκτελείται όταν ο χρήστης πατήσει με το δάχτυλο του το κουμπί `ConvertBtn`, τότε ελέγχεται αν το πλαίσιο κειμένου δεν είναι κενό, δηλαδή αν ο χρήστης έχει πληκτρολογήσει έστω και ένα γράμμα. Σε περίπτωση που αυτό συμβαίνει, ορίζεται η τιμή της μεταβλητής `textToConvert` να είναι ίση με το κείμενο που περιέχεται στο πλαίσιο κειμένου. Στη συνέχεια αποστέλλεται αίτημα HTTP GET στο API που περιέχει ως παραμέτρους τις τιμές των μεταβλητών `api_key` και `textToConvert`, δηλαδή το κλειδί επικοινωνίας με το API και το κείμενο που θα μετατραπεί σε κώδικα Braille. Η τελευταία εντολή αυτού του μπλοκ υπάρχει για να κλείσει το πληκτρολόγιο της συσκευής το οποίο δεν χρειάζεται πλέον να είναι εμφανές.

Η επεξεργασία του αρχείου JSON που στέλνει το API στην εφαρμογή και το οποίο περιέχει την μετάφραση του κειμένου σε κώδικα Braille πραγματοποιείται από το μπλοκ εντολών `when Web1.GotText`. Το μπλοκ αυτό είναι στην ουσία η μέθοδος χειρισμού του συμβάντος `GotText`, δηλαδή της λήψης του κειμένου, και περιέχει τέσσερις παραμέτρους με ονομασίες, `url`, `responseCode`, `responseType` και `responseContent`. Η πρώτη εντολή της μεθόδου ελέγχει αν η τιμή της παραμέτρου `responseCode` είναι ίση με 200, το οποίο σημαίνει ότι η επικοινωνία με τον διακομιστή του API διεξήχθη χωρίς προβλήματα. Αν συμβαίνει αυτό, τότε αρχικοποιούνται δυο τοπικές μεταβλητές με ονομασίες `getBrailleContent` και `list`, τύπου λεξικού και λίστας αντίστοιχα. Οι μεταβλητές αυτές είναι βοηθητικές για την ανάλυση της συμβολοσειράς JSON που αποστέλλεται στην εφαρμογή και την εξαγωγή των χαρακτήρων του κώδικα Braille που περιέχεται σε αυτήν. Η συμβολοσειρά JSON είναι αποθηκευμένη στην παράμετρο `responseContent` της μεθόδου και η ανάλυση της πραγματοποιείται με την αποκωδικοποίηση της σε μια δομή δεδομένων τύπου λεξικού η οποία γίνεται με την έτοιμη ενσωματωμένη μέθοδο που παρέχει το συστατικό Web του App Inventor και έχει όνομα `JsonDecodeWithDictionaries`.

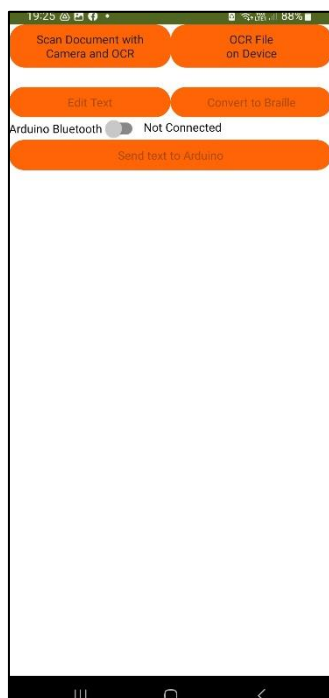
Το λεξικό το οποίο προκύπτει από την αποκωδικοποίηση της συμβολοσειράς JSON που περιέχει την μετάφραση, αποθηκεύεται στην μεταβλητή `getBrailleContent`. Στην συνέχεια η κάθε μια λέξη του κειμένου Braille αποθηκεύεται στην μεταβλητή `list` όπου με την χρήση της εντολής `join items using separator` όλες οι λέξεις συνενώνονται παρεμβάλλοντας ανάμεσα τους ένα κενό διάστημα για να μπορούν να ξεχωρίζουν και στην συνέχεια η παραγόμενη συμβολοσειρά ορίζεται ως το κείμενο της ετικέτας `BrailleLabel`. Αυτό έχει ως αποτέλεσμα να εμφανιστούν σε εκείνο το μέρος της οθόνης η κουκίδες του κώδικα Braille που αντιστοιχούν σε κάθε γράμμα του μεταφρασμένου κειμένου.

Το μπλοκ εντολών `When TypeTextScreen BackPressed` περιέχει τον κώδικα με τον οποίο η εφαρμογή οδηγεί τον χρήστη στην αρχική της οθόνη σε περίπτωση που αυτός πιάσει το κουμπί της προς τα πίσω πλοήγησης που υπάρχει στην συσκευή του.

5.3 Η οθόνη Screen2

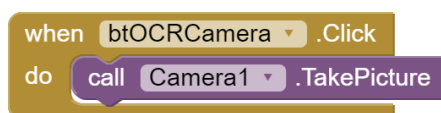
5.3.1 Λήψη εικόνας

Στην επόμενη εικόνα παρουσιάζεται το γραφικό περιβάλλον της διεπαφής του χρήστη, με την οθόνη Screen2.



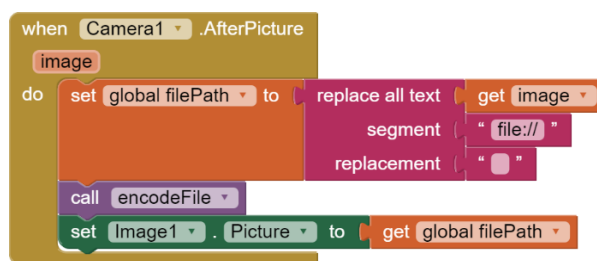
Εικόνα 33

Στο πάνω αριστερό άκρο της οθόνης υπάρχει ένα κουμπί το οποίο περιέχει το κείμενο «Scan Document with Camera and OCR». Το κουμπί αυτό δίνει στον χρήστη την δυνατότητα να τραβήξει με την κάμερα της συσκευής του μια φωτογραφία και στην συνέχεια να εξαχθεί στην εφαρμογή οποιοδήποτε κείμενο τυχόν περιέχεται στην εικόνα. Με το πάτημα αυτού του κουμπιού ανοίγει η κάμερα της κινητής συσκευής ώστε να είναι δυνατόν να πραγματοποιηθεί η λήψη εικόνας. Μόλις πραγματοποιηθεί με επιτυχία η λήψη η εικόνα παρουσιάζεται στην εφαρμογή και ξεκινάει αυτόματα η διαδικασία της οπτικής αναγνώρισης χαρακτήρων ενώ ταυτόχρονα εμφανίζεται ένα μήνυμα το οποίο ενημερώνει τον χρήστη ότι πρέπει να περιμένει μέχρι να τελειώσει η διαδικασία. Με την ολοκλήρωση της διαδικασίας το κείμενο που περιέχεται στην εικόνα εμφανίζεται στην εφαρμογή.



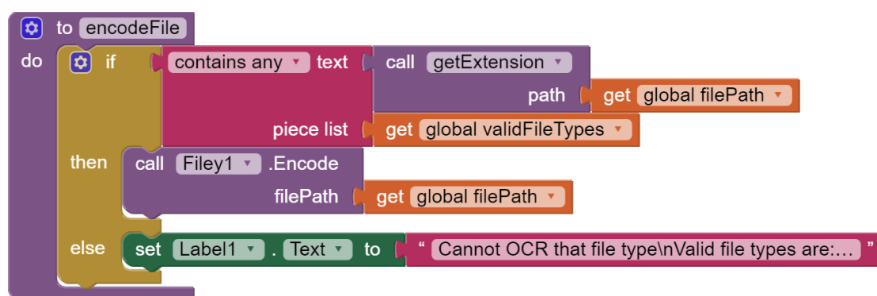
Εικόνα 34

Στην συνέχεια παρουσιάζονται και αναλύονται τα τμήματα κώδικα με τα οποία υλοποιείται η προαναφερόμενη λειτουργία. Με τις εντολές που φαίνονται στην προηγούμενη εικόνα ανοίγει η κάμερα της συσκευής μόλις πατηθεί το κουμπί που περιέχει το κείμενο «Scan Document with Camera and OCR» το οποίο ονομάστηκε btOCRCamera.



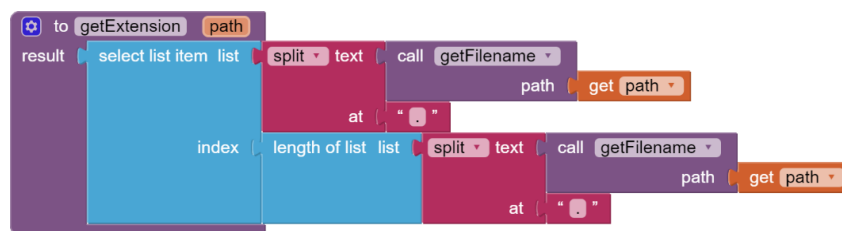
Εικόνα 35

Στην προηγούμενη εικόνα φαίνονται οι εντολές με τις οποίες η εφαρμογή διαχειρίζεται το συμβάν της λήψης φωτογραφίας από την κάμερα της συσκευής. Ο χειρισμός του συμβάντος πραγματοποιείται με την μέθοδο Camera1.AfterPicture που περιέχει την παράμετρο image τύπου συμβολοσειράς στην οποία είναι αποθηκευμένο το μονοπάτι στο σύστημα αρχείων της συσκευής το οποίο οδηγεί στην ληφθείσα φωτογραφία. Η τιμή της παραμέτρου image εκχωρείται στην καθολική μεταβλητή filePath και στην συνέχεια εκτελείται η μέθοδος encodeFile όπου μέσω αυτής κωδικοποιείται στην μορφή Base64 η φωτογραφία που τράβηξε ο χρήστης. Η τελευταία εντολή αυτού του μπλοκ εμφανίζει την εν λόγω φωτογραφία στην εφαρμογή.



Εικόνα 36

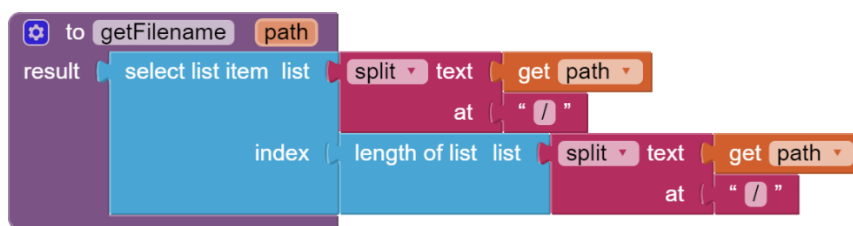
Η λειτουργία της μεθόδου encodeFile παρουσιάζεται στην προηγούμενη εικόνα. Η μέθοδος ελέγχει αν η επέκταση του ονόματος του αρχείου προς κωδικοποίηση είναι κάποια από της αποδεκτές μορφές (jpg, jpeg, png, gif, pdf). Ο έλεγχος γίνεται με την κλήση της μεθόδου getExtension η οποία περιγράφεται στην συνέχεια. Αν ο έλεγχος είναι επιτυχής, τότε το αρχείο κωδικοποιείται στην μορφή Base64. Σε διαφορετική περίπτωση εμφανίζεται μήνυμα στην εφαρμογή το οποίο αναφέρει ότι δεν μπορεί να πραγματοποιηθεί η οπτική αναγνώριση χαρακτήρων. Η κωδικοποίηση του αρχείου στην μορφή Base64 επιτυγχάνεται με την κλήση της μεθόδου File1.Encode. Η μέθοδος αυτή δεν υπάρχει στις ενσωματωμένες εντολές του περιβάλλοντος MIT App Inventor αλλά μπορεί να εισαχθεί με την λήψη από το διαδίκτυο της επέκτασης Filey. Η λήψη της επέκτασης μπορεί να πραγματοποιηθεί από την ηλεκτρονική διεύθυνση: https://drive.google.com/file/d/1Oow_o7Am65Cu9Q3NOVLFEw5E9CqcfVT6/view



Εικόνα 37

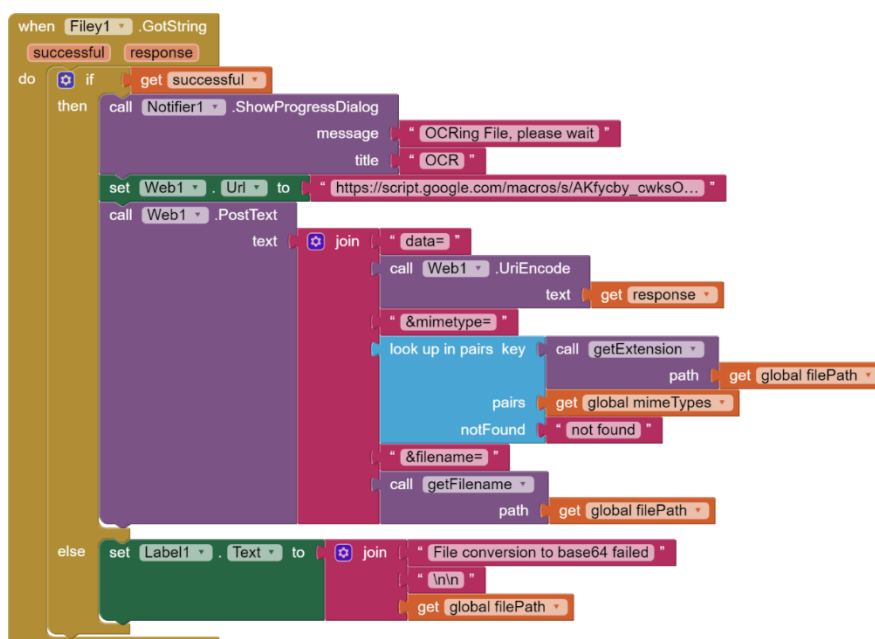
Στην προηγούμενη εικόνα φαίνεται το μπλοκ εντολών της μεθόδου getExtension η οποία δέχεται ως όρισμα την παράμετρο path και την μεταβιβάζει στην μέθοδο getFilename, ώστε να

μπορέσει να εντοπίσει με επιτυχία το όνομα του αρχείου του οποίου επιστρέφει την επέκταση. Το μπλοκ εντολών της μεθόδου `getFilename` απεικονίζεται στην επόμενη εικόνα.



Εικόνα 38

Όπως προαναφέρθηκε η μέθοδος `File1.Encode` της επέκτασης `Filey` κωδικοποιεί ένα αρχείο με τον αλγόριθμο `Base64`. Μετά την κωδικοποίηση του αρχείου ενεργοποιείται η μέθοδος `Filey1.GotString` η οποία διαχειρίζεται το συμβάν της κωδικοποίησης. Η μέθοδος αυτή περιέχει δύο παραμέτρους με ονομασίες `successful` και `response`. Η παράμετρος `successful` είναι τύπου `Boolean` και χρησιμεύει για να ελεγχθεί αν πραγματοποιήθηκε με επιτυχία η κωδικοποίηση, ενώ η παράμετρος `response` είναι τύπου συμβολοσειράς και περιέχει το αποτέλεσμα της κωδικοποίησης. Το μπλοκ της μεθόδου `Filey1.GotString` ξεκινάει ελέγχοντας αν πραγματοποιήθηκε επιτυχώς η κωδικοποίηση. Στην περίπτωση που αυτό συμβαίνει εμφανίζεται μήνυμα μέσω του συστατικού `Notifier` ότι έχει ξεκινήσει η διαδικασία της οπτικής αναγνώρισης χαρακτήρων στην φωτογραφία που λήφθηκε από τον χρήστη. Στη συνέχεια αποστέλλεται αίτημα `HTTP POST` σε σενάριο της γλώσσας προγραμματισμού `JavaScript`, το οποίο υλοποιήθηκε στο περιβάλλον `Google Apps Script` και εκτελείται στους διακομιστές της `Google`. Το σενάριο αυτό περιέχει τις εντολές με τις οποίες θα πραγματοποιηθεί η οπτική αναγνώριση χαρακτήρων με την χρήση της τεχνολογίας `Google OCR`. Το αίτημα `HTTP POST` υποβάλλεται με παραμέτρους την κωδικοποιημένη σε `Base64` συμβολοσειρά της φωτογραφίας που τράβηξε ο χρήστης, το όνομα και την επέκταση του ονόματος της εικόνας αυτής. Οι παράμετροι αυτοί φέρουν τις ονομασίες `data`, `filename` και `mimetype` αντίστοιχα. Σε περίπτωση που η κωδικοποίηση της εικόνας σε `Base64` δεν πραγματοποιήθηκε επιτυχώς, τότε εμφανίζεται μήνυμα στην εφαρμογή που πληροφορεί το χρήστη για το γεγονός αυτό. Το μπλοκ εντολών της μεθόδου `Filey1.GotString` οι οποίες πραγματοποιούν τις προηγούμενες λειτουργίες εμφανίζεται στην επόμενη εικόνα.



Εικόνα 39

5.3.2 Οπτική αναγνώριση χαρακτήρων

Οι εντολές του σεναρίου JavaScript με τις οποίες πραγματοποιείται η οπτική αναγνώριση χαρακτήρων είναι οι εξής:

```
1. // requires base64 encoded file, the file's mimetype, and a filename
2.
3. function doPost(e) {
4.   var data = Utilities.base64Decode(e.parameters.data);
5.   var blob = Utilities.newBlob(data, e.parameters.mimetype, e.parameters.filename);
6.
7.   //provide here a folder ID for the creation of the image file
8.   var fileID = DriveApp.getFolderById('189gJ1TU-hEMUXBQKkDx6ymSQvmA26fV8').createFile(blob).getId();
9.
10.  // 1tnKU8P20MDUt4UNq14kjRgSdjYq8beKh 189gJ1TU-hEMUXBQKkDx6ymSQvmA26fV8
11.  if (fileID !== "") {
12.    try {
13.
14.      // Fetch the image from drive
15.      var imageBlob = DriveApp.getFileById(fileID).getBlob();
16.      var resource = { title: e.parameters.filename, mimeType: imageBlob.getContentType() };
17.
18.      // OCR on .jpg, .png, .gif, (or .pdf uploads)
19.      var options = { ocr: true };
20.      var docFile = Drive.Files.insert(resource, imageBlob, options);
21.      var doc = DocumentApp.openById(docFile.id);
22.
23.      // Extract the text body of the Google Document
24.      var text = doc.getBody().getText().replace("\n", "");
25.
26.      // Send the documents to trash
27.      Drive.Files.remove(docFile.id);
28.      Drive.Files.remove(fileID);
29.      status = text;
30.    } catch (error) {
31.      status = "ERROR with OCR: " + error.toString();
32.    }
33.  } else {
34.    status = "ERROR with OCR: No image specified";
35.  }
36.
37.  return ContentService.createTextOutput(status);
38.
39. }
40.
```

Όπως προαναφέρθηκε το αίτημα HTTP POST υποβάλλεται στο διακομιστή με παραμέτρους data, mimetype και filename που αντιστοιχούν στην κωδικοποιημένη συμβολοσειρά Base64, τον τύπο και το όνομα του αρχείου. Το σενάριο εντολών JavaScript που χειρίζεται το αίτημα αρχίζει με την εντολή `var data = Utilities.base64Decode(e.parameters.data)` (γραμμή 4) η οποία κάνει χρήση της μεθόδου `base64Decode` από την κλάση `Utilities`. Η μέθοδος αυτή δέχεται ως παράμετρο την κωδικοποιημένη στην μορφή base64 συμβολοσειρά και την αποκωδικοποιεί στην αρχική της δυαδική μορφή, δηλαδή σε μια συμβολοσειρά που αποτελείται από τους χαρακτήρες 0 και 1. Το αποτέλεσμα της αποκωδικοποίησης αποθηκεύεται στην μεταβλητή `data`. Στην συνέχεια καλείται η μέθοδος `newBlob` της κλάσης `Utilities` που παίρνει παραμέτρους, την μεταβλητή `data`, τον τύπο του αρχείου (παράμετρος `mimetype` του αιτήματος HTTP POST) και το όνομα του αρχείου (παράμετρος `filename` του αιτήματος HTTP POST). Η μέθοδος αυτή δημιουργεί ένα νέο αντικείμενο BLOB, το οποίο είναι ουσιαστικά ένα κοντέινερ για δυαδικά δεδομένα που αντιπροσωπεύουν την εικόνα. Το επιστρεφόμενο αντικείμενο BLOB αποθηκεύεται στη μεταβλητή `blob`, για να μπορεί να χρησιμοποιηθεί αργότερα στον κώδικα.

Η εντολή της γραμμής 8 δημιουργεί ένα αρχείο στο GoogleDrive. Η μέθοδος `getFolderById` της κλάσης `DriveAPP` χρησιμοποιείται για την ανάκτηση ενός συγκεκριμένου φακέλου στο Google Drive ο οποίος έχει ως αναγνωριστικό του την συμβολοσειρά που δίνεται ως παράμετρος στην μέθοδο. Στη συνέχεια, η μέθοδος `createFile` καλείται για να δημιουργήσει το αρχείο

στον εν λόγω φάκελο με τα δεδομένα που δέχεται από την μεταβλητή blob την οποία παίρνει ως παράμετρο. Η μέθοδος `getId` ανακτά το αναγνωριστικό του αρχείου που μόλις δημιουργήθηκε. Το αναγνωριστικό αυτό είναι μια συμβολοσειρά η οποία αποθηκεύεται στη μεταβλητή `fileID`.

Στην γραμμή 15 με την εντολή `var imageBlob = DriveApp.getFileById(fileID).getBlob()` πραγματοποιείται ανάκτηση του αρχείου εικόνας από το Google Drive χρησιμοποιώντας την μέθοδο `getFileById` της κλάσης `DriveApp`. Η μέθοδος αυτή δέχεται ως παράμετρο την μεταβλητή `fileID` η οποία είναι το αναγνωριστικό του αρχείου. Η μέθοδος `getBlob()` χρησιμοποιείται για τη μετατροπή του ανακτημένου αρχείου σε ένα δυαδικό αντικείμενο (BLOB). Η δυαδική αυτή εικόνα αποθηκεύεται στην μεταβλητή `imageBlob`.

Η εντολή `resource = { title: e.parameters.filename, mimeType: imageBlob.getContentType() };` που υπάρχει στην γραμμή 16 του κώδικα δημιουργεί ένα αντικείμενο που ονομάζεται `resource` το οποίο περιέχει πληροφορίες σχετικά με τη δυαδική εικόνα που είναι αποθηκευμένη στη μεταβλητή `imageBlob`. Η ιδιότητα `title` του αντικειμένου `resource` ορίζεται ως η τιμή της παραμέτρου `filename` που μεταβιβάζεται στη συνάρτηση μέσω του αντικειμένου `e`, ενώ η ιδιότητα `mimeType` ορίζεται να είναι ο τύπος του δυαδικού αρχείου, ο οποίος λαμβάνεται χρησιμοποιώντας τη μέθοδο `getContentType()` του αντικειμένου `imageBlob`. Το αντικείμενο `resource` χρησιμοποιείται στα επόμενα βήματα του κώδικα για την εισαγωγή της δυαδικής εικόνας σε ένα νέο έγγραφο Google.

Με την εντολή `var options = { ocr: true };` της γραμμής 19 ορίζεται η μεταβλητή `options` η οποία καθορίζει τις επιλογές που θα χρησιμοποιηθούν κατά την εισαγωγή του δυαδικού αντικειμένου στο έγγραφο Google. Η μόνη επιλογή που ορίζεται σε αυτήν την περίπτωση είναι η `ocr: true`, η οποία υποδεικνύει στο Google Drive να εκτελέσει οπτική αναγνώριση χαρακτήρων (OCR) στο αρχείο εικόνας κατά τη δημιουργία του εγγράφου Google. Αυτό σημαίνει ότι το κείμενο από την εικόνα θα εξαχθεί αυτόματα και θα συμπεριληφθεί στο έγγραφο Google ως επεξεργάσιμο κείμενο, το οποίο στη συνέχεια μπορεί να υποβληθεί σε επεξεργασία στα επόμενα βήματα του κώδικα.

Στην γραμμή 20 καλείται η μέθοδος `Drive.Files.insert` με παραμέτρους τις μεταβλητές `resource`, `imageBlob` και `options`, η οποία επιστρέφει ένα αντικείμενο που αντιπροσωπεύει ένα έγγραφο Google που δημιουργείται με την χρήση της τεχνολογίας OCR. Το αντικείμενο αυτό αποθηκεύεται στην μεταβλητή `docFile`.

Η εντολή που υπάρχει στη γραμμή 21 ανοίγει το έγγραφο Google που δημιουργήθηκε από την δυαδική εικόνα, χρησιμοποιώντας τη μέθοδο `DocumentApp.openById(docFile.id)` μεταβιβάζοντας της το αναγνωριστικό του εγγράφου, το οποίο είναι αποθηκευμένο στην ιδιότητα `docFile.id`. Το αντικείμενο εγγράφου που προκύπτει αποθηκεύεται στη μεταβλητή `doc`, η οποία μπορεί να χρησιμοποιηθεί για πρόσβαση και χειρισμό των περιεχομένων του εγγράφου. Στην επόμενη γραμμή του κώδικα το κείμενο του εγγράφου εξάγεται χρησιμοποιώντας την εντολή `doc.getBody().getText().replace("\n", "")` και αποθηκεύεται στη μεταβλητή `text`. Η μέθοδος `getBody()` επιστρέφει το σώμα του εγγράφου και η μέθοδος `getText()` επιστρέφει το περιεχόμενο κειμένου του σώματος. Η μέθοδος `replace("\n", "")` χρησιμοποιείται για την αφαίρεση αλλαγών γραμμής από το κείμενο.

Στις γραμμές 27 και 28 η μέθοδος `Drive.Files.remove` καλείται δύο φορές για να διαγράψει τόσο το αρχείο εικόνας όσο και το έγγραφο Google από το Google Drive.

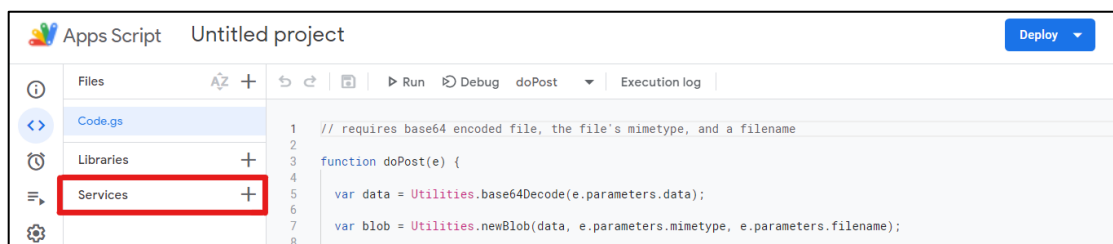
Το τμήμα κώδικα των γραμμών 29 έως 35 ορίζει την τιμή της μεταβλητής `status` με βάση το αποτέλεσμα της διαδικασίας OCR ως εξής:

- Εάν η διαδικασία OCR ήταν επιτυχής, η μεταβλητή `status` ορίζεται να είναι το εξαγόμενο κείμενο, το οποίο έχει αποθηκευτεί στη μεταβλητή `text`.

- Εάν υπήρξε σφάλμα κατά τη διαδικασία OCR, η μεταβλητή status ορίζεται σε ένα μήνυμα σφάλματος, ακολουθούμενο από την εξαίρεση που παρουσιάστηκε η οποία δημιουργείται χρησιμοποιώντας τη μέθοδο toString() του αντικειμένου error.
- Εάν δεν καθορίστηκε εικόνα, η μεταβλητή κατάστασης ορίζεται σε ένα μήνυμα σφάλματος που αναφέρει το γεγονός αυτό.

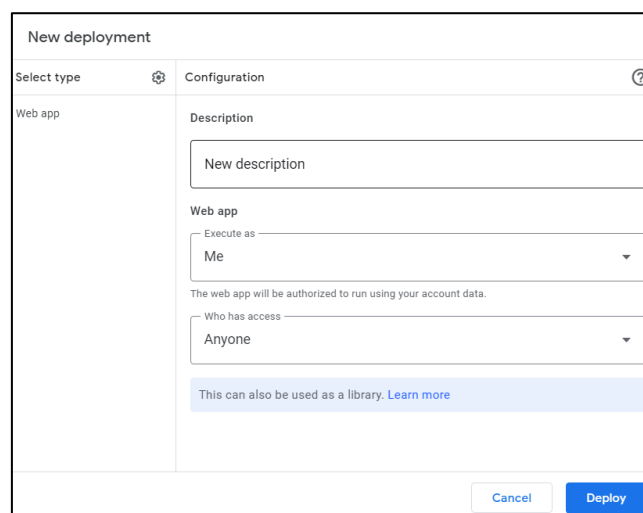
Τέλος στην γραμμή 37 του κώδικα η εντολή `return ContentService.createTextOutput(status);` επιστρέφει την τιμή της μεταβλητής status ως απόκριση στο αίτημα HTTP POST που ενεργοποίησε την εκτέλεση της συνάρτησης `doPost`.

Για να μπορέσει να εκτελεστεί επιτυχώς ο κώδικας του προηγούμενου σεναρίου και να επιτευχθεί η επικοινωνία με την εφαρμογή απαιτείται να γίνουν κάποιες επιπλέον ενέργειες στην πλατφόρμα Google Apps Script και στο Google Drive. Μια από τις ενέργειες αυτές είναι να προστεθεί από τα Services η υπηρεσία (Service) που ονομάζεται Drive και η οποία επιτρέπει να πραγματοποιηθεί η επικοινωνία με το Google Drive, για να είναι δυνατή η παροχή υπηρεσιών που έχουν να κάνουν με μεταφόρτωση, λήψη και αναζήτηση αρχείων.



Εικόνα 40

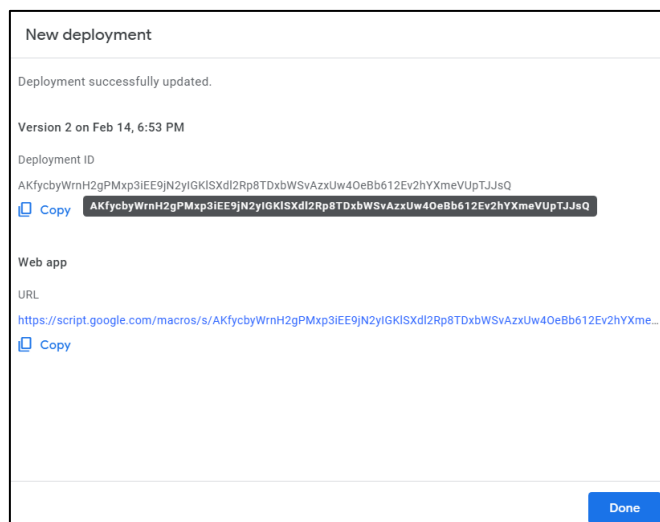
Επιπλέον είναι πολύ σημαντικό να γίνει Deploy στο σενάριο διότι η διαδικασία αυτή του επιτρέπει να επικοινωνεί και να ενσωματωθεί με άλλα συστήματα και εφαρμογές. Έτσι θα είναι δυνατή και η επικοινωνία του με την εφαρμογή. Η ενέργεια αυτή μπορεί να πραγματοποιηθεί από την επιλογή New deployment η οποία θα εμφανιστεί όταν πατηθεί το κουμπί με κείμενο Deploy που φαίνεται στην προηγούμενη εικόνα. Στο αριστερό μέρος του παραθύρου που θα ανοίξει αναγράφεται η φράση Select type, η οποία καθορίζει τον τύπο του σεναρίου, εκεί πρέπει να δοθεί η επιλογή Web app. Στο δεξί μέρος του παραθύρου στο πτυσσόμενο μενού με κείμενο «Execute as» πρέπει να δοθεί η επιλογή Me ενώ σε εκείνο που περιέχει το κείμενο «Who has access» επιλέγεται το Anyone. Για να γίνει αποθήκευση των επιλογών αυτών πρέπει να πατηθεί το κουμπί Deploy.



Εικόνα 41

Με το πάτημα του κουμπιού Deploy ανοίγει το επόμενο παράθυρο το οποίο δίνει την δυνατότητα αντιγραφής της διεύθυνσης URL στην οποία βρίσκεται το σενάριο. Η διεύθυνση αυτή

μπορεί στην συνέχεια να χρησιμοποιηθεί στο μπλοκ εντολών του περιβάλλοντος MIT App Inventor το οποίο διαχειρίζεται την επικοινωνία της εφαρμογής με το σενάριο.



Εικόνα 42

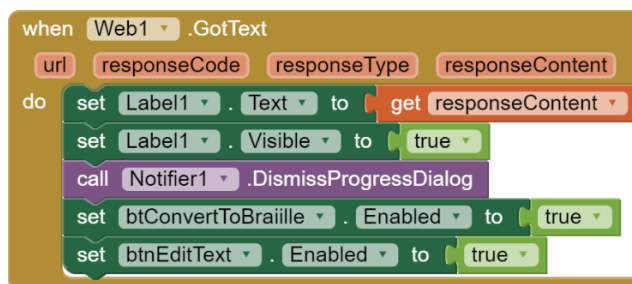
Επίσης, είναι απαραίτητο να δοθεί το αναγνωριστικό του φακέλου στο Google Drive στον οποίο αποθηκεύονται τα αρχεία που επεξεργάζεται το σενάριο. Ο φάκελος αυτός μπορεί να υπάρχει ήδη ή να δημιουργηθεί ένας νέος. Σε κάθε περίπτωση το αναγνωριστικό είναι δυνατόν να γίνει γνωστό από την διεύθυνση URL του φακέλου η οποία αναγράφεται στην γραμμή διευθύνσεων ενός οποιουδήποτε φυλλομετρητή ιστού και είναι το τμήμα της διεύθυνσης που υποδεικνύεται στην επόμενη εικόνα. Το αναγνωριστικό αυτό είναι απαραίτητο να δοθεί ως παράμετρος τύπου συμβολοσειράς την μέθοδο `getFolderById` όπως φαίνεται και στην γραμμή 8 του κώδικα που υλοποιεί το σενάριο.

drive.google.com/drive/u/0/folders/189gJ1TU-hEMUXBQKkDx6ymSQvmA26fV8

Εικόνα 43

5.3.3 Η απόκριση του διακομιστή

Αφού εκτελεστεί το σενάριο, και πραγματοποιηθεί η οπτική αναγνώριση των χαρακτήρων της εικόνας, η τιμή της μεταβλητής status που περιέχει το αποτέλεσμα επιστρέφεται ως απόκριση στο αίτημα HTTP POST. Η απόκριση αυτή χειρίζεται από την εφαρμογή η οποία κάνει λήψη του εξαγόμενου κειμένου από την εικόνα ώστε να είναι σε θέση να το επεξεργαστεί περαιτέρω και να το μετατρέψει σε κώδικα Braille. Το μπλοκ εντολών στο περιβάλλον MIT App Inventor με το οποίο γίνεται ο χειρισμός της απόκρισης φαίνεται στην εικόνα που ακολουθεί.



Εικόνα 44

Η πρώτη εντολή του μπλοκ ορίζει το κείμενο της ετικέτας Label1 να είναι το περιεχόμενο της απόκρισης του αιτήματος HTTP POST, δηλαδή το αποτέλεσμα που προέκυψε από την οπτική αναγνώριση των χαρακτήρων που περιέχονται στην φωτογραφία και το οποίο ήταν αποθηκευμένο στη μεταβλητή status του σεναρίου. Στην συνέχεια ορίζεται η επιλογή Visible για την

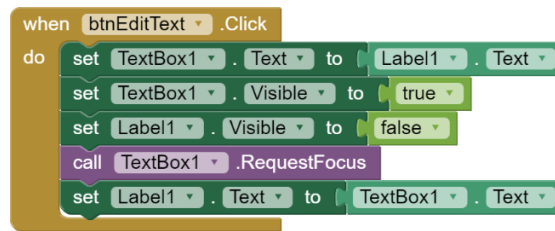
ετικέτα Label1 να έχει την τιμή true, έτσι ώστε η ετικέτα, επομένως και το κείμενο που αυτή περιέχει, να γίνουν ορατά στην οθόνη της συσκευής. Με τον τρόπο αυτό παρουσιάζεται στον χρήστη το αποτέλεσμα της οπτικής αναγνώρισης των χαρακτήρων, δηλαδή το κείμενο που περιέχει η φωτογραφία που τράβηξε με την κάμερα της συσκευής. Η εντολή που ακολουθεί κλείνει το αναδυόμενο παράθυρο που είχε εμφανιστεί στην εφαρμογή το οποίο ενημέρωνε τον χρήστη ότι πραγματοποιείται η διαδικασία της οπτικής αναγνώρισης των χαρακτήρων της φωτογραφίας και τον προέτρεπε να περιμένει μέχρι την ολοκλήρωση της. Οι δύο τελευταίες εντολές του μπλοκ δίνουν την τιμή true στην ιδιότητα Enable των κουμπιών με ονομασίες btConvertToBraille και btnEditText, ώστε τα κουμπιά αυτά να ενεργοποιηθούν στο γραφικό περιβάλλον της διεπαφής του χρήστη με την εφαρμογή και να είναι πλέον διαθέσιμα σε αυτόν. Η μορφή την οποία θα έχει η οθόνη της εφαρμογής μετά την ολοκλήρωση της προηγούμενης διαδικασίας παρουσιάζεται στην εικόνα που ακολουθεί όπου φαίνεται ότι το κείμενο το οποίο περιέχεται στο λογότυπο του MIT App Inventor έχει εμφανιστεί στην εφαρμογή μετά την οπτική αναγνώριση χαρακτήρων η οποία πραγματοποιήθηκε στην εικόνα του λογοτύπου και επιπλέον έχουν ενεργοποιηθεί τα κουμπιά με κείμενο «Edit Text» και «Convert to Braille».



Εικόνα 45

Το κουμπί στο οποίο αναγράφεται το κείμενο «OCR File on Device» επιτρέπει στον χρήστη να επιλέξει ένα αρχείο εικόνας σε μορφή jpg, jpeg, png, και gif ή ένα αρχείο pdf που βρίσκεται στην συλλογή της συσκευής του και στην συνέχεια να πραγματοποιηθεί οπτική αναγνώριση χαρακτήρων στο αρχείο αυτό σύμφωνα με την διαδικασία που αναφέρθηκε προηγουμένως.

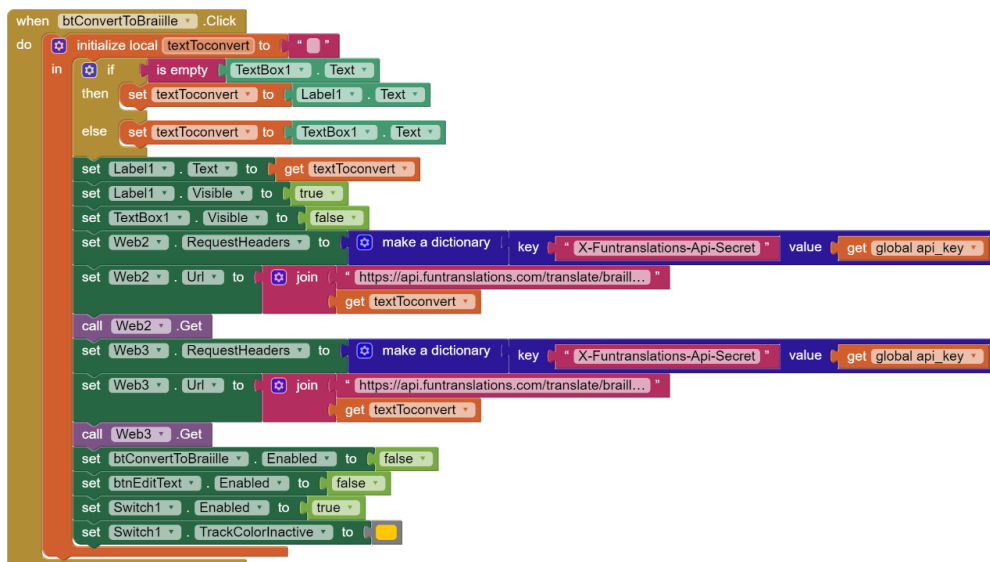
Το κουμπί btnEditText περιέχει το κείμενο «Edit Text» με το οποίο προτρέπει τον χρήστη της εφαρμογής να τροποποιήσει, σε περίπτωση που το επιθυμεί, το αποτέλεσμα της οπτικής αναγνώρισης των χαρακτήρων της φωτογραφίας που τράβηξε με την κάμερα της κινητής του συσκευής. Η διαδικασία αυτή είναι σε ορισμένες περιπτώσεις απαραίτητη διότι μπορεί να έχουν πραγματοποιηθεί λάθη στην οπτική αναγνώριση και το κείμενο που περιέχει φωτογραφία να εμφανιστεί αλλοιωμένο στην οθόνη. Το κουμπί αυτό πριν πραγματοποιηθεί η οπτική αναγνώριση των χαρακτήρων ήταν απενεργοποιημένο ενώ ενεργοποιείται με την ολοκλήρωση της. Το πάτημα του κουμπιού αυτού εμφανίζει ένα συστατικό πλαισίου κειμένου στο οποίο αναγράφεται το κείμενο που προέκυψε από την οπτική αναγνώριση χαρακτήρων, ώστε ο χρήστης να έχει την δυνατότητα να το τροποποιήσει αν επιθυμεί. Το μπλοκ των εντολών οι οποίες υλοποιούν την λειτουργία του κουμπιού btnEditText, παρουσιάζεται στην επόμενη εικόνα.



Εικόνα 46

5.3.4 Η μετατροπή σε κώδικα Braille

Το κουμπί btConvertToBraille περιέχει το κείμενο «Convert To Braille» με το οποίο προτρέπει τον χρήστη της εφαρμογής να μετατρέψει, αν επιθυμεί, σε κώδικα Braille το αποτέλεσμα που προέκυψε από την οπτική αναγνώριση των χαρακτήρων της φωτογραφίας που λήφθηκε από την κάμερα της συσκευής, είτε το κείμενο που προέκυψε από την τροποποίηση του αποτελέσματος αυτού από τον χρήστη. Η διαδικασία της μετάφρασης πραγματοποιείται με την αποστολή του κειμένου στο API με αίτημα HTTP GET και την διαβίβαση του αποτελέσματος της μετάφρασης από το API στην εφαρμογή σε μια συμβολοσειρά JSON η ανάλυση της οποίας θα προβάλλει σε κουκίδες Braille το αποτέλεσμα της μετάφρασης στην εφαρμογή. Ο τρόπος με τον οποίο υλοποιείται η επικοινωνία της εφαρμογής με το API αλλά και η διαδικασία ανάλυσης της συμβολοσειράς JSON είναι πανομοιότυπος με αυτόν που περιγράφηκε αναλυτικά σε προηγούμενη ενότητα. Το μπλοκ των εντολών με τις οποίες υλοποιείται η λειτουργία του κουμπιού btConvertToBraille, φαίνεται στην εικόνα που ακολουθεί.



Εικόνα 47

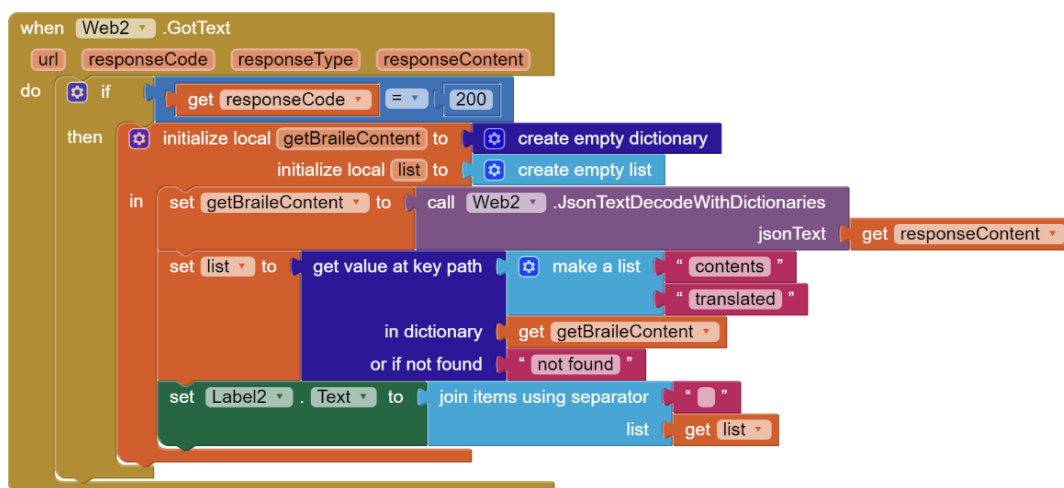
Το API με το οποίο πραγματοποιείται η μετάφραση δίνει στον προγραμματιστή διάφορες επιλογές με τις οποίες μπορεί να το καλέσει οι οποίες αντιστοιχούν σε διαφορετικές διευθύνσεις URL στις οποίες αποστέλλεται το αίτημα HTTP GET. Οι επιλογές αυτές, ανάλογα με την διεύθυνση κλήσης η οποία θα επιλεγεί, διαβιβάζουν η καθεμιά με τον δικό της διαφορετικό τρόπο το αποτέλεσμα της μετάφρασης του κώδικα Braille, πάντοτε όμως ενσωματωμένο σε μια συμβολοσειρά JSON. Το API δίνει την δυνατότητα επιστροφής του αποτελέσματος, σε εικόνες στις οποίες απεικονίζονται οι χαρακτήρες Braille, σε χαρακτήρες Unicode HTML Braille και σε αριθμούς, οι οποίοι υποδεικνύουν το νούμερο ή τα νούμερα των κουκκίδων κάθε χαρακτήρα, στο εξάστιγμο του κώδικα Braille. Αν ο προγραμματιστής επιθυμεί να λάβει το αποτέλεσμα της μετάφρασης σε χαρακτήρες Unicode HTML Braille θα πρέπει να καλέσει το API στην διεύθυνση: <https://api.funtranslations.com/translate/braille/html.json>, ενώ αν επιθυμεί η επιστροφή του αποτελέσματος να γίνει σε αριθμούς κουκκίδων Braille, τότε η κλήση του API γίνεται από την διεύθυνση: <https://api.funtranslations.com/translate/braille/dots.json>.

Το πάτημα του κουμπιού `btConvertToBraille` έχει ως αποτέλεσμα να αποσταλεί στο API το κείμενο που έχει εξαχθεί από την οπτική αναγνώριση χαρακτήρων της φωτογραφίας. Όπως φαίνεται στο μπλοκ κώδικα που υπάρχει στην προηγούμενη εικόνα το κουμπί πραγματοποιεί δυο διαφορετικές κλήσεις προς το API. Η πρώτη κλήση είναι προς την διεύθυνση URL η οποία επιστρέφει το αποτέλεσμα της μετάφρασης σε χαρακτήρες Unicode HTML Braille οι οποίοι μετά την ανάλυση της συμβολοσειράς JSON θα προβληθούν στην εφαρμογή, ώστε να είναι ορατό στον χρήστη το αποτέλεσμα της μετάφρασης σε κουκίδες Braille. Η δεύτερη κλήση είναι προς την διεύθυνση URL που επιστρέφει το αποτέλεσμα της μετάφρασης σε ακολουθίες αριθμών οι οποίοι αντιστοιχούν στις κουκίδες που αναπαριστούν κάθε χαρακτήρα Braille στο εξάστιγμο. Για παράδειγμα αν στην μετάφραση υπάρχει το αγγλικό γράμμα `i` το οποίο αναπαρίσταται σε κώδικα Braille από την δεύτερη και την τέταρτη κουκίδα του εξάστιγμου, τότε το API θα επιστρέψει το 24 ενώ για το γράμμα `a` το οποίο αντιστοιχεί στη πρώτη κουκίδα θα επιστραφεί η τιμή 1.

5.3.5 Ο προγραμματισμός της προσομοίωσης του κώδικα

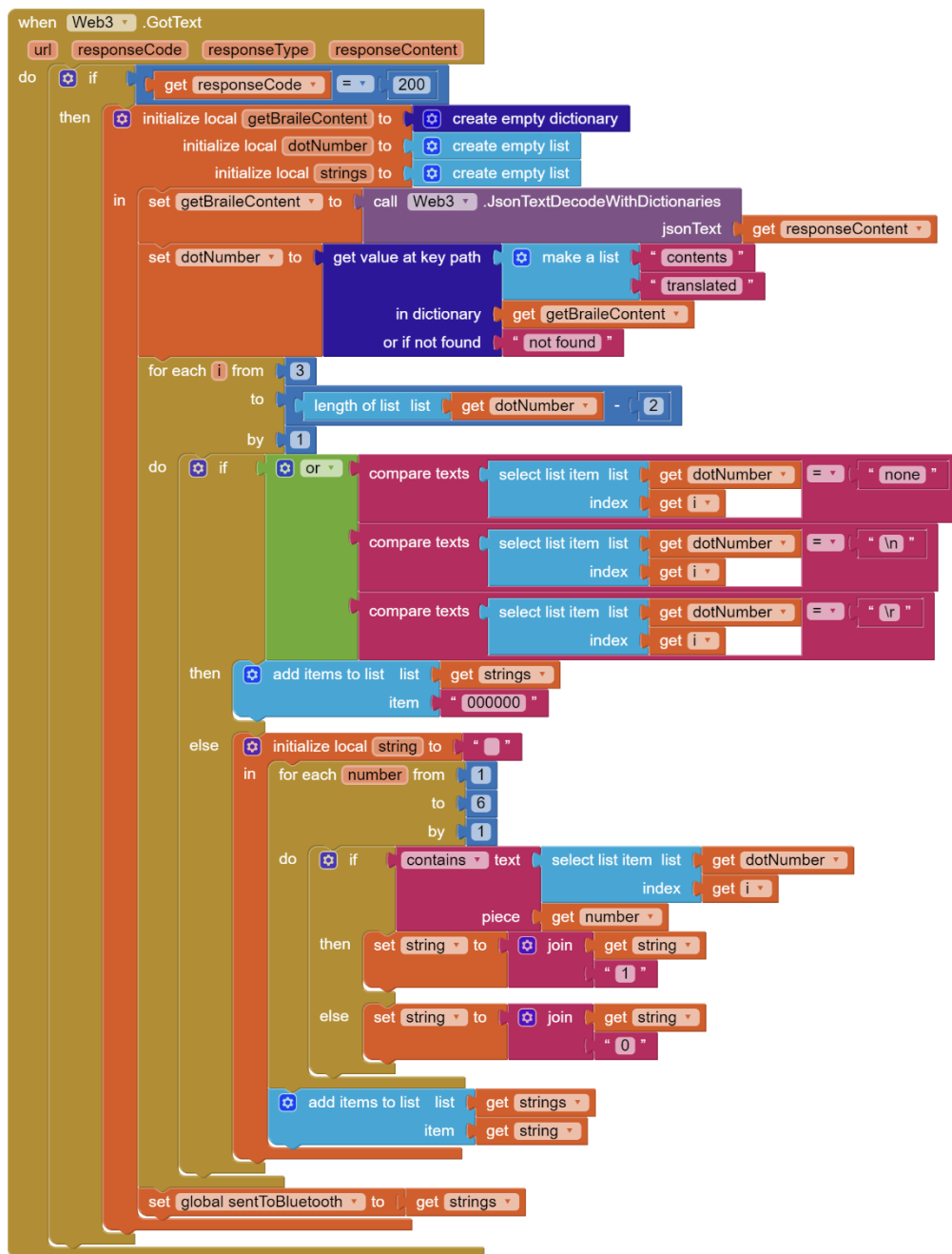
Η δεύτερη κλήση προς το API είναι απαραίτητη για να αναλυθεί το αποτέλεσμα που αυτό επιστρέφει σε συμβολοσειρές μήκους έξι χαρακτήρων οι οποίες θα αποτελούνται από τους αριθμούς 0 και 1. Στην συνέχεια κάθε μια από τις συμβολοσειρές αυτές θα αποσταλεί στον μικροελεγκτή Arduino μέσω της διασύνδεσης Bluetooth ώστε να προσομοιωθεί το αποτέλεσμα της μετάφρασης από έξι λυχνίες LED οι οποίες θα ανάβουν ή θα σβήνουν ανάλογα με τον χαρακτήρα που αναπαριστούν στο εξάστιγμο του κώδικα Braille. Για παράδειγμα στην προσομοίωση του γράμματος `i` θα ακτινοβολούν μόνο οι λυχνίες που αντιστοιχούν στις θέσεις 2 και 4 του εξάστιγμου ενώ εκείνες που αντιστοιχούν στις υπόλοιπες θέσεις θα είναι σβηστές.

Στην επόμενη εικόνα παρουσιάζεται το μπλοκ εντολών με το οποίο η εφαρμογή διαχειρίζεται τη συμβολοσειρά JSON που αποδίδει το αποτέλεσμα της μετάφρασης του κειμένου σε χαρακτήρες Unicode HTML Braille. Οι εντολές αυτές έχουν ως συνέπεια να προβληθεί το αποτέλεσμα της μετάφρασης σε κουκίδες Braille στην οθόνη της κινητής συσκευής. Η αναλυτική περιγραφή των εντολών που περιέχει αυτό το μπλοκ έχει πραγματοποιηθεί σε προηγούμενη ενότητα, καθώς η διαδικασία χειρισμού της απόκρισης από το API και η προβολή σε κουκίδες Braille του αποτελέσματος της μετάφρασης είναι πανομοιότυπη με αυτήν που πραγματοποιήθηκε στην οθόνη `TypeTextScreen`.



Εικόνα 48

Η εικόνα που ακολουθεί εμφανίζει το μπλοκ εντολών με το οποίο η εφαρμογή χειρίζεται την συμβολοσειρά JSON που επιστρέφει το API κατά την απόκριση του, και η οποία αποδίδει το αποτέλεσμα της μετάφρασης του κειμένου σε αριθμούς που υποδεικνύουν τις θέσεις των κουκίδων στο εξάστιγμο του κώδικα Braille που αντιστοιχούν σε κάθε γράμμα της μετάφρασης.



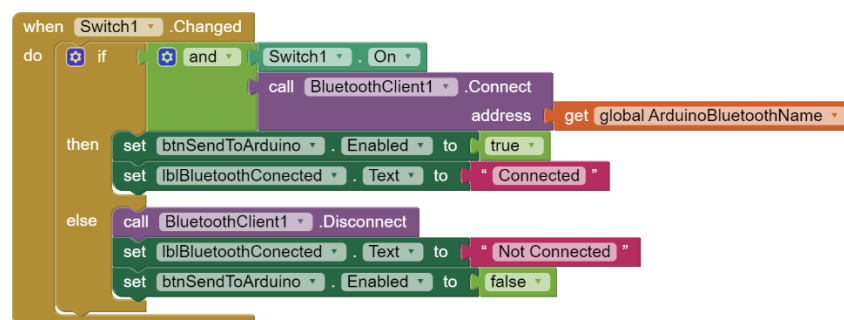
Εικόνα 49

Το μπλοκ ξεκινάει ελέγχοντας αν η τιμή της παραμέτρου responseCode είναι ίση με 200 πράγμα που σημαίνει ότι έχει πραγματοποιηθεί με επιτυχία η επικοινωνία με το διακομιστή του API. Στην συνέχεια, και με δεδομένο ότι η προηγούμενη συνθήκη είναι αληθής, αρχικοποιούνται τρεις τοπικές μεταβλητές η getBrailleContent τύπου λεξικού, η dotNumber και η strings τύπου λίστας. Η μεταβλητή getBrailleContent χρησιμοποιείται για να αποθηκεύσει σε ένα λεξικό την ανάλυση της συμβολοσειράς JSON που επιστρέφει το API κατά την απόκριση του. Το λεξικό που δημιουργήθηκε περιέχει στο εσωτερικό του μια λίστα. Τα δύο πρώτα στοιχεία της λίστας αυτής ορίζονται από το API να περιέχουν την αγγλική λέξη none. Κάθε άλλο στοιχείο, της λίστας από το τρίτο και μετά, είναι είτε μια συμβολοσειρά από αριθμούς οι οποίοι εκφράζουν την θέση των κουκίδων στο εξάστιγμο του κώδικα Braille για κάθε γράμμα του κειμένου που αποστάλθηκε στο API, είτε η λέξη none η οποία αντιστοιχεί στον χαρακτήρα του διαστήματος, είτε τα σύμβολα \n, \r τα οποία υποδεικνύουν την αλλαγή γραμμής. Η λίστα αυτή αποθηκεύεται στην μεταβλητή dotNumber.

Κάθε γράμμα του κώδικα Braille στον οποίο μεταφράζεται το κείμενο που στάλθηκε στο API θα προσομοιωθεί από έξι λυχνίες LED που θα έχουν ίδια διάταξη με τις κουκίδες στο εξάστιγμο και θα ανάβουν ή θα σβήνουν ανάλογα με το αν η αντίστοιχη κουκίδα χρησιμοποιείται στον χαρακτήρα Braille ή όχι. Για παράδειγμα το αγγλικό γράμμα i που αναπαρίσταται από τις κουκίδες 2 και 4, θα προσομοιωθεί ανάβοντας μόνο η δεύτερη και η τέταρτη λυχνία της αντίστοιχης διάταξης, ενώ οι υπόλοιπες θα παραμένουν σβηστές. Το άναμμα ή το σβήσιμο κάθε λυχνίας LED ελέγχεται από τον μικροελεγκτή Arduino. Για να πραγματοποιηθεί επιτυχώς η διαδικασία αυτή πρέπει ο κώδικας Braille στον οποίο μεταφράστηκε το κείμενο να αποσταλεί από την εφαρμογή στον Arduino μέσω Bluetooth σε μορφή που να είναι επεξεργάσιμη από αυτόν. Μια μορφή που ικανοποιεί αυτή την συνθήκη και συγχρόνως επιτρέπει τον εύκολο προγραμματισμό της προσομοίωσης είναι να αναπαρασταθεί κάθε χαρακτήρας του κώδικα Braille από μια συμβολοσειρά μήκους έξι θέσεων η οποία θα αποτελείται από τα ψηφία 1 και 0 ανάλογα με το αν η συγκεκριμένη κουκίδα χρησιμοποιείται για την αναπαράσταση του χαρακτήρα αυτού στο εξάστιγμο ή όχι. Για παράδειγμα το αγγλικό γράμμα i που αναπαρίσταται από τις κουκίδες 2 και 4 στο εξάστιγμο του κώδικα Braille θα μετατραπεί στην συμβολοσειρά 010100. Κάθε συμβολοσειρά μήκους έξι θα προσαρτηθεί στην λίστα strings, και στη συνέχεια κάθε ένα στοιχείο της λίστας αυτής θα αποσταλεί μέσω της διασύνδεσης Bluetooth από την εφαρμογή στον Arduino ώστε να πραγματοποιηθεί η προσομοίωση.

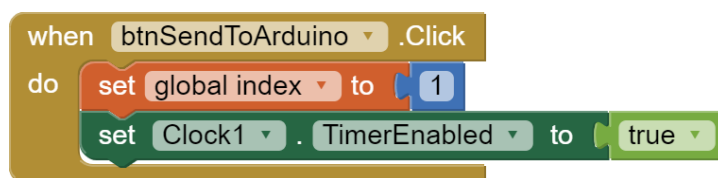
Οι εντολές του μπλοκ της εικόνας 28 που υπάρχουν μετά την αρχικοποίηση των τοπικών μεταβλητών χρησιμοποιούνται για να δημιουργηθεί η λίστα strings όπου κάθε στοιχείο της θα σταλεί στον Arduino. Μια δομή επανάληψης διατρέχει, από το τρίτο στοιχείο της μέχρι και δυο στοιχεία πριν το τέλος της, την λίστα dotNumber που περιέχει το αποτέλεσμα της μετάφρασης σε αριθμούς κουκίδων Braille. Αν το τρέχον στοιχείο είναι η λέξη none, ο χαρακτήρας \n ή ο χαρακτήρας \r, τότε προσαρτάται στην λίστα strings η συμβολοσειρά 000000. Διαφορετικά, αν το τρέχον στοιχείο της λίστας dotNumber δεν είναι κανένας από τους προαναφερόμενους χαρακτήρες, σημαίνει ότι αποτελείται από αριθμούς κουκίδων Braille. Στην περίπτωση αυτή αρχικοποιείται μια τοπική μεταβλητή τύπου συμβολοσειράς με ονομασία string. Στη συνέχεια μια νέα δομή επανάληψης ελέγχει αν κάθε αριθμός από το 1 μέχρι το 6 περιέχεται στο τρέχον στοιχείο της λίστας dotNumber και αν αυτό συμβαίνει, τίθεται στην αντίστοιχη θέση της συμβολοσειράς string ο αριθμός 1 αλλιώς τίθεται ο αριθμός 0. Μόλις ολοκληρωθεί η επανάληψη δημιουργείται μια συμβολοσειρά μήκους έξι χαρακτήρων όπου κάθε στοιχείο της είναι είτε ο αριθμός 0 είτε ο αριθμός 1. Με την ολοκλήρωση της δομής επανάληψης η συμβολοσειρά string, μήκους έξι που δημιουργήθηκε με την προηγούμενη διαδικασία προσαρτάται στο τέλος της λίστας strings. Η λίστα strings που δημιουργήθηκε από την εκτέλεση των προηγούμενων εντολών αποθηκεύεται στην καθολική (global) μεταβλητή με όνομα sentToBluetooth. Στη συνέχεια κάθε στοιχείο της θα αποσταλεί στον Arduino για να πραγματοποιηθεί η προσομοίωση του κώδικα Braille με λυχνίες LED.

Το γραφικό περιβάλλον της διεπαφής του χρήστη με την εφαρμογή που φαίνεται στην εικόνα 50 περιέχει ένα συστατικό τύπου διακόπτη (Switch), το οποίο χρησιμεύει για την ενεργοποίηση ή την απενεργοποίηση της σύζευξης Bluetooth μεταξύ της κινητής συσκευής και του Arduino. Το μπλοκ εντολών με το οποίο υλοποιείται η λειτουργία του διακόπτη παρουσιάζεται στην επόμενη εικόνα.



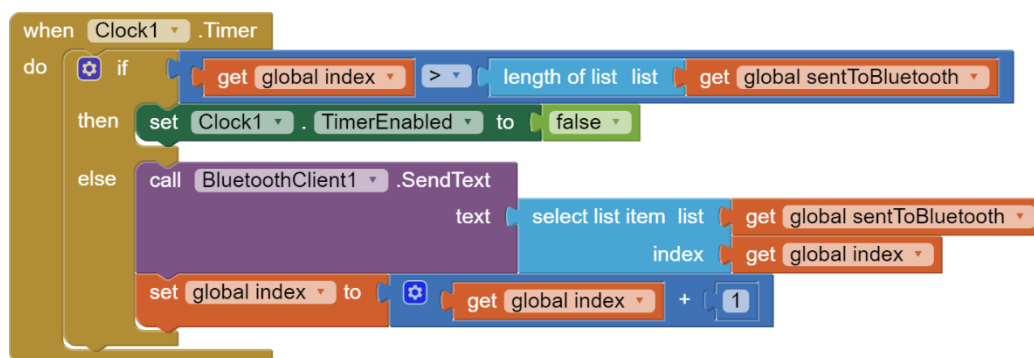
Εικόνα 50

Σύμφωνα με τις εντολές του μπλοκ εάν η κατάσταση του διακόπτη αλλάξει στη θέση «On» και έχει πραγματοποιηθεί η σύζευξη Bluetooth ανάμεσα στην κινητή συσκευή και τον Arduino τότε στην ετικέτα με ονομασία lblBluetoothConected αναγράφεται η λέξη «Connected» με την οποία πληροφορείται ο χρήστης ότι έχει επιτευχθεί η σύζευξη, ενώ ταυτόχρονα τίθεται η ιδιότητα Enabled του κουμπιού btnSendToArduino σε true, που σημαίνει ότι το κουμπί έχει γίνει διαθέσιμο στον χρήστη. Το πάτημα του κουμπιού αυτού έχει ως αποτέλεσμα να αποσταλεί στον Arduino κάθε συμβολοσειρά που είναι αποθηκευμένη στην μεταβλητή λίστας sentToBluetooth και η οποία θα χρησιμοποιηθεί για την προσομοίωση του κώδικα Braille από λυχνίες LED. Οι εντολές με τις οποίες υλοποιείται η λειτουργία του κουμπιού btnSendToArduino περιγράφονται παρακάτω. Εάν η κατάσταση του διακόπτη αλλάξει σε θέση διαφορετική από την «On» (θέση «Off»), τότε η σύζευξη Bluetooth μεταξύ συσκευής και Arduino διακόπτεται ενώ παράλληλα απενεργοποιείται το κουμπί btnSendToArduino και το κείμενο της ετικέτας lblBluetoothConected γίνεται «Not Connected» ώστε να πληροφορηθεί ο χρήστης ότι δεν έχει επιτευχθεί σύνδεση.



Εικόνα 51

Στην προηγούμενη εικόνα φαίνονται οι εντολές με τις οποίες υλοποιείται η λειτουργία του κουμπιού btnSendToArduino το πάτημα του οποίου έχει σαν αποτέλεσμα να ξεκινήσει η διαδικασία αποστολής στον Arduino των στοιχείων της λίστας sentToBluetooth που θα χρησιμοποιηθούν για την προσομοίωση. Αρχικά ορίζεται η τιμή της μεταβλητής index να είναι ίση με ένα και στην συνέχεια ενεργοποιείται το χρονόμετρο. Η μεταβλητή index χρησιμοποιείται ως δείκτης προς το στοιχείο της λίστας που πρόκειται να αποσταλεί στον Arduino. Η χρήση του χρονόμετρου είναι απαραίτητη για να υπάρχει μια μικρή καθυστέρηση, στην αποστολή των συμβολοσειρών της λίστας προς τον Arduino, ώστε αυτός να έχει την δυνατότητα να τις επεξεργαστεί όλες χωρίς να υπάρχει κίνδυνος υπερφόρτωσης της μνήμης του. Το χρονόμετρο έχει οριστεί έτσι ώστε η εφαρμογή να αποστέλλει στον Arduino ανά 2000 millisecond (δύο δευτερόλεπτα) καθεμία συμβολοσειρά που περιέχεται στη λίστα και η οποία θα χρησιμοποιηθεί για την προσομοίωση ενός χαρακτήρα του κώδικα Braille με λυχνίες LED.



Εικόνα 52

Η αποστολή της συμβολοσειράς στον Arduino πραγματοποιείται από το μπλοκ εντολών που χειρίζεται το συμβάν της ενεργοποίησης του χρονόμετρου και το οποίο παρουσιάζεται στην προηγούμενη εικόνα. Αρχικά ελέγχεται η συνθήκη αν η τιμή της μεταβλητής index είναι μεγαλύτερη από το μήκος της λίστας με ονομασία sentToBluetooth κάθε στοιχείο της οποίας θα αποσταλεί στον Arduino. Αν η συνθήκη αυτή είναι αληθής, τότε σημαίνει ότι έχει ολοκληρωθεί η αποστολή όλων των στοιχείων της λίστας, οπότε απενεργοποιείται το χρονόμετρο. Στην περίπτωση που η συνθήκη δεν ισχύει, δηλαδή η τιμή της μεταβλητής index είναι μικρότερη ή ίση από το μήκος της λίστας sentToBluetooth, τότε εκτελούνται κάθε δύο δευτερόλεπτα οι εντολές

που βρίσκονται στο τμήμα else της δομής if else. Η πρώτη από αυτές τις εντολές διαβιβάζει στον Arduino μέσω της διασύνδεσής Bluetooth την συμβολοσειρά μήκους 6 χαρακτήρων της λίστας sentToBluetooth. Η δεύτερη εντολή του τμήματος else αυξάνει την τιμή της μεταβλητής index κατά 1, ώστε να δείξει στο επόμενο στοιχείο που θα αποσταλεί. Η αύξηση της μεταβλητής index σταματάει όταν η τιμή της γίνει μεγαλύτερη από το μήκος της λίστας sentToBluetooth.

5.3.6 Ο προγραμματισμός της πλακέτας Arduino

Στην συνέχεια παρουσιάζεται το σενάριο εντολών σε γλώσσα προγραμματισμού C++ που υλοποιήθηκε στο περιβάλλον Arduino IDE με το οποίο προγραμματίζεται ο μικροελεγκτής Arduino UNO ώστε να πραγματοποιεί την προσομοίωση του κώδικα Braille από κύκλωμα με λυχνίες LED οι οποίες θα έχουν διάταξη αντίστοιχη των κουκίδων Braille στο εξάστιγμο και θα ανάβουν ή θα σβήνουν ανάλογα με το αν η αντίστοιχη κουκίδα χρησιμοποιείται ή όχι για την αναπαράσταση ενός συγκεκριμένου χαρακτήρα.

```
1. int braillePins[] = { 2, 4, 6, 8, 10, 12 };
2.
3. void setup() {
4.   Serial.begin(9600);
5.   for (int i=0; i < 6; i++) {
6.     pinMode(braillePins[i], OUTPUT);
7.     digitalWrite(braillePins[i], LOW);
8.   }
9. }
10.
11. void loop() {
12.   if (Serial.available()) {
13.     String str = Serial.readString();
14.     Serial.println(str);
15.     sendBinaryToArduino(str);
16.     delay(1000);
17.     offAll();
18.   }
19. }
20.
21. void offAll() {
22.   for (int i=0; i < 6; i++){
23.     digitalWrite(braillePins[i], LOW);
24.   }
25. }
26.
27. void sendBinaryToArduino(String binary){
28.   for ( int i = 0; i < binary.length(); i++ ) {
29.     char currentChar = binary.charAt(i);
30.     if ( currentChar == '1' ) {
31.       digitalWrite(braillePins[i], HIGH);
32.     } else if ( currentChar == '0' ) {
33.       digitalWrite(braillePins[i], LOW);
34.     }
35.   }
36. }
```

Ο σκοπός του σεναρίου είναι να λάβει τις δυαδικές συμβολοσειρές μήκους έξι χαρακτήρων που στέλνονται ανά δύο δευτερόλεπτα μέσω Bluetooth από την εφαρμογή και να τις μετατρέψει σε μοτίβο Braille. Ακολουθεί η ανάλυση του κώδικα:

Το σενάριο ξεκινάει με την δήλωση μιας μεταβλητής με ονομασία braillePins τύπου πίνακα ακεραίων στον οποίον αποθηκεύονται οι θέσεις των διόδων του μικροελεγκτή οι οποίες θα συνδέονται με καλώδια με τις λυχνίες LED.

Η μέθοδος setup (γραμμές 3 έως 9) ρυθμίζει την πλακέτα Arduino αρχικοποιώντας τη σειριακή επικοινωνία και διαμορφώνοντας τις διόδους ως ακροδέκτες εξόδου. Η μέθοδος loop είναι ο κύριος βρόχος του σεναρίου. Ελέγχει αν έχει γίνει η αποστολή μηνύματος από την διασύνδεση

Bluetooth, διαβάζει το μήνυμα αυτό, το τυπώνει στην οθόνη και το αποθηκεύει σε μια συμβολοσειρά με όνομα str. Στη συνέχεια, η συμβολοσειρά αυτή περνά ως όρισμα στη συνάρτηση sendBinaryToArduino η οποία την χρησιμοποιεί για να καθορίσει αν θα πρέπει να ανάψει ή να σβήσει καθεμία από τις έξι λυχνίες LED που προσδιορίζουν τις θέσεις των κουκκίδων του κώδικα Braille στο εξάστιγμο.

Η εντολή delay(1000) που υπάρχει στην γραμμή 16 του σεναρίου είναι μια ενσωματωμένη συνάρτηση στη γλώσσα προγραμματισμού Arduino, η οποία αναστέλλει την εκτέλεση του προγράμματος για καθορισμένο χρόνο σε χιλιοστά του δευτερολέπτου. Στην προκειμένη περίπτωση η delay(1000) προκαλεί παύση του προγράμματος για ένα δευτερόλεπτο (1000 χιλιοστά του δευτερολέπτου) πριν από την εκτέλεση της επόμενης γραμμής κώδικα. Εδώ είναι σημαντικό να τονιστεί ότι η τιμή που μπαίνει ως όρισμα στην συνάρτηση delay και η οποία καθορίζει το χρονικό διάστημα παύσης του σεναρίου, πρέπει να είναι μικρότερη από τον χρόνο που έχει καθοριστεί να αποστέλλει η εφαρμογή τις συμβολοσειρές μέσω Bluetooth στον Arduino. Στην προκειμένη περίπτωση ο χρόνος αυτός ανέρχεται στα δύο δευτερόλεπτα συνεπώς πληρείται η προηγούμενη απαίτηση. Η διαφορά αυτή ανάμεσα στα δύο χρονικά διαστήματα είναι απαραίτητη για να προλάβει ο μικροελεγκτής να διαβάσει και να επεξεργαστεί την τρέχουσα συμβολοσειρά πριν να έρθει η επόμενη, σε διαφορετική περίπτωση η προσομοίωση δεν θα πραγματοποιηθεί σωστά.

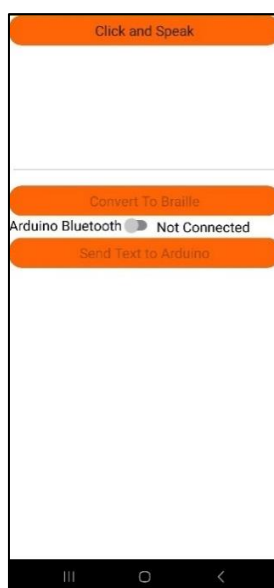
Στην τελευταία εντολή του βρόγχου loop (γραμμή 17) καλείται, η συνάρτηση offAll για να απενεργοποιήσει όλες τις διόδους του ελεγκτή και να σβήσει όλες λυχνίες LED είχαν ανάψει. Η συνάρτηση αυτή υλοποιείται στις γραμμές 21 έως και 25 του κώδικα. Στη συνέχεια, ο βρόχος loop ξεκινά ξανά και περιμένει για νέα είσοδο.

Στις γραμμές 27 έως και 36 του σεναρίου υλοποιείται η συνάρτηση sendBinaryToArduino η οποία δέχεται ως είσοδο μια δυαδική συμβολοσειρά μήκους έξι χαρακτήρων και προσομοιώνει την αναπαράσταση ενός γράμματος στο εξάστιγμο του κώδικα Braille. Αυτό πραγματοποιείται με μια δομή επανάληψης η οποία διατρέχει τη συμβολοσειρά χαρακτήρα προς χαρακτήρα και ορίζει την αντίστοιχη τιμή της διόδου που χειρίζεται την λυχνία LED είτε σε HIGH (αναμμένη) είτε σε LOW (σβησμένη), ανάλογα με το αν ο χαρακτήρας είναι 1 ή 0.

5.4 Η οθόνη Voice To Braille

5.4.1 Φωνητική πληκτρολόγηση

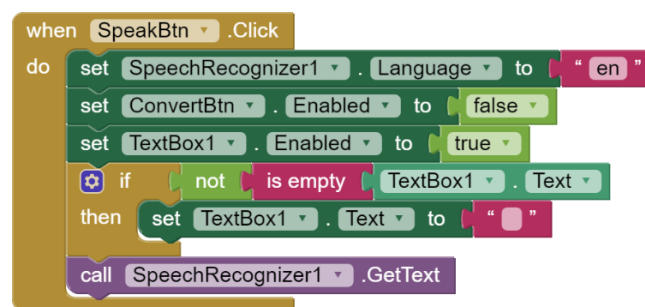
Στην επόμενη εικόνα παρουσιάζεται το γραφικό περιβάλλον της διεπαφής του χρήστη, με την οθόνη Voice To Braille.



Εικόνα 53

Η οθόνη αυτή δίνει την δυνατότητα στο χρήστη να εκφωνήσει λέξεις, φράσεις, η ακόμα και ολόκληρες προτάσεις οι οποίες στην συνέχεια με την χρήση της τεχνολογίας αναγνώρισης ομιλίας θα μετατραπούν σε κείμενο, το οποίο θα παρουσιαστεί στην εφαρμογή. Επιπλέον παρέχεται η δυνατότητα μετατροπής του κειμένου αυτού σε κώδικα Braille, κατόπιν ενέργειας του χρήστη, και το παραγόμενο αποτέλεσμα της μετατροπής να παρουσιαστεί στην οθόνη σε κουκίδες Braille. Μια ακόμα ενέργεια που μπορεί να πραγματοποιήσει ένας χρήστης ο οποίος χειρίζεται την οθόνη αυτή είναι η αποστολή του παραγόμενου κώδικα Braille στον μικροελεγκτή Arduino ώστε να πραγματοποιηθεί μια οπτική αναπαράσταση του κώδικα από λυχνίες LED.

Το πρώτο συστατικό της οθόνης αυτής είναι ένα κουμπί που περιέχει τη φράση «Click and Speak» η οποία προτρέπει τον χρήστη να το πατήσει και να αρχίσει να μιλάει. Το πάτημα του κουμπιού αυτού έχει ως συνέπεια να ξεκινήσει η διαδικασία αναγνώρισης ομιλίας η οποία παρέχεται από την συσκευή και να μετατραπεί η ομιλία σε κείμενο. Το μπλοκ εντολών στο περιβάλλον MIT App Inventor με τις οποίες υλοποιείται η διαδικασία αυτή παρουσιάζεται στην επόμενη εικόνα.



Εικόνα 54

Η πρώτη εντολή του μπλοκ ορίζει την γλώσσα στην οποία θα πραγματοποιείται η αναγνώριση της ομιλίας να είναι τα αγγλικά. Αυτό συμβαίνει διότι στην τρέχουσα έκδοση της εφαρμογής δεν υποστηρίζεται ακόμα η μετατροπή κειμένου σε κώδικα Braille από οποιαδήποτε άλλη γλώσσα εκτός της αγγλικής. Η αναγνώριση ομιλίας σε γλώσσα διαφορετική της αγγλικής θα είχε ως συνέπεια το κείμενο που θα παραχθεί να μην μπορεί να μετατραπεί σωστά σε κώδικα Braille.

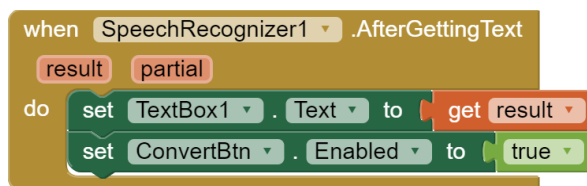
Η επόμενη εντολή ενεργοποιεί το κουμπί με ονομασία ConvertBtn το οποίο περιέχει τη φράση «Convert To Braille» και όπως φαίνεται στην εικόνα 53 αρχικά είναι απενεργοποιημένο. Το πάτημα του κουμπιού αυτού έχει ως αποτέλεσμα την μετατροπή σε κώδικα Braille του κειμένου που προέκυψε από την αναγνώριση ομιλίας. Η απενεργοποίηση του κουμπιού, ώστε να μην επιτρέπεται το πάτημα του, κατά το άνοιγμα της οθόνης είναι απαραίτητη διότι δεν υπάρχει ακόμη κείμενο το οποίο θα χρειαστεί να μετατραπεί σε κώδικα Braille.

Η εντολή που ακολουθεί έχει ως αποτέλεσμα την ενεργοποίηση του πλαισίου κειμένου με ονομασία TextBox1. Σε αυτό το πλαίσιο κειμένου πρόκειται να προβληθεί το κείμενο που θα προκύψει από την αναγνώριση ομιλίας. Το πλαίσιο είναι αρχικά απενεργοποιημένο, ώστε να μην επιτρέπεται στο χρήστη να έχει πρόσβαση σε αυτό, και επομένως την δυνατότητα πληκτρολόγησης κειμένου, πριν ολοκληρωθεί η διαδικασία της αναγνώρισης ομιλίας. Ο λόγος που προτιμήθηκε το συστατικό του πλαισίου κειμένου, έναντι του συστατικού της ετικέτας για την προβολή του κειμένου που προκύπτει από την αναγνώριση ομιλίας οφείλεται στο γεγονός ότι σε κάποιες περιπτώσεις το αποτέλεσμα που εξάγεται από την αναγνώριση ομιλίας μπορεί να περιέχει λάθη και να χρειάζεται να γίνουν ορισμένες τροποποιήσεις σε αυτό. Το συστατικό της ετικέτας απλά αναγράφει στην οθόνη το κείμενο που του παρέχει η εφαρμογή, χωρίς όμως να δίνει την δυνατότητα στον χρήστη να το τροποποιήσει. Αντιθέτως το συστατικό του πλαισίου κειμένου επιτρέπει στον χρήστη να παρέμβει στο κείμενο που περιέχεται σε αυτό και να πραγματοποιήσει αλλαγές ή τροποποιήσεις.

Το μπλοκ συνεχίζει με μια δομή ελέγχου if η οποία εξετάζει αν δεν είναι κενό το πλαίσιο κειμένου TextBox1, δηλαδή αν περιέχεται κείμενο σε αυτό. Στην περίπτωση που η προηγούμενη

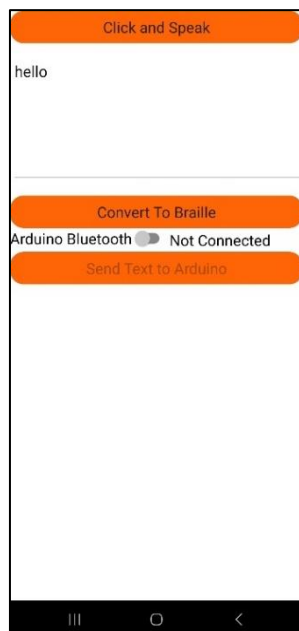
συνθήκη είναι αληθής, τότε τίθεται ως κείμενο στο πλαίσιο αυτό η κενή συμβολοσειρά. Αυτό που ουσιαστικά συμβαίνει είναι η διαγραφή οποιουδήποτε κειμένου τυχόν υπάρχει στο πλαίσιο αυτό. Αν ο χρήστης έχει ήδη χρησιμοποιήσει τουλάχιστον μια φορά την αναγνώριση ομιλίας τότε το πλαίσιο αναγράφει το κείμενο που προέκυψε από την διαδικασία αυτή. Αν ο χρήστης επιθυμεί να χρησιμοποιήσει ξανά την αναγνώριση ομιλίας, είναι απαραίτητο να καθαριστεί το πλαίσιο από το κείμενο της προηγούμενης διαδικασίας, ώστε να προβληθεί σε αυτό το νέο κείμενο που θα προκύψει.

Η τελευταία εντολή του μπλοκ καλεί την μέθοδο `SpeechRecognizer1.GetText` η οποία ζητάει από το χρήστη να μιλήσει και μετατρέπει την ομιλία σε κείμενο. Επιπλέον η μέθοδος αυτή ενεργοποιεί το συμβάν `AfterGettingText` όταν το αποτέλεσμα της αναγνώρισης ομιλίας είναι διαθέσιμο.



Εικόνα 55

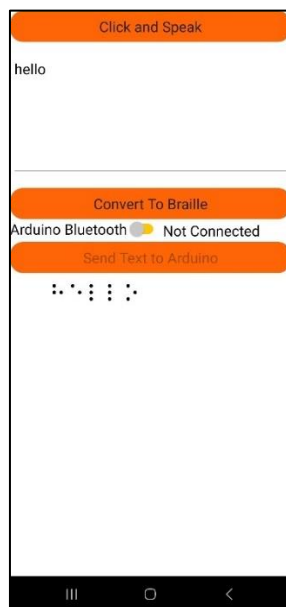
Στην προηγούμενη εικόνα παρουσιάζεται το μπλοκ εντολών με τις οποίες πραγματοποιείται ο χειρισμός του συμβάντος `AfterGettingText`. Η πρώτη εντολή του μπλοκ θέτει ως κείμενο του πλαισίου `TextBox1` την τιμή της παραμέτρου `result` που περιέχει η μέθοδος χειρισμού του συμβάντος. Αυτό έχει ως συνέπεια να προβληθεί στο πλαίσιο κειμένου το αποτέλεσμα που προέκυψε από την αναγνώριση της ομιλίας του χρήστη. Η επόμενη εντολή ενεργοποιεί το κουμπί με ονομασία `ConvertBtn`, ώστε να είναι διαθέσιμο προς τον χρήστη. Το πάτημα του κουμπιού αυτού έχει ως αποτέλεσμα να μετατραπεί σε κώδικα Braille το κείμενο που περιέχεται στο πλαίσιο `TextBox1`. Η μετατροπή του κειμένου σε κώδικα Braille πραγματοποιείται με την αποστολή του στο API μέσω αιτήματος HTTP GET. Η διαδικασία αυτή είναι πανομοιότυπη με εκείνη που πραγματοποιείται κατά την μετατροπή σε κώδικα Braille κειμένου που λήφθηκε από εικόνα και η οποία έχει περιγραφεί αναλυτικά στην προηγούμενη ενότητα.



Εικόνα 56

Η μορφή που παίρνει η οθόνη της εφαρμογής όταν ο χρήστης εκφωνήσει την λέξη «hello» παρουσιάζεται στην παραπάνω εικόνα στην οποία επίσης φαίνεται ότι έχει ενεργοποιηθεί το κουμπί με κείμενο «Convert To Braille».

Το αποτέλεσμα που προκύπτει όταν ο χρήστης πατήσει με το δάκτυλο του το κουμπί αυτό μπορεί να παρατηρηθεί στην επόμενη εικόνα όπου διακρίνονται οι κουκίδες του κώδικα Braille που αντιστοιχούν σε κάθε γράμμα της λέξης «hello». Επίσης στην εικόνα αυτή φαίνεται ότι έχει γίνει διαθέσιμη προς τον χρήστη και η λειτουργία του διακόπτη, με την αλλαγή θέσης του οποίου πραγματοποιείται η σύζευξη μέσω Bluetooth της εφαρμογής με τον μικροελεγκτή Arduino, στον οποίο στη συνέχεια μπορεί να αποσταλεί ο κώδικας Braille της λέξης «hello» ώστε να πραγματοποιηθεί μια οπτική αναπαράσταση του κώδικα αυτού από λυχνίες LED.



Εικόνα 57

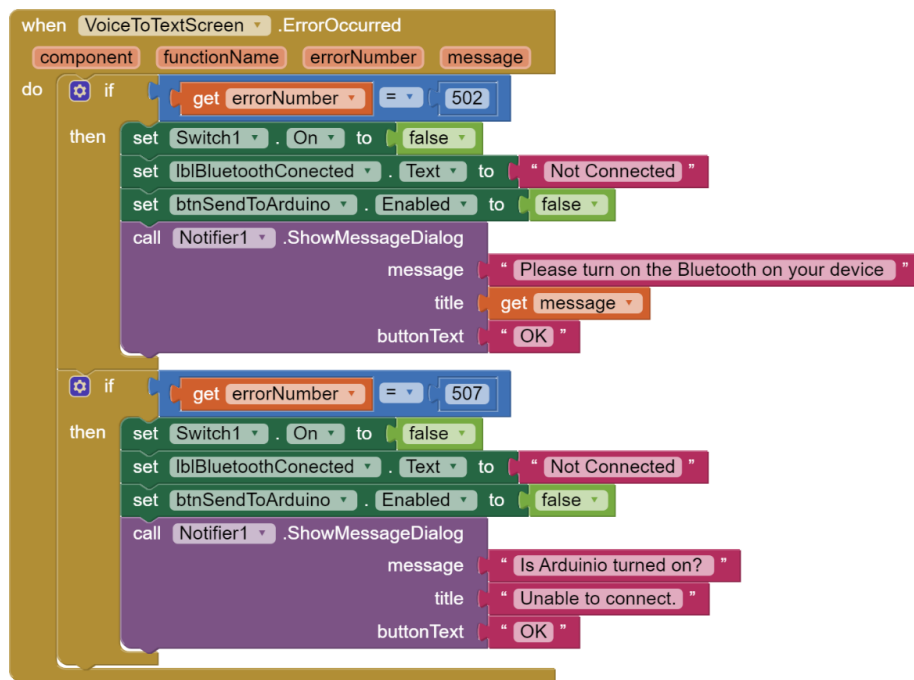
Η αλλαγή θέσης του διακόπτη έχει ως αποτέλεσμα να μεταβληθεί η φράση που υποδεικνύει την κατάσταση εγκαθίδρυσης σύνδεσης από «Not Connected» σε «Connected» και επιπλέον να ενεργοποιηθεί το κουμπί με κείμενο «Send Text to Arduino» με το οποίο ο χρήστης έχει την δυνατότητα να διαβιβάσει των κώδικα Braille στο μικροελεγκτή Arduino. Οι εντολές οι οποίες υλοποιούν την λειτουργία του διακόπτη, του κουμπιού αλλά και οι εντολές με τις οποίες ο Arduino πραγματοποιεί την προσομοίωση του κώδικα Braille έχουν περιγραφτεί αναλυτικά στην προηγούμενη ενότητα.

5.4.2 Εμφάνιση μηνυμάτων

Η εφαρμογή έχει προγραμματιστεί έτσι ώστε να εμφανίζει μηνύματα στο χρήστη στην περίπτωση που για διάφορους λόγους δεν είναι δυνατή η σύζευξη μέσω της διασύνδεσης Bluetooth ανάμεσα στην εφαρμογή και τον Arduino. Τα μηνύματα αυτά χρησιμεύουν ώστε να προτρέψουν τον χρήστη να προβεί στις απαραίτητες ενέργειες που απαιτούνται προκειμένου να πραγματοποιηθεί τελικά η εγκαθίδρυση της σύνδεσης. Στην συνέχεια θα παρουσιαστούν οι εντολές με τις οποίες υλοποιήθηκε η λειτουργία αυτή.

Δύο είναι οι βασικότεροι λόγοι για τους οποίους μπορεί να προκύψει αδυναμία ανάμεσα στην σύνδεση εφαρμογής και Arduino μέσω Bluetooth. Ο ένας από αυτούς έχει να κάνει με το γεγονός ότι υπάρχει πιθανότητα να μην είναι ενεργοποιημένη η λειτουργία Bluetooth στην συσκευή του χρήστη, οπότε σε αυτή την περίπτωση η εφαρμογή πρέπει να προτρέψει τον χρήστη να προβεί στην ενεργοποίησή. Ο δεύτερος εστιάζει στις περιπτώσεις κατά τις οποίες δεν έχει τροφοδοτηθεί με ρεύμα ο μικροελεγκτής ή δεν έχει συνδεθεί σωστά η μονάδα Bluetooth HC-05 στη πλακέτα αυτού οπότε η εφαρμογή ενημερώνει τον χρήστη για το γεγονός αυτό.

Η εμφάνιση των αντίστοιχων μηνυμάτων πραγματοποιείται από το συστατικό Notifier που προσφέρει το περιβάλλον ανάπτυξης MIT App Inventor για αυτό τον σκοπό. Οι εντολές με τις οποίες υλοποιήθηκαν οι διαδικασίες της εμφάνισης των μηνυμάτων παρουσιάζονται στην επόμενη εικόνα.

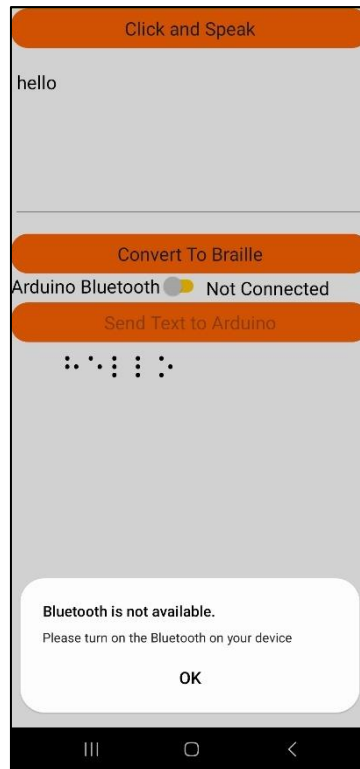


Εικόνα 58

Για το προγραμματισμό της διαδικασίας αυτής χρησιμοποιήθηκε η μέθοδος ErrorOccured της οθόνης. Η μέθοδος ξεκινάει με τον έλεγχο της συνθήκης αν η τιμή της παραμέτρου errorNumber είναι ίση με 502. Η τιμή αυτή αντιστοιχεί στον κωδικό του σφάλματος που πραγματοποιείται όταν η μονάδα Bluetooth της συσκευής δεν είναι ενεργοποιημένη. Αν η συνθήκη είναι αληθής, τότε ορίζεται η θέση on του διακόπτη να έχει την τιμή false. Στην ουσία η εντολή αυτή δεν επιτρέπει στον διακόπτη να μετακινηθεί στην θέση on η οποία υποδηλώνει ότι έχει πραγματοποιηθεί σύνδεση. Στην συνέχεια ορίζεται το κείμενο του συστατικού ετικέτας που έχει την ονομασία lblBluetoothConnecten να είναι το «Not Connected». Η εντολή που ακολουθεί απενεργοποιεί, στην περίπτωση που είναι ενεργοποιημένο, το κουμπί με ονομασία btnSendToArduino, το οποίο είναι το κουμπί εκείνο που με το πάτημά του από τον χρήστη μεταβιβάζει τον κώδικα Braille στον Arduino. Η δομή ελέγχου ολοκληρώνεται με την ενεργοποίηση του συστατικού Notifier1 το οποίο παραμετροποιείται έτσι ώστε να εμφανίζεται σε αυτό το μήνυμα «Please turn on the Bluetooth on your device» το οποίο προτρέπει τον χρήστη να ενεργοποιήσει τη διασύνδεση Bluetooth στην συσκευή του.

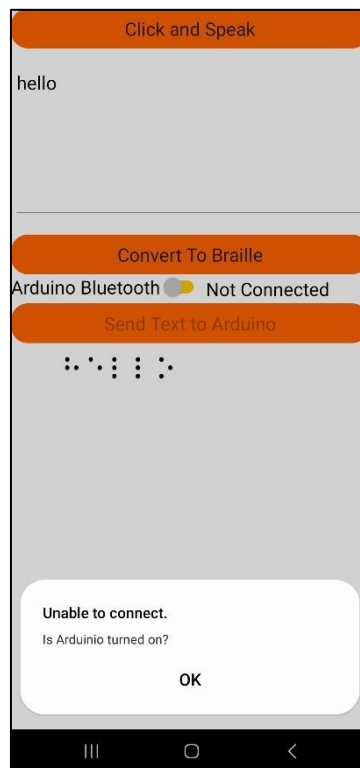
Το μπλοκ συνεχίζει με μια δεύτερη δομή ελέγχου if η οποία εξετάζει αν η τιμή της παραμέτρου errorNumber είναι ίση με 507. Η τιμή αυτή αντιστοιχεί στον κωδικό του σφάλματος που συμβαίνει όταν δεν έχει τροφοδοτηθεί με ηλεκτρικό ρεύμα ο μικροελεγκτής ή δεν έχει συνδεθεί σωστά η μονάδα Bluetooth HC-05 στη πλακέτα αυτού. Αν η συνθήκη είναι αληθής τότε ορίζεται πάλι η θέση on του διακόπτη να έχει την τιμή false και πραγματοποιείται επίσης απενεργοποίηση του κουμπιού btnSendToArduino, το οποίο δεν έχει λόγο να είναι ενεργοποιημένο αφού εξαιτίας της μη συνδεσιμότητας δεν είναι δυνατόν να διαβιβαστεί μήνυμα από την εφαρμογή στον Arduino. Η τελευταία εντολή και αυτής της δομής ελέγχου παραμετροποιεί το συστατικό Notifier με τέτοιο τρόπο που να εμφανίζει το μήνυμα «Is Arduino turned on ?» το οποίο προτρέπει τον χρήστη να ελέγξει αν η πλακέτα του Arduino έχει τροφοδοτηθεί με ηλεκτρικό ρεύμα ή αν υπάρχει κάποιο πρόβλημα στην συνδεσμολογία της μονάδας HC-05 με την πλακέτα.

Στην επόμενη εικόνα φαίνεται το μήνυμα που εμφανίζεται στην οθόνη μέσω του συστατικού Notifier όταν η μονάδα Bluetooth της συσκευής δεν είναι ενεργοποιημένη.



Εικόνα 59

Ενώ στην εικόνα που ακολουθεί παρουσιάζεται το μήνυμα που εμφανίζεται όταν η πλακέτα Arduino δεν έχει τροφοδοτηθεί με ηλεκτρικό ρεύμα.



Εικόνα 60

6 Αξιολόγηση, μελλοντικές επεκτάσεις και συμπεράσματα

Στο κεφάλαιο αυτό παρουσιάζονται τα αποτελέσματα που προέκυψαν από την διενέργεια αξιολόγησης της εφαρμογής ως προς τον προγραμματισμό και την ευχρηστία. Επίσης δίνονται προτάσεις που θα μπορούσαν εφαρμοστούν ως μελλοντικές επεκτάσεις της εφαρμογής που αναπτύχθηκε στα πλαίσια του Capstone Project. Τέλος γίνεται αναφορά στα συμπεράσματα που προέκυψαν από την ολοκλήρωση της εκπόνησης του project πραγματοποιώντας έτσι έναν τελικό απολογισμό του εγχειρήματος

6.1 Αξιολόγηση της εφαρμογής

Στην ενότητα αυτή παρουσιάζονται τα αποτελέσματα τα οποία προέκυψαν από την διενέργεια αξιολόγησης της εφαρμογής. Δεδομένου ότι η εφαρμογή δεν είναι δυνατόν σε αυτό το σημείο να αξιολογηθεί από άτομα με προβλήματα όρασης, πραγματοποιήθηκε προγραμματιστική αξιολόγηση της εφαρμογής καθώς επίσης και μια αξιολόγηση ευχρηστίας με τη μέθοδο SUS.

6.1.1 Προγραμματιστική αξιολόγηση

Η προγραμματιστική αξιολόγηση εστίασε στον έλεγχο της καλής λειτουργίας των επιμέρους τμημάτων της εφαρμογής και στην συνεργασία τους με τα APIs που χρησιμοποιούνται. Έτσι πραγματοποιήθηκαν 20 δοκιμές για κάθε μια από τις λειτουργίες Text to Braille, Voice to Braille και Image to Braille και ελέγχθηκε αν η λειτουργία τους έδινε τα σωστά αποτελέσματα σε κάθε περίπτωση.

Για την λειτουργία Text to Braille ο έλεγχος εστιάστηκε στο κατά πόσο το κείμενο που πληκτρολογεί ο χρήστης μεταφράζεται σωστά σε κώδικα Braille από το API. Οι έλεγχοι έδειξαν ότι σε όλες τις περιπτώσεις (ποσοστό 100%) το αποτέλεσμα της μετάφρασης ήταν το αναμενόμενο. Τα αποτελέσματα αυτά φαίνονται στον παρακάτω πίνακα.

Λειτουργία Text to Braille	
Διαδικασία	Ποσοστό επιτυχίας
Μετατροπή κειμένου σε κώδικα Braille	100 %

Πίνακας 1

Όσον αφορά την λειτουργία Voice to Braille ο προγραμματιστικός έλεγχος βασίστηκε σε δύο άξονες. Ο πρώτος από αυτούς είχε να κάνει με το κατά πόσο το κείμενο που προκύπτει από την φωνητική πληκτρολόγηση είναι το σωστό, δηλαδή αν οι λέξεις που εκφωνεί ο χρήστης αναγνωρίζονται επακριβώς από την λειτουργία αναγνώρισης ομιλίας. Οι έλεγχοι έδειξαν ότι σε 18 από τις 20 περιπτώσεις (ποσοστό 90%) το αποτέλεσμα ήταν το σωστό. Ενώ σε 2 μόνο περιπτώσεις (ποσοστό 10%) το αποτέλεσμα ήταν λανθασμένο, δηλαδή ο χρήστης έπρεπε να παρέμβει στο κείμενο που προέκυψε από την αναγνώριση ομιλίας και να το διορθώσει. Ο δεύτερος άξονας αξιολόγησης της λειτουργίας Voice to Braille επικεντρώθηκε στο γεγονός κατά πόσο το αποτέλεσμα της φωνητικής πληκτρολόγησης του χρήστη μετατράπηκε επιτυχώς σε κώδικα Braille από το API ακόμα και αν το κείμενο που προέκυψε από την φωνητική πληκτρολόγηση περιείχε λάθη. Οι έλεγχοι έδειξαν ότι σε όλες τις περιπτώσεις (ποσοστό 100%) το κείμενο που διαβιβάστηκε στο API μετατράπηκε σωστά σε κώδικα Braille. Τα αποτελέσματα συνοψίζονται στον πίνακα που ακολουθεί

Λειτουργία Speech to Braille	
Διαδικασία	Ποσοστό επιτυχίας
Φωνητική πληκτρολόγηση	90 %
Μετατροπή κειμένου σε κώδικα Braille	100 %

Πίνακας 2

Αναφορικά με την λειτουργία Image to Braille η αξιολόγηση βασίστηκε σε τρεις άξονες. Ο πρώτος από αυτούς εξέτασε αν η φωτογραφία η οποία λήφθηκε από την κάμερα της συσκευής,

διαβιβαζόταν με επιτυχία στο σενάριο της πλατφόρμας Google Apps Script που πραγματοποιεί την οπτική αναγνώριση χαρακτήρων. Οι έλεγχοι έδειξαν ότι στις 17 από τις 20 περιπτώσεις (ποσοστό 85%) το κείμενο διαβιβάστηκε με επιτυχία στο σενάριο, ενώ σε τρεις μόνο περιπτώσεις (ποσοστό 15%) η διαδικασία δεν ολοκληρώθηκε. Ο δεύτερος άξονας της αξιολόγησης εστίασε στο αποτέλεσμα της οπτικής αναγνώρισης χαρακτήρων, δηλαδή αν διαβιβάστηκε σωστά στην εφαρμογή το κείμενο που περιείχε η εικόνα η οποία στάλθηκε στο σενάριο της πλατφόρμας Google Apps Script. Οι έλεγχοι έδειξαν ότι σε όλες περιπτώσεις η εικόνα διαβιβάστηκε στο API, το κείμενο που υπήρχε σε αυτήν στάλθηκε στην εφαρμογή χωρίς να χρειαστούν διορθώσεις (ποσοστό 100 %). Ο τρίτος άξονας της αξιολόγησης επικεντρώθηκε στην σωστή μετατροπή σε κώδικα Braille του κειμένου που προέκυψε από την οπτική αναγνώριση χαρακτήρων και το οποίο στάλθηκε στο API. Οι έλεγχοι έδειξαν ότι σε όλες τις περιπτώσεις (ποσοστό 100%) το κείμενο που διαβιβάστηκε στο API μετατράπηκε χωρίς λάθη σε κώδικα Braille. Τα προηγούμενα αποτελέσματα παρουσιάζονται συνοπτικά και στον ακόλουθο πίνακα.

Λειτουργία Image to Braille	
Διαδικασία	Ποσοστό επιτυχίας
Αποστολή εικόνας στο σενάριο Google App Script	85%
Αποτέλεσμα οπτικής αναγνώρισης χαρακτήρων	100 %
Μετατροπή κειμένου σε κώδικα Braille	100 %

Πίνακας 3

Από τα προηγούμενα εξάγεται το συμπέρασμα ότι η προγραμματιστική λειτουργία της εφαρμογής είναι επιτυχημένη σε πολύ μεγάλο ποσοστό.

6.1.2 Αξιολόγηση ευχρηστίας

Η αξιολόγηση που πραγματοποιήθηκε με την μεθοδολογία SUS εστιάζει στην ταξινόμηση της ευκολίας χρήσης της εφαρμογής. Η Κλίμακα Ευχρηστίας Συστήματος SUS (System Usability Scale) παρέχει ένα γρήγορο, ακριβές και αξιόπιστο εργαλείο για τη μέτρηση της ευχρηστίας. Αποτελείται από ένα ερωτηματολόγιο 10 ερωτήσεων με πέντε επιλογές απάντησης για τους ερωτηθέντες, από το συμφωνώ απόλυτα έως το διαφωνώ απόλυτα. Δημιουργήθηκε αρχικά από τον John Brooke το 1986 και επιτρέπει την αξιολόγηση μιας μεγάλης ποικιλίας προϊόντων και υπηρεσιών, συμπεριλαμβανομένου του υλικού, του λογισμικού, των κινητών συσκευών, των ιστότοπων και των εφαρμογών.

Το SUS έχει γίνει ένα βιομηχανικό πρότυπο, με αναφορές σε πάνω από 1300 άρθρα και δημοσιεύσεις. Τα οφέλη από τη χρήση του SUS περιλαμβάνουν τα εξής:

- Είναι μια πολύ εύκολη κλίμακα για τη χορήγηση στους συμμετέχοντες.
- Μπορεί να χρησιμοποιηθεί σε μικρά μεγέθη δείγματος με αξιόπιστα αποτελέσματα.
- Είναι έγκυρη δηλαδή μπορεί να πραγματοποιήσει μια αποτελεσματική κατάταξη μεταξύ εύχρηστων και μη εύχρηστων συστημάτων.

Το ερωτηματολόγιο των 10 ερωτήσεων που δίνεται στους ερωτηθέντες οι οποίοι θα λάβουν μέρος στην έρευνα για την μέτρηση της ευχρηστίας υπάρχει στο παράρτημα.

Η κλίμακα SUS χρησιμοποιείται γενικά μετά την χρησιμοποίηση του συστήματος που αξιολογείται από τον ερωτώμενο, χωρίς όμως πριν να έχει γίνει οποιαδήποτε ενημέρωση ή συζήτηση. Θα πρέπει να ζητείται από τους ερωτηθέντες να καταγράφουν την άμεση απάντησή τους σε μια από τις πέντε πιθανές απαντήσεις που περιέχει κάθε ερώτηση, χωρίς να σκέφτονται για μεγάλο χρονικό διάστημα ποια απάντηση θα δώσουν. Πρέπει να απαντηθούν όλες οι ερωτήσεις. Εάν ένας ερωτώμενος αισθάνεται ότι δεν μπορεί να απαντήσει σε κάποια συγκεκριμένη ερώτηση, θα πρέπει να σημειώσει το κεντρικό σημείο της κλίμακας.

Το SUS δίνει έναν αριθμό που αντιπροσωπεύει ένα σύνθετο μέτρο της συνολικής χρηστικότητας του συστήματος που μελετάται. Οι βαθμολογίες για κάθε μεμονωμένη ερώτηση δεν έχουν νόημα από μόνες τους. Για τον υπολογισμό της βαθμολογίας SUS, αθροίζονται οι βαθμοί των απαντήσεων από κάθε ερώτηση. Η βαθμολογία της απάντησης κάθε ερώτησης κυμαίνεται από 0 έως 4. Για τις ερωτήσεις 1,3,5,7, και 9 ο βαθμός που δίνει κάθε απάντηση στη βαθμολογία είναι η θέση της απάντησης στην κλίμακα μείον 1. Για τις ερωτήσεις 2,4,6,8 και 10, ο βαθμός που δίνει κάθε απάντηση είναι 5 μείον τη θέση της απάντησης στην κλίμακα. Ο αριθμός που θα προκύψει από το άθροισμα της βαθμολογίας κάθε ερώτησης πολλαπλασιάζεται επί 2,5 για να ληφθεί το συνολικό σκορ SUS. Οι βαθμολογίες SUS έχουν εύρος από 0 έως 100. Γενικά μια βαθμολογία SUS άνω του 68 θεωρείται άνω του μέσου όρου και οτιδήποτε κάτω του 68 είναι κάτω του μέσου όρου.

Το ερωτηματολόγιο δόθηκε σε 10 ερωτηθέντες που χρησιμοποίησαν την εφαρμογή και ο μέσος όρος της βαθμολογίας SUS που προέκυψε από τις απαντήσεις τους ήταν 97,5 % γεγονός που σημαίνει ότι η εφαρμογή έχει ιδιαίτερα υψηλό δείκτη ευχρηστίας.

6.2 Μελλοντικές επεκτάσεις

Η εφαρμογή η οποία αναπτύχθηκε στα πλαίσια της εκπόνησης Capstone Project τρίμηνης διάρκειας ως τελικής Διπλωματικής Εργασίας του Προγράμματος Μεταπτυχιακών Σπουδών «Ψηφιακές Εφαρμογές και Καινοτομία» που προσφέρεται από το Τμήμα Πληροφορικής του Ιονίου Πανεπιστημίου, αποτελεί ένα ανθρώπινο δημιούργημα και συνεπώς δεν είναι τέλεια ούτε απαλλαγμένη από λάθη και παραλήψεις. Εξάλλου η υλοποίηση μιας ολοκληρωμένης εφαρμογής για κινητές συσκευές, η οποία θα βοηθάει τα άτομα με οπτική αναπηρία και όσους τους υποστηρίζουν εκπαιδευτικά, έχει απαιτήσεις που ξεφεύγουν από τα πλαίσια μιας Διπλωματικής Εργασίας Μεταπτυχιακού επιπέδου.

Στη συνέχεια αναφέρονται κάποιες προτάσεις οι οποίες θα μπορούσαν μελλοντικά να υλοποιηθούν, ώστε να βελτιώσουν την εφαρμογή έχοντας ως συνέπια να την αναβαθμίσουν καθιστώντας την περισσότερο λειτουργική, και να επεκτείνουν τις δυνατότητες και τις υπηρεσίες που προσφέρει.

Μια επέκταση που θα μπορούσε μελλοντικά να γίνει στην εφαρμογή θα ήταν να τροποποιηθεί το κύκλωμα το οποίο πραγματοποιεί την αναπαράσταση του κώδικα Braille, με τρόπο ώστε οι λυχνίες LED να αντικατασταθούν από ηλεκτρομαγνητικές βαλβίδες (Solenoids). Αυτό θα αποτελέσει ουσιαστικά όχι μόνο μια οπτική, αλλά και μια ακτική αναπαράσταση του κώδικα Braille. Έτσι θα υπάρξει η δυνατότητα κατασκευής ενός πρωτοτύπου για να δοκιμαστεί ο σχεδιασμός και η λειτουργικότητα κατασκευής μιας συσκευής ανάγνωσης του κώδικα Braille που θα χρησιμοποιείται από άτομα με οπτική αναπηρία.

Για την υλοποίηση του πλάνου της προηγούμενης πρότασης είναι απαραίτητο να μειωθεί και το χρονικό διάστημα το οποίο απαιτείται για να αποσταλεί μέσω Bluetooth από την εφαρμογή στον Arduino η αναπαράσταση ενός χαρακτήρα του κώδικα Braille. Η αναβάθμιση της εφαρμογής και ο προγραμματισμός της ώστε να φέρει σε πέρας την απαίτηση αυτή είναι μια ακόμα πρόταση μελλοντικής επέκτασης.

Η εφαρμογή μετατρέπει σε κώδικα Braille κείμενο το οποίο είναι γραμμένο μόνο στην αγγλική γλώσσα. Μια λειτουργία η οποία θα έκανε την εφαρμογή να μπορεί να χρησιμοποιηθεί από ένα πιο ευρύ κοινό θα ήταν να επεκταθεί κατάλληλα ώστε να μπορεί να μετατρέπει σε κώδικα Braille κείμενο που είναι γραμμένο σε άλλες γλώσσες εκτός της αγγλικής. Για να επιτευχθεί αυτό θα πρέπει είτε να βρεθεί ένα API το οποίο πραγματοποιεί μετάφραση κειμένου της εκάστοτε γλώσσας σε κώδικα Braille είτε να προγραμματιστεί κάνοντας χρήση κατάλληλων δομών δεδομένων που θα διευκολύνουν την λειτουργία αυτή.

Το προγραμματιστικό περιβάλλον MIT App Inventor στο οποίο υλοποιήθηκε η εφαρμογή, είναι μια πλατφόρμα ανάπτυξης εφαρμογών για κινητές συσκευές η οποία χρησιμοποιείται κατά βάση για εκπαιδευτικούς σκοπούς και για την υλοποίηση μικρών σε έκταση έργων. Ως εκ τούτου παρουσιάζει αρκετούς περιορισμούς και χαμηλές δυνατότητες συνεπώς απέχει αρκετά από

το να χαρακτηριστεί ένα περιβάλλον ανάπτυξης προηγμένων και επαγγελματικών εφαρμογών για έξυπνες κινητές συσκευές. Ένεκα του γεγονότος αυτού μια ακόμα πρόταση μελλοντικής βελτίωσης είναι να χρησιμοποιηθεί η ανάλυση και ο σχεδιασμός που πραγματοποιήθηκαν για την υλοποίηση της εφαρμογής αλλά η ανάπτυξη της να γίνει σε ένα προηγμένο και επαγγελματικό περιβάλλον ανάπτυξης εφαρμογών για έξυπνα κινητά τηλέφωνα, όπως το Android Studio ή το React. Αυτό θα προσφέρει μεγαλύτερη ευκολία στην συντήρηση της εφαρμογής και θα επεκτείνει σημαντικά τις δυνατότητες της ώστε να μπορέσει να γίνει εφικτή και η υλοποίηση των μελλοντικών επεκτάσεων που προτείνονται στην συνέχεια.

Η εφαρμογή έχει την δυνατότητα να εξάγει κείμενο από εικόνες μέσω της τεχνολογίας OCR και στην συνέχεια να μετατρέπει το κείμενο αυτό σε κώδικα Braille. Μια επέκταση η οποία θα της έδινε περισσότερο κύρος και ενδεχομένως πολύ μεγάλη δημοφιλία θα ήταν να έχει την δυνατότητα εκτέλεσης και της αντίστροφης διαδικασίας. Δηλαδή να χρησιμοποιεί την τεχνολογία μηχανικής όρασης για να ανιχνεύει χαρακτήρες Braille σε εικόνες και να τους μετατρέπει σε κείμενο. Αυτό είναι δυνατόν να υλοποιηθεί με την χρήση της βιβλιοθήκης ανοιχτού κώδικα OpenCV (Open Source Computer Vision) η οποία παρέχει εντολές για την επεξεργασία εικόνας και την αναγνώριση προτύπων. Η βιβλιοθήκη OpenCV δεν είναι διαθέσιμη στο MIT App Inventor, διατίθεται όμως σε προηγμένες γλώσσες προγραμματισμού, όπως η Java και συνεπώς μπορεί να χρησιμοποιηθεί στο Android Studio.

Μια λειτουργία η οποία θα μπορούσε να προσδώσει στην εφαρμογή πολύ μεγαλύτερες δυνατότητες είναι να επεκταθεί κατά τέτοιο τρόπο ώστε να μπορεί να συνδεθεί με μια οθόνη Braille η οποία είναι μια συσκευή που επιτρέπει σε άτομα με προβλήματα όρασης να διαβάζουν ψηφιακό κείμενο μέσω της αφής. Για να πραγματοποιηθεί αυτό είναι απαραίτητο η εφαρμογή να προγραμματιστεί κατά τέτοιο τρόπο ώστε να μπορεί να ανιχνεύει την οθόνη Braille και να διαμορφώνεται αυτόματα για το συγκεκριμένο μοντέλο της οθόνης. Αυτό διασφαλίζει ότι η εφαρμογή μπορεί να εμφανίσει τους σωστούς χαρακτήρες Braille και να μορφοποιήσει το κείμενο σωστά ανάλογα με το μέγεθος και τον προσανατολισμό της οθόνης. Επιπλέον, η εφαρμογή πρέπει να σχεδιαστεί με γνώμονα την προσβασιμότητα, πράγμα που σημαίνει ότι θα πρέπει να είναι εύκολη η πλοήγηση χρησιμοποιώντας πρόγραμμα ανάγνωσης οθόνης ή φωνητικές εντολές. Θα πρέπει επίσης να έχει τη δυνατότητα να προσαρμόζει το μέγεθος, την απόσταση και το στυλ των χαρακτήρων Braille ώστε να ανταποκρίνεται στις ανάγκες του χρήστη.

Ιδιαίτερα σημαντική επέκταση θα ήταν να δίνει η εφαρμογή στον χρήστη την δυνατότητα να πραγματοποιεί εκτύπωση του παραγόμενου κώδικα Braille σε χαρτί με την χρήση ενός εκτυπωτή Braille. Για να γίνει εφικτό αυτό, είναι απαραίτητο να υλοποιηθούν στον κώδικα της εφαρμογής διαδικασίες οι οποίες να επιτρέπουν την εναρμόνιση της με τα πρωτόκολλα επικοινωνίας και προγράμματα οδήγησης. Αυτό θα μπορούσε να περιλαμβάνει τη χρήση μιας προ υπάρχουσας βιβλιοθήκης λογισμικού που παρέχει ένα API για τον έλεγχο του εκτυπωτή Braille ή την ανάπτυξη ενός κώδικα επικοινωνίας χαμηλού επιπέδου. Επίσης είναι απαραίτητο να δημιουργηθεί και ένα κατάλληλο γραφικό περιβάλλον διεπαφής που θα επιτρέπει στον χρήστη να πραγματοποιεί την εκτύπωση.

Οι βάσεις δεδομένων παίζουν κυρίαρχο ρόλο στην ανάπτυξη συστημάτων και εφαρμογών επειδή παρέχουν έναν αποτελεσματικό τρόπο αποθήκευσης ανάκτησης και διαχείρισης δεδομένων, που διασφαλίζει την ασφάλεια τους, επιτρέπει την επεκτασιμότητα και διατηρεί την ακεραιότητα αυτών. Μια ακόμη πρόταση μελλοντικής επέκτασης της εφαρμογής θα ήταν να δημιουργηθεί μια βάση δεδομένων στην οποία θα αποθηκεύονται όλα τα κείμενα που έχει διαχειριστεί η εφαρμογή καθώς και η αντίστοιχη μετάφραση τους σε κώδικα Braille ώστε να είναι δυνατόν να ανακτηθούν από τον χρήστη ανά πάσα στιγμή το επιθυμεί. Αυτό θα έχει ως αποτέλεσμα την εξοικονόμηση χρόνου καθώς ένα κείμενο που έχει ήδη μετατραπεί σε Braille δεν θα χρειαστεί να υποβληθεί ξανά στην αντίστοιχη επεξεργασία.

6.3 Συμπεράσματα και τελικός απολογισμός

Η εφαρμογή που υλοποιήθηκε ως Capstone Project για τις ανάγκες της Διπλωματικής Εργασίας στα πλαίσια του Προγράμματος Μεταπτυχιακών Σπουδών «Ψηφιακές Εφαρμογές και

Καινοτομία» που προσφέρεται από το Τμήμα Πληροφορικής του Ιονίου Πανεπιστημίου καλύπτει μόνο ένα μικρό μέρος, του κενού το οποίο υπάρχει σχετικά με λογισμικά που παρέχουν υποστήριξη εκπαιδευτικών διαδικασιών σε άτομα με οπτική αναπηρία. Εκτιμάται όμως ότι θα αποτελέσει την απαρχή της δημιουργίας, στο μέλλον, ενός ολοκληρωμένου συστήματος υποστήριξης εκπαιδευτικών διαδικασιών σε άτομα με οπτική αναπηρία.

Θα ήταν προφανώς αυταπάτη ο ισχυρισμός ότι δημιουργήθηκε ένα λογισμικό απαλλαγμένο από προγραμματιστικά λάθη που να προσφέρει την καλύτερη δυνατή εμπειρία στους χρήστες του και μάλιστα στην πρώτη σοβαρή προσπάθεια δημιουργίας λογισμικού τέτοιου είδους. Εξάλλου η ανάπτυξη η συντήρηση και η αναβάθμιση μιας εφαρμογής λογισμικού δεν τελειώνει σχεδόν ποτέ! Υπάρχουν σίγουρα δυσλειτουργίες, λάθη που δεν ανακαλύφθηκαν ακόμα, και πολλά περιθώρια βελτίωσης, για τον λόγο αυτό είναι επιθυμητή οποιαδήποτε εποικοδομητική κριτική καθώς και προτάσεις για αλλαγές, προσθήκες και αναβαθμίσεις πέραν αυτών που αναφέρθηκαν εδώ.

Τελικά όμως ο πραγματικός στόχος του συγκεκριμένου εγχειρήματος επετεύχθη καθώς αποκτήθηκε σημαντική προγραμματιστική εμπειρία πάνω στην ανάπτυξη εφαρμογών για έξυπνες κινητές συσκευές, στην τεχνολογία οπτικής αναγνώρισης χαρακτήρων OCR, στην δημιουργία και τον έλεγχο ηλεκτρονικών κυκλωμάτων στην πλακέτα Arduino καθώς επίσης και στις τεχνολογίες που χρησιμοποιούνται για την ανταλλαγή δεδομένων μέσω αιτημάτων HTTP με την χρήση API. Επιπλέον η εφαρμογή αυτή είναι πολύ πιθανό να χρησιμοποιηθεί για το σκοπό που δημιουργήθηκε, ξεφεύγοντας έτσι από το να χαρακτηριστεί η δημιουργία της ως απλά μία αναγκαία διαδικασία για την απόκτηση ενός Μεταπτυχιακού Διπλώματος.

Πραγματοποιώντας έναν τελικό απολογισμό εξάγεται το συμπέρασμα ότι κατά την διάρκεια της εκπόνησης του Capstone Project, και την τελική ολοκλήρωση αυτού αποκτήθηκε σημαντική εμπειρία και τεχνογνωσία πάνω σε πρακτικά ζητήματα και προηγμένες τεχνολογίες του κλάδου της Πληροφορικής. Επιπλέον δόθηκε η ευκαιρία μεγαλύτερης ενασχόλησης, περισσότερης εμβάθυνσης και καλύτερης κατανόησης, πολλών εννοιών, τεχνολογιών και ζητημάτων που παρουσιάστηκαν κατά την διάρκεια της παρακολούθησης των μαθημάτων του Προγράμματος Μεταπτυχιακών Σπουδών. Όλα αυτά θα αποτελέσουν σημαντική παρακαταθήκη που θα βοηθήσει μελλοντικά στην ανάπτυξη εφαρμογών λογισμικού.

Βιβλιογραφία

- [1] <https://www.blindlook.com/blog/detail/the-population-of-blind-people-in-the-world>
- [2] <https://www.iapb.org/learn/vision-atlas/>
- [3] <https://www.maxmag.gr/eidiki-agogi/kodikas-braille-psilafizontas-tis-lexeis/>
- [4] <https://brailleworks.com/braille-resources/what-is-braille/>
- [5] <https://www.sciencedirect.com/science/article/pii/S1877050921023607>
- [6] https://www.academia.edu/80798850/Voice_to_Braille_Converter_Using_Arduino_Mega
- [7] https://www.researchgate.net/publication/343406954_Enhanced_Braille_Display_Use_of_OCR_and_Solenoid_to_Improve_Text_to_Braille_Conversion
- [8] https://www.researchgate.net/publication/338940968_Low-Cost_Braille_Printer_Prototype_Design_With_OCR_Technology
- [9] <http://ijream.org/papers/IJREAMV07I02SJ019.pdf>
- [10] <https://www.irjet.net/archives/V7/i8/IRJET-V7I8485.pdf>
- [11] <https://iopscience.iop.org/article/10.1088/1757-899X/995/1/012001/pdf>
- [12] <https://files.eric.ed.gov/fulltext/EJ1169379.pdf>
- [13] <https://github.com/topics/braille-translator>
- [14] <https://alexblackwell.ca/pages/braille/index.html>
- [15] <http://www.byronknoll.com/braille.html>
- [16] <https://www.brailletranslator.org/>
- [17] <https://twoblindbrothers.com/pages/braille>
- [18] <https://abcbraile.com/>
- [19] https://drive.google.com/file/d/1Oow_o7Am65Cu9Q3NOVLFew5E9CqcfVT6/view
- [20] <https://ai2.metricrat.co.uk/guides/ocr-images-and-pdfs-using-google-docs>
- [21] <https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html>
- [22] https://www.researchgate.net/publication/228593520_SUS_A_quick_and_dirty_usability_scale
- [23] <https://measuringu.com/sus/>
- [24] <https://uxpajournal.org/sus-a-retrospective/>
- [25] https://en.wikipedia.org/wiki/MIT_App_Inventor
- [26] <https://en.wikipedia.org/wiki/Base64>
- [27] https://en.wikipedia.org/wiki/Google_Apps_Script
- [28] https://en.wikipedia.org/wiki/Optical_character_recognition
- [29] https://en.wikipedia.org/wiki/Google_Drive
- [30] https://en.wikipedia.org/wiki/Speech_recognition

- [31] <https://funtranslations.com/api/braille>
- [32] <https://en.wikipedia.org/wiki/Arduino>
- [33] https://en.wikipedia.org/wiki/Arduino_Uno
- [34] <https://www.arduino.cc/en/software>
- [35] <https://en.wikipedia.org/wiki/Bluetooth>
- [36] https://en.wikipedia.org/wiki/Bluetooth_Low_Energy

Παράρτημα

Το ερωτηματολόγιο της μεθοδολογίας αξιολόγησης ευχρηστίας ενός συστήματος με το πρότυπο SUS.

	Διαφωνώ απόλυτα				Συμφωνώ απόλυτα
1. Νομίζω ότι θα ήθελα να χρησιμοποιώ αυτό το σύστημα συχνά.					
	1	2	3	4	5
2. Βρήκα το σύστημα περιττά πολύπλοκο.					
	1	2	3	4	5
3. Νομίζω ότι το σύστημα ήταν εύκολο στη χρήση.					
	1	2	3	4	5
4. Νομίζω ότι θα χρειαζόμουν την υποστήριξη ενός τεχνικού προσώπου για να μπορέσω να χρησιμοποιήσω αυτό το σύστημα.					
	1	2	3	4	5
5. Βρήκα ότι οι διάφορες λειτουργίες αυτού του συστήματος ήταν καλά ενσωματωμένες.					
	1	2	3	4	5
6. Νομίζω ότι υπήρχε πολύ μεγάλη ασυνέπεια σε αυτό το σύστημα.					
	1	2	3	4	5
7. Φαντάζομαι ότι οι περισσότεροι άνθρωποι θα μάθαιναν να χρησιμοποιούν αυτό το σύστημα πολύ γρήγορα.					
	1	2	3	4	5
8. Βρήκα το σύστημα πολύ δυσκίνητο στη χρήση.					
	1	2	3	4	5
9. Αισθάνθηκα πολύ σίγουρος για τη χρήση του συστήματος.					
	1	2	3	4	5
10. Χρειάστηκε να μάθω πολλά πράγματα προτού μπορέσω να ξεκινήσω με αυτό το σύστημα.					
	1	2	3	4	5