

**Πανεπιστήμιο Πατρών**  
**Διατμηματικό μεταπτυχιακό πρόγραμμα**  
**Συστήματα Επεξεργασίας Σήματος και Εικόνας(ΣΕΣΕ)**

**Ασύρματα τηλεπικοινωνιακά δίκτυα**

**Τελική εργασία του μαθήματος**

**Τζένος Δημήτρης Σ.Ε.Σ.Ε. (Α.Μ: 68)**

**Λάλος Αριστείδης Σ.Ε.Σ.Ε. (Α.Μ.: 73)**

**Σιώγκας Γιώργος Σ.Ε.Σ.Ε. (Α.Μ.: 82)**

**Καραφώτης Παναγιώτης Σ.Ε.Σ.Ε. (Α.Μ.: 70)**

**Αλεξανδρόπουλος Γιώργος Σ.Ε.Σ.Ε. (Α.Μ.: 78)**

**Μουράς Γεώργιος Μ.Δ.Ε. (Α.Μ.: )**

**Πάτρα, Μάρτιος 2004**

## **Περιεχομενα**

<b>1</b>	<b>Εισαγωγή</b>	<b>3</b>
<b>2</b>	<b>Θεωρία</b>	<b>3</b>
2.1	Ομόδυνος δυαδικός πομπός FSK . . . . .	4
2.2	Ετερόδυνος δυαδικός δέκτης FSK . . . . .	5
<b>3</b>	<b>Υλοποίηση</b>	<b>6</b>
3.1	Ο Κωδικοποιητής . . . . .	7
3.2	Ο Αποκωδικοποιητής . . . . .	8
<b>4</b>	<b>Περιγραφή λειτουργίας των εφαρμογών</b>	<b>9</b>
<b>5</b>	<b>Σύνδεση με ανώτερα επίπεδα δικτύου</b>	<b>10</b>

## 1 Εισαγωγή

Στα πλαίσια της εργασίας αυτής θα επιχειρήσουμε την επικοινωνία μεταξύ δύο υπολογιστών συνδέοντας τις κάρτες ήχου (το line in της μιας με το line out της άλλης). Η ομάδα μας θα ασχοληθεί αποκλειστικά με την υλοποίηση του φυσικού επιπέδου που χρησιμοποιείται για τη μεταφορά των δεδομένων. Η μετάδοση ψηφιακών δεδομένων ως γνωστό παραγματοποιείται χρησιμοποιώντας διαμόρφωση των εισερχόμενων δεδομένων πάνω σε ένα φέρον (συνήθως ημιτονικό) με καθορισμένα, απο το δίαυλο όρια συχνότητας. Η διαδικασία διαμόρφωσης που χρησιμοποιούμε περιλαμβάνει μεταλλαγή της συχνότητας του φέροντος σύμφωνα με τα εισερχόμενα δεδομένα. Θα εφαρμόσουμε επομένως δυαδική μεταλλαγής μετατόπισης συχνότητας (binary frequency-shift keying BFSK) αντιστοιχώντας το δυαδικό 0 στη συχνότητα των  $1.2\text{KHz}$  και το δυαδικό 1 στη συχνότητα των  $2.2\text{KHz}$ .

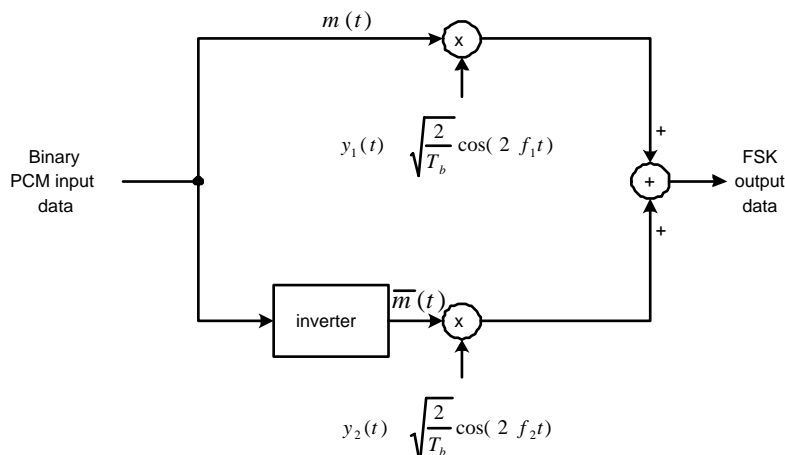
Πιο συγκεκριμένα παρακάτω θα περιγράψουμε τη λειτουργία της κωδικοποίησης<sup>1</sup> και πώς αυτή παραγματοποιείται στην εφαρμογή που αναπτύξαμε. Σημειώνουμε ότι η εφαρμογή που αναλαμβάνει την διαμόρφωση του ημιτονικού φέροντος πραγματοποιεί και τη μετάδοση απο την κάρτα ήχου των ζητούμενων bits με μορφή τόνων με ρυθμό (*bit-rate*) που ορίζεται απο το χρήστη. Πέρα απο τη λειτουργία κωδικοποίησης υλοποιήσαμε και μια εφαρμογή που αναλαμβάνει τη λειτουργία της αποκωδικοποίησης. Ουσιαστικά υλοποιούμε ένα ετερόδουνο δυαδικό δέκτη FSK. Είναι σημαντικό να αναφέρουμε ότι επειδή τα πειράματα που εκτελέστηκαν για να ελεγχθεί η ορθή λειτουργία των εφαρμογών μας έγιναν χρησιμοποιώντας μόνο ένα υπολογιστή και όχι δύο (δηλαδή δεν ήταν δυνατό να χρησιμοποιηθεί το line out της μιας κάρτας σαν είσοδο στην άλλη) πραγματοποιούμε κατα την αποκωδικοποίηση ηχογράφηση χρησιμοποιώντας σαν recording device όχι το line in αλλά το wave stream του υπολογιστή δημιουργώντας έτσι ένα τοπικό loopback.

## 2 Θεωρία

Στο τμήμα αυτό της αναφοράς μας θα αναφερθούμε πιο αναλυτικά στο πώς γίνεται η κωδικοποίηση και αποκωδικοποίηση των προς μετάδοση δεδομένων πραγματοποιώντας διαμόρφωση και αποδιαμόρφωση του ημιτονικού φέροντος. Όπως αναφέραμε ο κωδικοποιητής χρησιμοποιεί ένα δυαδικό πομπό FSK για τη διαμόρφωση των PCM δεδομένων τα οποία είναι αποθηκευμένα σε κάποιο *buffer* και αποτελούν το προς μετάδοση σήμα ενώ για την αποκωδικοποίηση υλοποιούμε ένα ετερόδουνο BFSK δέκτη. Στο σημείο αυτό είναι σημαντικό να αναφέρουμε ότι ο ομόδουνος BFSK δέκτης, αν και σχετικά πιο απλός, δεν μπορούσε να χρησιμοποιηθεί γιατί υποθέτει συγχρονισμό ανάμεσα στον πομπό και τον δέκτη. Επειδή στην περίπτωσή μας η υπόθεση αυτή δεν ισχύει, έπρεπε να χρησι-

---

<sup>1</sup>Τη λειτουργία δηλαδή του δυαδικού πομπού FSK



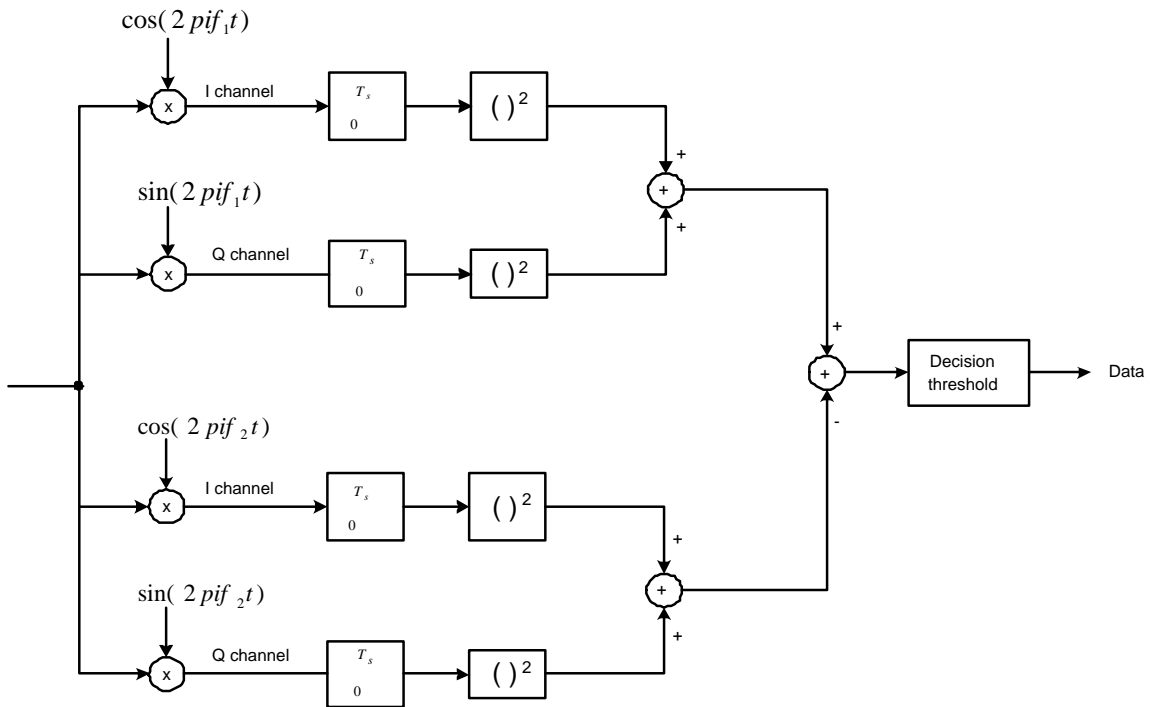
Σχήμα 1: Δομικό διάγραμμα για δυαδικό πομπό FSK.

μποποιηθεί ο ετερώδυνος. Παρακάτω εξηγούμε πιο αναλυτικά τη λειτουργία ενός BFSK πομπού (κωδικοποιητή) και ενός ετερώδυνου BFSK δέκτη (αποκωδικοποιητή).

## 2.1 Ομόδυνος δυαδικός πομπός FSK

Ένας ομόδυνος BFSK πομπός παρουσιάζεται στο σχήμα 1. Η δυαδική ακολουθία εισόδου παρουσιάζεται σε μια μορφή όπου το σύμβολο 1 αναπαρίστανται από ένα σταθερό πλάτος σε volt και το σύμβολο 0 αναπαρίστανται από 0 volt. Χρησιμοποιώντας αντιστροφέα (inverter) στο κάτω δίαυλο ουσιαστικά εξασφαλίζουμε ότι όταν έχουμε σύμβολο 1 στην είσοδο, ενεργοποιείται ο ταλαντωτής με συχνότητα  $f_1$  στο πάνω δίαυλο, ενώ ο ταλαντωτής με συχνότητα  $f_2$  στο κάτω δίαυλο απενεργοποιείται με αποτέλεσμα να μεταδίδεται η συχνότητα  $f_1$ . Αντίστροφα αν έχουμε το σύμβολο 0 στην είσοδο, ο ταλαντωτής στο πάνω δίαυλο απενεργοποιείται και ο ταλαντωτής στο κάτω δίαυλο ενεργοποιείται με αποτέλεσμα να μεταδίδεται η συχνότητα  $f_2$ . Οι δύο συχνότητες  $f_1$  και  $f_2$  επιλέγονται ώστε να είναι ακέραια πολλαπλάσια του ρυθμού bit  $\frac{1}{T_b}$ .

Στο πομπό υποθέτουμε ότι οι δύο ταλαντωτές είναι συγχρονισμένοι και επομένως οι έξοδοί τους ικανοποιούν τις ιδιότητες των δύο συναρτήσεων ορθοκανονικής βάσης  $y_1(t)$  και  $y_2(t)$ . Εναλλακτικά μπορεί να χρησιμοποιηθεί ένα απλός ταλαντωτής ελεγχόμενος από τάση. Και στις δύο περιπτώσεις η συχνότητα της διαμορφωμένης κυματομορφής μετατοπίζεται διατηρώντας τη φάση συνεχή σύμφωνα με τη δυαδική κυματομορφή εισόδου. Αυτό σημαίνει ότι, η συνέχεια της φάσης διατηρείται πάντα, συμπεριλαμβανομένων των χρονικών στιγμών αλλαγής μεταξύ bit. Αναφερόμαστε σε αυτό το τύπο ψηφιακής διαμόρφωσης σαν μεταλλαγή μετατόπισης συχνότητας συνεχούς φάσης (continuous-phase frequency-shift keying, CPFSK). Στη συνέχεια ακολουθεί μια θεωρητική



Σχήμα 2: Δομικό διάγραμμα για ετερόδουνο δυαδικό δέκτη FSK που χρησιμοποιεί τις I και Q συνιστώσες.

προσέγγιση της λειτουργείας ενός ετερόδουνου BFSK δέκτη.

## 2.2 Ετερόδουνος δυαδικός δέκτης FSK

Στην προηγούμενη ενότητα είδαμε ότι στη μεταλλαγή μετατόπισης συχνότητας, η συχνότητα μιας σταθερού πλάτους φέρουσας ποικίλει μεταξύ δύο τιμών που αντιστοιχούν στα δύο πιθανά επίπεδα του σήματος πληροφορίας 1 και 0 (high - low tones). Παραπάνω εξετάσαμε περιπτώσεις όπου η φάση ήταν συνεχείς, ωστόσο ανάλογα με τη διακύμανση της συχνότητας του μεταδιδόμενου σήματος, το FSK παρουσιάζει συνέχεια ή ασυνέχεια στη φάση. Το FSK σήμα σε αυτές τις περιπτώσεις διέπεται από τις σχέσεις:

$$S_{FSK}(t) = \begin{cases} U_H(t) = \sqrt{\frac{2E_b}{T_b}} \cos(2\pi f_c + 2\pi\Delta f)t, & 0 \leq t \leq T_b \text{ (binary 1)} \\ U_L(t) = \sqrt{\frac{2E_b}{T_b}} \cos(2\pi f_c - 2\pi\Delta f)t, & 0 \leq t \leq T_b \text{ (binary 0)} \end{cases} \quad (1)$$

όπου  $2\pi\Delta f$  είναι μία σταθερή απόκλιση από την κύρια συχνότητα της φέρουσας. Το μειονέκτημα

του FSK σήματος είναι η ασυνέχεια που παρατηρείται στη φάση του μεταδιδόμενου σήματος με συνέπεια το άπλωμα του φάσματος. Για να αποφευχθεί η ασυνέχεια αυτή απαιτείται ο αμέριστος αριθμός κύκλων στο  $T_s$ , Αυτό σημαίνει ότι η ελάχιστη διαφορά μεταξύ των συχνοτήτων θα είναι  $(\Delta f)_{min} = \frac{1}{T_s}$ .

Το βασικό πλεονέκτημα του FSK σήματος είναι η ετερόδυνη φώραση, όπου δεν απαιτείται ο συγχρονισμός μεταξύ πομπού και δέκτη. Στο σχήμα 2 φαίνεται ένας ετερόδυνος FSK δέκτης ο οποίος χρησιμοποιεί τα  $K$  και  $Q$  κανάλια κάνοντας χρήση της τριγωνομετρικής ιδιότητας (2) και χρησιμοποιείται για να επανακτηθεί η αρχική δυαδική ακολουθία από την ακολουθία που πέρνουμε στην έξοδο του ομόδυνου πομπού<sup>2</sup>.

$$\cos(2\pi ft - \phi) = \cos(\phi)\cos(2\pi ft) + \sin(\phi)\sin(2\pi ft) \quad (2)$$

Έτσι, κάθε διάυλος λειτουργεί σε μια από τις δύο συχνότητες, χρησιμοποιώντας όμως ο καθένας τις in-phase και quadrature συνιστώσες του σήματος που αποδιαμορφώνει. Για το λόγο αυτό οι συνιστώσες αυτές προστίθενται για κάθε συχνότητα και στη συνέχεια τα σήματα που προκύπτουν αφαιρούνται. Η προκύπτουσα διαφορά  $l$ , συγκρίνεται με κάποιο κατώφλι  $T$ . Εάν  $l > T$  ο δέκτης αποφασίζει υπέρ του 1 διαφορετικά αποφασίζει υπέρ του μηδέν. Στο σημείο αυτό είναι σημαντικό να τονίσουμε ότι το μετάδιδόμενο σήμα που προκύπτει στην έξοδο του πομπού είναι πλούσιο και στις δύο συχνότητες  $f_1$  και  $f_2$  συνεπώς μπορούμε να εξάγουμε τα  $y_1(t)$  και  $y_2(t)$  από το λαμβανόμενο σήμα εφαρμόζοντας το σε ένα ζεύγος φίλτρων στενής ζώνης, το ένα συντονισμένο στην  $f_1$  και το άλλο στην  $f_2$ .

Στην επόμενη ενότητα θα αναφερθούμε περισσότερο αναλυτικά στην υλοποίηση των εφαρμογών μας, η οποία βασίζεται στην θεωρία που αναφέραμε στην παρούσα ενότητα.

### 3 Υλοποίηση

Αρχικά είναι σημαντικό να αναφέρουμε ότι για τη διεπαφή με την κάρτα ήχου, χρησιμοποιήθηκε η βιβλιοθήκη SDL v1.2.6 σε περιβάλλον Windows 2000. Η βιβλιοθήκη SDL (Simple Directmedia Layer) είναι μια βιβλιοθήκη πολυμέσων που σχεδιάστηκε για να παρέχει χαμηλού επιπέδου πρόσβαση στην κάρτα ήχου το πληκτρολόγιο, το ποντίκι, σε 3D Hardware μέσω OpenGL. Χρησιμοποιείται ευρέως από προγράμματα αναπαραγωγής MPEG, από εξομειωτές και πολλά δημοφιλή παιχνίδια. Η SDL είναι γραμμένη σε C, αλλά χρησιμοποιείται και σε ρουτίνες γραμμένες σε C++.

Για την ανάπτυξη των εφαρμογών χρησιμοποιήθηκε το MS Visual Studio 6.0. Η βιβλιοθήκη αυτή

<sup>2</sup>Σημειώνουμε ότι στην ακολουθία αυτή θα μπορούσε να είχε προστεθεί και θόρυβος τον οποίο όμως δε λαμβάνουμε υπόψη μας

στη περίπτωση που θέλουμε να πετύχουμε διεπαφή με την κάρτα ήχου μπορεί να προσφέρει τα εξής:

1. Καθορίζει την αναπαραγωγή 8-bit ή 16-bit ήχου, μονοφωνικού ή στερεοφωνικού, χρησιμοποιώντας κατ'εκλογήν μετατροπή σε περίπτωση που το format δεν υποστηρίζεται από το Hardware.
2. Η παραγωγή ήχου εκτελείται σε ένα ανεξάρτητο thread, το οποίο υλοποιεί ένα callback μηχανισμό που ορίζεται από το χρήστη.
3. Έχει σχεδιαστεί για ειδικούς μίκτες ήχου που υλοποιούνται σε λογισμικό, αλλά περιέχει και μια πλήρη βιβλιοθήκη εξόδου ήχου, μουσικής.

Παρακάτω θα αναφέρουμε πιο αναλυτικά πως υλοποιούμε τον κωδικοποιητή και αποκωδικοποιητή.

### 3.1 Ο Κωδικοποιητής

Η λειτουργία του κωδικοποιητή παρουσιάζεται στο αρχείο *playtone.cpp*. Τα βήματα που ακολουθούνται είναι κατά σειρά τα παρακάτω:

1. Αρχικοποιούμε της βιβλιοθήκη ήχου.
2. Ανοίγουμε τη συσκευή ήχου και δεσμεύουμε μνήμη για μια struct τυπου *SDL\_AudioSpec* η οποία έχει σαν μέλη της χαρακτηριστικά που επιθυμούμε να έχει ο ήχος που θα παράγει η κάρτα ήχου (ρυθμός μετάδοσης, format, ...) και μια struct που αντιπροσωπεύει τα χαρακτηριστικά του ήχου που παράγεται τελικά.
3. Ο ρυθμός μετάδοσης που θέλουμε να πετύχουμε είναι *11025 HZ* που σημαίνει ποιότητα χαμηλότερη από FM ράδιο. Ο ήχος που θέλουμε να παραχθεί είναι μονοφωνικός ενώ χρησιμοποιούμε 16-bit signed integers για να αποθηκεύσουμε τα δείγματα του ήχου. Τέλος ορίζουμε το μέγεθος του buffer που αποθηκεύονται τα δείγματα ήχου (4096) λαμβάνοντας υπόψη ότι μεγάλοι buffers μειώνουν το ρίσκο να χαθεί πληροφορία αλλά αυξάνουν όμως το χρόνο απόκρισης.
4. Ορίζουμε το callback μηχανισμό ο οποίος δέχεται σαν όρισμα ένα δείκτη στον buffer που θα περιέχει τα δείγματα ήχου και το μήκος σε bytes του buffer. Ο μηχανισμός αυτό δημιουργεί από τα δεδομένα ήχου ένα audio stream.
5. Έστω ότι μεταδίδουμε (βγάζουμε ήχο από την soundcard) με  $x$  samples/second. Αν ζητείται μετάδοση με  $r$  bits/second, τότε ένα bit διαρκεί  $1/r$  seconds και επομένως θα του αφιερώσουμε

$x/r$  samples. Δειγματοληπτούμε ένα ημίτονο συχνότητας 1.2 ή 2.2 kHz με  $x$  samples/second για  $1/r$  seconds και τοποθετούμε τα δείγματα στον buffer. Επαναλαμβάνουμε για όλα τα bits της ακολουθίας.

6. Τέλος, δίνουμε εντολή στην κάρτα ήχου να παίξει τα PCM περιεχόμενα του buffer καλώντας την *SDL\_PauseAudio(0)* που επιτρέπει να εφαρμόσουμε κάποιες άλλες αρχικοποιήσεις που απαιτούνται πριν τρέξει η callback συνάρτηση. Αφού χρησιμοποιήσουμε την έξοδο της κάρτας ήχου, την κλείνουμε καλώντας τη συνάρτηση *SDL\_CloseAudio(.)*

### 3.2 Ο Αποκωδικοποιητής

Ο Αποκωδικοποιητής ουσιαστικά ηχογραφεί από το recording device που έχει οριστεί στο PC το οποίο γενικά είναι το line in, αλλά στη περίπτωση μας όπου δημιουργείται ένα τοπικό - loopback -, βάζουμε το recording device να είναι το wave stream του PC. Για την ηχογράφηση χρησιμοποιεί τη γενική μεθοδολογία της βιβλιοθήκης. Στη συνέχεια υλοποιούμε τον ετερόδυνο δυαδικό FSK δέκτη ακριβώς όπως τον περιγράψαμε στη προηγούμενη ενότητα. Η λειτουργία του αποκωδικοποιητή υλοποιείται στο αρχείο *fsk\_dg.c*. Ενώ το *fsk\_dg.h* είναι αρχείο επικεφαλίδας για το *fsk\_dg.c*. Στο αρχείο αυτό ορίζουμε μια δομή πληροφοριών, ορισμένα συμβολικά ονόματα για τις θέσεις των συντελεστών, ένα πίνακα που περιέχει τα μηνύματα λάθους και τα αντίστοιχα ονόματά τους και τις επικεφαλίδες των συναρτήσεων που χρησιμοποιούνται στο *fsk\_dg.c*. Παρακάτω αναφέρουμε τις συναρτήσεις που υλοποιούνται:

- *PrintWaveErrorMsg(.)*: Η συνάρτηση αυτή ανακτά και τυπώνει ένα μήνυμα λάθους για τον αριθμό του Wave In σφάλματος που μεταβιβάζεται.
- *setmute(.)*: Η συνάρτηση αυτή εκτελείται και στη περίπτωση που έχουμε mute διακόπτες. Το ποιοί απο αυτούς έχουν ενεργοποιηθεί το ελέγχουμε προσπαθώντας να ανακτήσουμε πληροφορία για οποιαδήποτε *mute control*. Επιστρέφει επίσης το control ID το οποίο θα χρειαστούμε αργότερα για να καθορίσουμε τη τιμή του.
- *check\_decode(.)*: Συνάρτηση ελέγχου ορθότητας της συνάρτησης αποκωδικοποίησης. Δέχεται σαν όρισμα εισόδου ένα δείκτη προς τη δομή πληροφοριών.
- *decode\_corr(.)*: Η συνάρτηση αυτή υλοποιεί την αποκωδικοποίηση με τη μέθοδο του συσχετιστή όπως περιγράψαμε στην ενότητα 2. Δέχεται σαν είσοδο ένα δείκτη προς τη δομή πληροφοριών και ένα προς το σήμα εισόδου. Επιστρέφει 0 ή 1 ανάλογα με το σήμα εισόδου.
- *error(.)*: Αναλαμβάνει την εκτύπωση λάθους και τερματίζει. Δέχεται σαν όρισμα ένα αλφαριθμητικό φόρμας και μεταβλητά ορίσματα, επιστρέφει στο περιβάλλον το μήνυμα *EXIT FAILURE*.



- *main()*: Κύρια συνάρτηση. Αρχικά γίνεται ο έλεγχος των ορισμάτων που δίνονται στη γραμμή εντολών, γίνονται οι κατάλληλες αρχικοποιήσεις και η ηχογράφηση απο το recording device που ορίζει ο χρήστης. Στη συνέχεια λαμβάνει χώρα η αποδιαμόρφωση και η εκτύπωση στην ορθή των αντίστοιχων bit.

Στην ενότητα που ακολουθεί περιγράφουμε τη λειτουργία των εφαρμογών που υλοποιήσαμε.

## 4 Περιγραφή λειτουργίας των εφαρμογών

Η εφαρμογή του κωδικοποιητή *playtone.exe* δέχεται ως ορίσματα στη γραμμή εντολών το πλάτος του ημιτονου που χρησιμοποιείται για τη διαμόρφωση, τη φάση σε ακίνια και το ρυθμό με τον οποίο μεταδίδουμε τα bits. Η συνταξη της είναι:

Syntax: `cmdline\Debug\playtone.exe <amplitude> <Phase (rad)> <Bitrate>`

π.χ. `cmdline\Debug\playtone.exe 1000 0 100`

Η εφαρμογή αρχικοποιείται και ζητάει το string με τα bits προς μετάδοση:

```
Tone Generator v0.1
(c) Dimitris Tzenos, 2003–2004
Program compiled to operate with sample rate 44100 Hz

Sine wave amplitude           : 1000
Sine wave phase (rad)        : 0
Bitrate                       : 100
Duration (msec)              : 10
Initializing SDL...           [Done]
Give sequence of bits to transmit (Ctrl–C to exit):
```

10

Δίνουμε ως input μια σειρά από bits, πχ 00000000000111010101011010001001001111001001. Με το πάτημα του enter, τα δεδομένα μεταδίδονται (ακούγονται οι αντίστοιχοι ήχοι από το PC).

Ο αποκωδικοποιητής δέχεται σαν όρισμα στη γραμμή εντολών μόνο το ρυθμό με τον οποίο μεταδίδονται τα bits απο το πομπό. Η σύνταξη της εφαρμογής είναι:

Syntax: `cmdline\decoder\decoder.exe <Bitrate>`

π.χ. `cmdline\decoder\decoder.exe 100`

Η εφαρμογή ξεκινά. Αν θέλουμε να κάνουμε κάποιο *interface mute*, το κάνουμε στην αρχική οθόνη. Στη γενική περίπτωση, απλά πατάμε enter. Από το σημείο αυτό ξεκινά να γίνεται decode των ήχων που λαμβάνονται από το recording device της κάρτας ήχου σε realtime.

## **5 Σύνδεση με ανώτερα επίπεδα δικτύου**

Απομονώνονται σε ανεξάρτητες συναρτήσεις οι encoder και decoder. Στη συνέχεια, μπορούν να ενσωματωθούν σε εφαρμογή που απαιτεί μετάδοση και λήψη διαδικών δεδομένων με σχετικά εύκολο και προφανή τρόπο. Θα μπορούσαν για παράδειγμα οι εφαρμογές αυτές να γίνουν attached από το λειτουργικό σύστημα NOS, το οποίο θα αναλαμβάνει τον διαχωρισμό δεδομένων σε πακέτα συγκεκριμένου μήκους και συγκεκριμένης δομής, που με τη σειρά τους θα αποστέλονται μέσω του φυσικού επιπέδου που υλοποιούν οι εφαρμογές που αναπτύξαμε.