

ΕΡΓΑΣΤΗΡΙΟ ΨΗΦΙΑΚΗΣ ΕΠΕΞΕΡΓΑΣΙΑΣ ΣΗΜΑΤΩΝ II

ΑΝΑΦΟΡΑ ΕΡΓΑΣΤΗΡΙΑΚΗΣ ΑΣΚΗΣΗΣ 1

ΑΘΑΝΑΣΟΥΛΑΣ ΓΙΩΡΓΟΣ 228264

ΚΑΡΑΠΙΠΕΡΗΣ ΑΛΕΞΑΝΔΡΟΣ 228070

ΑΚΟΛΟΥΘΕΙ ΜΙΑ ΠΕΡΙΓΡΑΦΗ ΤΗΣ ΛΟΓΙΚΗΣ ΤΟΥ ΚΩΔΙΚΑ
ΓΙΑ ΚΑΘΕ ΕΡΩΤΗΜΑ ΞΕΧΩΡΙΣΤΑ:

1.2) Χρησιμοποιούμε τις έτοιμες συναρτήσεις

```
Int16 DSK6713_AIC23_read(DSK6713_AIC23_CodecHandle hCodec, Uint32 *val)
```

```
Int16 DSK6713_AIC23_write(DSK6713_AIC23_CodecHandle hCodec, Uint32 val)
```

Με την 1^η λαμβάνουμε δεδομένα από τον AIC23 codec 32bit ανά φάση

Αφού έχουμε ρυθμίσει τη θύρα McBSP1 μέσω του καταχωρητή RCR (Γράφουμε 0x000000A0 στον RCR) και τα αποθηκεύουμε σε κάποια διεύθυνση μνήμης,

εώς ότου είμαστε έτοιμοι να μεταδώσουμε το δεδομένο στην έξοδο της σειριακής θύρας

ώστε να γίνει η εγγραφή των 32bit στον codec. Μεταδίδουμε 32 bit γιατί το έχουμε ρυθμίσει όμοια με τη λήψη (Γράφουμε 0x000000A0 στο XCR καταχωρητής McBSP1 θύρας).

Η διαδικασία αυτή υλοποιείται με τη μέθοδο του polling με τη βοήθεια των τιμών που επιστρέφουν οι 2 έτοιμες συναρτήσεις και έτσι επιτυγχάνεται ο έλεγχος της ετοιμότητας της θύρας για αποστολή ή λήψη.

Ο κώδικας εντός της main() για την υλοποίηση της παραπάνω λογικής:

```
while(1){  
    flagin=0;  
    while((flagin==0)){  
        flagin=DSK6713_AIC23_read(hCodec,&input32);  
    }  
}
```

```

flagout=0;
while(flagout==0){
    flagout = DSK6713_AIC23_write(hCodec,input32);
}

}

```

1.3)Ελέγχουμε αν έχουμε γράψει στο buffer μέχρι να τον γεμίσουμε μέχρι τότε η συνάρτηση μας επιστρέφει απλά στη main(),μόλις ο buffer γεμίσει επιστρέφουμε στην αρχική θέση του για να γράψουμε το πιο πρόσφατο και επιστρέφουμε το πιο παλιό στοιχείο του .

```

unsigned int delayed_input(unsigned int input ,unsigned int delay){

    if(count<delay*48000){
        delay_array[count]=input;           /*fill in the buffer until count=N*/
        count++;
        return 0;
    }
    else{
        delay_array[count%(delay*48000) ]=input;    /*return to the first cell
every time count>delay */
        count++;
        return delay_array[count%(delay*48000)+1];
    }
}

```

Κάνουμε την απαραίτητη ρύθμιση για να έχουμε είσοδο ήχου από το μικρόφωνο

Έπειτα εντός της main() χρησιμοποιούμε βοηθητικές μεταβλητές και την stereo_to_mono() για να μετατρέψουμε την 32bit είσοδο σε 16bit

όμοια και την παλιά τιμή του buffer έτσι ώστε να έχουμε $2 \times 16\text{bit}$ εξόδους με χρήση 2 καναλιών το ένα για τις παλιές τιμές (καθυστέρηση - ηχώ) το άλλο για τις real-time τιμές.

1.4)

Ανεβοκατεβάζοντας τους διακόπτες και

διαβάζοντας την τιμή `delay = user_switches_read();`

λαμβάνω μια τιμή από 0-7 την οποία περνάω σαν παράμετρο

στην `unsigned int delayed_input(unsigned int input, unsigned int delay)` .Με αυτή θα πολλαπλασιαστεί το μέγεθος του buffer το οποίο το διαιρώ με το 2 ($\text{delay} \times 24000$) για να πετύχω καθυστέρηση 0-3.5 sec με βήμα 0.5 sec.

Έτσι καθορίζουμε το που θα τελειώνει κάθε φορά ο buffer μας.

Επειδή η `printf()` παίρνει αρκετό βάζουμε συνθήκη για να εκτυπώνουμε την κατάσταση των switches stdout μόνο όταν αλλάζει η κατάσταση τους.