

אוניברסיטת בן-גוריון
הפקולטה להנדסה
המחלקה להנדסת חשמל ומחשבים

דו"ח מסכם לפרויקט גמר של מעבדת מיקרו מחשבים

אפנון אות ספרתי – גרסה 4

חברי קבוצה: אוהד שפירא, אור שמיר

המחלקה להנדסת חשמל ומחשבים, קורס מבוא מיקרו מחשבים

מדריך: אסף לביא

תאריך הגשה: 23.1.2020

תוכן עניינים

1.....	דף שער
2.....	תוכן עניינים
3.....	מטרה והגדרת הפרויקט
3.....	תיאור הפרויקט
4.....	תיאור כללי של ביצועי החומרה והתוכנה (מפרט טכני)
4.....	ביצועי החומרה
4.....	ביצועי התוכנה
5.....	רוטינות ותפקידן
6.....	הסבר מפורט של תוכנה ראשית ופונקציות
8.....	מיפוי משתנים בזיכרון
8.....	מסקנות והצעות לשיפורים

מטרה והגדרת הפרויקט:

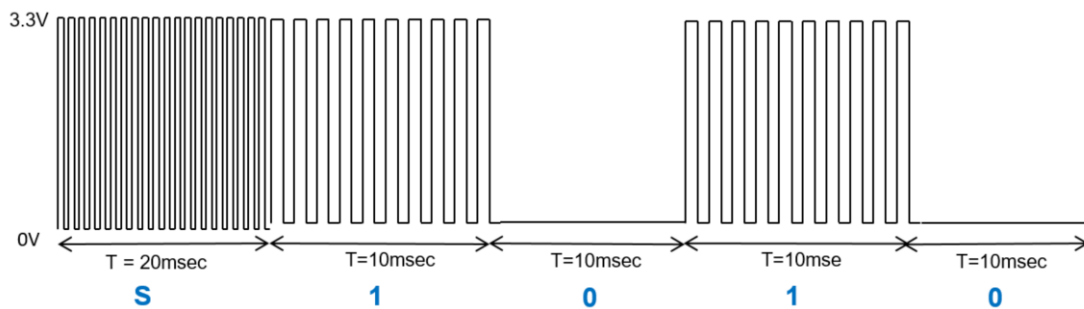
מטרת הפרויקט היא לבצע אפנון של אות ספרתי בגודל של 4 ביט בעזרת הבקר. בפרויקט אפשרנו קינפוג של תדר המודולציה ותדר הסנכרון ובהתאם להגדרות, לפי פקודה, הוצאת אות ספרתי ריבועי באורך של 60msec מרגל P2.2 של הבקר. את הגל הנוצר ניתן להציג על משקף התנודות.

במהלך פעולת האפנון, המשתמש יכול להשתמש ב-Keypad על מנת לבחור את האות שהוא מעוניין לאפן ובעזרת מסך ה-LCD יראה חיווי של הבחירות שביצע, ניווט מתבצע בעזרת כפתורי ה-Push Buttons.

תיאור הפרויקט:

הבקר מאפן אות ספרתי באורך 4 ביט בהתאם לבחירת המשתמש, כלומר, לאחר שהמשתמש בחר תדר מודולציה ותדר סנכרון, המשתמש יכול לבחור את האות שהוא מעוניין לאפן על ידי בחירת אות בלחיצה על ה-Keypad, במהלך פעולת האפנון, המשתמש יכול לשנות תדרים ואת האות שמאופן.

האות המאופן נראה בצורה הבאה:



כאשר S באורך 20msec והוא תדר הסנכרון ולאחריו, מופיע האות שאותו נדרשנו לאפן, בדוגמה זאת נבחר האות A אשר בייצוג בינארי מיוצג על ידי 1010.

במהלך כל התוכנית, מוצג למשתמש חיווי אודות התוכנית על גבי מסך ה-LCD, מעבר בין התפריטים ובחירה מאופציות מובנות מתבצע על ידי כפתורי ה-Push Buttons.

בתפריט הראשי קיימות מספר אפשרויות:

- מסך הוראות:
במסך זה מוצג למשתמש שם הפרויקט והוראות לאיזה רגל בקר לחבר את המשקף תנודות על מנת להציג את האות המאופן.
- תדר מודולציה:
במסך זה מוצג למשתמש מספר אפשרויות לבחירת תדר מודולציה f_1 מבין 2kHz, 4kHz, 6kHz.
- תדר סנכרון:
במסך זה מוצג למשתמש מספר אפשרויות לבחירת תדר מודולציה f_0 מבין 1kHz, 2kHz.
- התחלת פעולה:
לאחר כניסה לאופציה זאת, האות משודר מרגל P2.2 של הבקר.
- יציאה:
לאחר בחירת אופציה זאת, הבקר נכנס למצב שינה, עוצר את תהליך אפנון האות, יציאה ממצב זה מתבצעת אך ורק על ידי לחיצה על PB0.

תיאור כללי של ביצועי החומרה והתוכנה (מפרט טכני):

ביצועי החומרה:

- מסך LCD: הצגת חיווי למסך נעשית בעזרת מספר פונקציות שכתבנו בהשראת הסברים שניתנו מהאתר. חיבורים פיזיים לבקר נעשו באמצעות הרגליים P3.5, P3.6, P3.7 אל הפורטים RS, RW, E ובאמצעות PORT5 המשמש ל-Data.
- Keypad: פסיקות נעשות באמצעות חיבור הרגל IRQ לרגל P2.7 ובאמצעות חיבור מטריצת הלחצנים ל-PORT4. קריאת הביט שנבחר מתבצעת על ידי השיטה שתוארה בסרטון (מימוש Switch והתייחסות למקלדת כאל מטריצה).
- לדים: אנו כותבים לנוורות באמצעות PORT9.
- Push buttons: כל לחיצה ממופה לפסיקה ב-PORT1 וכך מתאפשר לנו לשלוט על המצב הנוכחי של ה-LCD ועל בחירות המשתמש.
- TimerA – אנו משתמשים בטיימר זה על מנת ליצור פסיקה קבועה במרווחים של 10msec כדי לבדוק האם נדרש לשנות את תדר האות הריבועי.
- TimerB – אנו משתמשים באופציית Output mode אשר באמצעות הטיימר אנו מוציאים אות PWM ריבועי.
- משקף תנודות: הצגת האות המאופנן היוצא מרגל P2.2.

ביצועי התוכנה:

באמצעות התוכנה, אנו קבענו את הלוגיקה שהתוכנית מבצעת, לדוגמה:

- יצירת פסיקות עבור מחזורי שעון
- יצירת פסיקות ופונקציות הפועלות עבור לחיצה על PB
- שינוי תצוגת הלדים בהתאם לבחירת האות.
- הצגת טקסט על מסך ה-LCD
- קליטת תווים מה-Keypad
- ניהול חישוב האות.

רוטינות ותפקידן:

שם	קובץ	סוג	פירוט
TAX_ISR	Main	פסיקה	Timer_A, יוצר פסיקה כל 10msec
TBX_ISR	Main	פסיקה	Timer_B, יוצר PWM
Port1_ISR	functions	פסיקה	יוצר פסיקה מלחצני PB
Manage	functions	רוטינה	ממשמש מכונת מצבים עבור שימוש בלחצן PBO
Quit	functions	רוטינה	מגדיר כניסה למצב quit אשר נכנס למצב שינה ומכבה את האות
Action	functions	רוטינה	מאפשר פסיקות מה-Timer-ים ומוציא אות ריבועי. מאפשר קבלת קלט מה-Keypad
debounce	functions	Macro	יוצר השהייה עבור לחצני PB
Port2_ISR	KP_input	פסיקה	יוצר פסיקה עבור קלט מה-keypad
debounce	KP_input	Macro	יוצר השהייה עבור לחצני KP
Setup_TA	Signal_proc	רוטינה	מגדיר את Timer_A
Setup_C1	Signal_proc	רוטינה	מגדיר את Timer_B
Frequ	Signal_proc	רוטינה	מגדיר את התדר שבו נשתמש עבור Timer_B
Curr_disp	Lcd_display	רוטינה	מעדכן את תצוגת ה-LCD
Start_menu	Lcd_display	רוטינה	חזרה למסך הראשי בתצוגת ה-LCD
Show_signal	Lcd_display	רוטינה	הצגת הקלט שנבחר ב-Keypad
Start_inst	Lcd_display	רוטינה	מעדכן מסך לתפריט "הוראות"
Start_mod	Lcd_display	רוטינה	מעדכן מסך לתפריט "תדר מודולציה"
Start_sync	Lcd_display	רוטינה	מעדכן מסך לתפריט "תדר סנכרון"
Start_act	Lcd_display	רוטינה	מעדכן מסך לתפריט "אקטיבציה"
quit_display	Lcd_display	רוטינה	מעדכן מסך לתפריט "ציאה"
Delay	Lcd_display	Macro	מייצר השהייה עבור lcd
lcd_cmd	Lcd_display	Macro	ביצוע כלל הפקודות הקשורות לראש הכותב (הזזת ראש הכותב להתחלה וכו')
Lcd_data	Lcd_display	Macro	מציג תו בודד במסך
lcd_strobe	Lcd_display	רוטינה	מופעלת על ידי המקרו LCD_cmd לטובת כלל הפקודות הקשורות לראש הכותב
Lcd_setup	Lcd_set	רוטינה	מגדיר את תצורת ה-LCD
lcd_strobe	Lcd_set	רוטינה	מופעלת על ידי המקרו LCD_cmd לטובת כלל הפקודות הקשורות לראש הכותב
lcd_cmd	Lcd_set	Macro	ביצוע כלל הפקודות הקשורות לראש הכותב (הזזת ראש הכותב להתחלה וכו')
Lcd_data	Lcd_set	Macro	מציג תו בודד במסך
Delay	Lcd_set	Macro	מייצר השהייה

הסבר מפורט של תוכנה ראשית ופונקציות (ברמת תרשימי זרימה של התוכנה)

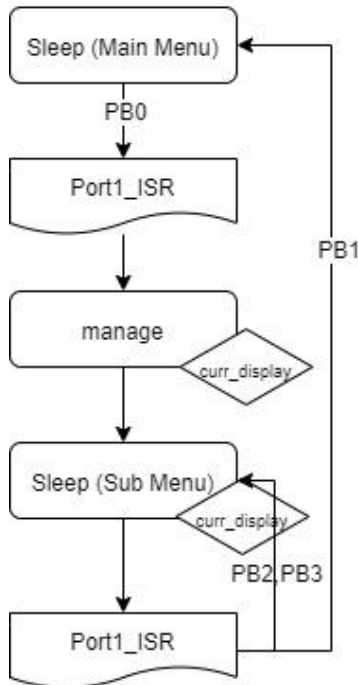
כפי שצינו בפרקים הקודמים קיימים לנו בתוכנית מספר ענפים שניתנים להיכנס אליהם מהתפריט הראשי, ולכן נפרט עליהם בהרחבה:

דפי מידע, בחירת תדר מודולציה ותדר סנכרון

כאשר המשתמש מעוניין להכנס לדף ההוראות הוא מבצע את התהליך כפי שהראינו בתרשים זרימה בדוח המכין, ברמה התוכניתית מתבצעות הפעולות הבאות:

ברגע שהמשתמש לוחץ על מקש הבחירה (PB0), פסיקה מתרחשת ובעקבות הבחירה ובעזרת משתני העזר שלנו, הפונקציה בשם manage בודקת האם נבחרה אפשרות בתת תפריט. לאחר שהתברר שלא נבחרה מכיוון שאין בחירת אפשרויות בתת תפריט של מסך הוראות, מגיעים לפונקציה curr_display שמעדכנת את תצוגת ה-LCD. במצב זה המשתמש רשאי לדפדף בין האפשרויות של התת תפריט באמצעות לחיצה על PB2, PB3 או חזרה לתפריט הראשי באמצעות PB1.

נציין כי בתהליך בחירת תדר מודולציה או סנכרון, התהליך הזה פרט לעובדה שלאחר שהמשתמש מדפדף בתת התפריט ולאחר מכן בוחר תדר, באמצעות מקש PB0, פונקציית manage תשנה את ערך הרגיסטר הרלוונטי לערך התדר שהמשתמש בחר.



הפעלה (Action)

כאשר המשתמש מעוניין להתחיל את תהליך אפנון האות (ולוחץ על PB0), מתבצעים השלבים הבאים ברמת התוכנה:

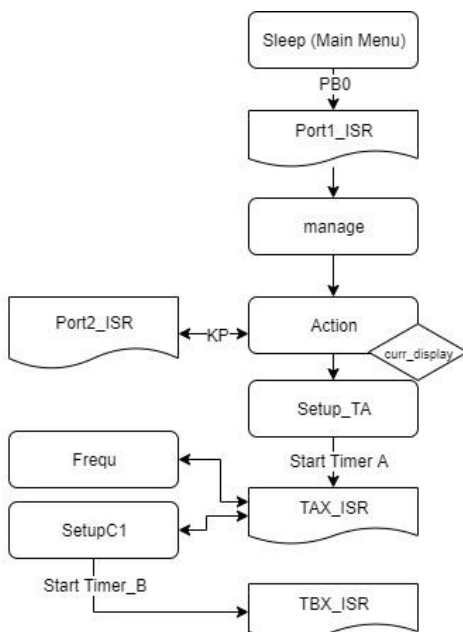
מתרחשת פסיקה בעקבות בחירת המשתמש ובעזרת משתני העזר שלנו, הפונקציה בשם manage יודעת להפנות את הבקר לפונקציית המתאימה לעדכון ה-LCD ומריצה את פונקציית Action.

פונקציה זו למעשה מאפשרת פסיקות מה-Keypad עבור המשתמש שיכניס אות אפנון.

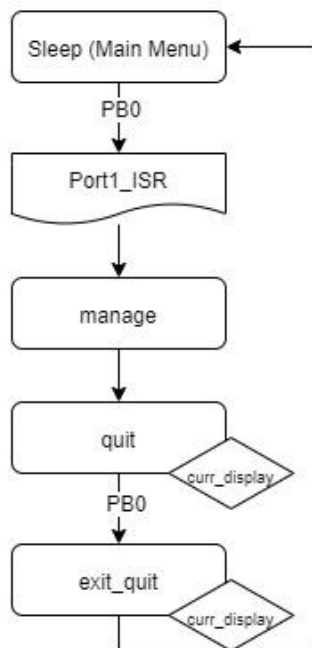
בנוסף, הפונקציה מקנפגת את Timer_A ומאפשרת פסיקות אשר מתוזמנות כל 10msec.

באינטרפט TAX_ISR אנו מבצעים 2 רוטינות:

1. רוטינת Frequ אשר בעקבות הערכים הנתונים לה מגדירה את התדר של PWM עבור Timer_B
2. רוטינת SetupC1 מאפשרת פסיקות מ-Timer_B.



Quit



כאשר המשתמש מעוניין להכנס למצב שינה ולכבות את המערכת הוא מבצע את התהליך כפי שהראינו בתרשים זרימה בדוח המכין, ברמה התוכניתית מתבצעות הפעולות הבאות:

ברגע שהמשתמש לוחץ על מקש הבחירה (PB0), פסיקה מתרחשת ובעקבות הבחירה ובעזרת משתני העזר שלנו, הפונקציה בשם manage יודעת להפנות את הבקר לפונקציית המתאימה בשם quit.

בפונקציית quit אנו למעשה מכבים את האות הריבועי ומבטלים את האפשרות לבצע פסיקות על ידי PB1, PB2, PB3 ועל ידי ה-keyboard.

בנוסף אנו מבצעים עדכון ל-LCD באמצעות הפונקציה curr_display.

במצב זה, רק מקש PB0 זמין, וכאשר לוחצים עליו אנו מגיעים לפונקציה בשם exit_quit שתפקידה זה להחזיר אותנו לתפריט הראשי ולאפשר פסיקות מכל ה-PB.

מיפוי משתנים בזיכרון:

<u>משתנה</u>	<u>שימוש</u>
quit_flag	בדיקה עבור תפריט יציאה
R4	Debounce
R5	Counter בשימוש עבור ביט אפנון נוכחי
R6	Counter עזר עבור בדיקה האם אנחנו באות סנכרון
R7	מייצג את האות המאופנן
R8	התדר הנוכחי שמוזן עבור Timer_B
R9	רגיסטר עזר עבור בדיקה ומחזוריות של הסיגנל
R10	אופציה נוכחית בתפריט הראשי
R11	אופציה נוכחית בתת-תפריט
R13	מייצג את f_0 - תדר הסנכרון
R14	ערך של f_1 - תדר המודולציה
R15	ערך ב-ASCII שנקלט מה-Keypad

**** נציין כי הזנו ערך ברירת מחדל עבור המשתנים המכילים את האות המאופנן, תדר מודולציה ותדר סנכרון.**

מסקנות והצעות לשיפורים

למעשה, במעבדה זאת נבחנו על הידע שנלמדנו בכלל המעבדות ולכן התנסינו בכתיבת תוכנה עצמאית המבצעת את הנדרש מאיתנו.

בראשית התהליך חשבנו כיצד אנחנו רוצים לממש את התוכנית ואת המבנה הכללי שלה, לאחר שהתחלנו לממש נתקלנו במספר מכשולים ולכן היינו צריכים לפתור אותם תוך כדי התהליך, למשל, ניהול מצבים עבור המסכים והתפריטים שלנו, בתחילת התהליך תכננו לנהל את כלל התפריטים על ידי משתנה בודד אך נתקלנו במכשולים ולכן הוספנו משתנה נוסף שינהל לנו את המצבים בתתי התפריטים.

למעשה, שינויים במימוש הוא בלתי נמנע תוך כדי התהליך ולכן על מנת לשפר את התהליך ואת התוכנית הנוכחית הייתי מוסיף משתני סביבה נוספים שינהלו את המצבים שאנחנו נמצאים בהם (כמו שעשינו עבור מצב quit) ובכך ניהול התוכנית יהיה פשוט וקל יותר להבנה.