

Table of Contents

A. הוראות כלליות לעבודה עם ערכות הפיתוח במעבדה:	2
B. <i>חומר עזר:</i>	2
C. חלק תיאורטי – Basic Timer (משפחה MSP430x4xx):	2
D. חלק תיאורטי – Advanced Timers (משפחה MSP430x4xx):	3
E. חלק מעשי – כתיבת קוד מערכת פורטאבילי בשפת אסמבלי:	3
F. הבהרות:	4
G. צורת הגשה דוח מכין:	4

דו"ח מכין מס' 5 – ניסוי Basic and Advanced Timers

A. הוראות כלליות לעבודה עם ערכות הפיתוח במעבדה:

השלט הממוסגר הבא נמצא בכל עמדה בכיתת המעבדה 204/33, רלוונטי החל מניסוי מספר 3 ואילך **בעבודה על**

ערכת הפיתוח במעבדה.

1. סדר פעולות בסיום יום העבודה:

- ביצוע shut down מלא למחשב.
- כיבוי מכשירי המדידה.

2. במידה והתקבלה בחלון סביבת IAR אחת ההודעות:

"Failed to initialize"

"Communication error"

נתק למשך 5 שניות את החיבור בין שני כבלי ה-USB (מאחורי ערכת הפיתוח של MSP430).

B. חומר עזר:

❖ Basic Timer1

בקובץ מעבדה MSP430x4xx user guide עמודים 425 – 433

Tutorial 5.1 (חומר כתוב + וידאו).

❖ :Advanced Timers

1. קובץ הכנה Tutorial No.6 (חומר כתוב + וידאו).

2. Timer B - קריאה מקדימה

בקובץ מעבדה MSP430x4xx user guide עמודים 473 – 498

3. חומר עזר עבור ערכת פיתוח אישית במודל הנמצא תחת לשונית "Personal Evaluation Kit"

C. חלק תיאורטי – Basic Timer (משפחה MSP430x4xx):

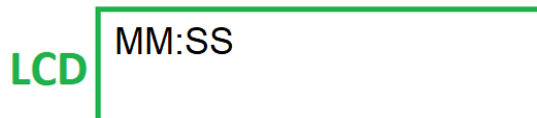
1. רשום את ערך רגיסטר BTCTL לצורך ביצוע פסיקה במרווחי זמן של 1sec.
 2. מהו ערכם של הרגיסטרים BTCNT1 ו- BTCNT2 לאחר שנייה זו.
 3. הסבר את שני אופני העבודה של טיימר Basic Timer1. ואת היתרון של כל אחד מהם.
 4. רשום את ערך רגיסטר BTCTL לצורך ביצוע פסיקה במרווחי זמן של 8usec (הכי קרוב שאפשר).
 5. הסבר כיצד פועלת פסיקת Basic Timer1 ועל הצורך בה.
 6. באופן העבודה של שני טיימרים נפרדים בגודל 8-bit.
- נתונים: $clk1 = ACLK, clk2 = SMCLK, BTIP = 0x7$
- חשב והסבר, בכמה מתקדם הערך BTCNT1 בין פסיקות עוקבות של הטיימר.

D. חלק תיאורטי – Advanced Timers (משפחה MSP430x4xx):

1. מנה מס' אפליקציות שניתן לבצע בעזרת Timer_B.
2. מנה את אופני העבודה של Timer_B.
3. מה משמעות ומטרת השימוש של אופן **Capture** ואופן **Compare**.
4. הסבר מהי מטרת יחידת **Output Unit** ומה היתרון שלה?
5. הסבר רעיונית, כיצד ניתן למנות תדר לא ידוע של שעון חיצוני המחובר למעבד?
6. כאשר Timer_B מוזן ע"י ACLK (32768Hz) והוא מקונפג ל-**Compare mode** מהו ערך הרגיסטר **TBCCR0** לצורך אפשרור פסיקה פעם בשנייה.
7. פרט והסבר שני הבדלים פונקציונאליים בין TimerA במשפחה MSP430x2xx לבין TimerB במשפחה MSP430x4xx

E. חלק מעשי – כתיבת קוד מערכת פורטאבילי בשפת אסמבלי:

- את הלחצנים PB1, PB0 נדרש לחבר לרגלי הבקר P2.0, P2.1 בהתאמה.
 - את מסך ה- LCD נדרש לחבר לפורט P1 עבור D0-D7 ואת שלושת קווי הבקרה לרגליים P2.5, P2.6, P2.7
- הערה:** חומר עזר בנושא LCD נמצא תחת לשונית LAB3.
- ארכיטקטורת התוכנה של המערכת נדרשת להיות מבוססת **Simple FSM** (כמתואר בדו"ח מכין 4, סעיף E) המבצעת אחת מתוך ארבע פעולות בהינתן בקשת פסיקה חיצונית של לחיצת לחצן מתוך שלושת הלחצנים.
 - **קוד המערכת נדרש להיות מחולק לשכבות** (כמתואר בדו"ח מכין 4, סעיף D) **כך שהוא יהיה נייד (portable) בקלות בין משפחות MSP430x2xx, MSP430x4xx ע"י החלפת שכבת ה-BSP בלבד.**
- המשמעות:** קוד המערכת נדרש לרוץ על ערכת הפיתוח האישית וגם על ערכת הפיתוח במעבדה (**בניסוי מעבדה 5 בסמסטר זה, קוד המערכת ייבחן על גבי ערכת פיתוח אישית בלבד**). ליידע כללי, בגישה זו רוב שלבי הפיתוח ייעשו על גבי הערכת פיתוח האישית ובדיקת שלב התאמת ה-BSP ייעשה על ערכת הפיתוח במעבדה.
- כתיבת פונקציות ה-driver של ה-LCD צריכות להיות ממוקמות ב-HAL בעוד שפונקציה לכתיבת מחרוזת המבוססת עליהן צריכה להיות ממוקמת בשכבת ה-API.
 - טרם שלב כתיבת הקוד נדרש לשרטט גרף דיאגרמת FSM **מפורטת** של ארכיטקטורת התוכנה של המערכת ולצרפה לדו"ח מכין. המצבים אלו הצמתים והקשתות אלו המעברים ממצב למצב בגין בקשות פסיקה.
 - משלב זה ואילך, אסור לבצע השהייה ע"י שימוש ב-polling למעט עבור debounce ברוטינת שירות של **בקשות פסיקה בגין לחצנים.**
 - **להלן דרישת מצבי המערכת:**
 - (state=idle=0): הבקר נמצא/חוזר למצב שינה (Sleep Mode).
 - בלחיצה על לחצן PB0 (state=1):
- נדרש לבצע על גבי מסך LCD שעון התראה של שתי דקות במניית שניות (המצב ההתחלתי בכניסה למצב הוא 00:00), הקפידו על רמת הדיוק המנייה ועל מיקום המנייה על מסך ה-LCD כמופיע באיור הבא:

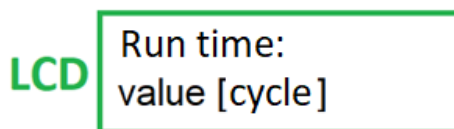


המצב מוגדר להסתיים בהגעה לערך מנייה 02:00, מצב זה נדרש לאפשר לחתוך ע"י שאר הלחצנים, במשפחה MSP430x2xx המנייה חייבת להתבצע בשימוש פסיקות מודול Timer_A0 בלבד

בלחיצה על לחצן PB1 (state=2):

נדרש לממש cycle counter מבוסס תכונת input capture למדידת זמן ריצה של קטע קוד נתון מראש. לצורך כך נדרש להגדיר שתי פונקציות בשכבת ה-APP בשם tic, toc אשר משתמשות בפונקציות משכבת ה-HAL שמייצרות event פנימי (בשימוש אפשרויות VCC, GND של ערוץ טיימר במצב input capture) ומחזירות ערך capture של ליבת טיימר בגין אותו event (מצב ה capture מוגדר להיות on both – עלייה וירידה). פונקציית tic דוגמת ערך התחלתי יחסי של השעון ופונקציית toc דוגמת ערך סופי יחסי של השעון וכוללת בתוכה את הדפסת זמן הריצה הנמדד למסך ה LCD ביחידות [cycle] של שעון $2^{20} Hz$.

שלב הפיתוח: בחרו קטע קוד שרירתי, עטפו אותו בפונקציות tic, toc והשוו את התוצאות המתקבלות בעזרת מכשיר סקופ (כתיבת פקודת xor לרגל GPIO פנויה לפני tic ואחרי toc כדי לבדוק את רמת הדיוק, מדידת רוחב הפולס הנוצר מהווה את זמן הריצה של הקוד)



המצב מוגדר להסתיים בלחיצה על לחצן המשויך למצב אחר. המימוש נדרש להיות מבוסס אופן עבודה Input Capture, במשפחה MSP430x2xx של Timer_A1.

F. הבהרות:

נדרש לארגן את הקוד בצורה מסודרת בלפחות שני קבצים ולהפריד בין קובצי המקור של הרוטינות והתוכנית הראשית (main).

ערך תדר ברירת המחדל של שעון MCLK הוא:

$$f_{MCLK} = 32 \cdot 32768 = 2^{20} = 1,048,576 \text{ Hz} \rightarrow T_{MCLK} = \frac{1}{2^{20}} \approx 0.954 \mu\text{sec}$$

G. צורת הגשה דוח מכין:

- הגשת מטלת דוח מכין תיעשה ע"י העלאה למודל של תיקיית zip מהצורה **id1_id2.zip** (כאשר $id1 < id2$), רק הסטודנט עם הת"ז id1 מעלה את הקבצים למודל.
- התיקיה תכיל את שני הפרטים הבאים בלבד:
 - ✓ קובץ pre_labx.pdf – מכיל תשובות לחלק תיאורטי דו"ח מכין
 - ✓ תיקייה בשם IAR - מכילה את קובצי המקור בלבד (קבצים עם סיומת *.s43) של מטלה מעשית דוח מכין.

בהצלחה