

F. חלק מעשי – כתיבת קוד מערכת פורטאבילי :

כיול חומרה :

1. רמת הדיוק של המערכת תלויה בידיעת ערך התדר של שעון SMCLK המזין את הרכיבים הפריפריאליים של הבקר. לצורך מדידת תדר SMCLK נדרש לאפשר אותו כמוצא לרגל P1.4 (בעזרת קוד + קובץ בינארי לצריבה הנמצא בקישור הבא: [SMCLK output](#)) ולאחר מכן לחברו לערוץ בסקופ הנמצא בעמדות המעבדה (רוחב פס של הסקופ הנמצא בערכה לא מאפשר מדידת תדרים של MHz ומעלה), לכוון את הטריג'ר לתוך טווח המתחים של האות ולאחר מכן למצע את האות הנמדד בסקופ ע"י הבחירה הבאה:

א. לחץ על לחצן Acquire

ב. בתחתית המסך בחר את המלבן השמאלי ביותר בשם Acquire Menu

ג. בחר באפשרות של Averaging

ד. לאחר ביצוע Stop ניתן לקחת את ערך התדר הנמדד (ערך המדידה יציב עד כדי ערך אחדות ה-kHz)

2. מתח Vcc בערכת הפיתוח האישית נע בין 3.5v-3.65v (בשונה מערכת הפיתוח במעבדה Vcc=3.3v) **והוא תלוי לאיזה מחשב מחוברת ערכת הפיתוח (קיימת שונות של מתח ה-Vcc בחיבור בין מחשבים שונים).**

כדי לבדוק את ערך Vcc במדויק, ניתן למדוד רמת מתח בעזרת DMM באחת מהדרכים הבאות:

- רמת מתח באחד מרגלי SW0-SW3 כאשר הם במצב 1 לוגי
- רמת מתח באחד מרגלי PB0, PB1

חיבורי חומרה של המערכת:

1. מתג SW0 מחובר לרגל P2.0
2. לחצנים PB0 - PB2 מחוברים לרגלי הבקר P1.0 – P1.2 בהתאמה
3. כניסה אנלוגית של הבקר P1.3 (כניסה A3) נדרש לחבר לפוטנציומטר (רגל POT על גבי הכרטיס, ראו סעיף G)
4. מסך LCD נדרש לחבר את D7-D4 לרגליים P1.4-P1.7 בהתאמה (אופן עבודה של ה-LCD בארבע סיביות של מידע) + שלושת קווי הבקרה של ה-LCD לרגליים P2.5, P2.6, P2.7 (קוד עבור LCD נתון במודל, עליכם לעדכן לפי צרכיכם).
5. מוצא Generator מחובר לרגל P2.4 במצב של (Input capture)
6. רגל P2.2 במצב של Output compare (PWM) מחוברת לערוץ הסקופ CH2 ול-Buzzer (במקביל)

ארכיטקטורת תוכנה של המערכת:

1. ארכיטקטורת התוכנה של המערכת נדרשת להיות מבוססת *Simple FSM* (כמתואר בדו"ח מכין 1) המבצעת אחת מתוך ארבע פעולות בהינתן בקשת פסיקה חיצונית של לחיצת לחצן מתוך שלושת הלחצנים.
2. קוד המערכת נדרש להיות כתוב בצורה גנרית ומחולק לשכבות אבסטרקציה 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. בגישה זו רוב שלבי הפיתוח ייעשו על גבי הערכת פיתוח האישית ובדיקת שלב התאמת ה-BSP ייעשה על ערכת הפיתוח במעבדה.

3. 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.
4. משלב זה ואילך, אסור לבצע השהייה ע"י שימוש ב poling (כפי שבוצע בניסוי LAB1) למעט עבור debounce ברוטינת שירות של בקשות פסיקה בגין לחצנים.
5. מאחר והבקר אינו מכיל יחידת FPU (Floating Point Unit) שימוש בטיפוסים float , double פוגעת בביצועי המערכת ולכן אסור, על כן עבור שימוש במספרים שאינם שלמים נדרש להשתמש ב- Q Format (כפי שנלמד בקורס מבוא), ראו חומר עזר הנמצא באתר הקורס תחת לשונית LAB2 פריט בשם: HW Emulators and Q-Format
- שימו לב:** שימוש ב Q-Format בשפת C לא מצריך מימוש שלכם עבור קוד אמולציה לפעולות כפל וחילוק מאחר והקומפיילר מבצע את האמולציה עבורכם (ניתן לראות את התרגום בחלון disassembly).

דרייברים שכבת ה HAL תוספות לתמיכה במערכת:

1. כתיבת פונקציות ה driver של ה LCD צריכות להיות ממוקמות ב HAL בעוד שפונקציה לכתיבת מחרוזת המבוססת עליהן צריכה להיות ממוקמת בשכבת ה API .
2. מצב 1 של המערכת מבוסס תכונת Input capture , הגדירו לכך דרייבר מתאים.
3. מצב 2 של המערכת מבוסס תכונת Basic Timer , הגדירו לכך דרייבר מתאים.
4. מצב 3 של המערכת מבוסס תכונת Output compare , הגדירו לכך דרייבר מתאים.

דרישת מצבים של גרעין ההפעלה במערכת:

מצב ברירת מחדל (state=idle=0):

בלחיצת RESET או בסיום ביצוע כל המצבים, הבקר נמצא/חוזר למצב שינה (Sleep Mode).

בלחיצה על לחצן PB0 (state=1):

נדרש לממש counter (מונה תדר) למדידת תדר אות שעון חיצוני f_{in} המוזן ממחולל האותות לרגל הבקר P2.4 (ובערכת הפיתוח במעבדה P2.3) את ערך התדר הנמדד יש להציג על גבי מסך LCD, לפי הפירוט הבא:
 הצגת התדר ביחידות של Hz בצורה **דינאמית** (ללא הצגת היסטוריית המדידות), כאשר ערך המדידה מתעדכן נדרש לעדכן רק את שדה value ולא לרענן את כל המסך (שיקולים הנדסיים של ביצועים וצריכת הספק).
 התדר המחושב יוצג על גבי מסך ה- LCD, עבור תחום $f_{in} \in [50\text{Hz} - 20\text{kHz}]$ ברמת דיוק של תדר שלם

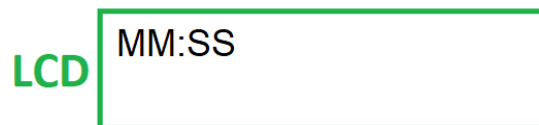
LCD $f_{in} = 19349 \text{ Hz}$

- ✓ המימוש יהיה לפי השיטה המבוססת על input capture בלבד, לצורך השגת זמן ריענון ודיוק מקסימאליים (קבעו את שעון המזין את הטיימר להיות SMCLK).
- ✓ פשטו את הביטוי לחישוב f_{in} להיות מינימאלי, מהו הביטוי? כמה מחזורי שעון אורך החישוב? השוו את התוצאה למדידת תדר בעזרת מכשיר ה-scope?
- ✓ סינון רעשים (כלומר, במידה והתדר לא משתנה המרחק הנמדד לא ישתנה).

המצב מוגדר להסתיים בלחיצה על לחצן המשויך למצב אחר. המימוש נדרש להיות מבוסס אופן עבודה Input Capture, במשפחה MSP430x2xx של Timer_A1. לפני חיבור המחולל לערכה, וודא שהאות במוצא המחולל הוא גל ריבועי עם ערכי קצוות 0v-3v. הקפידו לחבר את המחולל בקוטביות נכונה, בננה שחורה ל-GND ובננה אדומה לפין הרצוי.

בלחיצה על לחצן PB1 (state=2):

נדרש לבצע על גבי מסך LCD שעון עצר של מניית שניות מבוסס צורת מנייה של Up/Down Mode. תחילת מנייה של שעון העצר כאשר SW0='1' ועצירת מנייה כאשר SW0='0' (בכניסה למצב ערך המנייה מאותחל 00:00, במקרה של הפעלת שעון עצר מספר פעמים המנייה ממשיכה מהערך הקודם). הקפידו על רמת הדיוק המנייה ועל מיקום המנייה על מסך ה-LCD כמופיע באיור הבא:



המצב מוגדר להסתיים בלחיצה על לחצן המשויך למצב אחר. במשפחה MSP430x2xx המנייה חייבת להתבצע בשימוש פסיקות מודול Timer_A0 בלבד.

בלחיצה על לחצן PB2 (state=3):

נדרש לממש מחולל Tones על בסיס אות כניסה v_{in} של הפוטנציומטר עם ערכי קצוות 0v-3.3v נדרש להוציא מרגל הבקר P2.2 המחוברת ל Buzzer את ריבועי בתדר f_{out} באופן הבא:

$$f_{out}: [ADC \text{ conversion values}] \xrightarrow{\text{linear transformation}} [1kHz - 2.5kHz]$$

$$(1) \text{ Physical equation: } f_{out} = f(N_{ADC}) = 1kHz + 1.5kHz \cdot \frac{N_{ADC}}{0x3FF - 0x000}$$

$$(2) \text{ MCU (DSP) equation: } N_{f_{out}} = f(N_{ADC}) = m \cdot N_{ADC} + n$$

הסבר: נדרש לחשב ידנית את הערכים m, n עבור חישוב בזמן אמת בקוד המערכת של ערך $N_{f_{out}}$ עבור כל שינוי

בערך דגימה N_{ADC} מהפוטנציומטר. (מצ"ב קישור לקוד Python המכיל בנייה וחישוב של משוואה 2 ואת

השחזור של משוואה 1 כולל הגרפים שלהם)

המצב מוגדר להסתיים בלחיצה על לחצן המשויך למצב אחר.