תוכן עניינים:

DMA (Direct Memory Access)

A. נושאי המעבדה:

בניסוי מעבדה זה נעסוק בנושא DMA.

באופן כללי בעזרת מודול DMA נוכל להעביר בלוקי מידע **מ**מודלי-חומרה (peripheral hardware) או מהזיכרון **אל** מודלי-חומרה או לזיכרון וזאת ללא שימוש ב- CPU. מאחר ומבחינה סטטיסטית העברת מידע רב היא בעלת התנהגות לוקאלית (המידע מתפרס על פני ערכי כתובות סמוכים) שימוש ב- CPU מיותר ו"גוזל" זמן והספק יקרים הנדרשים לצורך **ייבוא נתונים**.

בבקר MSP430 ישנו מודול DMA אחד בעל שלושה ערוצים שונים בלתי-תלויים, כאשר בו זמנית ניתן להפעיל רק אחד מהם. קובצי קוד לדוגמה בשימוש DMA נמצאים במודל.

B. חומר הכנה למעבדה:

- סיכום הנמצא ב- MOODLE של הפרקים 22,23 בספר הבקר "KL25 Sub-Family User Manual" בנושאים DMA Controller, DMAMUX.
- להפעיל את הקוד לדוגמא ולהבין אותו לעומק.
- על בסיס הנ"ל לבצע את מטלת המעבדה.

C. שאלות הכנה – KL25Z128VLK4 DMA module:

מודול DMA :

1) הסבר/י בפירוט את המוטיבציה לשימוש ב DMA ביחס ל- CPU לצורך העברת נתונים (מהו ה Tradeoff).
2) הסבר/י בפירוט את שיטות המעון, רשמו דוגמה מתאימה עבור כל אחת מהשיטות.
3) הסבר/י בפירוט את השיטות להעברת מידע בשימוש DMA, רשמו דוגמה מתאימה עבור כל אחת מהשיטות.
4) הסבר/י בפירוט את המושג circular buffer. האם השדות SSIZE, DSIZE חייבים להיות שווים ומה השפעתם על ההתנהגות של circular buffer.
5) הסבר/י את תפקידו של ביט DMA_DSR_BCRn[DONE] ומה שימושו.
6) הסבר/י באילו תנאים תתבצע בקשת פסיקה של מודול DMA.
7) הסבר/י את תפקיד ביט DMA_DCRn[LINKCC]. רשום/י דוגמה לצורך בשימוש ביט זה.

מודול DMAMUX0:

1) הסבר/י באופן כללי, מה משמש המודול DMAMUX0.
2) מנה/י את אופני העבודה והסבר/י את יעודם.
3) סווג/י את סוגי המקורות בכניסה למודול DMAMUX0 ואת ייעודם.

D. תרגול בעזרת קוד לדוגמא:

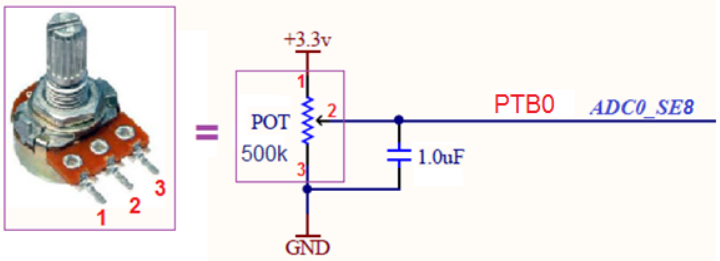
בדוגמה זו, נעביר מידע ממודול ADC0 ישירות למערך נתונים הנמצא בזיכרון RAM.

בעזרת מודול ADC0 נדגום מתח אנלוגי מפין PTB0 בשימוש נגד משתנה (מדידת מתח אנלוגי בתחום של 0v - 3.3v דרך רגל הבקר כמופיע באיור) ואת ערך הדגימה נעביר בעזרת DMA לתא במערך בשם value (בגודל 8 תאי int).

- תזמון קצב ביצוע הדגימות יהיה **דגימה אחת לשנייה**, בעזרת שעון PIT. הטריגר לביצוע הדגימה הוא בתוכנה בעזרת כתיבה לרגיסטר ADC0_SC1A בתוך ISR של ה- PIT. לצורך חיווי ביצוע הטריגר נבצע פעולת הדלקה לסירוגין בין שני הלדים הירוק והאדום, בכל כניסה ל- ISR של ה- PIT.
- דגימת ADC0 היא באורך 16 ביט.
- ה- CPU מבצע את ה- ISR של DMA0 בכל פעם שערך BCR מגיע ל-0 **(**למעשה בסיום כל דגימה. ניתן לבצע דגימות ללא טריגר תוכנה אלא בעזרת טריגר חומרה כפי שלמדתם בפרק ADC, בצורה זו ניתן לגרום ל- CPU להיכנס ל- ISR של DMA0 כל n דגימות**)**, במיוחד שאופן העבודה הוא DMA_DCR0[CS]=1.
- בסיום רצף של שמונה דגימות המועברות ממודול ADC0 לזיכרון בעזרת DMA ה- CPU נכנס לפעולה ומבצע מיצוע של שמונה הדגימות ומכניס לתוך משתנה **avg**.

- מדידת המתח תיעשה <u>בצורה הבאה</u>:

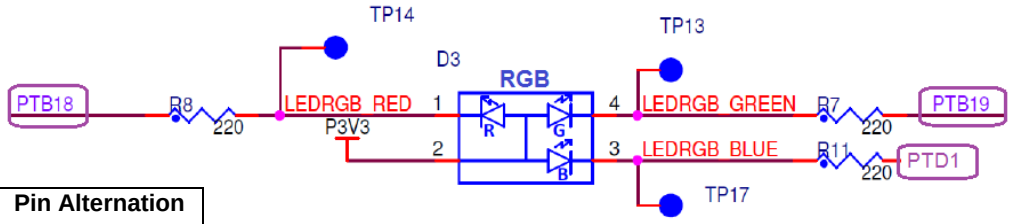
נחבר נגד משתנה של 500kΩ (כמתואר באיור). ההתנגדות בין רגליים 1,3 היא קבועה וערכה הוא 500kΩ. ההתנגדות בין הרגליים 2,3 **או** 2,1 היא התנגדות משתנה ואפשרית בטווח של 1Ω - 500kΩ כתלות בסיבוב הזרוע. סכום ערכי 2 ההתנגדויות המשתנות הוא 500kΩ. כלומר: $R_{1,3} = R_{2,1}+R_{2,3}$



- מעבר לתחילת התוכנית ניתן ללחוץ על לחצן RESET.

<u>תזכורת:</u>

האיור הבא מתאר את חיבור **RGB led** בכרטיס הבקר.



RGB LED	KL25Z128	Pin Alternation
Red Cathode	PTB18	ALT3 = **TPM2_CH0**
Green Cathode	PTB19	ALT3 = **TPM2_CH1**
Blue Cathode	PTD1	ALT4 = **TPM0_CH1**

E. חלק מעשי – כתיבת קוד המערכת (באופן גנרי ופורטאבילי):

הקדמה:

בניסוי זה, נממש מערכת בעלת יכולת של הקלטת צלילים בעזרת Keypad והשמע נגינה דרך Buzzer (זמזם).
כידוע לכם, זמזם זהו רכיב אלקטרוני אשר כניסתו היא אות ריבועי בתדר משתנה ומוצאו צליל (tone) בתדר הכניסה. תדר הצליל נובע מרעידות של ממברנה (קול נוצר כאשר גוף מתנודד מהר, כך שכתוצאה מתנודתו נוצר גל מתקדם בתווך החומרי) בתדר אות הכניסה בדומה לפריטה על מיתר של כלי נגינה.
כשפורטים על מיתר של גיטרה למשל, תדר התנודות הוא פונקציה של אורך המיתר, עוביו, מתיחותו וסוג החומר ממנו הוא עשוי. כמו בכלי נגינה אנו נוכל בעזרת שינוי תדר אות הכניסה של הזמזם (אות PWM עם Duty Cycle = 50%) לשלוט על הצליל היוצא ממנו, וכך להפיק מנגינה מבוססת tones כרצוננו.
הערה: רוחב הפס של תדרי הקול אותם מסוגל לשמוע האדם נעים בתחום 20Hz – 20kHz

דרישת חיבורי החומרה:

- הלחצנים PB0 - PB2 מחוברים לרגלי הבקר P1.0 – P1.2 בהתאמה
- מסך LCD נדרש לחבר את D7-D4 לרגליים P1.7-P1.4 בהתאמה (אופן עבודה של ה- LCD בארבע סיביות של מידע) + שלושת קווי הבקרה של ה- LCD לרגליים P2.7, P2.6, P2.5 (*קוד עבור LCD נתון במודל, עליכם לעדכנו לצרכיכם*).
- רגל P2.2 במצב של Output compare (PWM) ל- Buzzer
- חיבור ה- Keypad לרגלי PORT כלשהוא לבחירתכם תחת שיקולים הנדסיים (חיבור ישירות לכרטיס KL25Z) ואת רגל הפסיקה שלו לרגל P2.1 (ראו נספח בנושא Keypad)
- 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.
- **קוד המערכת נדרש להיות מחולק לשכבות כך שהוא יהיה נייד (portable) בקלות בין משפחות MSP430 ע"י החלפת שכבת ה- BSP בלבד.**
- **כתיבת פונקציות ה driver של ה LCD וה- Keypad צריכות להיות ממוקמות ב HAL בעוד שפונקציה לכתיבת מחרוזת / קלט מחרוזת המבוססת עליהן צריכות 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.**
- **תזכורת: אסור לבצע השהייה ע"י שימוש ב poling למעט עבור debounce ברוטינת שירות של בקשות פסיקה בגין לחצנים.**

דרישת מבנה המערכת מבוססת גישת Simple FSM:

נדרש להגדיר ארבעה מערכים סטטיים בשכבת ה main מטיפוס int בשמות ***song1, song2, song3*** המאחסנים את רצף הטונים לניגון שלושת המנגינות הנתונות בנספח Tones based music ומערך סטטי נוסף ***recorder*** לאחסון הקלטת מנגינה המנוגנת ע"י המשתמש דרך ה- Keypad.

- התווים בארבעת המערכים יהיו מיוצגים ע"י תדר הגל הריבועי המרכיב את התו.
- משך זמן ניגון כל תו נגינה הינו 345msec .
- גודל כל אחד מהמערכים הנ"ל נדרש להיות נפרד ומנוהל בצורה גנרית בשימוש *#define*. גודל מערכי $song_i$ יקבעו לפי מספר התווים לכל מנגינה המפורטים בנספח. גודל מערך *recorder* הוא 32.

(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 tone recorder . נדרש להדפיס בקשה למשתמש דרך ממשק מתאים על גבי מסך ה- LCD , לנגן (באמצעות הקלדה) עד מקסימום 32 תווים דרך ה- Keypad (הסבר בנספחים מצורפים). במהלך הנגינה נדרש לבצע חיווי למשתמש באמצעות ה- Buzzer ולאחסן במערך *recorder* את תווי המנגינה שמכניס המשתמש ובסיום ההקלטה להדפיס הודעה מתאימה למשתמש הכוללת את כמות התווים שהוקלטו.

הערה: המצב מוגדר להסתיים בלחיצה על לחצן המשויך למצב אחר.

בלחיצה על לחצן PB1 (state=2):

נדרש לממש Audio player. נדרש להדפיס בקשה למשתמש דרך ממשק מתאים על גבי מסך ה- LCD , לבחירת מנגינה מתוך ארבעת המערכים ***song1, song2, song3, recorder*** באמצעות ה- Keypad (שורה לכל מנגינה כאשר גלילת שורות במסך תיעשה באמצעות PB2).

נגינת השיר תתבצע העברת בלוק מידע ממערך בזיכרון RAM למודול TimerB (להפקת אות מוצא ריבועי) בעזרת DMA, **ללא שימוש ב- CPU** כאשר אופן DMA Transfer Mode נתון לבחירתכם תחת ביצוע מטבי של דרישות המשימה כלומר, העברת המידע תהיה בנפרד תא אחר תא במערך לרגל P2.2 בעזרת מודול TimerB **בשימוש DMA** ותזמון המעבר יהיה בעזרת טריגר של מודול PIT במרווחי זמן של 345ms.

הערה: המצב מוגדר להסתיים בלחיצה על לחצן המשויך למצב אחר.

F. 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.

- התיקייה תכיל את שני הפרטים הבאים בלבד:
  - ✓ **קובץ pre_lab$_x$.pdf** – מכיל תשובות לחלק תיאורטי דו"ח מכין
  - ✓ תיקייה בשם **CW** - מכילה שתי תיקיות, אחת בשם ***Sources*** של קובצי source (קבצים עם סיומת **.c***) והשנייה בשם ***Project_Headers*** של קובצי header (קבצים עם סיומת **.h***).

G. צורת הגשה דוח מסכם:

- 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.

- התיקייה תכיל את שני הפרטים הבאים בלבד:
  - ✓ **קובץ final_lab$_x$.pdf** – מכיל תיאור והסבר לדרך הפתרון של מטלת זמן אמת.
  - ✓ תיקייה בשם **CW** - מכילה שתי תיקיות, אחת בשם ***Sources*** של קובצי source (קבצים עם סיומת **.c***) והשנייה בשם ***Project_Headers*** של קובצי header (קבצים עם סיומת **.h***).

בהצלחה