

תוכן עניינים:

2	A. הקדמה:
3	B. חומר הכנה למעבדה:
3	C. שאלות הכנה:
3	D. תרגיל קוד לדוגמה:
3	E. חלק יישומי – כתיבת קוד (על גבי ערכת הפיתוח האישית מבוססת בקר KL25Z):
5	F. צורת הגשה דוח מכין:
5	G. צורת הגשה דוח מסכם:

Universal Asynchronous Receiver/Transmitter (UART0)

A. הקדמה:

בתרגיל זה עליכם ליצור תקשורת בין מחשב PC ולבין בקר מסוג MKL25Z128 (בערכת פיתוח אישית).

עליכם להגדיר בכל צד (מחשב, בקר) את ההגדרות המתאימות לצורך ביצוע התקשרות בניהם.

באופן פיזי, חיבור התקשורת בין המחשב לבקר נעשה דרך חיבור USB הקיים בערכה (דרכו אתם מתחברים לסביבת הפיתוח IAR לצורך פיתוח קוד המערכת – Active application , Debug). באופן כללי, ישנו צ'יפ על הכרטיס המבצע המרה מפרוטוקול USB לפרוטוקול טורי JTAG עבור תמיכה בפעולת Debug והמרה לפרוטוקול טורי RS-232 עבור תמיכה בתקשורת טורית אסינכרונית עם הבקר בשימוש רכיב UART פריפריאלי (בעזרת פרוטוקול המגדיר את שכבת Data-Link) ובצד המחשב בעזרת פרוטוקול RS-232 (מגדיר את השכבה הפיזית).

באופן פרטני עבור בקר מסוג MKL25Z128, דרך חיבור USB מתבצעת בפועל תקשורת מול המחשב דרך חיבור פנימי

לרגלי הבקר PTA1-RX, PTA2-TX – אין לכם צורך לחבר כלום מלבד כבל USB רק זכרו שרגלי הבקר , P1.1

P1.2 תפוסים עבור התקשורת).

- **צד בקר** - התקשורת בצד הבקר תהיה בעזרת מודול UART, הקוד יהיה מבוסס interrupt driven בשידור וקליטה. כלומר עליכם לכתוב תוכנית שתומכת בשידור וקליטה בשימוש פסיקות בלבד (ראו באתר המודל קודים לדוגמה תחת לשונית *Personal Evaluation Kit*).
- **צד מחשב** - בעזרת כתיבת אפליקציה על גבי המחשב המבוססת קוד בשפה עילית לבחירתכם, כגון: C#, MATLAB, Python, C++, C, JAVA, etc (במודל נתון לכם שלושה קובצי קוד לדוגמה מבוסס Python – **צד בקר + צד מחשב**). באופן כללי, בכל לחיצת תו במקלדת המחשב, נשלח תו מהמחשב לבקר.

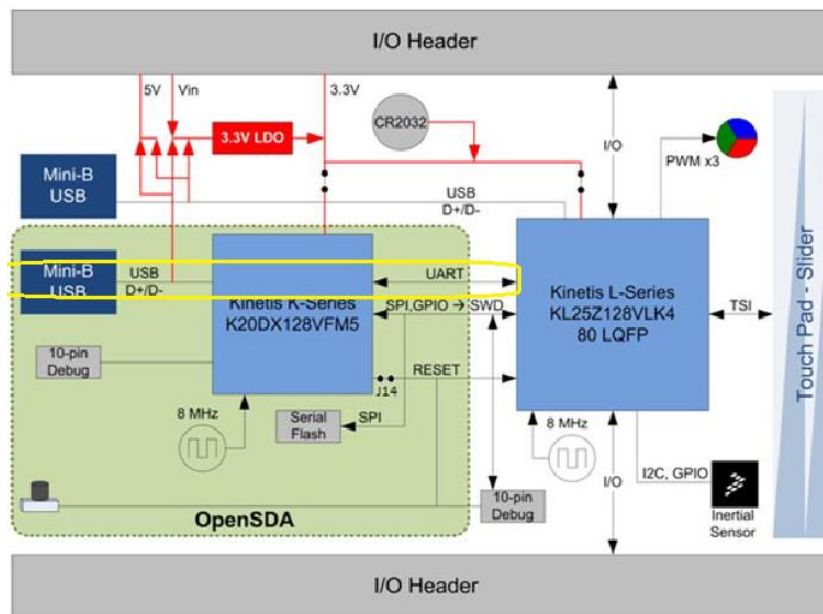


Figure1 – USB to UART adaptation

B. חומר הכנה למעבדה:

1. הרצאה פרונטלית תיאורטית ע"י מרצה הקורס.
2. בקובץ כרטיס הבקר "FRDM-KL25Z User's Manual" לקרוא את כל Chapter 39.
3. להפעיל את הקוד לדוגמא ולהבין אותו.
4. חידוד הנושאים בשעות הקבלה של המדריכים.
5. על בסיס הנ"ל לבצע את מטלת המעבדה.

C. שאלות הכנה:

1. הסבר את אופן הפעולה של הרכיב הפריפריאלי UART ומהי מטרת שימוש.
2. הסבר את השוני בין UART ו- RS-232 וכיצד כל אחד מהם מתאים למודל שבע השכבות.
3. מהי מטרת השימוש ב- Parity Bit וכיצד מטפלת בכך המערכת.
4. הסבר את המושגים Baud Rate ו- Modulation.
5. במצב של קליטה, כיצד קובעת המערכת את ערכו של כל ביט במידע שמתקבל.
6. הסבר ופרט את מבנה ופעולת בקר הפסיקות עבור קליטה ושידור.
7. עבור 9600-8N1 (שמשמעותו '8' סיביות מידע - 'N', ללא סיבית זוגיות, '1' - אורך סיבית הסיום היא (משך הזמן של סיבית יחידה בתדר 9600bps). רשום את הרגיסטרים וערכם לצורך ביצוע דרישות אלו.

D. תרגיל קוד לדוגמה:

בצד המחשב נכתוב אפליקציה מבוססת פרמטרים 9600-8N1. בצד הבקר נעבוד באופן עבודה UART0 ונחבר את הבקר למחשב דרך כבל USB (בחיבור openSDA).
בלחיצה על לחצן PTD7 תצא הודעה לביצוע Echo. בלחיצת תו במקלדת המחשב יופיעו 2 תווים (אחד שנשלח מהמחשב לבקר ואחד שנשלח בחזרה מהבקר למחשב).
הסבר: בכל לחיצת תו במקלדת המחשב נשלח תו מהמחשב לבקר. בקליטת תו זה בבקר, הבקר שולח את התו הנקלט חזרה למחשב.

E. חלק יישומי – כתיבת קוד (על גבי ערכת הפיתוח האישית מבוססת בקר KL25Z):

- ארכיטקטורת התוכנה של המערכת נדרשת להיות מבוססת **Simple FSM** המבצעת קטע קוד השייך לאחד ממצבי המערכת בהינתן בקשת פסיקה המגיעה ממחשב PC לבקר דרך ערוץ התקשורת למודל UART.
- קוד המערכת נדרש להיות מחולק לשכבות כך שהוא יהיה נייד (portable) בקלות בין משפחות KL25Z ע"י החלפת שכבת ה- **BSP בלבד**.
- 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.
 - אסור לבצע שהייה ע"י שימוש ב **poling** למעט עבור **debounce** ברוטנית שירות של בקשות פסיקה בגין לחצנים.
 - בתחילת התוכנית, הבקר נמצא במצב שינה.

חיבורי חומרה נדרשים:

- חיבור לד RGB: **R, G, B**
- חיבור POT (פוטנציומטר)
- מסך LCD נדרש לחבר את D7-D4 (אופן עבודה של ה-LCD בארבע סיביות של מידע) + שלושת קווי הבקרה של ה-LCD

עליכם לבצע תקשורת בין הבקר ולבין ה-PC, תחת הדרישות הבאות:

1. בלחיצת מספר במקלדת המחשב של השורה המתאימה בתפריט, תתבצע הפעולה המתאימה בבקר
2. להלן הגדרת התפריט להדפסה על מסך המחשב
3. כל שורה בתפריט מהווה מצב במערכת כך שהמערכת מכילה שמונה מצבים שונים כאשר ערך כל מצב הוא מספר שורה בתפריט. הקשתות בגרף המערכת בגיין מתקבלת החלטה למעבר ממצב אחד למצב אחר נעשית בשכבת ה-HAL ב-ISR של RX עקב קלט מידע (מידע נקלט מכונה Command לצורך ברירת מצב המערכת). בכניסה למצב, המידע הנקלט משמש כ Data למצב הנמצא בביצוע. בהתאם לאמור לעיל, יש צורך לסווג את המידע הנקלט ל Command ול Data (כתחליף לשימוש בלחצנים כפי שבצעתם בניסויים LAB1-LAB3).

Menu

1. Blink RGB LED, color by color with delay of X[ms]
2. Count up onto LCD screen with delay of X[ms]
3. Count down onto LCD screen with delay of X[ms]
4. Get delay time X[ms]:
5. Potentiometer 3-digit value [v]
6. Clear LCD screen
7. Show menu
8. Sleep

הסבר סעיפי התפריט:

נתון שערך ברירת המחדל של X הוא 500ms (כלומר X=500).

1. נדרש להבהב את לד RGB צבע אחר צבע עם השהיה של X בין כל החלפת צבע.
2. מנייה מעלה בעזרת משתנה מטיפוס *int* תוך שמירת ערך המנייה במעבר בין מצבים (ראו הערה בסעיף 6)
3. מנייה מטה בעזרת משתנה מטיפוס *int* תוך שמירת ערך המנייה במעבר בין מצבים (ראו הערה בסעיף 6)
4. קליטת ערך X מהמשתמש (היחידות של המספר X הן ms). בלחיצת 4 בתפריט נדרש לקלוט מהמשתמש מחרוזת ספרות (המייצגת את X) כאשר תו הסיום ENTER .
5. מדידת ערך מתח מרגל הפוטנציומטר **בצורה דינאמית**, בדיוק שלוש ספרות (ביחידות של volt).
6. ניקוי המסך + אתחול משתני ערך המניות של סעיפים 2,3 לערכים 0, 65535 בהתאמה
7. הצגת התפריט על גבי מסך צד מחשב
8. כניסה למצב שינה (המוגדר ב idle state)

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 **שלושת** הפרטים הבאים בלבד:
 - ✓ קובץ **pre_labx.pdf** – מכיל תשובות לחלק תיאורטי דו"ח מכין
 - ✓ תיקייה בשם **CW** - מכילה שתי תיקיות, אחת של **קובצי source** (קבצים עם סיומת *.c) והשנייה של **קובצי header** (קבצים עם סיומת *.h).
- ✓ תיקייה בשם **PC_side** - המכילה קובצי מקור של אפליקציית צד מחשב + קובץ ReadMe המתאר בקצרה מה תפקיד כל קובץ מקור במימוש האפליקציה.

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 **שלושת** הפרטים הבאים בלבד:
 - ✓ קובץ **final_labx.pdf** – מכיל תיאור והסבר לדרך הפתרון של מטלת זמן אמת.
 - ✓ תיקייה בשם **CW** - מכילה שתי תיקיות, אחת של **קובצי source** (קבצים עם סיומת *.c) והשנייה של **קובצי header** (קבצים עם סיומת *.h).
- ✓ תיקייה בשם **PC_side** - המכילה קובצי מקור של אפליקציית צד מחשב + קובץ ReadMe המתאר בקצרה מה תפקיד כל קובץ מקור במימוש האפליקציה.

בהצלחה.