מגמת הנדסת אלקטרוניקה ומחשבים

מדריך למורה – התמחות
מיקרו מעבדים

VHDL

תוכן עניינים

יחידה 1: מיקרו מעבדים

היקף: 180 שעות עיוני

רציונל ודגשים בתכנית הלימודים

הרציונל

המטרה העיקרית של תכנית הלימודים **בלימודי ההתמחות** של המגמה **הנדסת אלקטרוניקה ומחשבים** היא להעניק ללומד חוויה טכנולוגית מעשירה בסביבת למידה עדכנית בדגש על 3 עקרונות מרכזיים:

1. טיפוח לומד בעל הכוונה עצמית בלמידה
2. חינוך לחשיבה במסגרת למידה מבוססת פרויקטים
3. פיתוח חשיבה מסדר גבוה וטיפוח מיומנויות עבודת צוות.

לימודי ההתמחות כוללים 3 מרכיבים:

1. התמחות ב**אחד** מנושאי החלופה (חובה) בהיקף של 180 שעות
   - **א.** התמחות במקצועות מיקרו מעבדים
   - **ב.** או שפת - VHDL (בחירה של אחד מבין השני המקצועות)
2. המקצועות מיקרו מעבדים ושפת VHDL כשפת תכנות של רכיבים מתכנתים מהווים בסיס לעבודה על הפרויקט ומעבדת פרויקטים ליישום ואינטגרציה בין הנושאים שנלמדו בהיקף של 180 שעות
3. בתי הספר יכולים לבחור את נושא ההתמחות הספציפי של המגמה מתוך 7 החלופות להלן בהתאם לחזון בית הספר בהיקף של 120 שעות עיוני+60 שעות התנסות:
   - **א.** תקשורת במערכות אלקטרוניות
   - **ב.** לוחמה אלקטרונית
   - **ג.** אלקטרואופטיקה
   - **ד.** הנדסה רפואית
   - **ה.** בקרה ורובוטיקה
   - **ו.** תחבורה חכמה
   - **ז.** או תכנית לימודים ייחודית באישור המפמ"ר.

במהלך הלימודים בכיתה י"ב התלמיד ייפתח עבודת גמר המבוססת על נושאים הנלמדים במסגרת ההתמחויות הנ"ל (השונות).

פרקי תכנית הלימודים

פרק 1 – תכנות Embedded C

יעדים

פרק זה מהווה חזרה על תכנים מתוכנית הלימודים מהמקצוע המוביל "מערכות משובצות מחשב" והעמקה בנושאים נבחרים.

תכנים

1. מאפיינים ייחודיים בפיתוח תוכנה בסביבת Embedded C.
2. התנסות בכתיבת הכוללות מבני בקרה וביצוע חוזר של פעולות.
3. עבודה עם מערכים.
4. עבודה עם פונקציות.
5. שילוב פונקציות ומערכים בפיתוח תוכנה בסביבת Embedded C.

פרק 2 – חומרת הבקר

יעדים

פרק זה מהווה חזרה על תכנים מתוכנית הלימודים מהמקצוע המוביל "מערכות משובצות מחשב".

תכנים

1. ארכיטקטורת הבקר.
2. כניסות ויציאות של הבקר(זרמי ומתחי כניסה ומוצא ,סוגי מוצא).
3. הגדרת מאפייני ממיר(A/D רזולוציה ,מתח יחוס ,זמן המרה).
4. הגדרת מפתח הפלט בתוכנה תוך שימוש בפעולה pinMode.
5. שליטה על Duty-Cycle בשיטת PWM.
6. מונים וטיימרים – השהיות מדויקות ,מדידת רוחב פולס ,מדידת תדר.
7. פסיקות.

פרק 3 – תקשורת נתונים תקן RS-232

יעדים

פרק זה מהווה הרחבה של הפרק "תקשורת טורית UART בין מיקרו-בקר למחשב" מתוכנית הלימודים "מערכות משובצות מחשב".

תכנים

1. פרוטוקול התקשורת ,צורת הסנכרון ,מהירויות העברת מידע ,מרחק השידור, יתרונות חסרונות. דוגמאות למימוש.
2. חיבור בין שני מחשבים או מיקרו-מעבדים בפרוטוקול זה.

פרק – 4 פרוטוקול TWI/I²C

יעדים

בפרק זה התלמידים יכירו את מאפייני פרוטוקול התקשורת הטורית I²C ויתנסו בחיבור של התקנים אל המיקרו בקר באמצעות פרוטוקול זה.

תכנים

1. מאפיינים כללים.
2. קצב העבודה של הפרוטוקול.
3. הפרוטוקול להעברת נתונים (תחילת תשדורת, סיבית אישור, סיבית סיום).
4. הדקי קווי התקשורת SDA ו- SCL.
5. שליחת כתובת לרכיב.
6. פעולת כתיבה וקריאה מרכיבי slave.
7. התנסות עם רכיב PC8574 או אחר.
8. התנסות עם רכיב DS1307 או אחר

פרק 5 – פרוטוקול SPI

יעדים

בפרק זה התלמידים יכירו את מאפייני פרוטוקול התקשורת הטורית SPI ויתנסו בחיבור של התקנים אל המיקרו בקר באמצעות פרוטוקול זה.

תכנים

1. מאפיינים כללים.
2. קצב העבודה של הפרוטוקול.
3. הפרוטוקול להעברת נתונים.
4. הדקי קווי התקשורת.
5. התנסות עם רכיב MAX7219.
6. מסך גרפי.
7. מסך מגע.

פרק 6 – עבודה עם רכבי חומרה ייחודיים לתחום ההתמחות

יעדים

בפרק זה התלמידים יכירו רכיבי חומרה על פי החלופה שנבחרה על ידי בית הספר.

לדוגמה:

- חלופת תקשורת במערכות אלקטרוניות ניתן ללמד כרטיס תקשורת ASK 455MHz או כרטיס Ethernet Shield או כרטיס GSM.

- בחלופה אלקטרואופטיקה ניתן ללמד עבודה עם סיבים אופטיים או מודול מצלמה כדוגמת OV7670.
- בחלופה הנדסה רפואית ניתן ללמד חיישן גלי מוח.
- בחלופה בקרה ורובוטיקה ניתן ללמד בקר PID.
- בחלופה לוחמה אלקטרונית ניתן ללמד מודול מצלמה כדוגמת OV7670.

תכנים

1. רקע תיאורטי ועקרונות פיזיקליים של הרכיב.
2. תכונות אלקטרוניות של הרכיב אופן חיבור למיקרו-מעבד תוך כדי קריאת דפי נתונים רלוונטיים.
3. שרטוט מערך החיבורים בין רכיב לבקר.
4. כתיבת מחלקה/חקירת מחלקה המטפלת הפעלת הרכיב.
5. כתיבת תוכנה המשלבת את המחלקה לתקשורת בין הרכיב לבקר.
6. אינטגרציה בין הרכיב הנלמד לרכיב נוסף כמו צג ,תצוגה ,חיישן אחר.

פרק 7 – עקרונות במערכות בקרה ממוחשבות

יעדים

בפרק זה התלמידים ילמדו מושגים מתחום הבקרה הקלאסית (תכנים 3-1) ואת תאורה של מערכת בקרה בחוג סגור. בהמשך הפרק התלמידים יכירו מבנה של מערכת בקרה ממוחשבת (תכנים 8-4).

תכנים

1. מושגי יסוד בבקרה ,בקרה בחוג פתוח ובקרה בחוג סגור – משוב.
2. תיאור מערכת בקרה בחוג סגור באמצעות תרשים מלבנים והאותות במערכות :אות רצוי ,אות מצוי ,הפרעות ,מדידה והתמרה ,השוואה ושגיאה..
3. דוגמאות למערכות בקרה טכנולוגיות.
4. תיאור מערכת בקרה ממוחשבת הכוללת :מעבד ,חיישן ומעגל וויסות.
5. בקרת טמפרטורה או אור דו מצבית – קריאת נתון מחיישן אנלוגי באמצעות A/D או חיישן דיגיטלי והפעלה דו מצבית של צרכן חום או אור.
6. בקרת אור או חום בשיטת – PWM קריאה מחיישן אנלוגי באמצעות A/D או חיישן דיגיטלי וויסות צרכן האור או הטמפרטורה באמצעות בקרת.PWM
7. בקרת מהירות מנוע DC בשיטת - PWM קריאת נתוני מהירות המנוע באמצעות אינקודר אופטי או מגנטי ושליטה על מהירות המנוע באמצעות בקרתPWM .
8. בקרת זווית -קריאת נתון מחיישן זווית ושליטה על מנוע סרבו או מנוע צעד.

פרק 1: תכנות Embedded C

זהו פרק חזרה והרחבה של תכנית הלימודים "מערכות משובצות מחשב". הפרק מציג את עקרונות התכנות שפת Embedded C תוך השוואה בינה לבין שפת תכנות סטנדרטית למחשב האישי. דגש מיוחד הושם על מגבלות התכנות הקיימות בשפה זו. כמו כן קיימת התייחסות לאופן בו ניתן לאחסן נתונים במרחב הזיכרון מסוג flash תוך שימוש בפקודות מאקרו (למשל F) או באמצעות שימוש בפעולות אחרות. בפרק מבוצעת השוואה בין אחסון בזיכרון רגיל (ram) לבין הזיכרון מסוג flash.

מכיוון שבמקביל למקצוע הקדם "מערכות משובצות מחשב", נלמד גם את המקצוע "תכנות בשפת #C" התלמיד נחשף לשתי שפות שונות שלהן מושגים מעט שונים. בפרק זה, כמו בפרקים הבאים, בחרנו להשתמש במונחים מתחום התכנות בשפת #C. כמו למשל המונח "פעולה" המקביל למונח "פונקציה" בשפת Embedded C.

הפרק מציג גם את **מבני הבקרה** השונים ובהם משפטי תנאי, משפט ברירה מרובה (switch-case), משפט תנאי טרינארי, לולאות, מערכים ופעולות. כדאי להראות את השימוש במשפט ברירה מרובה, למשל בהקשר של מכונת מצבים. ניתן דגש על הנושא מערך וכן על הנושא פעולות.

הדגשים המרכזיים של פרק זה (וכן של הפרקים הבאים) הוא על:

- כתיבה מודולארית, תוך שימוש בריבוי קבצים.
- פעולות הייחודית לשפה זו, תוך כדי שימוש בחומרה המחוברת אל המיקרו בקר.

בסעיפים על המערכים והפעולות מומלץ למורה לבצע השוואה בין שפת תכנות זו לבין שפת #C שנלמדת במקביל. כדאי להדגיש את ההבדל בין הגדרת מערך בשפה זו (הקצאה סטטית) לבין ההגדרה בשפת #C (הקצאה דינמית). כמו כן, מומלץ להדגיש את ההבדל בין הגדרת פעולה (פונקציה) בשפה זו לבין הגדרת פעולות בשפת #C.

חשוב מאוד להדגיש לתלמיד שגם שפה זו היא שפה מונחית עצמים בדומה ל- #C ולכן אנחנו משתמשים בתחביר ומושגים מעולם זה.

הפרק מסתיים בהצגת פרויקט מערכתי המציג באופן מפורט את שיטת הפיתוח של מערכות משובצות מחשב הן מהיבט החומרתי והן מההיבט התיכנותי.

פתרון לשאלה 1.16

בפרויקט לוח מקשים ותצוגת 7 מקטעים, ישנה דרישה להרחיב את הפרויקט כך שיכלול מנוע חשמלי. לוח המקשים יקבע את מהירות המנוע, והיא תהיה בתחום 0 עד 99. כאשר ערך של המהירות 0 המנוע במנוחה, וכאשר הערך הוא 99 המנוע יפעל במהירות הגבוהה ביותר.

א. השלימו את תרשים המלבנים על פי הדרישה.

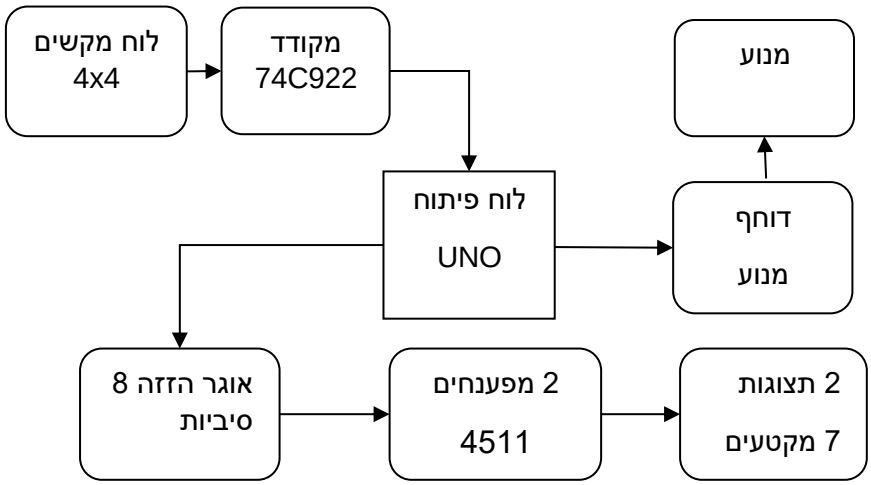
ב. השלימו את תרשים החשמלי לחיבור המנוע.

ג. הוסיפו קובץ נפרד לטיפול במנוע.

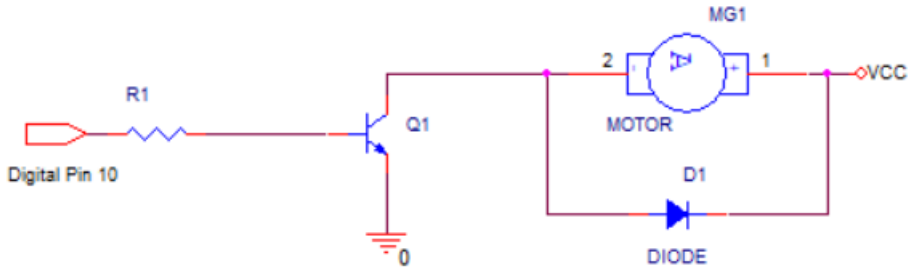
ד. שנו את התוכנית הראשית כך שהמערכת תפעל על פי הדרישה.

פתרון

א. נוסיף לתרשים המלבני דוחף זרם ומנוע חשמלי:



ב. להלן תוספת לתרשים החשמלי. נעשה שימוש במעגל בסיסי – טרנזיסטור כמתג. יש להסביר לתלמיד את הנחיצות של הדיודה במעגל.



ג. קובץ לטיפול במנוע

המנוע יופעל בשיטת PWM, לכל ערך בתחום מ-0 עד 99 נתאים ערך מ-0 עד 255. כאשר הערך המזערי הוא 0 (מנוע במנוחה) והערך המרבי הוא 255 (מנוע מסתובב במהירות מרבית).

Project2 | Disp7Seg | Keyboard | Motor

```
1  const int M = 10;
2
3  void InitMotor(){
4    pinMode(M, OUTPUT);
5  }
6
7  void stopMotor(){
8    analogWrite(M,0);
9  }
10
11 void runMotor(int percent){
12   int motorSpeed = map(percent, 0, 99, 0, 255);
13   analogWrite(M, motorSpeed);
14 }
```

ביאור התוכנית

שורה 1: הגדרת הדק החומרה

שורות 3-5: פעולת InitMotor

שורה 4: הגדרת הדק חומרה כפלט

שורות 7-9: פעולת stopMotor

שורה 8: הפסקת אות PWM

שורה 11-14: פעולת runMotor המקבלת כפרמטר את אחוז הפעולה של המנוע

שורה 12: העתקה לינארית של הערך באחוזים לערך המתאים ל- PWM

שורה 13: הפעלת המנוע בערך הרצוי

ד. התוכנית הראשית

בתוכנית נוספו השורות הבאות:

שורות 6-7 בפעולת ה- setup לאתחול המנוע

שורה 17 הפעלת המנוע בפעולה loop.

Project2 | Disp7Seg | Keyboard | Motor

```
unsigned int val;
void setup() {
  InitKbd();
  InitShiftReg();
  val = 0;
  InitMotor();
  stopMotor();
}
void loop() {
  char key;
  if (keyAvailiable()>0){
    key = getKey();
    if (isDigit(key)){
      val = val * 10 + (key - '0');
      val = (val>100)? val % 100 : val;
      show(val);
      runMotor(val);
    }
  }
}
```

פרק 2: חומרת הבקר

פרק זה מהווה חזרה על ארבעה פרקים של תכנית הלימודים "מערכות משובצות מחשב":

1. מבואות ומוצאים ספרתיים
2. מבואות ומוצאים תקביליים
3. מיקרו בקרים
4. פסיקות וקוצבי זמן.

מטרת הפרק הינה סקירה של התכנים שנלמדו בשנה הקודמת ותרגול השימוש במיקרו בקר, תוך חשיפה להתקנים חדשים ובהם חיישן טמפרטורה LM36, חיישן PIR, חיישן אולטרה-סוני וחיישן LDR. בתהליך ההתנסות עם הרכיבים נדרש התלמיד לזהות את סוג החיישן או את התקן ההפעלה ולקבוע את אופן חיבורו למיקרו בקר: חיישן תקבילי לכניסה תקבילית, חיישן ספרתי לכניסה ספרתית. באותו אופן עליו לקבוע גם ליחידות הפעלה.

הפרק פותח בתיאור עקרוני של מיקרו בקר באופן כללי וממשיך בהתמקדות במיקרו בקר ATmega329P, תוך פירוט מרכיביו ותכונותיו. בהמשך אנו סוקרים של ההדקים הספרתיים של המיקרו בקר, תוך הצגת מאפייניהם החשמליים וההוראות בשפת Embedded C. זאת אנו עושים על מנת שנוכל להשתמש בהם כהדקי מבוא או כהדקי מוצא (לפי הצורך).

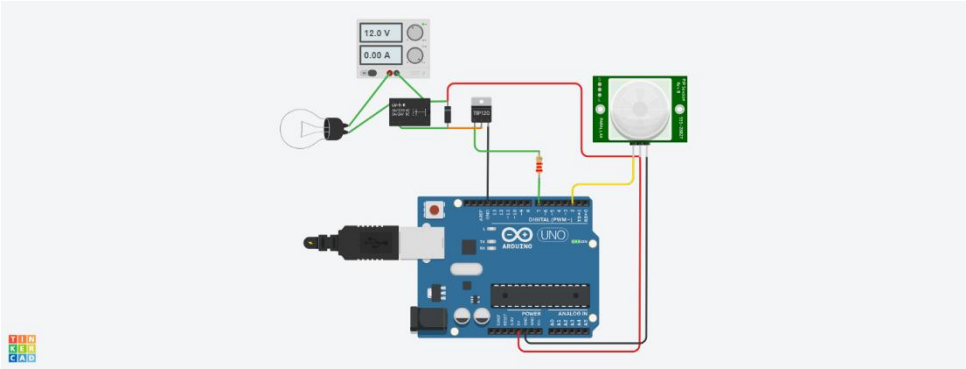
לאחר סקירת ההדקים הספרתיים אנו סוקרים את הדקי המבוא התקבילי, המספקים ערכים אל יחידת ההמרה ADC המובנית במיקרו בקר. הסקירה כוללת התייחסות למאפייניהם החשמליים של המבואות וכן חישוב של הרזולוציה של הממיר, תוך התחשבות במתח הייחוס. לאחר הסקירה של המבואות התקביליים מתבצעת סקירה של המוצאים התקביליים הפועלים על פי שיטת PWM. לאחר סקירת המאפיינים החשמליים מתנסה התלמיד בכתיבת תכניות המשלבות חיישן טמפרטורה LM36 והפעלת נורת LED בעוצמה משתנה, תוך שימוש בתוכנת סימולציה TinkerCad.

בהמשך הפרק אנו חוזרים על שיטות העבודה תשאול (polling) ופסיקה (interrupt), ושימוש במונים. בחלק זה של הספר מוסבר המבנה הפנימי של המונים במיקרו בקר ותפקידם בסביבת הפיתוח ליצירת אותות תיזמון כמו שעון המערכת. מסיבה זו אין בספר התייחסות ישירה למונים של המיקרו בקר אלא באמצעות פעולות המסופקות על ידי סביבת הפיתוח. לאחר סקירת המאפיינים החשמליים מתנסה התלמיד בכתיבת תכניות המשלבות חיישן PIR והפעלת צליל התראה באמצעות רמקול תוך שימוש בתוכנת סימולציה TinkerCad. בהתנסות יכתבו התלמידים פתרון, הן בשיטת התשאול והן בשיטת הפסיקה.

בנוסף התלמידים יתנסו בכתיבת תכניות למדידת מרחק באמצעות חיישן אולטרה-סוני ומד תדר. בשתי תכניות אלה יתנסו התלמידים בשימוש בהוראה pulseIn.

פתרון שאלה 2.4

נתון התרשים החשמלי הבא:



התרשים מתאר מערכת להפעלת תאורה אוטומטית. כאשר מזוהה תנועה, מופעלת התאורה למשך זמן של 1 דקה. אם מזוהה תנועה תוך כדי ההפעלה של התאורה, יחודש זמן ההפעלה למשך 1 דקה. אם כעבור דקה, לא מזוהה תנועה התאורה תופסק.

א. מה תפקידו של הטרנזיסטור הביפולרי TIP120?

ב. מדוע נחוץ ממסר?

ג. כתבו תכנית למימוש המערכת.

פתרון

ניתן למצוא את הפרויקט[1] באתר בקישור שנמצא כאן.

א. תפקידו של הטרנזיסטור הוא למתג את הממסר. לוח הפיתוח לא יכול לספק את זרם ההפעלה של הממסר באמצעות הדקי המוצא שלו (מגבלת זרם מרבית של 40mA) ולכן זרם ההפעלה מסופק על ידי הטרנזיסטור המחובר בחיבור ישיר למקור המתח של כרטיס הפיתוח.

ב. הממסר משמש לבקרת הספק. הסיבות לשימוש בממסר הן:

1. מתח חילופים. לא ניתן למתג באמצעות טרנזיסטור (פועל בזרם ישר בלבד).
2. מתח ישר. בטרנזיסטור קיימת מגבלת מתח והספק.
3. בידוד גלווני. השימוש בממסר מפריד בין הצרכן – העומס לבין מערכת הבקרה – בקר.

ההפעלה של הצרכן הוא 12V ואילו מתח ההפעלה של המיקרו בקר הוא 5V ולכן במקרה זה גם הטרנזיסטור יכול לשמש כמתג ואז הממסר מיותר.

[1] על מנת להשתמש בפרויקט זה עליכם להיות משתמשים רשומים באתר.

ג. התוכנית מבוססת על דוגמה 2.4 המוצגת בספר הלימוד. בתוכנית הגדרנו את הדק המוצא אליו מחובר הטרנזיסטור והפעלנו את הנורה בכל תזוזה של העצם מול החיישן.

```
void setup()
{
  pinMode(2, INPUT);
  pinMode(13, OUTPUT);
  pinMode(7, OUTPUT);
  attachInterrupt(digitalPinToInterrupt(2), myISR, CHANGE);
}

void myISR(){
  if (digitalRead(2) == HIGH) {
    digitalWrite(13, HIGH);
    digitalWrite(7, HIGH);
    delay(1000);
  }
  else {
    digitalWrite(13, LOW);
    digitalWrite(7, LOW);
  }
}

void loop() {
}
```

פרק 3: תקשורת נתונים תקן RS-232

פרק זה מהווה חזרה על הפרק תקשורת טורית של תכנית הלימודים "מערכות משובצות מחשב". מטרת הפרק היא לסקור את תכני הלימודים שנלמדו בשנה הקודמת ולהתייחס לצורך בערוצי תקשורת בנוסף לערוץ התקשרת המובנה Serial. התלמידים נחשפים בפרק זה למימוש של ערוצי תקשורת נוספים תוך שימוש במחלקה באמצעות המחלקה SoftwareSerial.

הפרק פותח בתיאור של התקן RS-232 (או בשמו האחר TIA232). התיאור ברמה הפיסית כולל התייחסות אל קווי החיבור המלא לציוד תקשורת, קווי חיבור החיבור המינימאליים הנדרשים ליצירת תקשורת, אורך הקווים וציון תחום המתחים בקווי התקשורת. התיאור ברמה הלוגית של ההתקן מתמקד במבנה מסגרת שידור, בדיקת שגיאות וקצב שידור ביחידות של סיביות לשנייה ותווים לשנייה.

בנושא הבא של הפרק מוצגים הדקי התקשורת המובנית (Serial) בלוח הפיתוח ואת האפשרות להשתמש בהדקים אחרים בלוח הפתוח, היכולים לשמש כערוצי תקשורת תוך שימוש במחלקת תוכנה (SoftwareSerial) למימוש עקרונות התקשורת. בחלק זה מוצגות באופן חלקי הפעולות של המחלקה Serial וכן של המחלקה SoftwareSerial. דגש מיוחד הושם על השימוש במחלקה SoftwareSerial ומגבלות השימוש בה.

נקודות מומלצות לדיון בכיתה

- הבדל בין המחלקה Serial לבין המחלקה SoftwareSerial מבחינת יצירת עצם.
- מגבלות השימוש במחלקה SoftwareSerial
- מומלץ למורה לבצע השוואה בין יצירת עצם בשפת C# לבין יצירת עצם בשפת Embedded C (שלוב המחלקה ופעולת היצירה).

החלק הבא של הפרק מתאר את אופן השימוש במחלקה SerialPort של שפת C#, חיבור בין לוח פיתוח למחשב האישי וחיבור שני לוחות פיתוח זה לזה בתקשורת טורית.
המדריך למורה של "מערכות משובצות מחשב" מפרט בהרחבה שימושים אלה ושימושים נוספים.

פתרון שאלה 5 – נוסח מתוקן

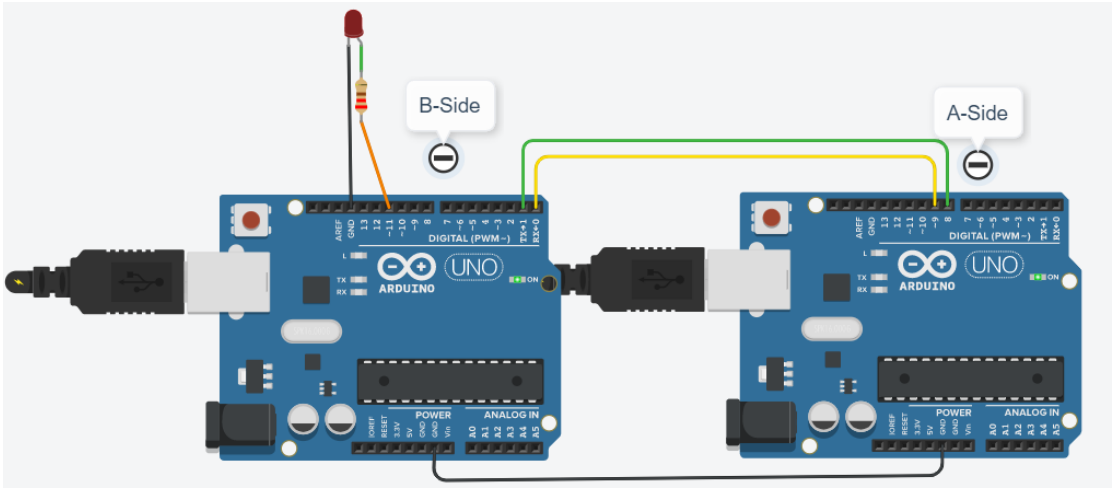
כתבו תכנית ללוח הפיתוח UNO (B-Side) המפעילה LED, המחובר להדק 11 בלוח הפיתוח ומוגדר כיציאה אנאלוגית. ההפעלה תתבצע באמצעות ערך מספרי שיתקבל דרך ערוץ התקשורת הטורית של הלוח המחובר ללוח פיתוח UNO אחר (A-Side). הערך המספרי שיתקבל יהיה בתחום בין 0 ל-255.

בצד האחר (A-Side) ייקלט ערך מספרי מהמשתמש באמצעות המסוף הטורי (monitor), שישלח דרך ערוץ התקשורת הטורית המקשר בין הלוחות.

פתרון

ניתן למצוא את הפרויקט[2] באתר בקישור שנמצא כאן.

להלן תרשים החיבורים:



[2] על מנת להשתמש בפרויקט זה עליכם להיות משתמשים רשומים באתר.

תכנית A-Side

```
// A-Side
#include <SoftwareSerial.h>
SoftwareSerial Serial1(8, 9); // Rx, TX

void setup(){
  Serial.begin(9600);
  Serial1.begin(9600);
}

void loop(){
  if (Serial.available()>0){
    int val = Serial.parseInt();
    Serial1.print(val);
  }
}
```

תכנית B-Side

```
// B-Side
const int LED = 11;

void setup(){
  pinMode(LED, OUTPUT);
  Serial.begin(9600);
}

void loop(){
  if (Serial.available()>0){
    int val = Serial.parseInt();
    analogWrite(LED, val);
    Serial.print(val);
  }
}
```

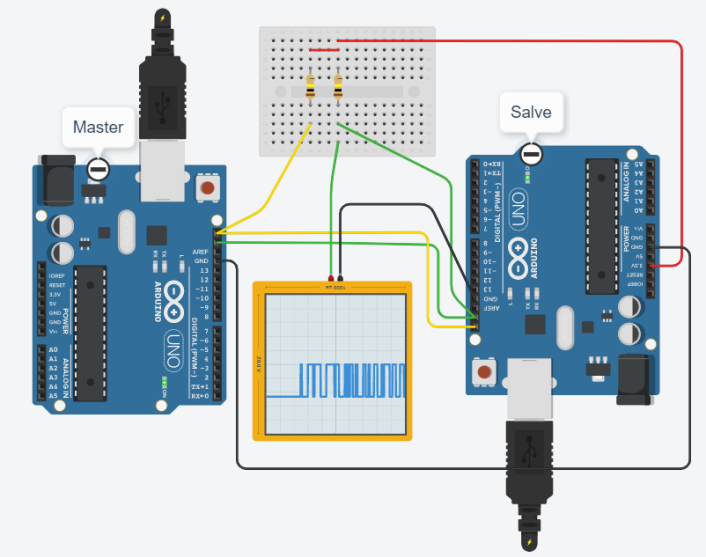
פרק 4: פרוטוקול TWI/I2C

בפרק זה התלמידים יכירו את מאפייני פרוטוקול התקשורת הטורית I^2C ויתנסו בחיבור של התקנים אל המיקרו בקר באמצעותו.

- הפרק פותח בתיאור עיוני של ארכיטקטורת פרוטוקול I^2C או בשמו האחרTWI. התיאור ברמה הפיסית כולל התייחסות אל קווי החיבור הנדרשים ליצירת תקשורת בין ההתקנים השונים והמתחים בקווי התקשורת, תוך דגש על נגדי pull-up הנחוצים לקיום הדרישות החשמליות. התיאור ברמה הלוגית של הארכיטקטורה מתייחס לתפקידים של ההתקנים: שליט-עבד (או בביטוי הלועזי master-slave). התיאור הלוגי דן באיתותי הבקרה (Start ו- Stop) היזומים על ידי התקן ה- master וממשיך בתיאור האותות המשמשים להעברת סיביות בין ההתקנים השונים. המידע עובר במסגרות של כתובת או נתונים שבסיומן משודרת סיבית אישור ACK. בסופו של החלק העיוני מוצג קצב התקשורת האפשרי בפרוטוקול זה.

- לאחר התיאור העיוני של הפרוטוקול מוצג המימוש שלו במיקרו בקר ATmega329P של לוח הפיתוח UNO תוך התייחסות למגבלת הקצב הקיימת בו. סביבת הפיתוח מספקת מחלקת שירות Wire המטפלת בכל מאפייני התקשורת של פרוטוקול זה. הפרק כולל תיאור מפורט של פעולות המחלקה ואת אופן ההגדרה של לוח הפיתוח כהתקן master או כהתקן slave.

- ניתן למצוא כאן קישור למימוש דוגמה 4.1 המופיעה בספר הלימוד באמצעות תוכנת הסימולציה Tinkercad. הסימולציה מרחיבה את הדוגמה שבספר באמצעות הצגת אות SDA באמצעות משקף תנודות.



- בהמשך הפרק מופיעות דוגמאות נוספות לחיבור בין שני לוחות לפי תקן זה כאשר מחברים רכיבים נוספים אל אחד הלוחות.

- בפרק נכלל ניסוי שמומלץ שיבוצע באמצעות לוחות פיתוח ולא באמצעות סימולציה.

- הפרק כולל חיבור של התקנים שונים: מד מרחק אולטרה-סוני SRF08, הרחבת IO PCF8574 ושעון זמן אמת DS1307. לשני ההתקנים האחרונים קיימים ניסויים שמומלץ שיבוצעו על ידי התלמיד.

- פרוטוקול זה, כמו פרוטוקול SPI המוצג בפרק הבא, שונה מפרוטוקול RS-232 שנלמד בפרק הקודם. בעוד שפרוטוקול RS-232 נועד לחיבור ציוד קצה של תקשורת או חיבור בין שתי מערכות ממוחשבות, פרוטוקול TWI משמש לחיבור מספר רב של התקנים למחשב מארח (127 כתובות אפשריות). מומלץ לציין זאת מיד בתחילת הפרק ואף לבצע השוואה בין שני הפרוטוקולים מבחינת כמות התקנים, מספר קווי חיבור, אותות חשמליים וקצב התקשורת.

- החלק העיוני בפרק עמוס בפרטים טכניים ומומלץ להמחיש את התיאור בצורה מעשית או באמצעות סימולציה. בכל מקרה, באופן מעשי, התלמיד לא יצטרך לטפל במרבית המאפיינים העיוניים של הפרוטוקול חוץ מהצורך בנגדי pull-up, התייחסות לכתובתו של התקן העבד וקצב התקשורת.

- מומלץ למורה לאתגר את התלמיד בלימוד של התקנים המתחברים לפי פרוטוקול זה על פי דפי המפרט וליישם את המידע בהם לתוכנית תוך שימוש במחלקה Wire.

פתרון שאלה 4.10

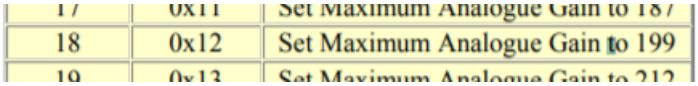
היעזרו בקישור של הרכיב SRF08 וענו על השאלות הבאות:

א. מעוניינים בהגבר אנאלוגי של כ- 200. מהו הערך שנדרש לשלוח לאוגר ההגבר?

ב. מהו הערך שנדרש להעביר לאוגר 2 על מנת שטווח המדידה יהיה עד לטווח של 1 מטר?

פתרון

א. נפתח את דפי המפרט בעמוד 6. מתוך הטבלה נבחר באפשרות הקרובה ביותר 199



18	0x12	Set Maximum Analogue Gain to 199

הערך המתאים להעביר לאוגר הוא 0x12.

ההוראות המתאימות הן:

```
Wire.beginTransmission(0x70); // 0xE0 >> 1
Wire.write(byte(0x01)); // אוגר ההגבר
Wire.write(byte(0x12));
Wire.endTransmission();
```

ב. על פי המתואר בפרק ניתן לקבוע את תחום המדידה באמצעות כתיבה לאוגר התחום בכתובת 2. קביעת תחום המדידה ביחידות של מ"מ נקבעת על פי הנוסחה הבאה:

$$Range = (\text{Range\_Register} * 43\text{mm}) + 43\text{mm}$$

ומכאן שהערך יהיה:

$$Range\_Register = \frac{Range - 43mm}{43mm} = \frac{1000 - 43}{43} = 22.2558 = 22 = 0x16$$

פתרון שאלה 4.11*

רוצים לקבוע לחיישן כתובת חדשה 0xF8 במקום ברירת המחדל 0xE0.

א. על פי דפי המפרט של החיישן מהו רצף הפעולות הנדרש?

ב. רשמו פעולה בשם changeAddress לקביעת הכתובת החדשה.

פתרון

א. על מנת לענות על השאלה נצטרך לפתוח את דפי המפרט של הרכיב הנמצא כאן.

נעבור אל דף 7 בדפי המפרט:

Changing the I2C Bus Address
To change the I2C address of the SRF08 you must have only one sonar on the bus. Write the 3 sequence commands in the correct order followed by the address. Example; to change the address of a sonar currently at 0xE0 (the default shipped address) to 0xF2, write the following to address 0xE0; (0xA0, 0xAA, 0xA5, 0xF2). These commands must be sent in the correct sequence to change the I2C

על פי המפרט יש לוודא שקיים התקן אחד בלבד. הרצף שיש להעביר לכתובת הוא:

0xA0, 0xAA, 0xA5, 0xF8

ב. להלן הפעולה

```
11 void changeAddress(int neeAddress){
12   Wire.beginTransmission(0x70); // 0xE0 >> 1
13   Wire.write(byte(0xA0));
14   Wire.write(byte(0xAA));
15   Wire.write(byte(0xA5));
16   Wire.write(byte(0x7C)); // 0xF8 >> 1
17   Wire.endTransmission();
18 }
```

ניתן למצוא כאן ספרייה מוכנה לטיפול ברכיב SRF08.

כמו כן, על המורה ללמד במידת הצורך את המאפיינים הנוספים של הרכיב על פי דפי המפרט.

פרק 5: פרוטוקול SPI

בפרק זה התלמידים יכירו את מאפייני פרוטוקול התקשורת הטורית SPI ויתנסו בחיבור של התקנים אל המיקרו בקר באמצעות פרוטוקול זה.

- הפרק פותח בתיאור עיוני של ארכיטקטורת פרוטוקול SPI. התיאור ברמה הפיסית כולל התייחסות אל קווי החיבור הנדרשים ליצירת תקשורת בין ההתקנים השונים והמתחים בקווי התקשורת הנחוצים לקיום הדרישות החשמליות. התיאור ברמה הלוגית של הארכיטקטורה מתייחס לתפקידים של ההתקנים: שליט-עבד. התיאור כולל שתי צורות חיבור אפשריות. התיאור הלוגי דן באיתותי הבקרה באמצעות קו הבחירה SS ואות השעון SCLK היזומים על ידי התקן ה- master. המידע בין ה- master לבין ה- slave מתבצע באמצעות הדקים MOSI ו- MISO.

- לאחר התיאור העיוני של הפרוטוקול מוצג המימוש שלו במיקרו בקר ATmega329P של לוח הפיתוח UNO תוך התייחסות למגבלת הקצב הקיימת בו. סביבת הפיתוח מספקת מחלקות שירות SPISettings ו- SPI המטפלות בכל מאפייני התקשורת של פרוטוקול זה. הפרק כולל תיאור מפורט של פעולות המחלקות ואת אופן ההגדרה של לוח הפיתוח כהתקן master (לא ניתן להגדיר כ- slave). בלוח הפתוח ניתן להגיע לקצב גבוה של עד 8MHz.

- הפרק כולל חיבור של התקנים שונים: דוחף תצוגה MAX7219, מסך גרפי 12864B ומסך מגע. הפרק כולל דוגמאות רבות לכל אחד מההתקנים.

- מומלץ לבצע מגוון של דוגמאות ותרגילים באופן מעשי בהתקנים כאלה או דומים להם.

- פרוטוקול זה, כמו פרוטוקול I^2C aהוצג בפרק הקודם, שונה מפרוטוקול RS-232 שנלמד בפרק הקודם. בעוד שפרוטוקול RS-232 נועד לחיבור ציוד קצה של תקשורת או חיבור בין שתי מערכות ממוחשבות, פרוטוקול SPI משמש לחיבור מספר רב של התקנים למחשב מארח (כמספר קווי האפשור SS הזמינים). מומלץ לציין זאת מיד בתחילת הפרק ואף לבצע השוואה בין שלושת הפרוטוקולים מבחינת כמות התקנים, מספר קווי חיבור, אותות חשמליים וקצב התקשורת.

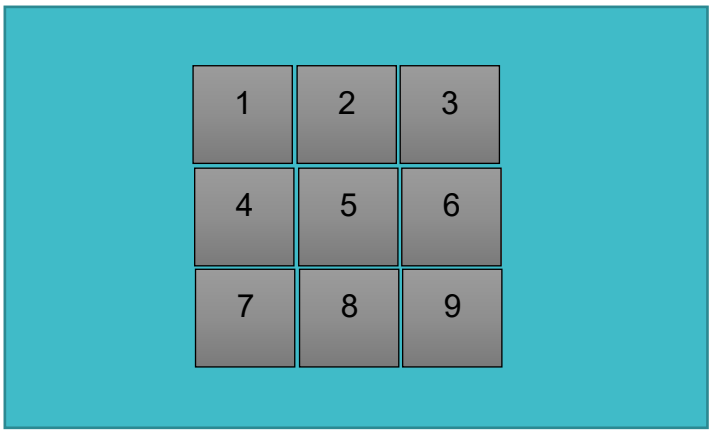
- החלק בו לומדים על התצוגה הגרפית מורכב למדי. מכיוון שמדובר ברכיב בעל מגוון רחב של אפשרויות עבודה התלמיד עלול לאבד את דרכו. לפיכך, מומלץ לשקול האם להשתמש בכל האפשריות הגלומות בו לבחור במגוון מצומצם יותר.

- בפרק זה מוצג גם מסך מגע המתחבר כהתקן אנאלוגי אל לוח הפתוח. יש מקום לחבר את מסך המגע עם הפעלת התצוגה הגרפית.

- במידה ומציגים תמונות נדרש לתת סקירה של פורמטים נפוצים של תמונות (BMP, GIF, JPG).

פתרון שאלה 5.17

תלמיד תכנן ליצור משחק איקס-עיגול. הוא ביקש לחלק את המסך כך שבמרכזו יוצגו 9 ריבועים. איור 5.19 מתאר את מבנה המסך הנדרש. גודלו של המסך הוא 240x320.



מבנה חלוקת המסך

א. קבעו את ערכי הגבולות של כל אחת מהמשבצות המופיעות במרכז מסך המגע. הציגו באמצעות טבלה.

ב. כתבו פעולה המזהה את המשבצת שבה זוהתה הלחיצה. הפעולה תחזיר ערך מספרי בין 1 לבין 9 אם זיהוי תקין, וערך שלילי אם הזיהוי שגוי.

ג. שכתבו את התוכנית מדוגמה 5.5 כך שתציג את מספר המשבצת בה בוצעה לחיצה.

פתרון

א. נקבע כל משבצת בגודל 60x60. כל תשע המשבצות יהיו בגודל 180x180.

בציר x יישארו 140=180-320, כלומר שוליים מהצדדים של 70.

בציר y יישארו 60=180-240, כלומר שוליים מהצדדים של 30.

להלן טבלת ערכי המשבצות לפי הסימון לעיל:

Ymax	Ymin	Xmax	Xmin	משבצת
89	30	129	70	1
89	30	189	130	2
89	30	249	190	3
149	90	129	70	4
149	90	189	130	5
149	90	249	190	6
209	150	129	70	7
209	150	189	130	8
209	150	249	190	9

ב. להלן הפעולה

```
int identify(int x, int y){
  if (y >= 30 && y < 90) { // 1, 2, 3
    if (x >= 70 && x < 130) return 1;
    if (x >= 130 && x < 190) return 2;
    if (x >= 190 && x < 250) return 3;
  }
  if (y >= 90 && y < 150) { // 4, 5, 6
    if (x >= 70 && x < 130) return 4;
    if (x >= 130 && x < 190) return 5;
    if (x >= 190 && x < 250) return 6;
  }
  if (y >= 150 && y < 210) { // 7, 8, 9
    if (x >= 70 && x < 130) return 7;
    if (x >= 130 && x < 190) return 8;
    if (x >= 190 && x < 250) return 9;
  }
  return -1; // error
}
```

ג. התוכנית המתוקנת תכלול את הפעולה לזיהוי המשבצת מסעיף ב ותיקון של ההצגה בפעולה הראשית:

```
void loop() {
    // הגדרת הדקי לוח הפיתוח כדי לזהות לחיצה על פי תוכנית קודמת
  pinMode(XM, OUTPUT);
  digitalWrite(XM, LOW);
  pinMode(YP, INPUT_PULLUP);
  if (digitalRead(YP) == LOW){  // בדיקה אם קיימת לחיצה
      readTouch();       // פעולה לקריאת הקואורדינטה של הלחיצה
         // מיפוי ערכי קואורדינטות גולמיים לקואורדינטות מסך בפיקסלים
      xpos = map(x, left, right, 0, touchWidth);
      ypos = map(y, top, bottom, 0, touchHeight);
      int id = identify(xpos, ypos);
      if (id != -1){
        Serial.print("panel - ");
        Serial.println(id);
      }
      else
        Serial.println("Error");
  }
  delay(200);
}
```

פרק 6: עבודה עם רכיבי חומרה ייחודיים לתחום ההתמחות

על פי היעדים שנקבעו בתוכנית הלימודים בפרק זה יכירו התלמידים רכיבי חומרה על פי החלופה שנבחרה על ידי בית הספר. חלופת תקשורת במערכות אלקטרוניות יכיר התלמיד כרטיס Ethernet Shield. בחלופות אלקטרואופטיקה ובחלופה לוחמה אלקטרונית יכיר התלמיד מודול מצלמה OV2640.

פרק זה מחולק באופן הבא:

א. סעיפים 6.1-6.3 עוסקים בתקשורת מחשבים, כרטיס התקשורת ומחלקות תמיכה בתקשורת.

ב. סעיף 6.4 עוסק במודול המצלמה

תקשורת מחשבים

- בחלק זה של הספר מוצגת סקירה של מושגי יסוד בתקשורת מחשבים ובהם המונחים LAN, WAN, מודל 5 השכבות ופרוטוקולים TCP/IP. מוצגת שיטת הפנייה להתקני קצה באמצעות כתובת IP (גרסה 4 בלבד) ומספר פורט (port) המאפיין יישום בהתקן הקצה.

- כרטיסי ההרחבה Ethernet Shield בגרסאות 1 ו- 2 מוצגות בהמשך הפרק וכן ההבדלים בין הגרסאות. בשני הכרטיסים קיימת תושבת עבור כרטיס זיכרון SD אולם אין בפרק התייחסות לכך. הכרטיסים מתחברים אל לוח הפיתוח UNO באמצעות פרוטוקול SPI.

- זה המקום לציין שאמנם בספר קיימת התייחסות לכרטיס הרחבה קווי (wired) בעוד שכרטיסי הרחבה אלחוטיים (wifi) הפכו להיות נפוצים יותר, אבל אין בכך כדי לפגום בתכנים. היסודות המוצגים בפרק תקפים גם לכרטיס האלחוטי. בעת כתיבת הפרק Arduino כבר החלה בשיווק של לוח פיתוח הנקרא UNO WIFI הכולל כרטיס תקשורת אלחוטי מובנה. לכרטיס תכונות דומות לזה של לוח הפיתוח UNO הנלמד על פי תכנית הלימודים. כאן ניתן לקרוא מידע על לוח הפיתוח UNO WIFI.

- בחלק השלישי של חלק זה מתוארים באופן מפורט כל מחלקות התקשורת: IPAddress, Ethernet.h, Ethernet2.h, EthernetClient.h, EthernetServer.h. כן קיימת בפרק התייחסות לשרת אינטרנט והפרוטוקול HTTP

- בחלק זה דוגמאות להפעלת כרטיס הרשת המקבל כתובת באופן אוטומטי (DHCP) או בצורה סטטית תוך השוואה בין השיטות.
- בסוף חלק זה שתי דוגמאות מסכמות בטכנולוגית client-server:
  - מימוש באמצעות שרת אינטרנט והלקוחות הם דפדפנים
  - הקמת רשת מקומית של מספר לוחות פיתוח כאשר לוח אחד הוא השרת ועד 4 (מגבלת תמיכה של כרטיס ההרחבה) לוחות נוספים המשמשים כלקוחות.

- מומלץ שהתלמיד יתנסה בהקמת שרת אינטרנט לפי הדוגמה שהוצגה בספר.

מודול מצלמה

- בחלק זה מוצגת סקירה של חיישן מצלמה תך הגדרת מונח הפיקסל וממדי החיישן. מתבצעת השוואה בין חיישנים בעלי ממדים שונים וכמות פיקסלים שונה. מההשוואה מתקבלת המשמעות של איכות התמונה המתקבלת מהחיישן.

- בחלק הבא מתוארות התכונות העיקריות של מודול המצלמה OV2640 המשתמש בשתי שיטת תקשורת שונות I2C (הגדרת החיישן) ו- SPI (הוראות הפעלה וזרימת נתונים).

- החלק האחרון עוסק בספריות התמיכה במודול המצלמה וכיצד לקבל התמונה לאפליקציה חיצונית המותקנת במחשב לשם צפייה בלבד.

- הטיפול באחסון תמונה בכרטיס SD או עיבוד תמונה חורג ממסגרת פרק זה.

- פרק זה נועד לספק דוגמאות וכלים לתלמיד העוסק בפיתוח עבודת גמר בתחום ההתמחות.

פרק 7: עקרונות במערכות בקרה ממוחשבות

על פי היעדים שנקבעו בתוכנית הלימודים, בפרק זה יכירו התלמידים מושגים מתחום הבקרה הקלאסית ויבחינו בין מערכת בקרה בחוג פתוח לבין מערכת בקרה וחוג סגור. כמו כן, התלמידים יתארו מערכות בקרה בחוג סגור באמצעות תרשים מלבנים תוך ציון האותות של המערכת. לאחר ההכרות עם מערכת הבקרה הקלאסית ירחיבו התלמידים את הידע למערכות בקרה ממוחשבות הכוללות מעבד, חיישן ומעגל ויסות. את המימוש של מערכת הבקרה הממוחשבת יבצעו התלמידים באמצעות לוח הפיתוח UNO תוך חיבור חיישנים ספרתיים ואנאלוגיים וכן יחידות ויסות ספרתיות או אנאלוגיות.

פרק זה מהווה יישום של כל פרקי הלימוד בתחום הידע של בקרה ממוחשבת. בפרק זה ינתחו התלמידים מערכת בקרה ממוחשבת, יזההו את החיישנים הנדרשים ואת תכונותיהם, ויכתבו תכנית לבקרת המערכת הממוחשבת.

פרק זה מחולק באופן הבא:

א. סעיפים 7.1-7.2 מתארים מערכת בקרה קלאסית

ב. סעיף 7.3 מתאר מערכת בקרה ממוחשבת מבוססת מיקרו בקר

ג. סעיף 7.4 מתאר מערכות בקרה מבוססות מיקרו בקר הכוללים חיישנים ומעגלי ויסות

מערכת בקרה קלאסית

- בחלק זה לומדים את מושגי היסוד בבקרה ובהם אות הערך הרצוי ואות המשתנה המבוקר. בהמשך נסקרות מערכות בקרה בחוג פתוח וחוג סגור המתוארים בתרשים מלבנים:
  - **אותות**: ערך רצוי, אות משוב, אות השגיאה, אות הפרעה ואות מצוי.
  - **מערכות**: משווה, בקר, אלמנט הפעלה, תהליך והמדידה.

- מערכת הבקרה בחוג סגור מתוארת באופן מפורט תוך שימוש בדוגמה מלווה.

- מומלץ למורה לקיים דיון בנושא זה ולבקש מהתלמידים דוגמאות של מערכות בקרה ביתיות (דוד חשמל, מזגן, מקרר וכדומה) ולהציג אותם באמצעות תרשים תוך פירוט חלקי התרשים. יש להבחין בדיון בין מערכות בקרה קלאסיות לבין מערכות בקרה ממוחשבות (כיום, ייתכן וחלק ממערכות הבקרה הביתיות ממוחשבות).

מערכת בקרה ממוחשב

- בחלק זה מתוארים המבנה והמאפיינים של מערכת בקרה ממוחשבת תוך הבלטת היתרונות של מערכת זו על פני המערכת הקלאסית. כמו כן, מתוארים קשרי הגומלין בין רכיבי מערכת הבקרה. לבסוף, מתוארים אביזרי הקצה (חיישנים או מעגלי ויסות) המחוברים לבקר הממוחשב והצורך בהמרת אותות; אותות תקביליים לספרתיים ולהיפך, אותות ספרתיים לתקביליים.

מערכות בקרה מבוססות מיקרו בקר

- בחלק זה מתוארים יישומים שונים של מערכות בקרה ממוחשבות תוך התייחסות לשיטות בקרה דו-מצבית (on-off) ובקרה רציפה והמימוש שלה במיקרו בקר תוך התייחסות לאלמנט ההפעלה. בכל אחת משיטות הבקרה מתוארות מערכות המספקות ערכים ספרתיים דו-מצביים של רכיב המדידה (חיישן) וערכים תקביליים של רכיב המדידה.

- הפרק כולל סקירה של מנוע לזרם ישר, מקודד ציר אופטי, דוחף מנוע L293D, מנוע צעד וחיישן זווית. כחלק מתיאור מנוע הצעד מוצגות פעולות המחלקה Stepper התומכות בהפעלתו.

- הדוגמאות המלוות את הפרק
  - מערכת לבקרת טמפרטורה דו-מצבית עם חיישן ספרתי
  - מערכת לבקרת טמפרטורה דו-מצבית עם חיישן תקבילי
  - מערכת לבקרת טמפרטורה תקבילית PWM עם חיישן תקבילי
  - מערכת לבקרת מהירות מנוע DC בשיטת PWM באמצעות מקודד אופטי (encoder)
  - הפעלת מנוע צעד בסיבוב מלא
  - בקרת זווית במנוע צעד

- מומלץ לבצע את הדוגמאות באופן מעשי או באמצעות תוכנת הסימולציה Tinkercad.

פתרון שאלה 7.11

שנו את התוכנית מדוגמה 7.6 כך שהמנוע יפעל במהירות של 180 סל"ד ויבצע תנועה של 90 מעלות בכל כיוון עם השהייה של רבע שנייה בכל החלפת כיוון פעולה.

פתרון

נשנה תחילה את המהירות ל- 180: myStepper.setSpeed(180)

נקבע את כמות הצעדים: 50 = const int stepsPerRevolution 50 צעדים מהווים רבע סיבוב.

הקוד לאחר השינויים הנדרשים.

```
#include <Stepper.h>
const int stepsPerRevolution = 50;
Stepper myStepper(stepsPerRevolution, 8, 9, 10, 11);
void setup() {
  myStepper.setSpeed(180);
}

void loop() {
  myStepper.step(stepsPerRevolution);
  delay(250);
  myStepper.step(-stepsPerRevolution);
  delay(250);
}
```

פתרון שאלה 7.16

הציעו שינוי בתוכנית כך שהתנודות ייפסקו עם ההגעה ליעד.

פתרון

על מנת למנוע את התנודות נבדוק אם המרחק בין הערך הרצוי לערך המצוי גדול מסף מסוים, למשל 2 ולא להסתפק ביחס גדול או קטן בלבד הגורם לתנודות. שיטה זו נקראת בקרת on-off בעלת תחום מת (dead zone). ניתן לדמות את שיטת הבקרה כמשווה מסוג שמידט שבו, כידוע, נקודות המפנה קובעות את רוחב החלון ובעצם את התחום המת.

קוד בתוכנית הקיימת

```
18  if (presentVal>previoustVal)
19    stepper.step(2);
20  if (presentVal<previoustVal)
21    stepper.step(-2);
```

הקוד לאחר הצעת השינוי

```
18  if (presentVal - previoustVal > 2)
19    stepper.step(2);
20  if (presentVal - previoustVal < -2)
21    stepper.step(-2);
```

יחידה 2: *VHDL*

מבוא

למידת שפת מחשב אינה פשוטה מהסיבה שנדרש ניסיון רב בלמידה עצמית בכלל והעמקה בנושאים מרובי דעת. שפת V.H.D.L דורשת יכולת תכנות בשפה עילית וכן ידע רב במערכות ספרתיות.

במתמטיקה קיימות הנחות יסוד ואקסיומות עליהן מתבסס תחום זה כגון, שני ישרים מקבילים לעולם לא נפגשים או סכום הזוויות במשולש הוא מאה שמונים מעלות. הנחות אלה הן המתודולוגיה, היסודות עליהן מתבסס הנושא בו דנים. ללא יסודות אלה לא ניתן להבין את המתמטיקה ואת הפיסיקה.

כאשר ניגשים ללמוד תחום חדש עלינו לברר תחילה את המתודולוגיה עליה הוא מבוסס. פרק ההקדמה מנסה להתמודד בהפשטת הנושא למבנים המוכרים לתלמיד, שילוב ייחודי של תוכנה וחומרה יחדיו ותיאור חומרה על ידי תוכנה .

בספר שלנו, שפת V.H.D.L היא שפה לתיאור מערכות לוגיות מורכבות וכמו בכל שפה, אי אפשר להמציא יש מאין. זו נקודה עדינה ולעיתים קשה להבנה, לכן בשפה עלית ישנם מבנים המנחים אותנו כיצד לכתוב תחביר נכון (לדוגמה, מבנה לולאה בעזרת for while). כיון ש-V.H.D.L נחשבת שפה עילית נוטים לשכוח כי היא אינה כפופה רק לאילוצי השפה, אלא בראש ובראשונה לכללי בניית רכיבים לוגיים. לדוגמה, האופרטור if בורר בין שתי אפשרויות או יותר ועלינו למצוא מבנה המתאים לו. הפתרון הבסיסי ביותר הוא מרבב (multiplexer), עליו למדנו במסגרת לימודי מערכות ספרתיות.

שפת V.H.D.L מאפשרת לנו גמישות מחשבתית, יחד עם זאת עלינו להבין כיצד קוד זה מתורגם למערכות ספרתיות. תרגום הקוד משפה עילית לשערים לוגים מתבצע בהתאם להנחות היסוד, כלומר על פי שיקולי תכנון חומרתיים.

הדוגמה הבאה מבהירה את חשיבות החומרה לפני שמתכנתים בשפת V.H.D.L, שכן השפה היא שילוב של חומרה ושפה עילית.

```
If(x==5)

Then ……
```

הפתרון – אם x לא שווה חמש, אזי התוכנית תמשיך הלאה.

לא מומלץ לתכנת בשפת V.H.D.L כך גם אם לא תופיע שגיאת תחביר, כיוון שה-compiler יבצע את המשימה עצמאית והתוצאה תהיה לא רצויה.

כדי להימנע מטעויות נרצה לסיים את הקוד בפקודת else, דבר שיאפשר ל – synthesizer להתחשב בכל האפשרויות של המרבב.

דוגמה נוספת לאופרטור מטעה היא האופרטור for בלולאות. הסיבה לכך היא שמערכות ספרתיות לא כוללות לולאות כפעולה בסיסית. כדי להתמודד עם בעיה זו אלה מתורגמים למבנים ידועים ועלינו להבין ולהכיר אותם.

נושא תרגום הקוד מוביל אותנו לעסוק בהבדל בין כתיבת קוד הניתן לסינתזה (synthesizable) לבין קוד שלא ניתן לסינתזה (non synthesizable). כאשר כותבים קוד לצריבה נתרגם אותו לשערים לוגיים כלומר, הוא יהיה קוד ניתן לסנתוז. כאשר נכתוב קוד לסימולציה, נכתוב בקוד שלא ניתן לסינתזה שכן זה רץ בסביבת מחשב תוך ניצול משאבי המחשב והקוד גמיש יותר.

דוגמה זו תסייע לנו להמחיש את הנאמר:

בשפת V.H.D.L קיים משתנה מסוג time ובמערכות לוגיות, כדוגמת מונים, איתו אפשר לספור את מספר מחזורי השעון בפרק זמן מסוים. יש לשים לב שאין הכוונה לאופרטור (פעולה בסיסית), אלא להגדרת מספר מחזורים באמצעות יחסי זמנים. דוגמה: היחס 30 ms/20ns זהה לכתיבת 1500000.

על מה יש לשים דגש בתכנון לוגי?

1. יש להקפיד על חיבור רגל reset לכל הדלגלגים ללא לוגיקה על רגל זו, מה שמוגדר כ-skew. גישה זו כמובן נכונה גם לגבי השעון
2. במקרה של if או case יש להימנע מכיסוי כל האפשרויות
3. יש להימנע מלהחסיר משתנים מרשימת ה- sensitivity list במקרה של process
4. להימנע מלולאות מכוננות

דרכי הוראה:

1. נדרשת חזרה על נושא מערכות ספרתיות
2. יש לתרגל בפרויקטים אישיים כל נושא נלמד. לדוגמה, כתיבת קוד וסימולציה במבנים פשוטים עד כדי שערים.
3. לכל תכנון רצוי להוסיף חיווי, כמו נוריות ותצוגת שבעה מקטעים.
4. יש לבצע צריבות על נושאים נבחרים ובדיקה באמצעות אוסילוסקופ.

5. על מנת לתרגל בצורה מיטבית, יש לתכנן שעון של 24 שעות הכולל דקות ושניות זאת בנוסף לתצוגת שבעה מקטעים. תרגיל זה מהווה הבנה יסודית של תכנון זמנים.
6. יש לשים דגש על עבודה בסביבות שונות בו זמנית כגון, הבנת מערכות לוגיות וכתיבת קוד בשפה עילית. יש לגשת לנושא בהדרגה. בשלב ראשון להתרכז בתרגילים המתמקדים בנושא אחד, לדוגמה בסביבת העבודה או מערכות לוגיות ולבסוף לשלב את הנושאים למכלול אחד.
7. יש להכביר בשימוש בסימולציות כמשוב לתלמיד.
8. יש להיזהר באמצעי חיווי ויזואליים כאמצעי לבדיקת קוד שכן הם עלולים לגרום לנו לפספס מעברים לוגיים .
9. סביבת FPGA הינה סביבה מורכבת ותכנון בה תובעני ודורש הבנה עמוקה במערכות ספרתיות. יש לשים דגש על הפשטה והסברים ברורים וריבוי תרגילים, דרכם תוודאו כי התלמידים הבינו את הנלמד. בהמשך זה יקל על המעבר לתרגילים ונושאים מתקדמים.
   למרות הקשיים שפורטו לעיל לסביבת FPGA יתרונות רבים:
   א. סביבות עבודה לא עומדות בפני עצמן ונדרשות סביבות תומכות. בשונה מסביבות אחרות, הדורשות חיבור באמצעות רכיבים פיזיים, בסביבת FPGA ניתן להוסיף או לגרוע כל תכנון מבלי לשנות את המבנה הבסיסי.
   ב. סביבת הסימולציה מצליחה לחזות בצורה מיטבית מכשולים אפשריים בתכנון.
   ג. ניתן לייצר כל סביבה שנדרשת או לתאר התנהגות של חיישנים.

הדוגמא הבאה תמחיש את הנאמר על סביבת העבודה:

כאשר בונים פרויקט עם חיישן מרחק srf04, לחיישן זה שתי רגליים מסוג trigger ו-echo. ניתן לייצר בסימולציה את התנהגות ה-trigger ובעקבות זאת לייצר שוב בסימולציה את echo וכל זאת טרם בנינו את הפרויקט. היכולת לחזות בהתנהגות רכיבי הפרויקט טרם בנייתו מאפשרת תכנון נכון וקצר יותר מבלי צורך לחזור על פעולות מיותרות.

10. כאשר ניגשים ללמד תכנון מקבילי, יש לתת את הדעת על כך שתכנון פולס שעון מחייב חשיבה מסדר גבוה.

ארגון חומר הלימוד

להוראת נושא זה קיימות שתי סביבות עיקריות QUARTOS MODELSIM ואת שתיהן ניתן להוריד מאתר ALTERA.COM. נושא ההתקנה יורחב בהמשך.

פרק 1: הכרת כלי סינתזה

בפרק זה התלמידים ילמדו להכיר את תוכנת QUARTOS, לבחור את הרכיב הנכון בהתאם לערכת העבודה ולבחור את הפרויקט האקטואלי משלל הפרויקטים שנמצאים בספריה.
בנוסף, הם ילמדו לקרוא דו"ח סופי הכולל את שם הפרויקט וכמה אחוזי משאבים נצרכו מהרכיב לטובת הפרויקט.

דרכי הוראה

1. פתיחת פרויקט ובחירת הרכיב הרצוי על פי לוח העבודה שברשות בית הספר
2. כתיבת קוד פשוט, דוגמת שער אחד בלבד, הכרת המקום לכתיבת כניסות ויציאות – entity וכן את מיקום הגדרת התכנים הלוגים
3. תיקון טעויות תחביר
4. קריאת דו"ח יישום הפרויקט על גבי הרכיב
5. חשוב בשלב זה להימנע מכל אתגר של מערכות לוגיות ולהתרכז בהכרת התוכנה בלבד.

דגשים

חשוב לשלוט טכנית בסביבת העבודה QUARTOS. פרק זה מבוסס על מעגלים לוגים פשוטים על מנת לנטרל קושי במערכות ספרתיות.

פרק 2: הכרת כלי סימולציה

יעדים

1. הכרת תוכנת הסימולציה MODELSIM והפונקציות שלה.
2. יצירת קובץ סימולציה מסביבתQUARTOS .
3. פתיחת פרויקט בתוכנת MODELSIM.
4. העברת קבצי הקוד והסימולציה מסביבת QUARTOS לסביבת הMODELSIM.
5. הגדרת כניסות ויציאות של הסימולציה.
6. קומפלציה.
7. טעינת קוד.
8. קביעת זמן ריצה.
9. הרצת הסימולציה.

דגשים

1. יש לשים דגש על תוכנת הסימולציה כיון שזו כלי עיקרי לאיתור טעויות בכתיבת הקוד.
2. בסביבה זו כותב הקוד חייב לתת את הדעת על פרטים כגון: חיבור בין מערכות או סנכרון בין המערכות. בתעשייה יש לתת את הדעת גם על עמידה בתדרים גבוהים, אך אין הדבר נדרש כאן, בגלל הרכיבים האיטיים באופן יחסי.
3. פרק זה בנוי כ- tutorial על מנת להקל על הכרת הסביבה לצורך הסימולציה. חשוב מאוד שהתלמיד יכיר היטב וישלוט בסביבת העבודה לפני שממשיכים לפרקים הבאים. יש לתרגל כמה פעמים את נושא הסימולציה בתחומים לוגים פשוטים.

פרק 3: עקרונות השפה

יש להתעכב על כך ששפת V.H.D.L ייחודית בכך שיש בה מספרים המתאימים לחומרה.
בנוסף, קיימת חפיפה בין סוגי מספרים בשפה זו למספרים בשפות עיליות אחרות.

יעדים

1. הכרת סוגי המספרים
2. הבנת הצורך בסוגי המספרים השונים

דגשים

1. קביעת תחום קיום של כל מספר
2. המרה בין סוגי מספרים
3. הגדרת מערכים

פרק 4: מבוא לקומפילציה בסיסית

פרק זה מתמקד בקומפילציה, הכרת כלים נוספים של תוכנת QUARTOS, הפיכת קוד כתוב לשרטוט חשמלי והכרת הכלי RTL -register transmit logic.

דרכי הוראה

1. יצירת פרויקט ובחירת רכיב בהתאם ללוח העבודה בבית הספר
2. הכרת כלים גרפים להגדרת כניסות ויציאות של מערכת לוגית
3. קומפלציה של הקוד ותיקון שגיאות תחביר (אם קיימות)
4. הפיכת קוד למערכת שערים לוגים

פרק 5: מבוא בסיסי לסימולציה

לאחר הכרת כלי הסימולציה בפרק 2, בפרק זה נכתוב קוד אשר יבצע בדיקות על הקוד הנתון וימדוד את הזמנים הנדרשים בכלים של תוכנת Modelsim. בנוסף, נכיר מבנים נוספים בעזרתם ניתן לחולל אותו מסוג שעון ועוד.

יעדים

1. הבנת הקשר בין הקוד הנבדק לקוד הסימולציה
2. כתיבת קוד סימולציה test bench עם דרישות מורכבות יותר
3. כתיבת קוד סימולציה ע"י process ללא sensitivity list
4. הכרת כלי מדידה של זמנים בתוך הסימולציה
5. הגדרת זמני ריצת תוכנה בהתאם לדרישות הקוד

דגשים

1. כתיבת קוד סימולציה דורשת את הבנת הסביבה הפיסקלית בה הפרויקט אמור לפעול. לדוגמה, הפעלת מנוע דורשת פולס מסוג –PWM.
2. במערכת סינכרונית נדרש ליצור אות מסוג reset או פולס מסוג trigger לכל אחד מהאותות ויש לבדוק האם הם מחזוריים. אם לאו, האם ההתנהגות משתנה לאורך הזמן.

פרק 6: אבני בנייה לסינתזה

בפרק זה נעסוק בהבחנה בין כתיבה לצורך סינתזה לבין כתיבה לצורך סימולציה.

יעדים

1. הבנת ההבדל בין כתיבה לצורך סינתזה לעומת כתיבה לצורך סימולציה.
2. הכרת מבנים לסינתזה.
3. הכרת ה- process
4. הבנת משמעות של process לצורך מבנים אסינכרוניים וסינכרוניים.
5. הכרת אופרטורים מחוץ ל-process

דרכי הוראה

1. יישום מערכות לוגיות עם process
2. בדיקת התחביר בתוך ה- process – if ו-case-wait-for.
3. בדיקת תחביר מחוץ ל-process - after.when.

פרק 7: מכונת מצבים

מכונות מצבים הן כלי חשוב בפתרון בעיות לוגיות. מומלץ להקדיש תשומת לב רבה לנושא זה. בדרך כלל נתכנן מכונות מסוג MOOR או Mealy . הסיבה לכך היא שהיישום שלהן מתקיים בסביבת חיישנים. ל-Moor יש יתרונות רבים, כיוון שחלק ה-output מנותק מה-current וה-next. שינוי בו אינו מחייב שינוי בשניהם.

לא נעסוק כאן בשיטות קידוד. נושא זה חורג ממסגרת הלימוד

יעדים

1. אפיון של בעיה נתונה על ידי דיאגרמת בועה.
2. כתיבת קוד מכונת מצבים לפי Moor או Mealy על פי דיאגרמת בועות.

דרכי הוראה

1. כתיבת קוד מכונת מצבים
2. הגדרת מצבי מכונה על ידי constant
3. הגדרת מצבי מכונה על ידי type

פרק 8: מערכים ורכיבי זיכרון

בשפת V.H.D.L ניתן לתרגום מעריכים למערכת זיכרון. ברכיבי FPGA קיימים רכיבי זיכרון פנימיים וניתן לקבוע את התצורה שלהם באמצעות כלים מובנים או כתיבת קוד אשר יגרום ל- synthesizer לזהות את המבנה ולהשתמש ברכיבים הפנימיים. רוב הפרויקטים בלימודי תיכון דורשים זיכרונות קטנים אשר ניתנים באמצעות קוד.

יעדים

1. כתיבת מערכים חד ממדיים
2. כתיבת מערכים דו ממדיים
3. כתיבת מערכת דו ממדיים כפרוש למערכת זיכרון

דרכי הוראה

1. תכנון מערכים מגודל נתון
2. תכנון על ידי כלי תוכנת Quartus

פרק 9: פונקציות ופרוצדורות

פרק זה עוסק בכתיבת שפה גבוהה בדומה לכתיבת Object oriented. לצורך כך, יש לשים לב שנדרש ניסיון רב באפיון פרויקטים.

יעדים

1. הבחנה בין כתיבה לצורך סינתזה ולסימולציה בלבד.
2. הגדרת גדלים לסינתזה על פי גדלים שונים
3. הגדרת פונקציות לצורך ארגון הפרויקט ופונקציות לסינתזה.
4. הגדרת package והכללתו בפרויקט

דרכי הוראה

פרויקט אישי בהנחיה המורה

הדוגמה הבאה, המשתמשת בפונקציה ופרוצדורה, היא דוגמה לכתיבה מסדר גבוה. היסודות לכתיבה זו מובאים בפרקי הספר.

1. נגדיר package מסוג MyPackage .
2. בתוך ה-package נגדיר פונקציה בשם log2 .
3. כיוון שאין פעולת log בשפת V.H.D.L עלינו ליצור אותה וזה יתבצע באמצעות את לולאת for.
4. לולאת for רצה מ-i=1 עד הערך d שנקלט. i**2 היא פעולת חזקה.
5. הקורא יעקוב אחר הקוד. שימו לב, מבנים דומים קיימים גם בשפות אחרות.

```
package MyPackage is
 function log2 (constant N :integer) return integer;
end package MyPackage;

package body MyPackage is
 function log2 (constant d :integer) return integer is
 begin
   for i in 1 to d loop
               if ((2**i)>= N) then
                      return i;
               end if;
       end loop;
   return 1;
 end function log2;
end package body MyPackage;
```

קריאה ל- package מתבצע ע"י ההוראות הבאות :

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
USE ieee.std_logic_arith.all;
USE ieee.std_logic_unsigned.all;

library work;
use work.MyPackage.all;

ENTITY Distance IS
```

ההגדרות הבאות מגדירות תקופות סוג המשתנה time .

```
GENERIC (EchowidthPulse : time:=18 ms;
                      clockPirode    : time:=20 ns );

PORT
(
clk : IN STD_LOGIC;
rst : IN STD_LOGIC;
echo :IN STD_LOGIC;
UsDistance : out
```

ההגדרה הבאה של הווקטור מוגדרת ע"י ה- package שהוגדר

```
STD_LOGIC_VECTOR(log2(EchowidthPulse/clockPirode)-1 downto 0)
);
```

נספח - מדריך לאתר של ALTERA.COM

נספח זה מתאר את תהליך ההרשמה והורדת התוכנה אשר תשמש את התלמידים במהלך לימודי שפת V.H.D.L. בנוסף, תמצאו כאן מדריך להכרת הלוח וחיבורו, כך תוכלו למנוע טעויות וקצרים בבניית הפרויקטים.

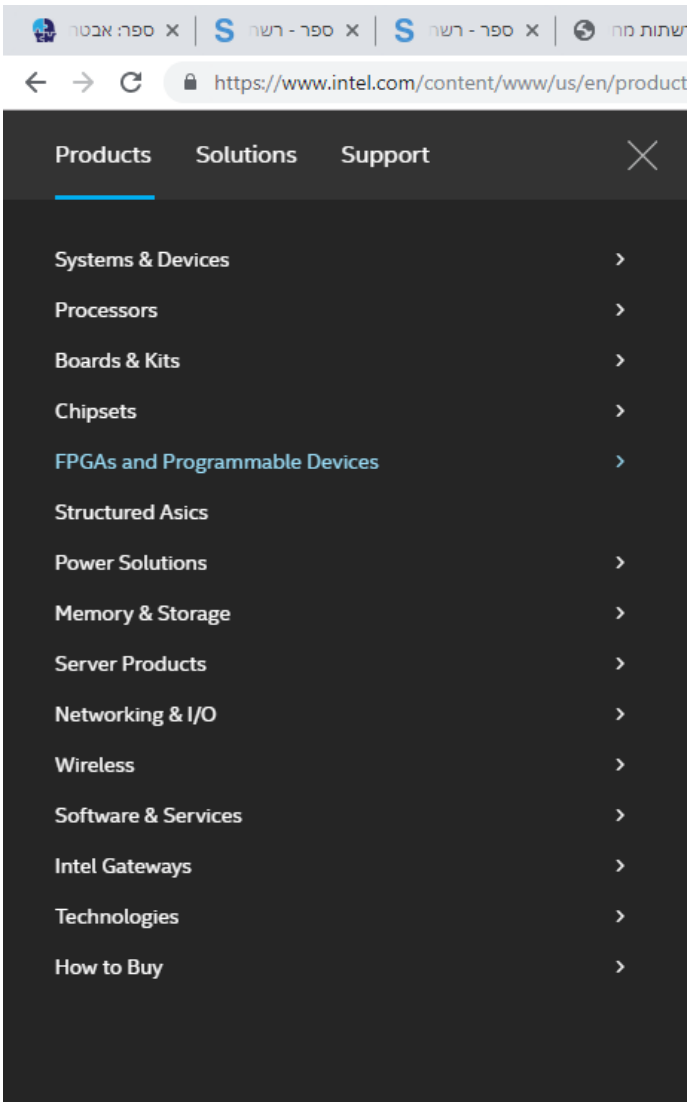
הורדת התוכנה:

להורדה יש להירשם באתר של Altera intel בקישור הבא:

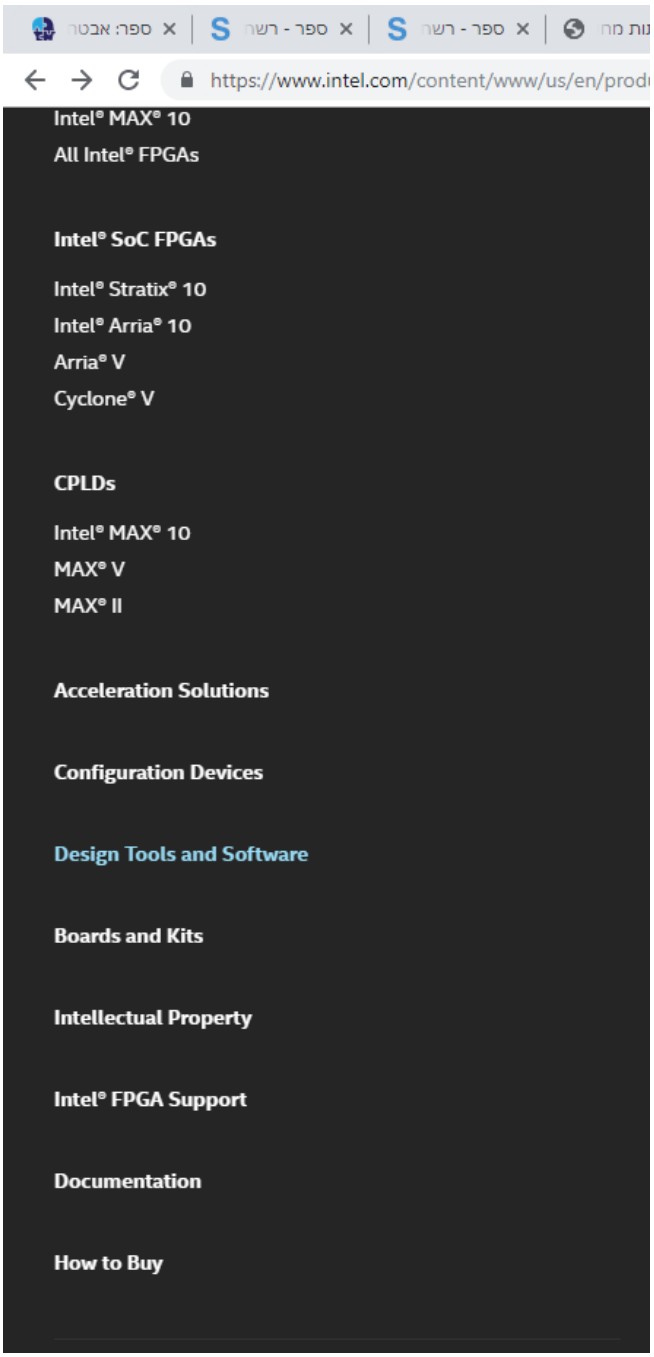
https://www.intel.com/content/www/us/en/products/programmable/fpga.html

תהליך ההרשמה:

1. לחצו על product ובחרו את הקישור FPGA and Programing Devices



2. בחרו באפשרות הבאה Design Tools and Software



3. בחרו באפשרות החינמית של התוכנה - Intel Quarus Prime Lineedition

לאחר ההתקנה עליכם לבחור את התוכנה בהתאם לחומרה.

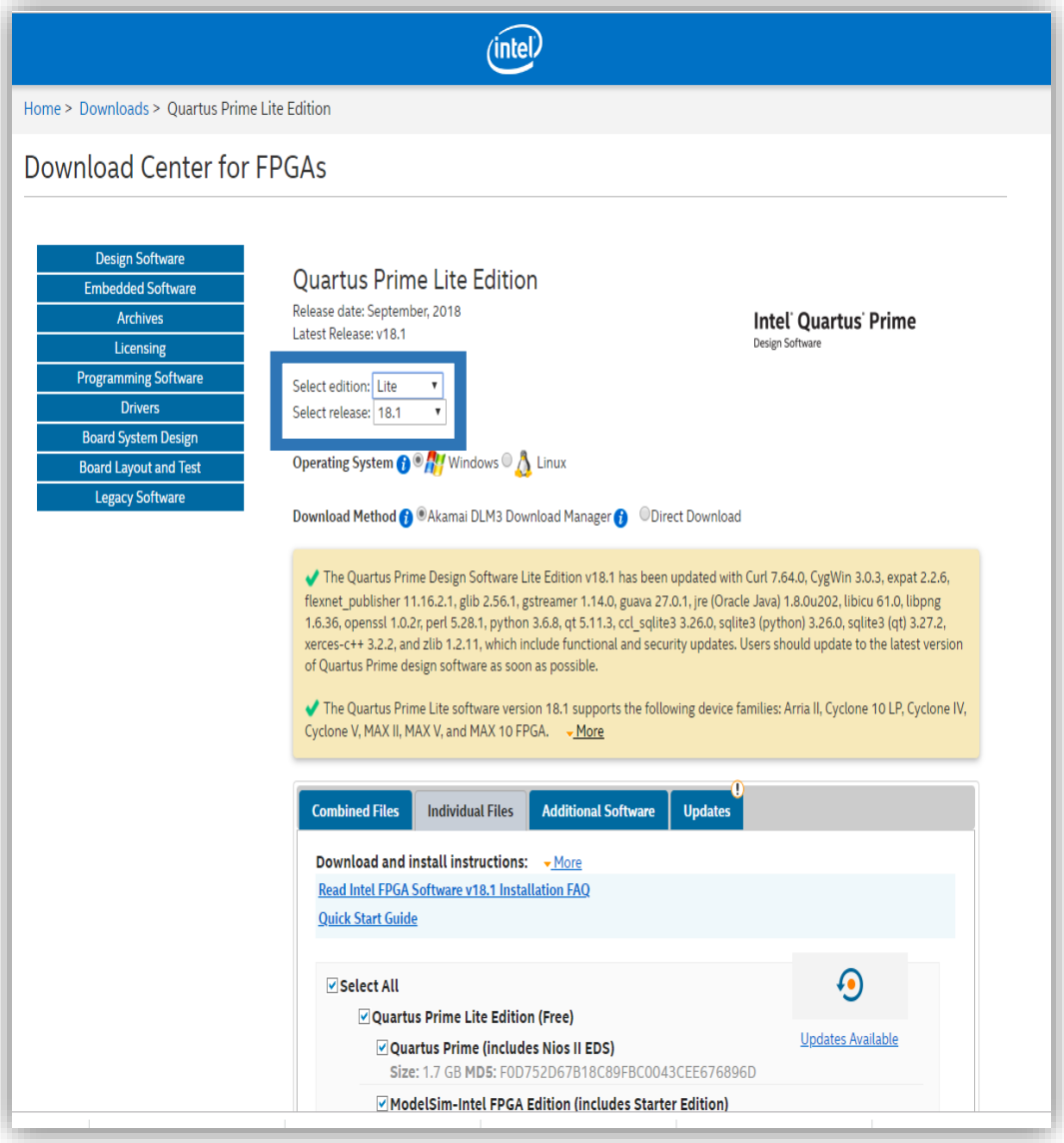
לכל לוח ישנו רכיב FPGA שונה. עליכם לבדוק מראש באיזה רכיב מדובר, על דפי הסבר המצורפים לערכה שנרכשה.

לצורך ההדגמה נשתמש ברכיב cyclone ii, הרכיב שנמצא על לוח שנקרא DE2.

התקינו לפי השלבים הבאים:

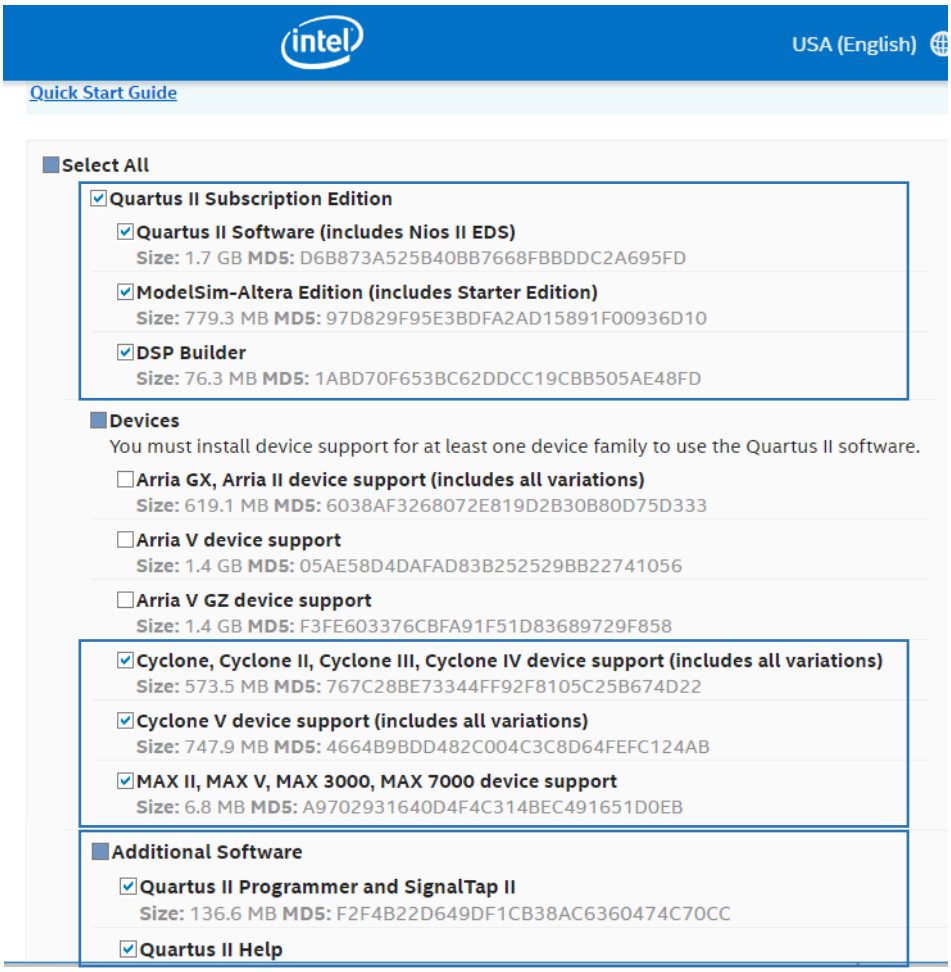
1. ב-select edition בחרו standard במקום Lite

   ב-Select release בחרו גרסה ל-13 במקום 18.

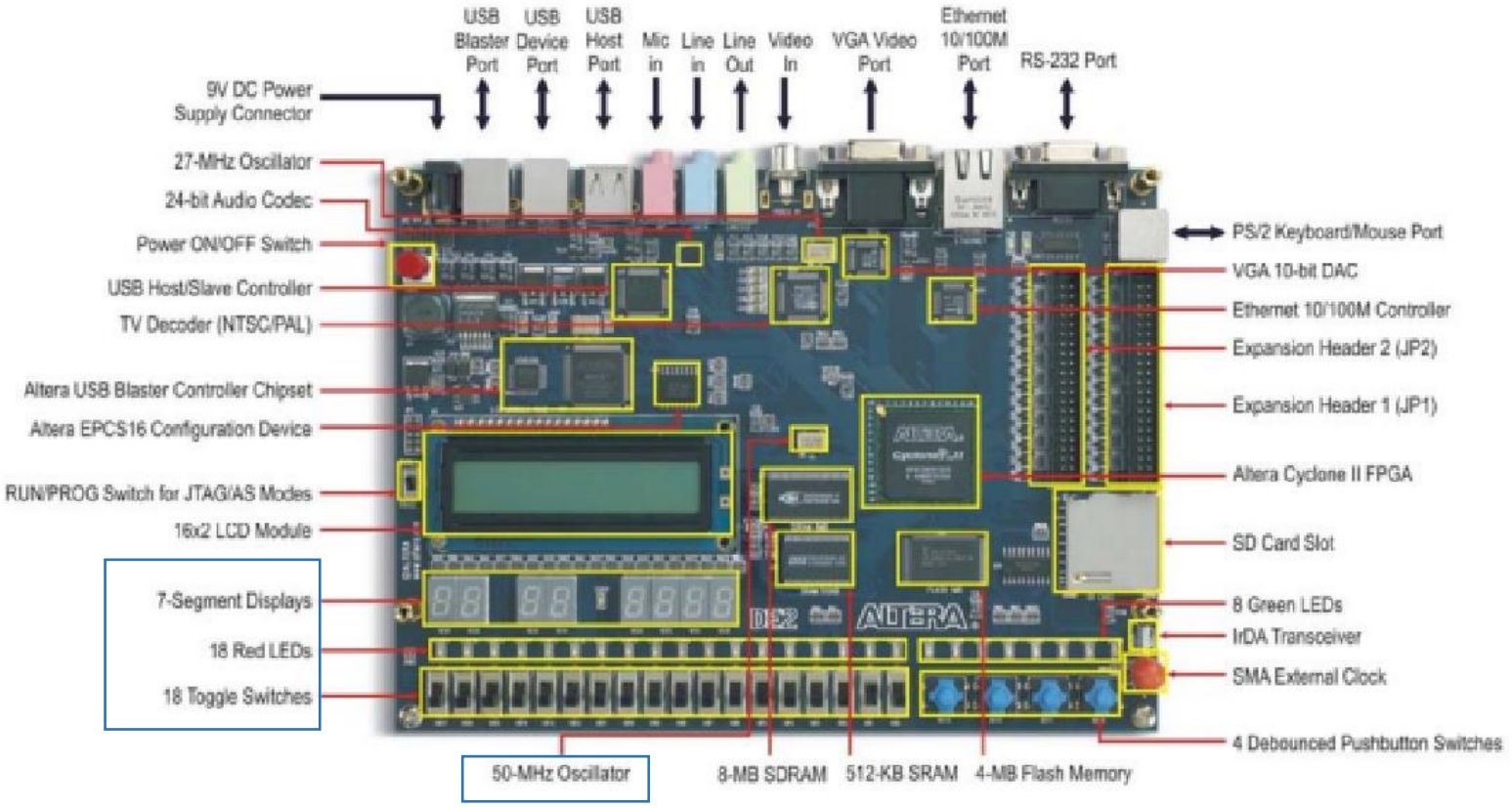


2. בשלב הבא תוכלו לסמן את הסעיפים הרצויים לכם. המלצתנו מסומנת מטה.

   שימו לב, שגם QUARTUS וגם Modelsim סומנו לצורך הורדה.



הכרת הלוח DE2



בצילום למטה מסומנים המתגים והנוריות ב- seven segment הגביש 50MHz.

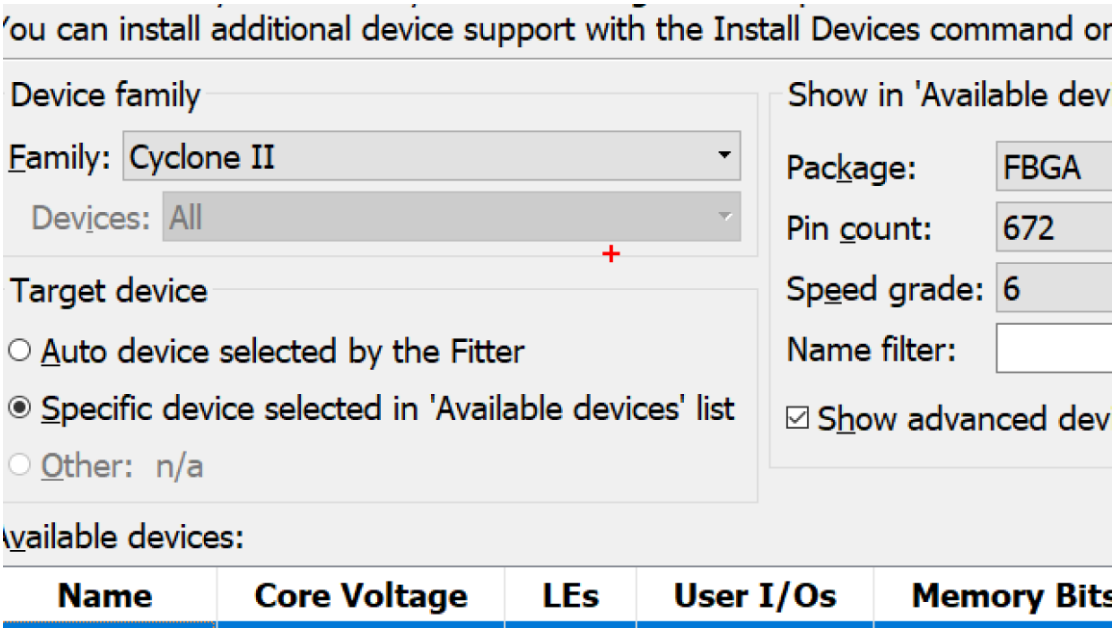
אלה חיבורים פיסיים ולכל חיבור יש ייצוג בטבלאות חיבורים לרגליים של ה- FPGA. מצורף דף של Altera DE2 Board. עלינו לאתר את הרגל שיש להגדיר בתוכנה. בדוגמה להלן המתג sw0 מחובר לרגל pin_N25 ברכיב. כאן ניתן לתרגל הדלקת מנורה על ידי מתג לצורך הכרת הלוח.

Signal Name	FPGA Pin No.	Description
SW[0]	PIN_N25	Toggle Switch[0]
SW[1]	PIN_N26	Toggle Switch[1]
SW[2]	PIN_P25	Toggle Switch[2]
SW[3]	PIN_AE14	Toggle Switch[3]
SW[4]	PIN_AF14	Toggle Switch[4]
SW[5]	PIN_AD13	Toggle Switch[5]
SW[6]	PIN_AC13	Toggle Switch[6]
SW[7]	PIN_C13	Toggle Switch[7]
SW[8]	PIN_B13	Toggle Switch[8]
SW[9]	PIN_A13	Toggle Switch[9]
SW[10]	PIN_N1	Toggle Switch[10]
SW[11]	PIN_P1	Toggle Switch[11]
SW[12]	PIN_P2	Toggle Switch[12]
SW[13]	PIN_T7	Toggle Switch[13]
SW[14]	PIN_U3	Toggle Switch[14]
SW[15]	PIN_U4	Toggle Switch[15]

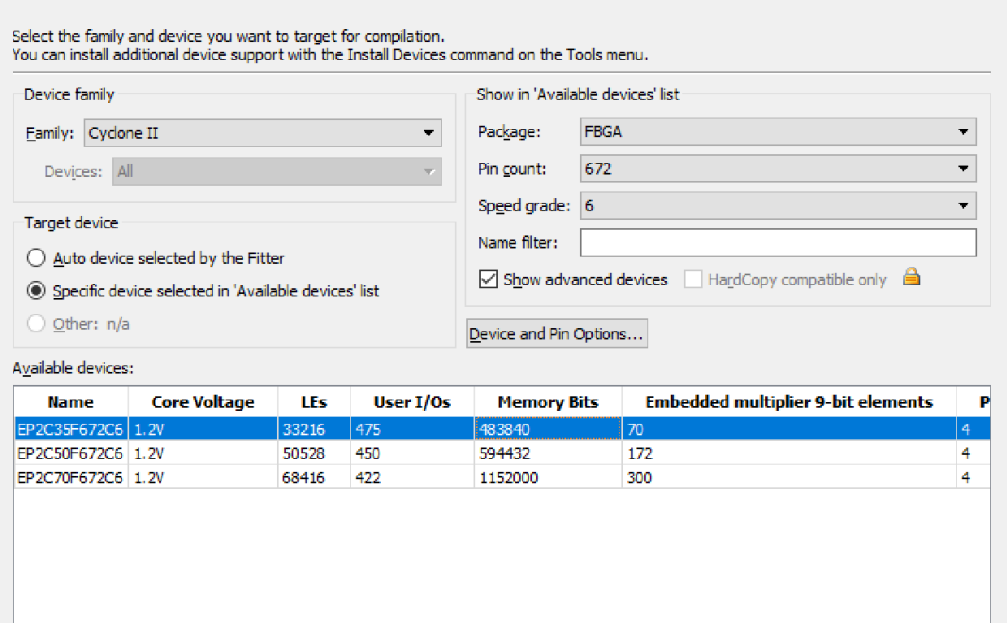
Signal Name	FPGA Pin No.	Description
LEDR[0]	PIN_AE23	LED Red[0]
LEDR[1]	PIN_AF23	LED Red[1]
LEDR[2]	PIN_AB21	LED Red[2]
LEDR[3]	PIN_AC22	LED Red[3]
LEDR[4]	PIN_AD22	LED Red[4]
LEDR[5]	PIN_AD23	LED Red[5]
LEDR[6]	PIN_AD21	LED Red[6]
LEDR[7]	PIN_AC21	LED Red[7]
LEDR[8]	PIN_AA14	LED Red[8]
LEDR[9]	PIN_Y13	LED Red[9]
LEDR[10]	PIN_AA13	LED Red[10]
LEDR[11]	PIN_AC14	LED Red[11]
LEDR[12]	PIN_AD15	LED Red[12]
LEDR[13]	PIN_AE15	LED Red[13]
LEDR[14]	PIN_AF13	LED Red[14]

תהליך הדלקת נורה באמצעות מתג

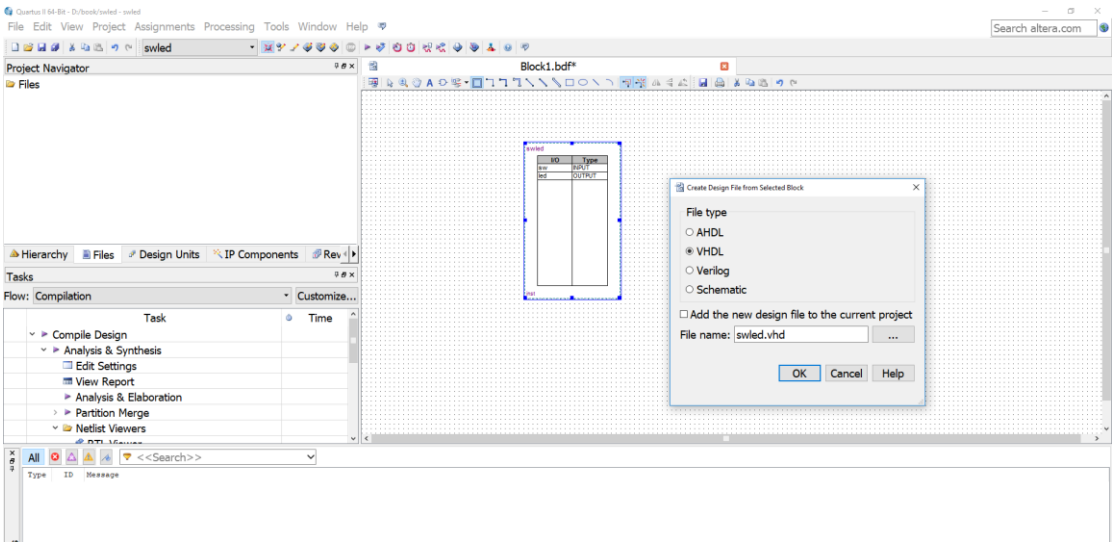
1. פתחו פרויקט כפי שלמדנו בפרק 1 וקראו לו swled.
2. בחרו את הרכיב המתאים כפי שמתואר בתמונה הבאה:

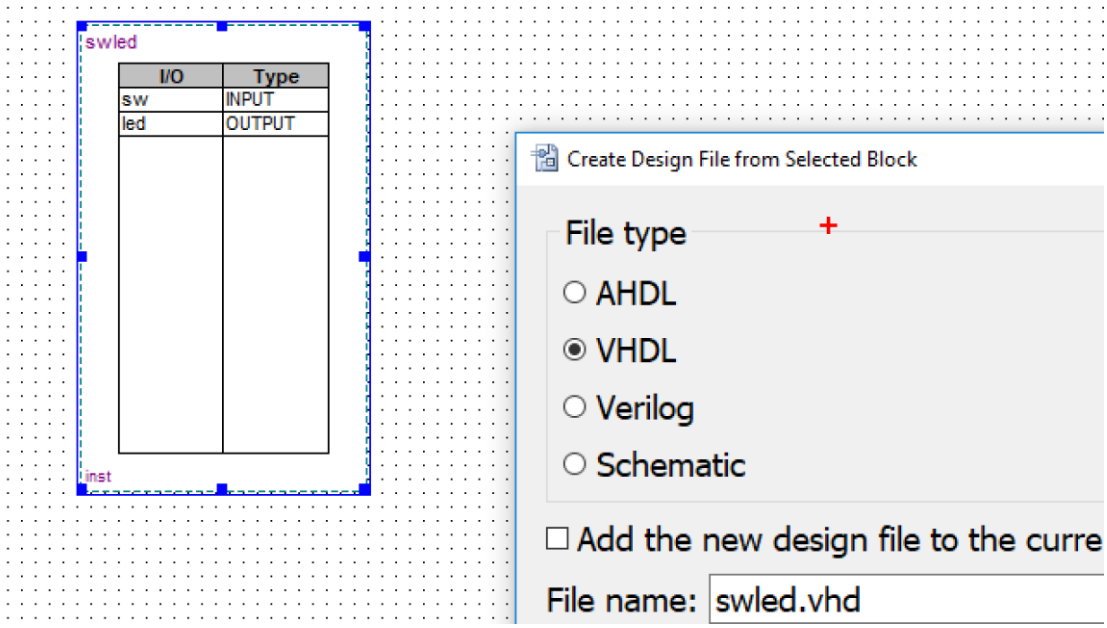


3. בחרו את הרכיב עליו צורבים:

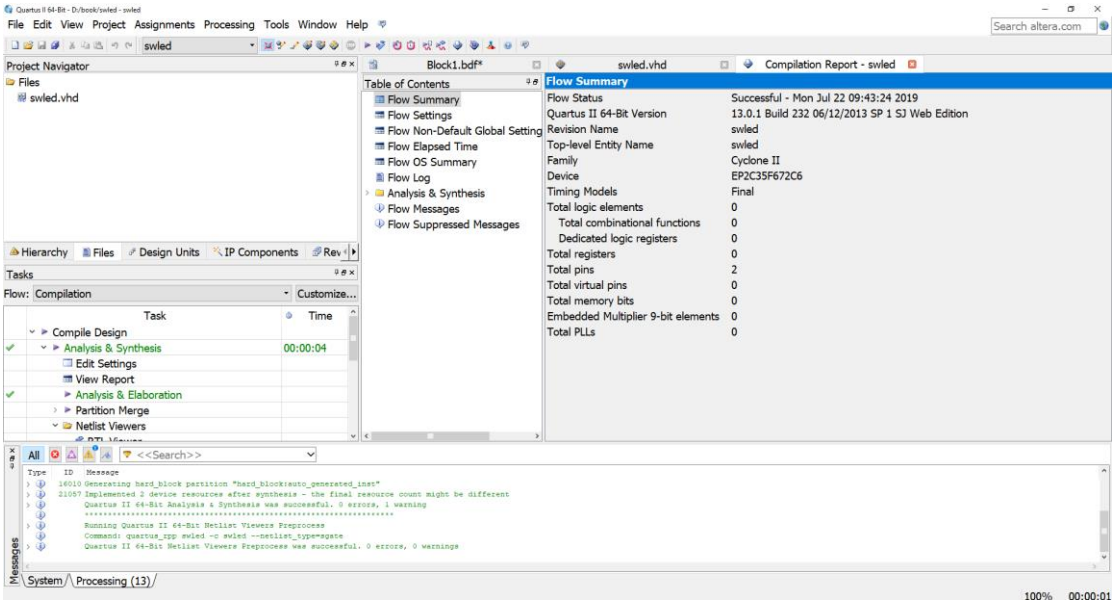


4. התוצאה שתתקבל:



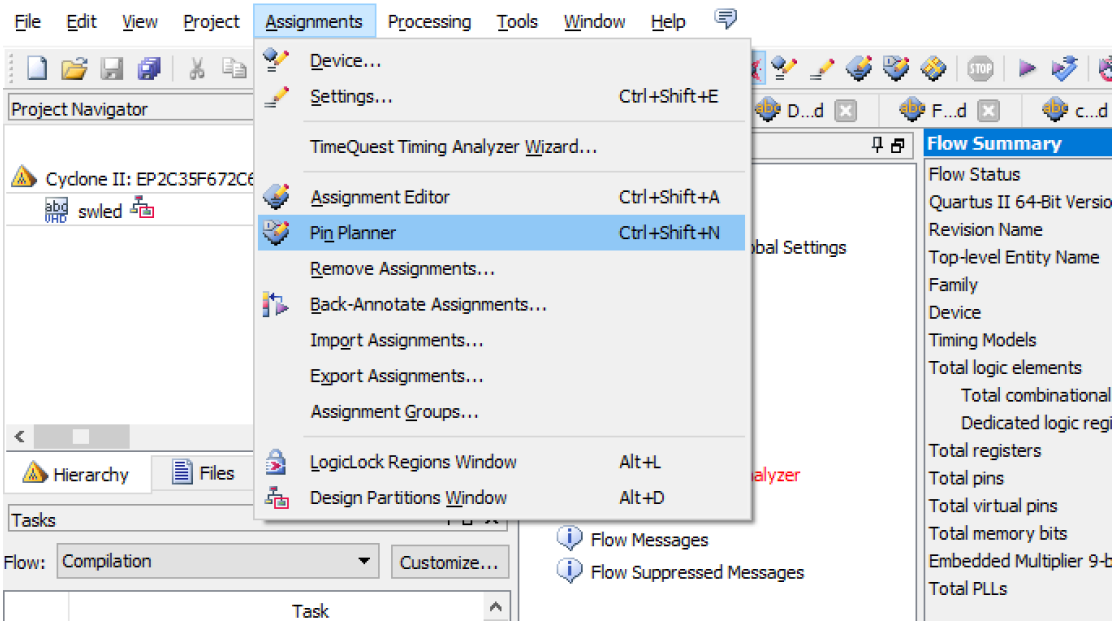


5. בסיום ההידור נקבל:

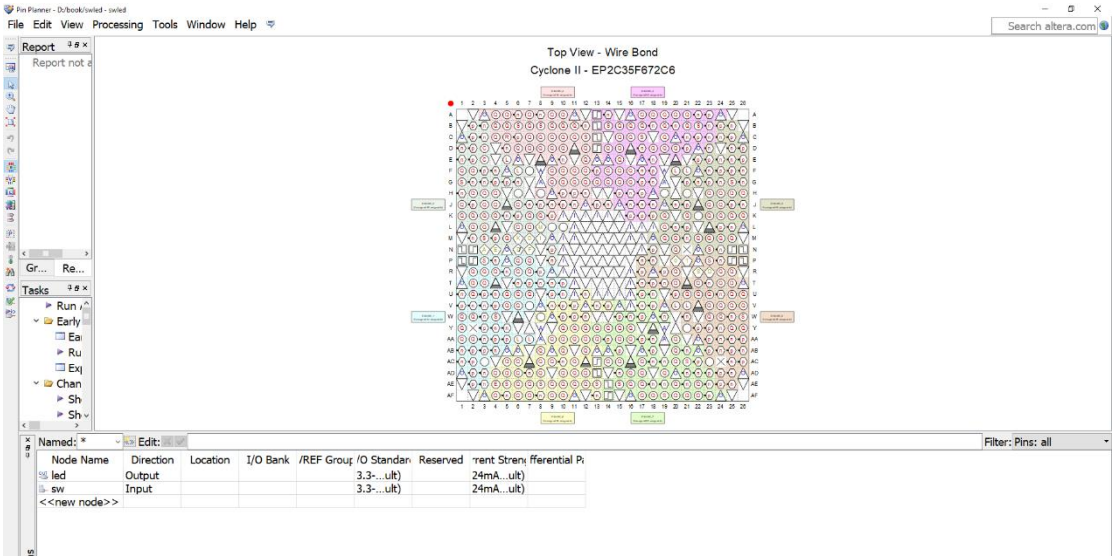


6. להלן ה- RTL של התכנון:

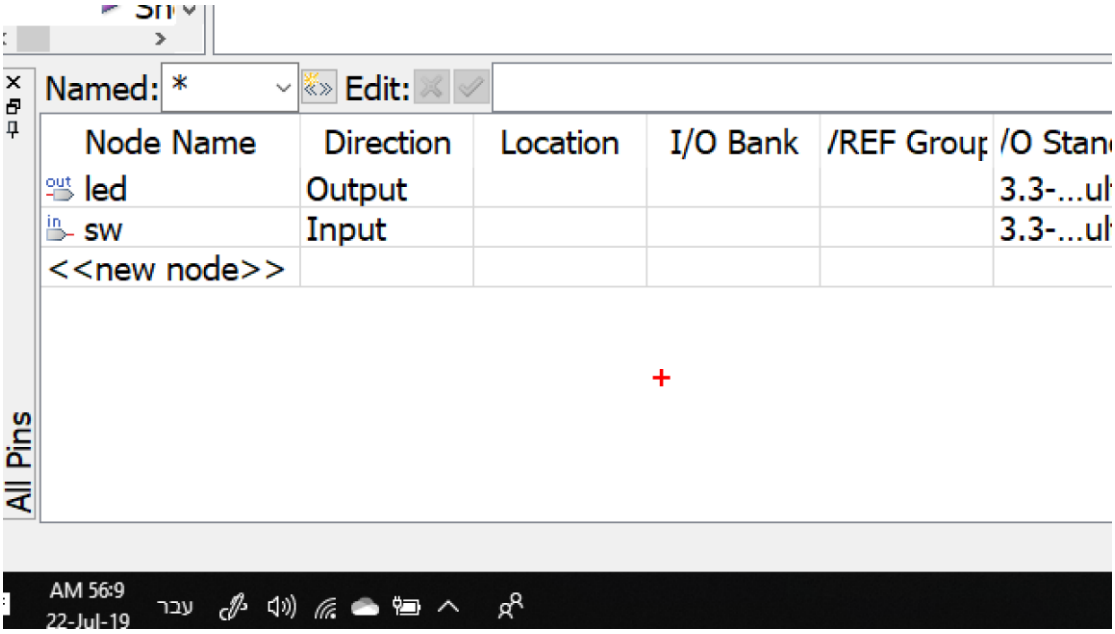
7. כעת נבצע קישור בין רגלי התכנון לבין רגלי הרכיב. מתוך תפריט ה-assignment נבחר ב- pin planer:



8. ונקבל:



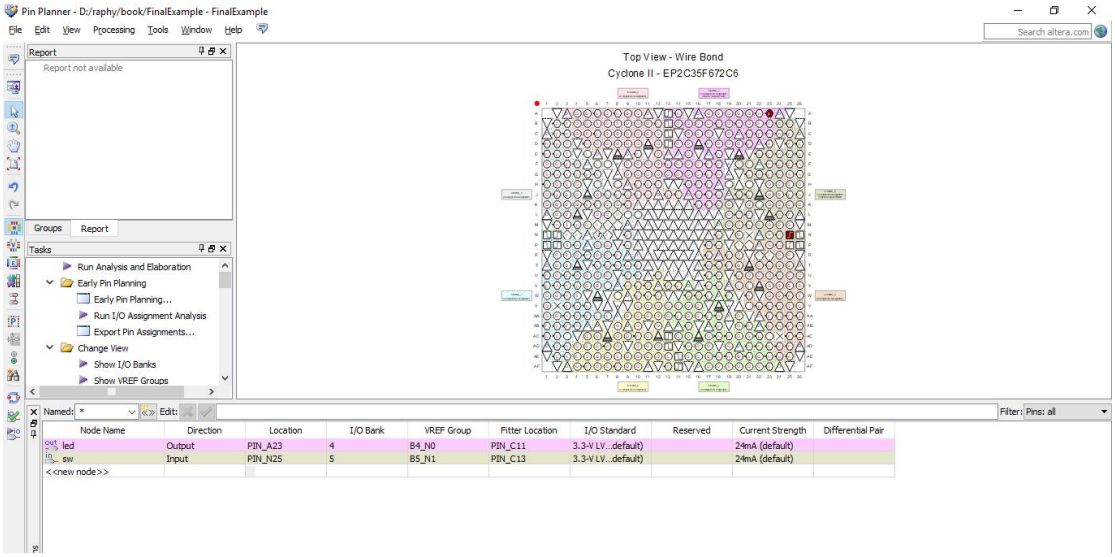
9. בדקו את הרגלים שהוגדרו בתכנון:



10. כפי שציינו למעלה, הרגלים על פי הטבלה

- Sw0=pin_n25
- Ledr0=pin_AE23

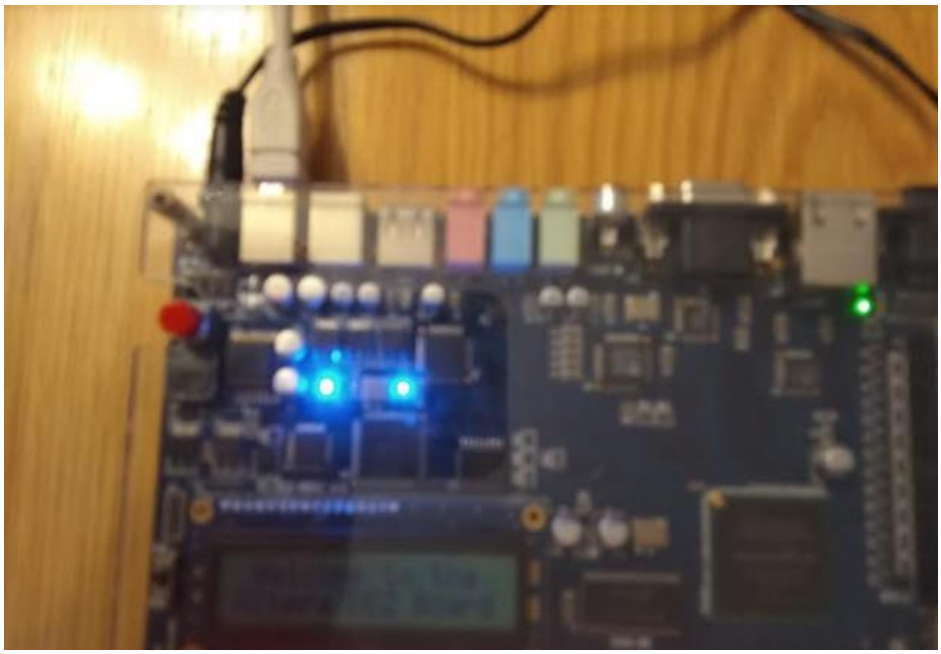
נכתוב אותם ב – location:



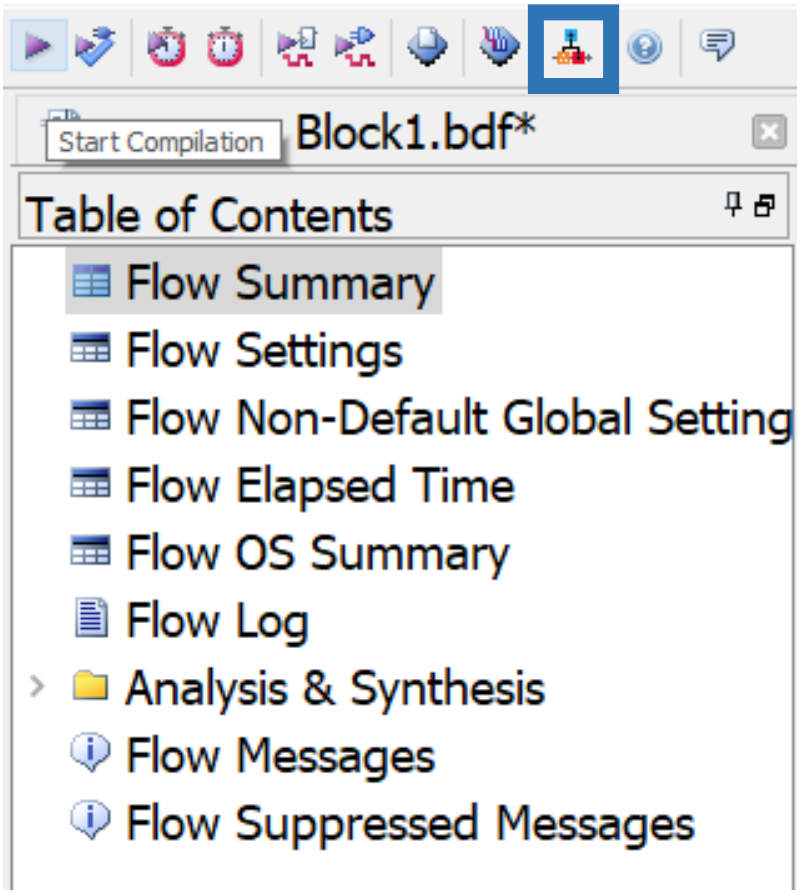
Signal Name	FPGA Pin No.	Description
LEDR[0]	PIN_AE23	LED Red[0]
LEDR[1]	PIN_AF23	LED Red[1]
LEDR[2]	PIN_AB21	LED Red[2]
LEDR[3]	PIN_AC22	LED Red[3]
LEDR[4]	PIN_AD22	LED Red[4]
LEDR[5]	PIN_AD23	LED Red[5]
LEDR[6]	PIN_AD21	LED Red[6]
LEDR[7]	PIN_AC21	LED Red[7]
LEDR[8]	PIN_AA14	LED Red[8]
LEDR[9]	PIN_Y13	LED Red[9]
LEDR[10]	PIN_AA13	LED Red[10]
LEDR[11]	PIN_AC14	LED Red[11]
LEDR[12]	PIN_AD15	LED Red[12]
LEDR[13]	PIN_AE15	LED Red[13]
LEDR[14]	PIN_AF13	LED Red[14]

Signal Name	FPGA Pin No.	Description
SW[0]	PIN_N25	Toggle Switch[0]
SW[1]	PIN_N26	Toggle Switch[1]
SW[2]	PIN_P25	Toggle Switch[2]
SW[3]	PIN_AE14	Toggle Switch[3]
SW[4]	PIN_AF14	Toggle Switch[4]
SW[5]	PIN_AD13	Toggle Switch[5]
SW[6]	PIN_AC13	Toggle Switch[6]
SW[7]	PIN_C13	Toggle Switch[7]
SW[8]	PIN_B13	Toggle Switch[8]
SW[9]	PIN_A13	Toggle Switch[9]
SW[10]	PIN_N1	Toggle Switch[10]
SW[11]	PIN_P1	Toggle Switch[11]
SW[12]	PIN_P2	Toggle Switch[12]
SW[13]	PIN_T7	Toggle Switch[13]
SW[14]	PIN_U3	Toggle Switch[14]
SW[15]	PIN_U4	Toggle Switch[15]

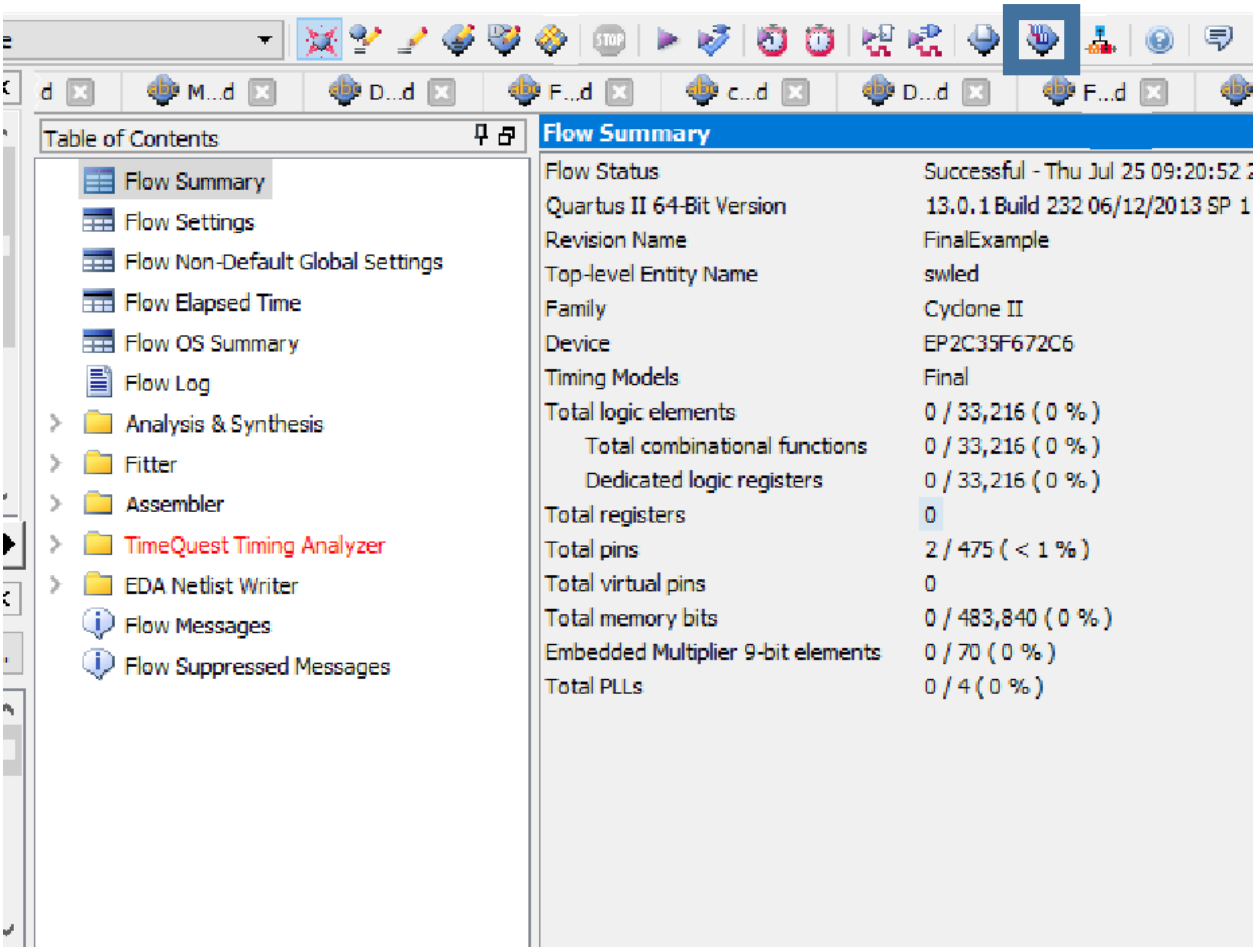
11. נוודא חיבור usb לערכה וחיבור החשמל:



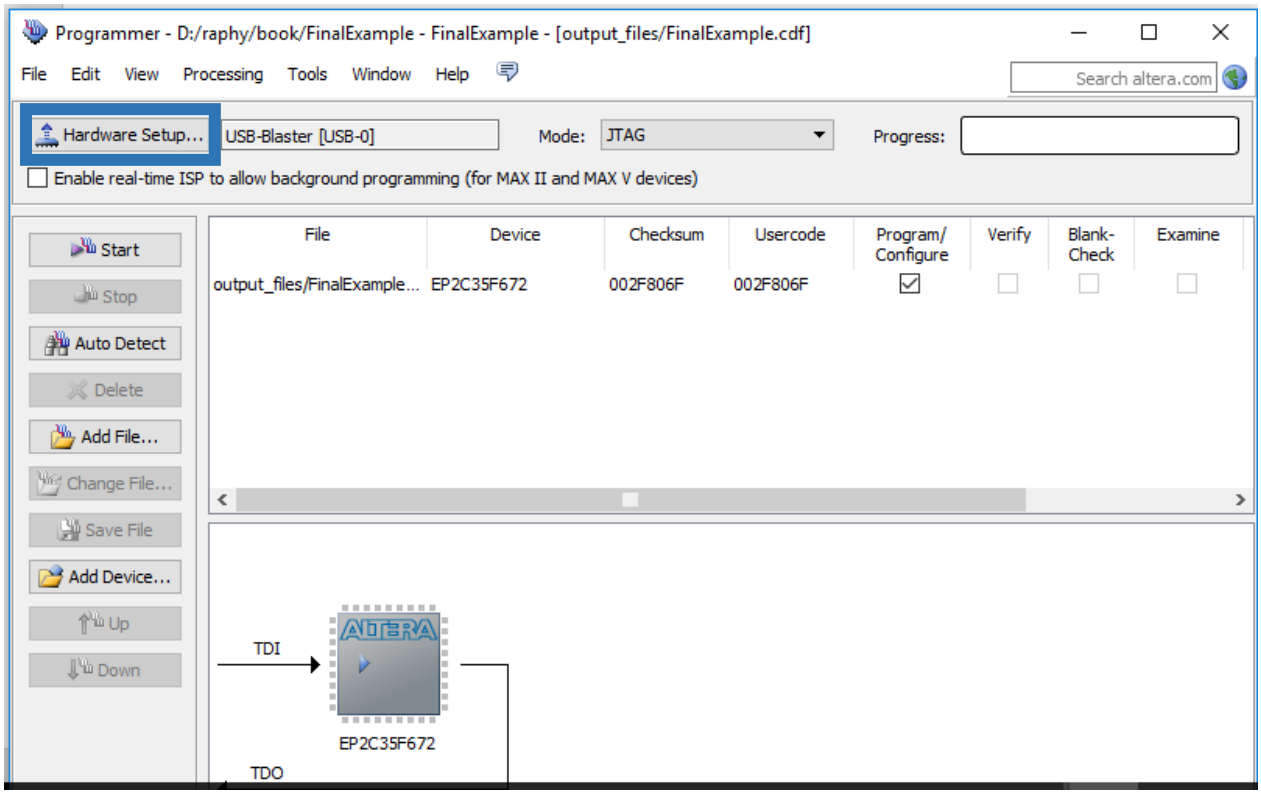
12. נבצע קומפילציה נוספת:



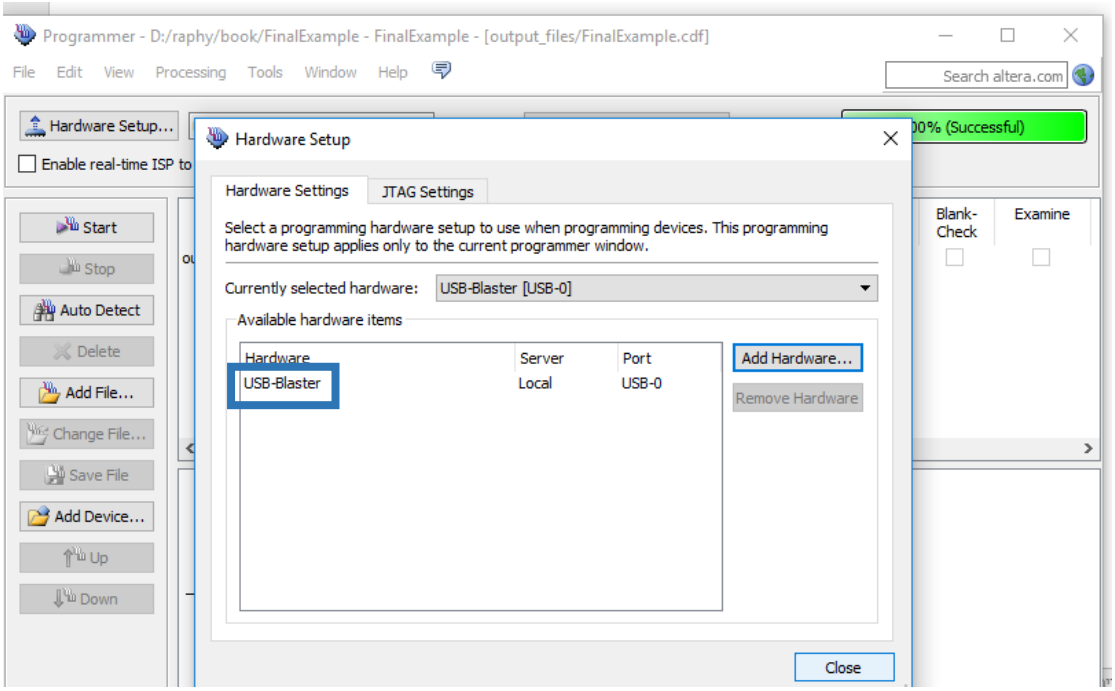
13. נבצע צריבה לרכיב מהצלמית:



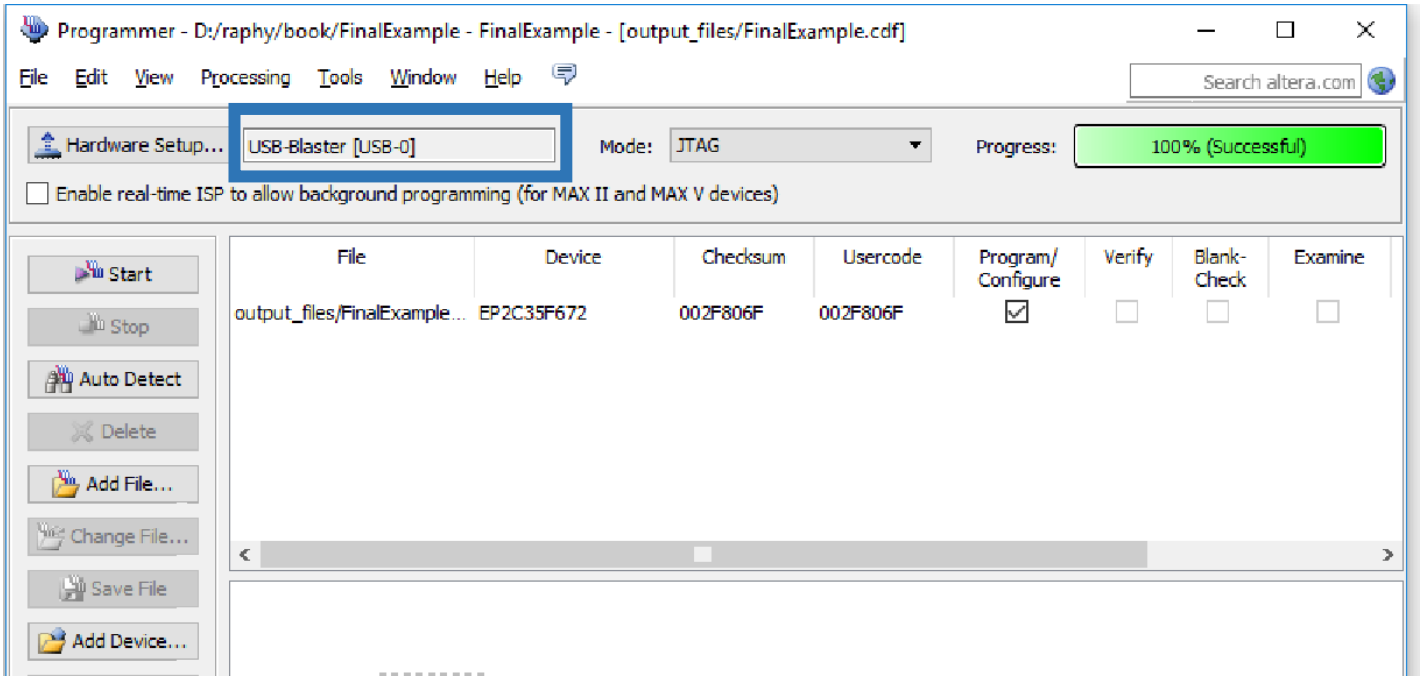
14. במידה שאינכם רואה את החומרה – הקליקו על hardware setup:



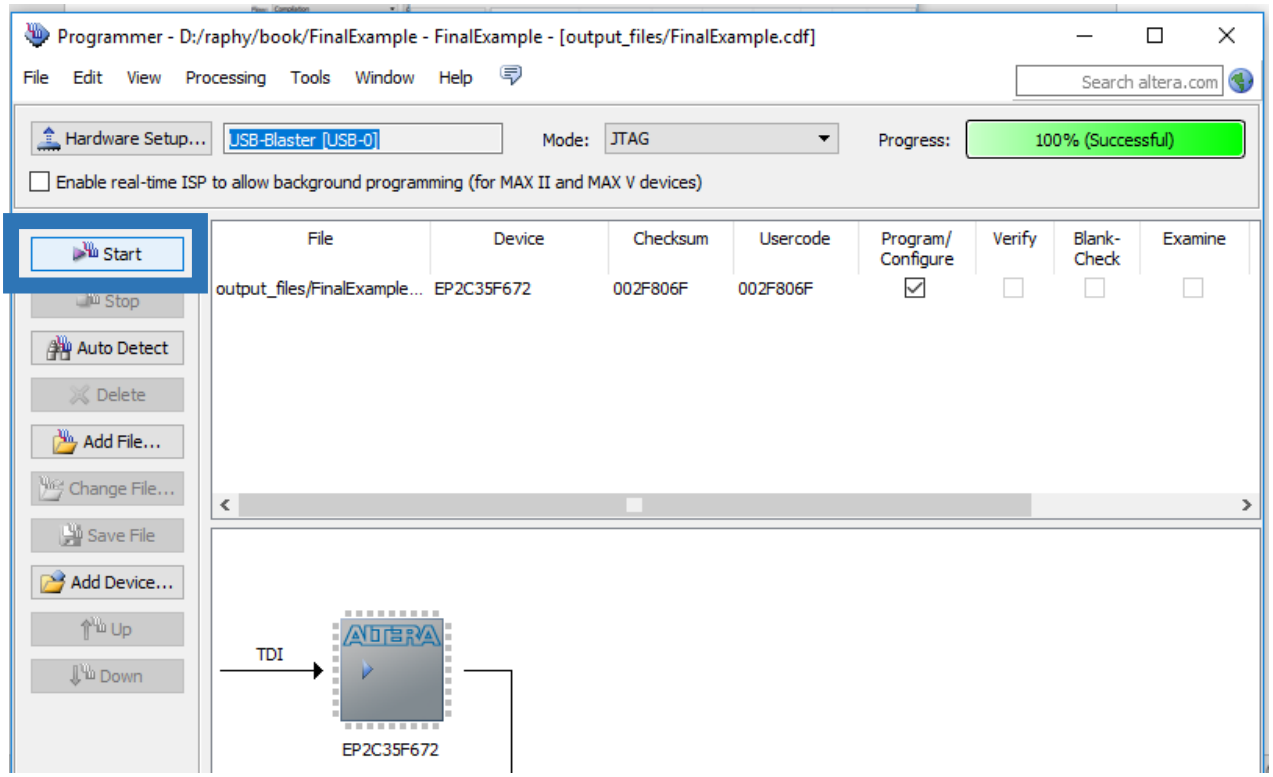
15. בחרו באפשרות Usb Blaster. שימו לב, חלון זה יעלה רק אם הערכה מחוברת ודלוקה:



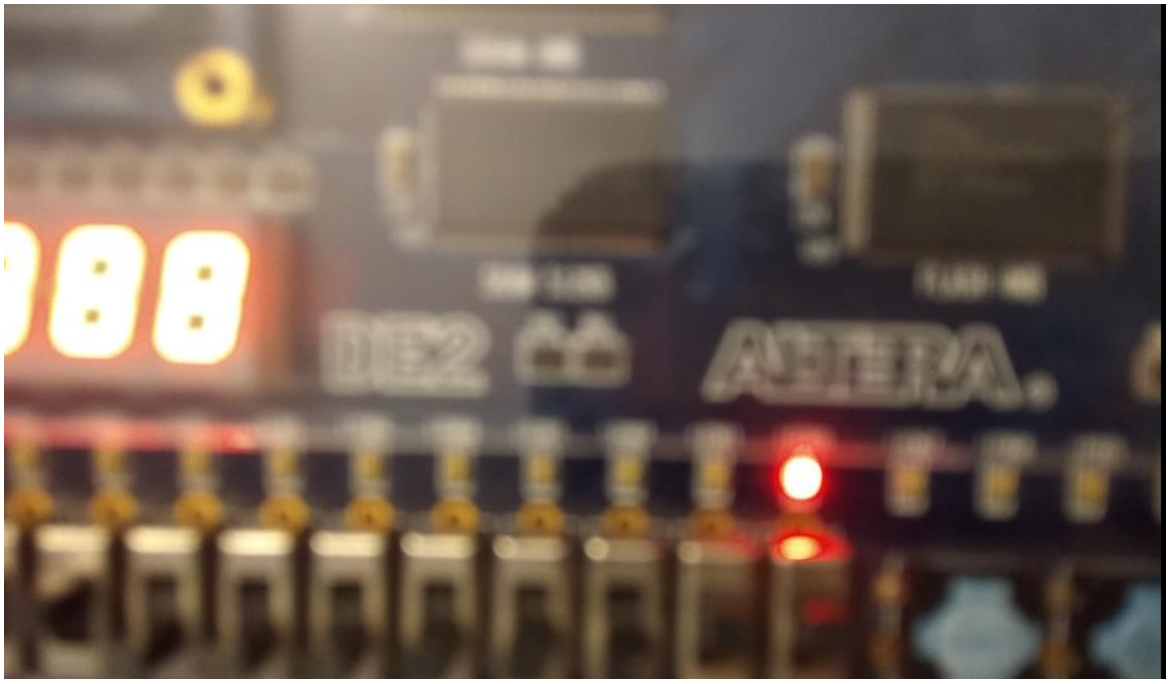
16. שימו לב שה- Usb Blaster מתקבל:



17. הקליקו על Start על מנת להתחיל צריכה:



18. בסיום התהליך תדלק הנורית הרצויה בהתאם לשינויים במתג:



אתר מומלץ:

https://www.nandland.com/vhdl/tutorials/index.html