

חיישני מרחק אולטרה סוני SRF05 + פסיקות

החיישן נראה באיור 1 .



איור 1 : חיישן מרחק אולטרה סוני. בצד שמאל srf05 ובצד ימין HC_SR04.

הסבר לפעולת החיישן ניתן למצוא באתר : <http://www.arikporat.com/projects/SRF05.doc>

1. מבוא - עיקרון מדידת מרחק

נחזור בקצרה על עקרון הפעולה. גל המשודר בתווך כלשהו, מתפשט במרחב ופוגע בעצם, מוחזר בחזרה אל המקור המשודר.

מהירות של גל קול תלויה בתווך בו עובר הקול, בטמפרטורה ובלחץ. בגובה פני הים, בטמפרטורה של 24 מעלות צלסיוס, מהירות הקול היא 1232 ק"מ/שעה שהם כ 344 מטר לשנייה.

חישוב המרחק מבוצע על ידי חישוב בעזרת הנוסחה: $S = V * t$ הזמן מרגע ששודר את אולטרה סוני ועד לרגע שמקבלים הד חוזר. נכפיל זמן זה במהירות הקול ואז נדע את המרחק הלוחך וחוזר. נחלק ב 2 ונמצא את הטווח.

ניתן לבצע את החישוב בדרך פשוטה יותר. אם מרחק העצם מחיישן המרחק הוא 1 מטר, אז גל הקול מבצע דרך של 2 מטר (הלוחך וחוזר) ולכן הזמן עבור מרחק של 1 מטר יהיה הדרך שגל הקול מבצע חלקי מהירות הקול, כלומר

$$t = s / v \quad \longrightarrow \quad t = 2 / 344 = 0.00581 = 5.81\text{Msec}$$

כלומר כ 5.8 מילי שניות לטווח של 1 מטר (2 מטר הלוחך וחוזר). או 58 מיקרו שניות ל 1 ס"מ.

עבור מרחק של 1 ס"מ נקבל 58 מיקרו שנייה. כלומר אם ניקח מונה שתדר פולסי השעון שיגיעו לספירה הם 1MHz אז עבור כל ס"מ של מרחק המונה יספור 58 פולסי שעון. נוכל לומר שניתן לאבחן מרחק של 1/58 של ס"מ.

לחיישן יש הדק דרבון (התנעה - Trig). זמן דופק הדרבון לפי הוראת היצרן צריך להיות מינימום 10 מיקרו שנייה. מרגע סיום הדרבון החיישן ישדר 8 מחזורים של אות אולטרה סוני, בתדר של כ 40 קילו הרץ- מעבר לתחום שמע של אדם.

הדק נוסף הוא דופק הד חוזר : רוחב הדופק הוא ביחס ליניארי למרחק העצם מהחיישן והוא שווה ל 5.8 msec עבור 1 מטר.

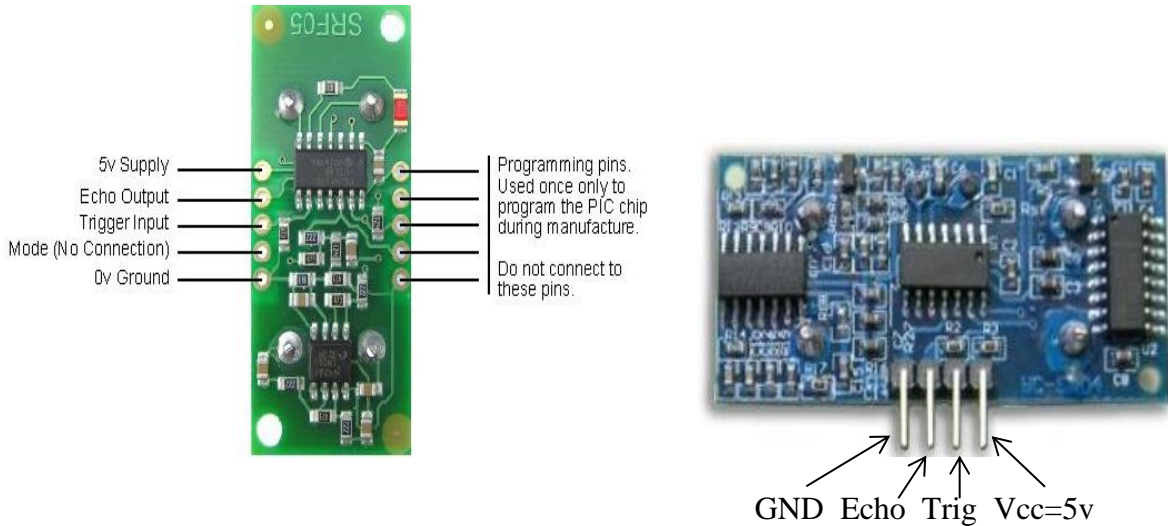
ניתן להשתמש בנוסחה:

$$2 / \text{מהירות הקול} * \text{הזמן של המצב הגבוה בהדק ECHO} = \text{טווח}$$

בחישובים ניקח את מהירות הקול לפי 340M/S

חיבורי ההדקים :

באיור 2 מופיעים חיבורי הרגלים של המעגל.

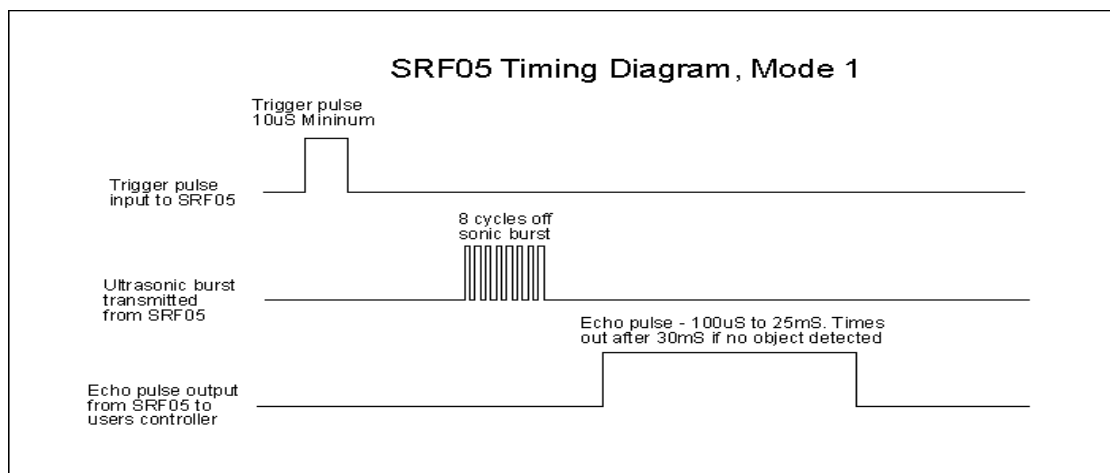


איור 2 : ההדקים של חיישן המרחק. משמאל srf-05 ומימין HC-sr04 .

לרכיב יש שימוש ב 4 הדקים:

- מתח ספק של 5 וולט
- אדמה (ה - של מתח הספק)
- פולס יציאה של ההד החוזר. נקרא Echo Output .
- פולס כניסה להפעלת המשדר. נקרא Trigger Input .
- את החיבורים מצד ימין באיור 2 , של הרכיב srf05 שבצד מימין לא מחברים כלל. הם נועדו לתכנון התחלתי של החיישן על-ידי היצרן.

תרשים זמנים של אופן פעולת החיישן :



שרטוט 1 : תרשים זמנים של אופן פעולת החיישן

בשרטוט מספר 1 ניתן לראות את תרשימי הפעולה בעזרת 3 צורות הגלים :

- בצורת הגל בחלק העליון, רואים את פולס ההפעלה של המערכת. זהו הדק כניסה. יש להכניס בהדק זה פולס שצריך להיות לפחות ברוחב 10 מיקרו שניות.
- מתחתיו רואים את הגל האולטרה סוני המשודר כתוצאה מפולס ההפעלה. אות השידור מורכב מ 8 מחזורים של גל מרובע בתדר של 40 קילו הרץ. מיד אחרי שידור 8 מחזורים אלו, מעגלי החיישן מחכים זמן קצר ביותר כדי שהמחזור האחרון יעזוב את המשדר ללא השפעות חוזרות והמערכת תתייצב ותירגע ואז הדק ההד החוזר Echo (עולה ל 1). כאשר המחזור השמיני מסתיים נוהגים להפעיל טיימר שיספור זמן עד שמגיע הד חוזר. עכשיו המערכת עוברת למצב של קליטה.
- צורת הגל התחתונה היא ההד החוזר להדק ה Echo. כל עוד לא נקלט הד אז המתח בהדק ה Echo הוא '1'. כשמגיע הד חוזר ממטרה, אז המתח עובר ל '0'. אם לא מגיע הד ממטרה, אז המעגל מוריד את מתח ההד החוזר אחרי 25 ms המתאים למרחק של 4.166 מטר. רוחב הפולס החוזר, תלוי אם כך, במרחק של ההד. ככל שהוא קרוב יותר אז רוחב הפולס קטן יותר. יש להמתין 10ms מסיום ההד הנקלט עד שנותנים פולס הפעלה חדש.

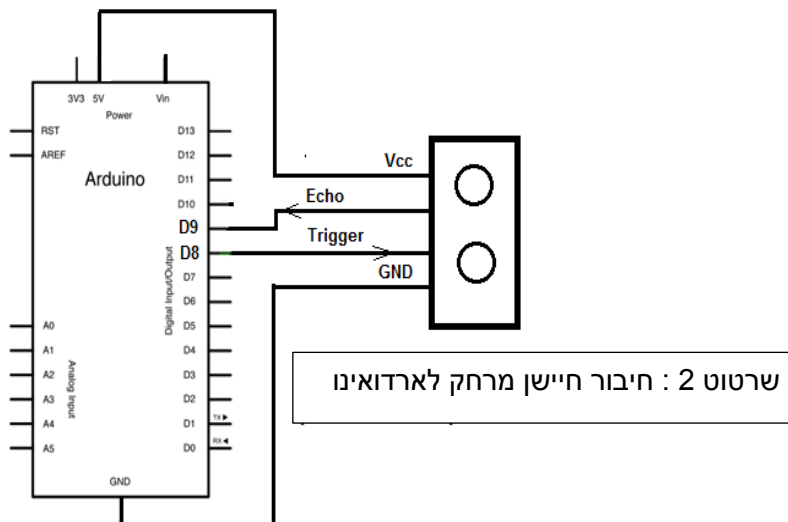
לרכיב SRF-05 יש אופן עבודה נוסף הנקרא MODE 2. באופן זה חוסכים הדק אחד. הדק ה TRIG (התנעה, דרבון) משמש אחרי ההתנעה כהדק ההד החוזר Echo. ניתן דוגמא בפרק התכנה גם על אופן עבודה זה. הסבר מפורט על החיישן ניתן למצוא באתר : <http://www.arikporat.com/projects/SRF05.doc>

מאפיינים

- מתח ספק כוח - 5 וולט .
- זרם – 30 ma אופייני, מקסימום 50ma (ב HC-SR04) זרם סטאטי (ללא שידור) 2ma .
- תדירות – 8 פולסי שידור בתדר 40KHz .
- טווח מקסימאלי – יותר מ 4 מטר.
- טווח מינימאלי – 2 ס"מ .
- זווית שידור לא גדולה מ 15 מעלות.
- פולס התנעה – פולס של מינימום 10 מיקרו שניות ברמת מתח TTL .
- פולס הד – אות TTL חיובי ברוחב התלוי בטווח.
- דיוק של 0.3 ס"מ .
- מידות קטנות – 43mm*20mm*17mm

2. חומרה

חבר את החיישן כמתואר בשרטוט מספר 2:



3. תכנה

בחישוב הטווח בפרק המבוא, ראינו שטווח של 1 ס"מ מתאים ל 58 מיקרו שניות.

נשתמש בתרגיל זה ב 2 פונקציות :

1. פונקציה (`micros()`) המחזירה ערך מטיפוס `unsigned long` שמציין את מספר המיקרו שניות מאז שהארדואינו התחיל להריץ את התכנית הנוכחית. המספר גולש ועובר ל 0 אחרי כ 70 דקות. בלוחות ארדואינו עם גביש של 16MHz יש לפונקציה זו רזולוציה של 4 מיקרו שניות (הערך המוחזר הוא תמיד מכפלה של 4). בלוחות עם 8MHz הרזולוציה היא של 8 מיקרו שניות.
2. פונקציה (`pulseIn()`) המודדת רוחב פולס (גבוה או נמוך) בהדק כלשהו. לדוגמא : אם הערך הוא HIGH - גבוה, אז הפונקציה מחכה שההדק יעלה ל HIGH ומתחילה למדוד זמן במיקרו שניות. היא עוצרת את הזמן כאשר ההדק יורד ל LOW - נמוך. הפונקציה מחזירה את אורך הזמן של הפולס במיקרו שניות. ניתן לרשום כמה זמן נרצה לחכות עד שההדק יעלה/יורד לגבוה/נמוך (אם נרשום HIGH/LOW). הפונקציה מחזירה 0 אם אין התחלת פולס בפרק הזמן שרשמנו. התזמון של פונקציה זו נקבע בצורה אמפירית ויכול לגרום לשגיאות בפולסים ארוכים. כדאי לעבוד עם פולסים באורך זמן של 10 מיקרו שניות עד 3 דקות.

בתחביר SYNTAX : 2 אפשרויות

```
pulseIn(pin, value)
pulseIn(pin, value, timeout)
```

pin הוא מספר ההדק. **value** - טיפוס הפולס שיש לקרוא : או HIGH או LOW (מטיפוס `int`).
timeout - הוא אופציונאלי ובו ניתן לקבוע כמה מיקרו שניות לחכות להתחלת הפולס. ברירת המחדל הוא שנייה אחת. הערך החוזר הוא מטיפוס `unsigned long` הוא מספר המיקרו שניות או 0 אם לא התחיל פולס לפני ה `timeout` שקבענו.

תכנית להפעלת חיישן מרחק

התוכנית שלהלן מדגימה שימוש בחיישן מרחק. היא מציגה על גבי המוניטור את המרחק בס"מ מהחיישן אל מטרה.

```
int trig=8, echo=9; // קביעת מספר ההדקים של הדרבון ושל ההד החוזר

void setup() {
  pinMode(trig,OUTPUT); // קביעת הדק 8 כמוצא. הדק הדרבון - התנעה - למבוא החיישן
  pinMode(echo,INPUT); // קביעת הדק 9 כמבוא. הדק לקבלת הד חוזר
  Serial.begin(9600); // קצב התקשורת הטורית עם המוניטור
}

int time, distance; // הגדרת 2 משתנים שיכילו את הזמן והמרחק

void loop() {
```

```

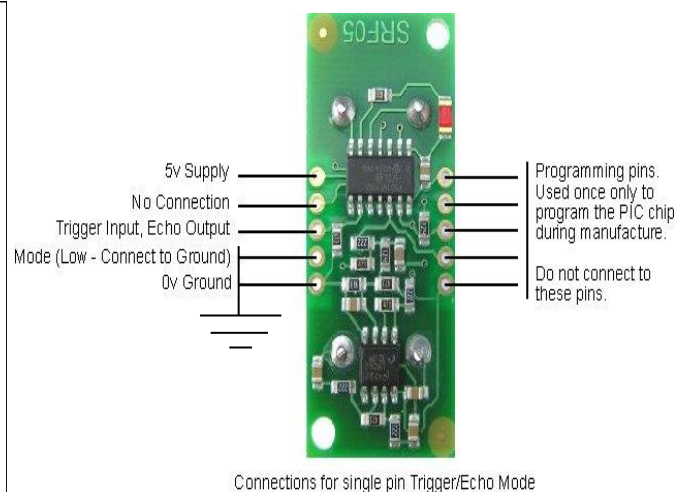
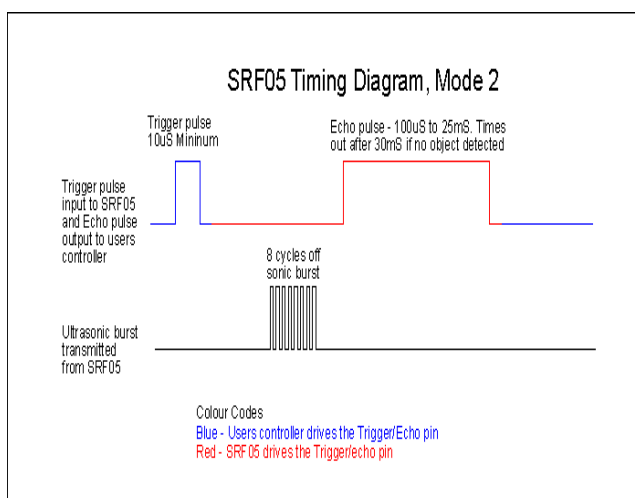
digitalWrite(trig,HIGH); // '1' לנו מעלים את הדק 8 ל
delayMicroseconds(10); // למשך 10 מיקרו שניה
digitalWrite(trig,LOW); // '0' מורידים את הדק 8 יורד ל
//while(digitalRead(echo)==0); // '1' יעלה ל Echo המתנה שהדק
time=pulseIn(9,HIGH); // קרא אל המשתנה time את הערך ב micro second מרגע העליה ל 1 עד 0
//while(digitalRead(echo)==1); // המתן עד לירידת ה Echo ל '0'
// time=micros( ) - time; // הצב במשתנה time את פרק הזמן שחלף
// בפעם הבאה שנקרא שוב נ
distance=time/58; // חשב את המרחק בס"מ
Serial.print ("distance = ");
Serial.println(distance); // הדפס במוניטור את המרחק ועבור שורה
delay(100);
}

```

שימוש באופן 2 של ה SRF-05 - Mode 2

כאשר בונים פרויקט מורכב שבו יש גם חיישן, גם תצוגה, גם לוח מקשים ועוד רכיבים נוספים, נוצרת מצוקה של הדקים כי כמות ההדקים של הארדווינו מוגבלת. יש אפשרות לחסוך הדק אחד על ידי הפעלת החיישן באופן 2. לשם כך יש לבצע את הדברים הבאים:

- לחבר לאדמה את ההדק כפי שנראה בשרטוט מספר 3.
- את הדק ההד החוזר Echo לא לחבר ולהשאיר באוויר.
- הדק ההתנעה TRIG ישמש גם כהדק ההד החוזר - Echo.



שרטוט מספר 3 : חיבור חיישן SRF-05 ב MODE 2 עם תרשים הזמנים באופן זה.

כדי שההדק ישמש כהדק מוצא כאשר הוא משמש כהדק TRIG וגם כהדק קלט כאשר הוא משמש כהדק Echo, יש להפוך את מצב ההדק בתכנה ממוצא למבוא ושוב למוצא כאשר הוא משמש ל-TRIG.

נרשום תכנית המפעילה את החיישן עם הדק משותף להתנעה TRIG ולהד חוזר – Echo. הדק זה מחובר ליציאה דיגיטאלית מספר 8 - D8. את קריאת המרחק נציג במוניטור.

```
#define TrigEchoPin = 8 // הדק משותף גם להתנעה וגם להד חוזר
void setup() {
  Serial.begin(9600); // אתחול התקשורת הטורית של המוניטור
}
void loop()
{
  long distance; // הגדרת משתנה
  pinMode(TrigEchoPin, OUTPUT); // הגדרת ההדק כמוצא
  delayMicroseconds(2); // הורדת ההדק ל 0 ל 2 מיקרו שניות כדי להיות בטוח שהוא 0
  digitalWrite(TrigEchoPin, HIGH); // התחלת ההתנעה – העלאה ל 1
  delayMicroseconds(10); // המתנה של 10 מיקרו שניות
  digitalWrite(TrigEchoPin, LOW); // סיום פולס התנעה – הורדת ההדק ל 0
  pinMode(TrigEchoPin, INPUT); // החלפת ההדק לקלט
  // נקרא לפונקציה המעבירה את אורך הפולס מרגע שההדק עולה ל 1 (גבוה) עד שירד ל 0
  distance = pulseIn(TrigEchoPin, HIGH)/58;
  Serial.print("Distance = ");
  Serial.print(distance);
  Serial.println(" cm");
  delay(100); // השהייה של עשירית שנייה שהמוניטור יספיק להדפיס
}
```

בשתי התוכניות שכתבנו המיקרו בקר עסק במדידת המרחק ושליחת התוצאה של המרחק להדפסה במסך המחשב בתכנית המוניטור. מרגע שהמיקרו סיים לתת בהדק TRIG את פקודת ההתנעה (דרבון), מופעלת הפונקציה pulseIn() ומחכים שהדק ההד החוזר יעלה ל 1 וירד ל 0 כאשר תתגלה מטרה. רק אחרי שהגל המשודר פוגע במטרה וחוזר, ההדק יורד ל 0 והזמן במיליוניות שנייה מועבר אל המשתנה distance. במשך כל הזמן שהפונקציה עובדת המיקרו למעשה ממתין ולא מבצע כל פעולה אחרת. אם זוהי מטרת התרגיל אז דיינו והתכנית שכתבנו פועלת כראוי. אולם, בהרבה פרויקטים מד המרחק משמש רק כרכיב אחד בתוך מכלול גדול יותר והזמן הוא משאב חשוב בתוך התכנית הגדולה של הפרויקט.

קיימת אפשרות לנצל את פרק הזמן שבו המיקרו בקר ממתין שרגל ההד החוזר – Echo – תרד ל 0 ולעשות בינתיים פעולות נוספות. הדבר אפשרי בעזרת פסיקה Interrupt. בפרק המבוא לארדואינו הפסיקה. נזכיר כאן כיצד מפעילים את פונקציות הפסיקה.

הפונקציה attachInterrupt ()

הפונקציה מציינת לאיזו תכנית טיפול - ISR - יקרא המיקרו בקר כאשר קורית הפסיקה. התחביר :

attachInterrupt(interrupt, ISR, mode)

interrupt – מספר הפסיקה .

ISR – שם הפונקציה שיש לקרוא לה (התכנית שתבוצע) כשקורית פסיקה . הפונקציה לא מקבלת פרמטרים ולא מחזירה ערך .

mode – מגדיר מה צריך להיות בהדק כדי לקבל פסיקה. יש 4 אפשרויות :

- א. **LOW** – להפעיל פסיקה כאשר בהדק יש 0 .
- ב. **CHANGE** – להפעיל פסיקה כל פעם שיש חילוף מצב בהדק.
- ג. **RISING** - להפעיל פסיקה כשיש עליה מ 0 ל 1 .
- ד. **FALLING** - להפעיל פסיקה כשיש ירידה מ 1 ל 0 .

הפונקציה איננה מחזירה ערך.

לדוגמה : הפקודה `attachInterrupt (1,monitor,FALLING);` אומרת שמדובר בפסיקה מספר 1 (מתחברת להדק D3) , כשהיא תופעל תבוצע פונקציה שנקראת `monitor` . הפסיקה תקרה אם יש בהדק ירידה מ 1 ל 0 .

דוגמה :

נחבר מפסק להדק דיגיטאלי 2 של הארדואינו (שישמש אותנו כהדק פסיקה מספר 0) . במצב רגיל יש בהדק זה '1' . לחיצה על המפסק תעביר '0' אל ההדק. נכתוב תכנית שתהבהב את מצב ה led שבהדק 13 שבארדואינו לפי הלחיצות במפסק. כל לחיצה תחליף את מצב ה led. לחיצה ראשונה תדליק אותו, לחיצה שנייה תכבה אותו, הלחיצה השלישית תדליק אותו וחוזר חלילה.

```
#define pin 13 // הגדרה של מספר ההדק שבו מחוברת ה led

volatile int state = LOW; // הגדרת משתנה מטיפוס שלם שערכו יכול להשתנות בפסיקה
void setup()
{
  pinMode(pin, OUTPUT); // הגדרת הדק 13 כמוצא

  // נגדיר פסיקה שמספרה 0 (מתחברת להדק דיגיטאלי 2), שתפעיל את הפונקציה blink והיא תופעל
  // כאשר יש שינוי בהדק מ 0 ל 1 או מ 1 ל 0
  attachInterrupt(0, blink, CHANGE);
}

void loop()
{
  digitalWrite(pin, state); // הוצאת הערך שבמשתנה ל led שבהדק 13
}
// ----- פונקציית הפסיקה אליה מגיעים בכל פעם שיש שינוי בהדק D2 -----
void blink()
{
  state = !state; // הפוך מצב המשתנה . אם היה בו 0 יהיה 1 ולהפך
}
}
```

פונקציה נוספת הקשורה לפסיקות היא : detachInterrupt()

הפונקציה מפסיקה (חוסמת) את פעולת הפסיקה שמספרה רשום בסוגריים.

התחביר : detachInterrupt(interrupt)

Interrupt הוא מספר הפסיקה שרוצים לחסום/ להפסיק .

לדוגמה, הפקודה : detachInterrupt(1); חוסמת/מפסיקה את פסיקה מספר 1 המתחברת להדק D3 .

דוגמה : חיבור חיישן מרחק עם פסיקה

באחת הדוגמאות הקודמות חיברנו חיישן מרחק ללא פסיקה. כאן ניתן דוגמה לעבודה עם החיישן בעזרת פסיקה.

ההבדל בחומרה יהיה לנתק את מוצא Echo (של החיישן) מהדק 9 וחבר אותו להדק 2 (המשמש מבוא interrupt 0 של ארדואינו).

בתוכנית שלהלן המיקרו בקר מבצע שתי משימות במקביל.

- i. מדידת המרחק באופן מחזורי (ע"פ המלצת היצרן כל 50mS).
- ii. הדלקת הled שבהדק 13 כאשר המרחק קטן מ 15 ס"מ.

```
#define trigPin 8 // בהדק 8 נותנים את ה'דרבון' של 10uS
#define echoPin 2 // חיבור מוצא החיישן Echo להדק 2 ( interrupt 0 )
#define led 13 // להדק 13 מחובר led (על הכרטיס)

unsigned long time=0;
unsigned long duration; // משתנה עבור משך הזמן של רוחב הפולס
int distance=0; // משתנה עבור המרחק בס"מ

//----- פונקציה trigUs() נותנת 'דרבון' של 10uS במבוא trigger של החיישן -----
void trigUs() {
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);
    attachInterrupt(0, startCount ,RISING); // startCount פונקציה
}

//----- פונקציה startCount() מאפשרת את מדידת הזמן -----
void startCount() {
    duration=micros(); // קרא זמן נוכחי למשתנה duration
```



```
detachInterrupt(0); // חסום פסיקה מספר 0 .
attachInterrupt(0, measurement, FALLING); // measurement() : בצע: , ב'ירידה',
}
//----- פונקציה measurement() מודדת את הזמן ומחשבת את המרחק -----
void measurement()
{
  duration=micros()-duration; // ('עליה') לזמן הקודם (ב'ירידה')
  distance=duration/58; // distance מ"מ והצבה למשתנה
  detachInterrupt(0); // חסום פסיקה מספר 0 .
}
void setup()
{
  Serial.begin(9600);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(led, OUTPUT);
  trigUs(); // כדי להתחיל את התהליך צריך לתת פעם אחת 'דרבון' של 10uS ואפשר פסיקה
}
void loop()
{
  if(millis()-time>50) { // 50mS מדידת מרחק כל
    Serial.print(distance); // הדפס את המרחק בס"מ
    Serial.println(" cm"); // הדפס ס"מ ועבור שורה
    trigUs(); // יצירת 'דרבון' של 10uS
    time=millis();
  }
  if (distance<15 ) // אם המרחק קטן מ 15 ס"מ
    digitalWrite(led,HIGH); // הדלק את הled
    else // אחרת
    digitalWrite(led,LOW); // כבה את הled
}
```

תרגול

1. בתרגיל הדוגמה עם הפסיקה : מה צריך לשנות בחומרה ומה בתכנה אם רוצים לעבוד עם פסיקה חיצונית 1 במקום עם פסיקה חיצונית 0 .
2. מה צריך לשנות בתוכנית כדי שהלד יידלק רק כאשר המרחק קטן מ 20 ס"מ.
3. רשום תכנית ללא פסיקה, שתציג את המרחק בעזרת הלד שמחובר להדק 13 בצורה הבאה : אם המרחק קטן מ 10 ס"מ הלד דולק באופן קבוע. בין 10 ל 20 ס"מ הלד מהבהב בקצב של שנייה. בין 20 ל 30 הלד מהבהב בקצב של 2 פעמים בשנייה. בין 20 ל 30 ס"מ הלד מהבהב 3 פעמים בשנייה וכך הלאה בהתאמה. אחרי 1 מטר הלד לא דולק כלל.
4. את התכנית שבסעיף הקודם לבצע עם פסיקה.