

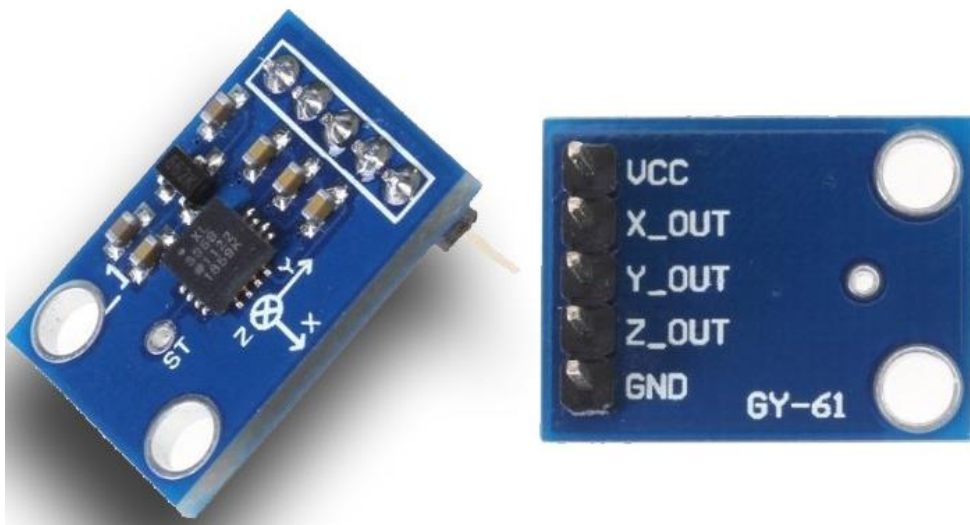
## מד תאוצה ADXL335

### 1. כללי

כולנו מכירים את התכונה של הטלפון הנייד שיודע להזיז או להפוך את התמונה שעל המסך ימינה או שמאלה או להפוך אותה. בתוך הטלפון הנייד קיים מד תאוצה - Accelerometer - שיכול לחוש כאשר אנחנו מטים אותו מצד לצד. כך הטלפון מבין באופן אוטומטי מתי להעביר את פריסת המסך לכיוון שרוצים. המאמר כאן מתאר מד תאוצה ונעזר בקישור :

<https://lastminuteengineers.com/adxl335-accelerometer-arduino-tutorial/>

האיור הבא מתאר את המודול ADXL335 Module שנקרא גם GY-61



איור 1 : המודול ADXL335 משני צדדיו. מצד שמאל – צד הרכיבים. מצד ימין צד הקונקטור.

נתחיל בהסבר הפיזיקלי של תאוצה. החוק השני של ניוטון אומר:  $\mathbf{F} = \mathbf{m} * \mathbf{a}$  כלומר הכוח הפועל על גוף שווה למכפלה של המסה שלו בתאוצה. ניתן לעשות שינוי בנוסחה ולקבל:

$$\mathbf{a} = \mathbf{F} / \mathbf{m}$$

התאוצה שגוף נע שווה ליחס בין הכוח המופעל עליו והמסה שלו.

### 2. כיצד פועל מד תאוצה?

כדי לדעת כיצד פועלים מדי תאוצה, לעתים קרובות כדאי לדמיין כדור בתוך קובייה תלת-ממדית הנמצאת בחלל החיצון. האיור הבא מסביר את המקרה :

במצב של חוסר משקל  
כמו בחלל החיצון

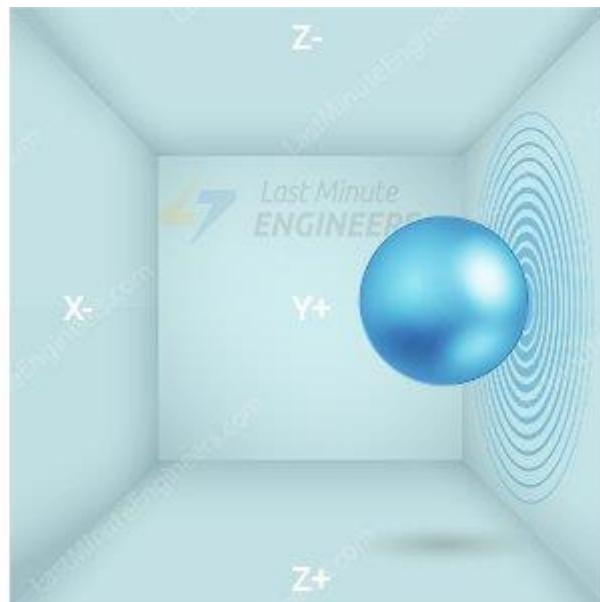


$X=0g$   
 $Y=0g$   
 $Z=0g$

איור 2 : קובייה בחלל החיצון

נניח שהקובייה נמצאת בחלל החיצון שבו הכול במצב חוסר משקל, הכדור פשוט יצוף באמצע הקובייה. נדמיין שכל קיר מייצג ציר מסוים. אם נזיז לפתע את התיבה שמאלה עם תאוצה  $1g$  (כוח  $G$  יחיד  $1g$  שווה ערך לתאוצת הכבידה  $9.8$  מטר לשנייה<sup>2</sup>), אין ספק שהכדור יפגע בקיר  $X$ . אם נמדוד את הכוח שהכדור מפעיל על הקיר  $X$ , נוכל לקבל ערך פלט של  $1g$  על ציר  $X$  כפי שמתואר באיור הבא:

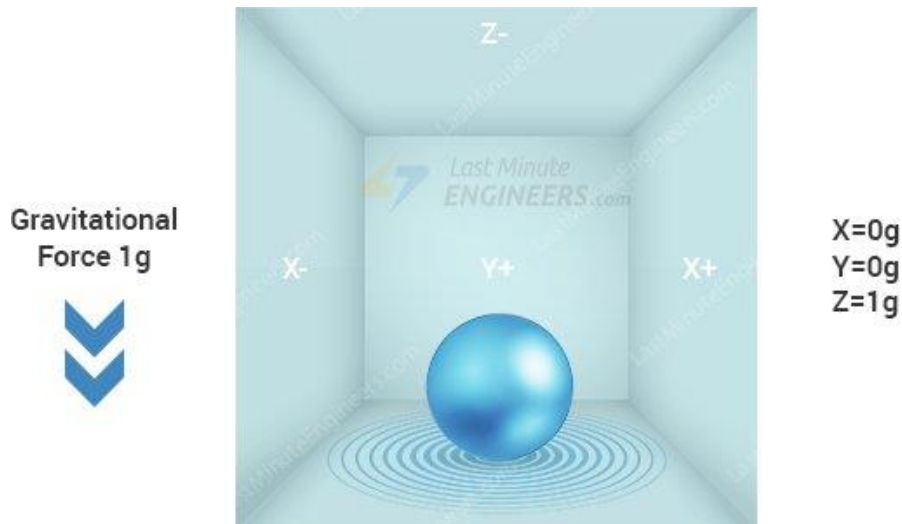
  
Force  $1g$



$X=1g$   
 $Y=0g$   
 $Z=0g$

איור 3 : הכדור פוגע בדופן הימני של הקובייה כאשר הזננו את הקופסה שמאלה בכוח של  $1g$ .

מה יקרה אם נשים את הכדור הצף על כדור הארץ? הכדור פשוט ייפול על הקיר  $Z$  ויפעיל כוח של  $1g$ , כפי שמוצג בתמונה למטה:



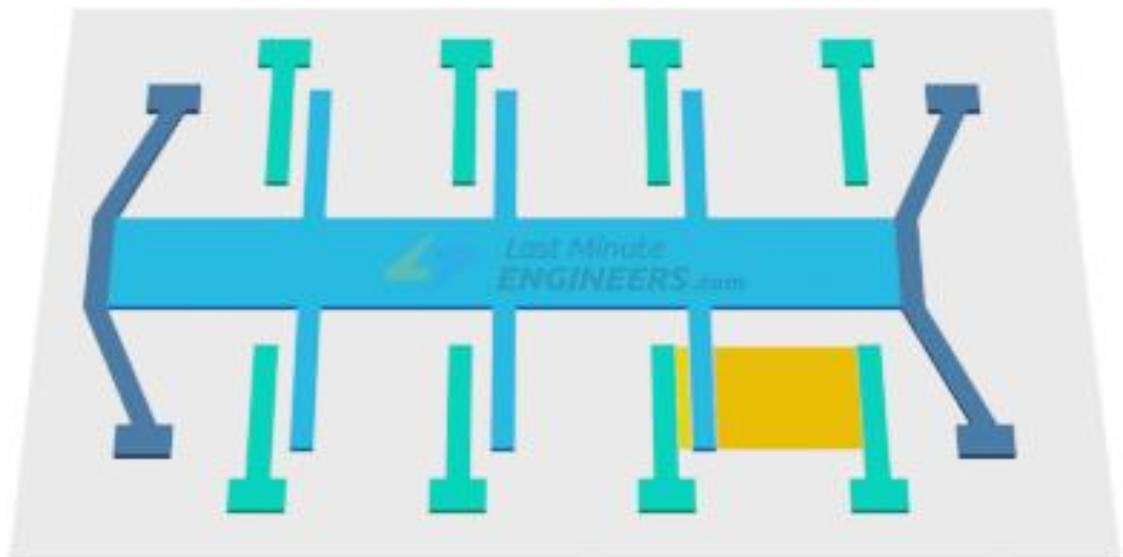
איור 4 : הכדור פוגע בדופן Z של הקובייה בכוח של 1g .






במקרה זה התיבה לא זזה אבל אנחנו עדיין מקבלים קריאה של 1g על ציר Z. הסיבה לכך היא שכוח הכבידה מושך את הכדור כלפי מטה בכוח 1g. מד התאוצה מודד את התאוצה הסטטית של כוח הכבידה ביישומים של חישת הטיה וגם תאוצה דינאמית כתוצאה מתנועה, הלם - shock - או רטט.

### 3. כיצד עובדות מערכות מדי תאוצה מיקרו אלקטרומכניות - MEMS ?

מד תאוצה MEMS (מערכות מיקרו אלקטרו מכניות) מורכב ממבנה מיקרו-מכונה הבנוי על גבי פרוסת סיליקון .

האיור הבא מתאר מבנה כזה:



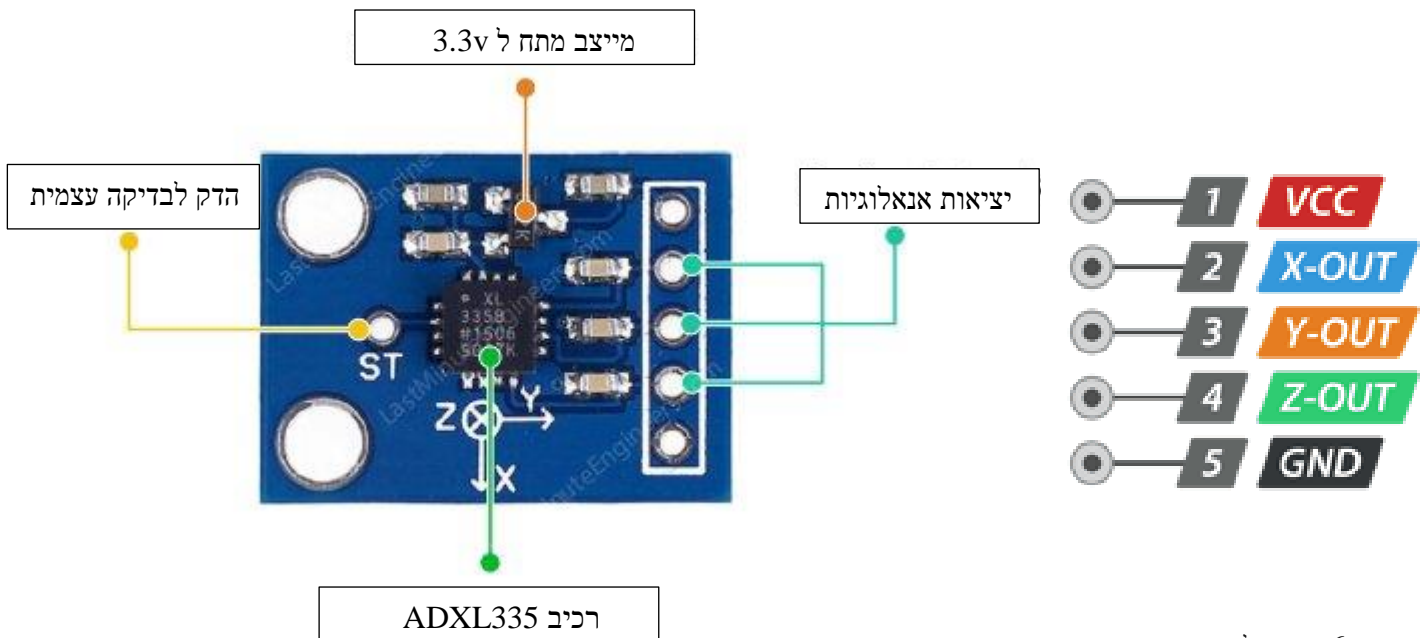
	מסה תלויה		לוחיות קבועות		פרוסת - מצע - סיליקון
	קפיצים מפולי סיליקון		שינוי בקיבול		

איור 5 : MEMS – מבנה מערכת מד תאוצה מיקרו מכנית

המבנה תלוי על קפיצי פולי סיליקון וזה מאפשר למבנה להסיט בזמן שבו התאוצה מוחלת על הציר המסוים. בשל הסטייה משתנה הקיבול בין לוחות קבועים ולוחות המחוברים למבנה התלוי. שינוי זה בקיבוליות הוא פרופורציונלי לתאוצה בציר זה. החיישן מעבד את השינוי הזה בקיבוליות וממיר אותו למתח יציאה אנלוגי. החיישן מעבד את השינוי הזה בקיבוליות וממיר אותו למתח יציאה אנלוגי.

#### 4. סקירת חומרה של מד תאוצה ADXL335

בלב המודול נמצא מד תאוצה - MEMS – קטן בעל 3 צירים בהספק נמוך של חברת Analog Devices עם רעש נמוך במיוחד שנקרא ADXL335. לחיישן יש טווח חישה מלא של  $\pm 3g$ . הוא יכול למדוד את התאוצה הסטטית עקב כוח הכבידה ביישומי חישת הטיה, כמו גם תאוצה דינמית הנובעת מתנועה, זעזוע או רטט. האיור הבא מתאר את מודול מד התאוצה:



איור 6 : מודול מד התאוצה

החיישן פועל עם מתח ספק של בין 1.8 וולט ל-3.6VDC (3.3 וולט אופטימלי), ובדרך כלל צורך רק  $350\mu A$  של זרם. עם זאת, מייצב מתח ל 3.3V מובנה מאפשר התחברות עם מיקרו-בקרים של 5V כמו הארדואינו. במעגל יש 5 הדקים להלחים מחבר הכולל 3 יציאות אנלוגיות עבור מדידות צירי X, Y, Z - 1 ו-2 הדקים למתח ספק כוח. הדק נוסף (לא בקונקטור) הוא לבדיקה עצמית המאפשר לך לבדוק את תפקוד החיישן ביישום הסופי. הפלטים האנלוגיים הם רציומטריים – משתנים יחסית לכניסה. תפוקת המדידה של 0g שווה באופן נומינלי למחצית ממתח הספק של 3.3 וולט (כלומר 1.65 וולט),  $-3g$  הוא 0 וולט ו  $3g$  הוא 3.3 וולט ובהתאמה המתחים בין  $-3g$  ל  $3g$ .

#### 4.4 תפקיד ההדקים

**VCC** - הדק VCC מספק הכוח למד התאוצה. ניתן לחברו ל-5V בארדואינו.

**X-Out** - יציאת מתח אנלוגי פרופורציונלי לתאוצה המופעלת על ציר X.

**Y-Out** - יציאת מתח אנלוגי פרופורציונלי לתאוצה המופעלת על ציר Y.

**Z-Out** - יציאת מתח אנלוגי פרופורציונלי לתאוצה המופעלת על ציר Z.

**GND** - הדק אדמה, מחובר ל-GND בארדואינו

**ST(Self-Test)** - שולט בתכונת הבדיקה העצמית. על תכונה זו נדון בפירוט בסוף.

הטבלה הבאה מתארת את המפרט – מאפיינים של מד התאוצה ADXL335 .

Operating Voltage	מתח הפעלה	1.8V – 3.6V
Operating Current	זרם פעולה	350μA (typical)
Sensing Range	תחום חישה	±3g (Full Scale)
Temperature Range	תחום טמפרטורה	-40 to +85°C
Sensing axis	צירי חישה	3 axis
Sensitivity	רגישות	270 to 330mV/g (Ratiometric)
Shock Resistance	עמידות לכבידה	Up to 10,000g
Dimension	מידות	4mm x 4mm x 1.45mm

טבלה 1 : מאפייני ADXL335

## 5. חיבור ADXL335 אל ארדואינו

ההדקים הבאים מתחברים בין ADXL335 והארדואינו אונו. כמובן שניתן לחבר גם לארדואינו נאנו או ארדואינו מגה:

ARDUINO UNO	ADXL335
<b>5V</b>	<b>VCC</b>
<b>A0</b>	<b>X-out</b>
<b>A1</b>	<b>Y-out</b>
<b>A2</b>	<b>Z-out</b>
<b>GND</b>	<b>GND</b>
<b>3.3v ↔ AREF</b>	

**שימו לב :** החיבור האחרון שרשמנו הוא לקבלת תוצאות מדויקות, עלינו לשנות את מתח הייחוס האנלוגי (AREF) של הארדואינו. ניתן לעשות זאת על ידי חיבור הדק 3.3V בארדואינו להדק AREF.

## 6. כתיבת תוכנית

התוכנית הבאה מציגה את פלט מד התאוצה בכל ציר במסך הטורי של הארדואינו .

```
#define xInput A0

#define yInput A1

#define zInput A2

// קביעת מינימום ומקסימום בכל ציר

int RawMin = 0;

int RawMax = 1023;

// מבצעים 10 דגימות כדי להוריד רעש

#define sampleSize 10

void setup()

{

  analogReference(EXTERNAL); // 3.3v = AREF לדיגיטלי מתח הייחוס לממיר מאנלוגי לדיגיטלי
  הוא המתח ב

  Serial.begin(9600); // 9600 ביטים בשנייה אתחול קצב התקשורת עם המוניטור הטורי ל
  9600 ביטים בשנייה

}

void loop()

{

  // קריאת הערך ביציאת כל ציר

  int xRaw = ReadAxis(xInput);

  int yRaw = ReadAxis(yInput);
```

```
int zRaw = ReadAxis(zInput);

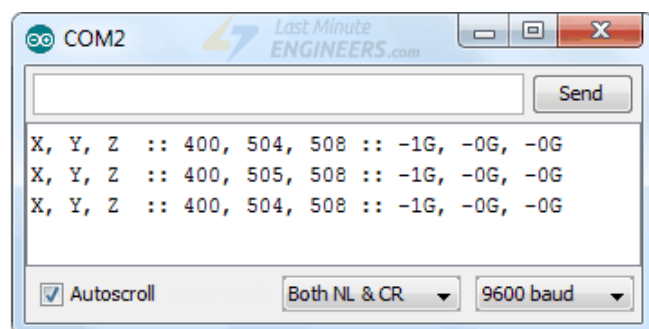
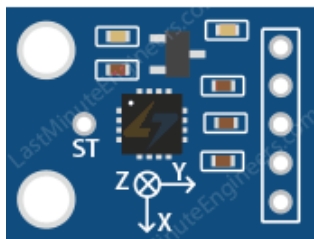
// המרה של הערכים שקראנו בעזרת פונקציית map למילי ג
long xScaled = map(xRaw, RawMin, RawMax, -3000, 3000);
long yScaled = map(yRaw, RawMin, RawMax, -3000, 3000);
long zScaled = map(zRaw, RawMin, RawMax, -3000, 3000);

// re-scale to fractional Gs
float xAccel = xScaled / 1000.0;
float yAccel = yScaled / 1000.0;
float zAccel = zScaled / 1000.0;

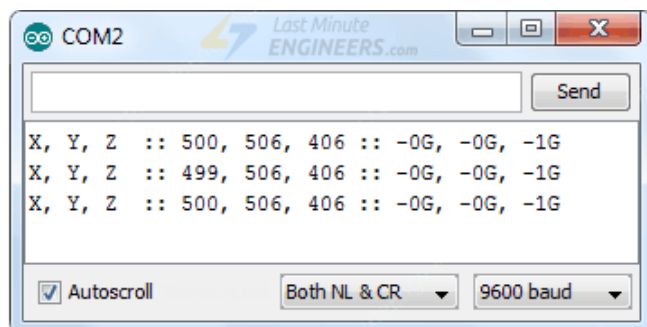
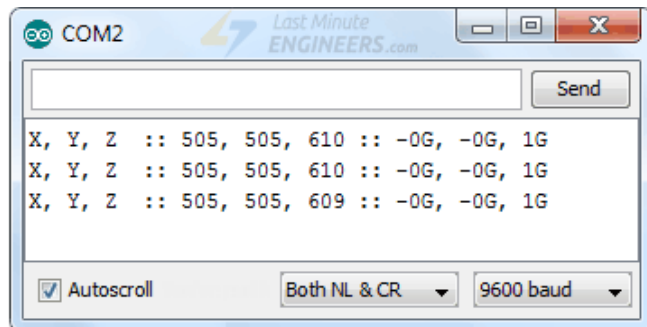
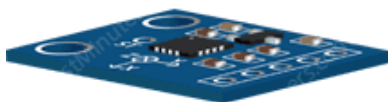
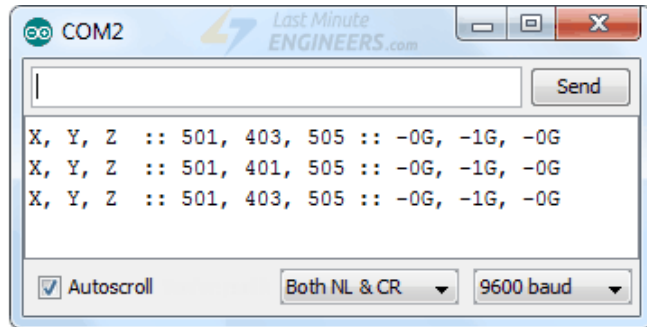
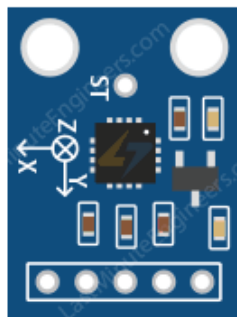
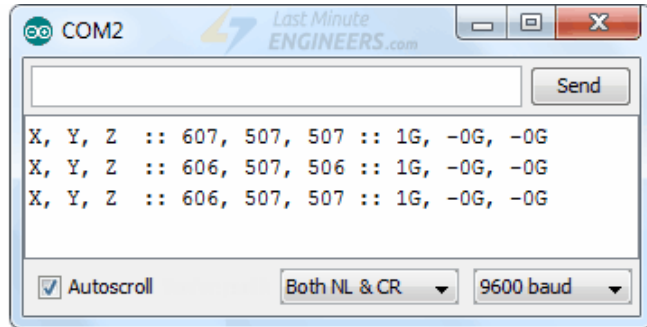
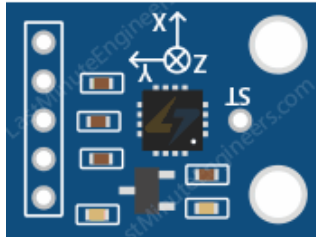
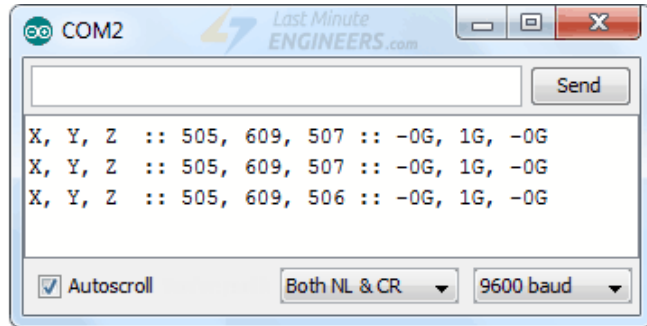
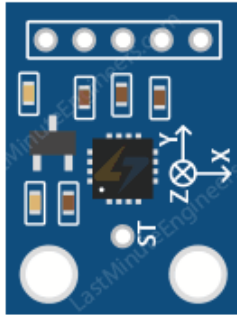
Serial.print("X, Y, Z :: ");
Serial.print(xRaw);
Serial.print(", ");
Serial.print(yRaw);
Serial.print(", ");
Serial.print(zRaw);
Serial.print(" :: ");
Serial.print(xAccel,0);
Serial.print("G, ");
Serial.print(yAccel,0);
Serial.print("G, ");
Serial.print(zAccel,0);
Serial.println("G");
```

```
        delay(200);  
    }  
  
    // ביצוע של 10 דגימות והחזרת הממוצע  
    int ReadAxis(int axisPin)  
    {  
        long reading = 0;  
        analogRead(axisPin);  
        delay(1);  
        for (int i = 0; i < sampleSize; i++)  
            {  
                reading += analogRead(axisPin);  
            }  
        return reading/sampleSize;  
    }  
}
```

התמונות הבאה מציגה את פלט מד התאוצה בצג הטורי במיקומים שונים.







איור 7 : התוצאות המתקבלות במוניטור הטורי

## 7. הסבר התוכנית

התוכנית מתחילה בהכרזה על הדקי הקלט האנלוגיים של Arduino שאליהם מחוברים פני ה- $X$ ,  $Y$  ו- $Z$  של החיישן. לאחר מכן, אנו מגדירים את הערכים המינימליים והמקסימליים שהארדואינו יכול לתת ( ה-ADC בארדואינו הוא של 10 ביט ולכן הערכים הם בין 0 ל-1023). ה-ADC ימיר את מתחי המוצא של החיישן בין 0 ל-3.3 וולט לערכים שלמים בין 0 ל-1023. זו הסיבה ש RawMin מוגדר ל-0 ו-RawMax מוגדר ל-1023.

sampleSize אומר לארדואינו לקחת 10 דגימות של כל המרה כדי לקבל תוצאות מדויקות יותר (אם במקרה נכנס רעש הוא "ייחלש" או פחות ישפיע בגלל חישוב הממוצע).

בפונקציית ה-`setup`

```
void setup()
```

```
{
  analogReference(EXTERNAL); // AREF הוא המתח ב
  Serial.begin(9600); // 9600 ביטים בשנייה
}
```

קובעים שמתח הייחוס ל-ADC הוא מתח חיזוני שהוכנס ברגל AREF. במקרה שלנו הוא 3.3 וולט כי קיצרנו את הדק AREF להדק ה-3.3 וולט של הארדואינו ומאתחלים את התקשורת בין הארדואינו והמוניטור הטורי של ה-IDE ל-9600 ביטים בשנייה.

אזהרה: אם לא נצליח לקרוא לפונקציה `analogReferencd(EXTERNAL)` יתבצע קצר בין מתח הייחוס הפנימי ל-AREF דבר שעלול להזיק לארדואינו.

בפונקציית ה-`loop` אנחנו קוראים לפונקציה `ReadAxis()` ושולחים לה איזה ציר אנחנו רוצים לקרוא.

```
int xRaw = ReadAxis(xInput);
int yRaw = ReadAxis(yInput);
int zRaw = ReadAxis(zInput);
```

הפונקציה `ReadAxis` מבצעת 10 דגימות בכל ציר ומחזירה את הממוצע של 10 הדגימות ל-`xRaw`, `yRaw` ו-`zRaw` בהתאמה. אחרי שקראנו את הקריאות בכל 3 הצירים אנחנו משתמשים בפונקציה הנקראת `map()` של הארדואינו שתפקידה להמיר את הקריאות הדיגיטליות של הממיר ADC של המתח האנלוגי בכל אחד מ-3 הצירים ליחידות של כבידה `g`. השורה

```
long xScaled = map(xRaw, RawMin, RawMax, -3000, 3000);
```

אומרת להעביר את הערך שיש ב-`xRaw` שהוא בין 0 ל-1023 (`RawMin`, `RawMax`) לתחום ערכים חדש בין -3000 ל-3000 שזה התחום שהחיישן שלנו עובד מבחינת כבידה  $\pm 3g$ . הכפלנו את  $\pm 3g$  ב-1000 כדי לקבל תוצאות באלפיות `g`.

1. כאשר החיישן מוציא 0 וולט על ציר x כלומר  $x_{Raw}=0$ , פונקציית  $map()$  תחזיר -3000 המייצגת -3g.
2. כאשר החיישן מוציא 3.3 וולט על ציר x כלומר  $x_{Raw}=1023$ , פונקציית  $map()$  תחזיר 3000 המייצגת +3g.
3. כאשר החיישן מוציא 1.65 וולט על ציר x כלומר  $x_{Raw}=511$ , פונקציית  $map()$  תחזיר 0 המייצגת 0g.

לקראת הסוף מבצעים חלוקה ב 1000 כדי לקבל גם חלקים של g ( כמו 1.25g או 0.9g וכו').

```
float xAccel = xScaled / 1000.0;
```

```
float yAccel = yScaled / 1000.0;
```

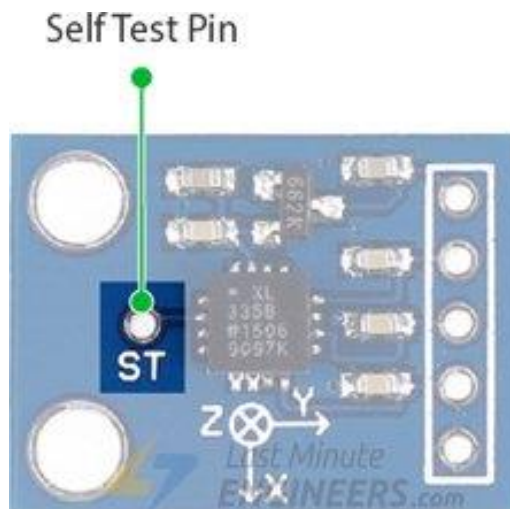
```
float zAccel = zScaled / 1000.0;
```

ולבסוף מבצעים הדפסה של התוצאות למסך הטורי כפי שרואים בתמונות בעמודים קודמים.

## 8. בדיקה עצמית Self Test .

מד התאוצה ADXL335 כולל תכונת בדיקה עצמית המאפשרת לך לבדוק את תפקוד החיישן ביישום הסופי.

האיור הבא מתאר את הדק הבדיקה העצמית.



איור 8 : הדק בדיקה עצמית

כאשר נחבר את הדק ST למתח של 3.3V - מופעל כוח אלקטרוסטטי על קרן מד התאוצה באופן פנימי. התנועה המתקבלת של הקרן מאפשרת למשתמש לבדוק אם מד התאוצה מתפקד. השינוי האופייני בפלט הוא :

• בציר X  $-1.08\text{ g } (-325\text{ mV})$

• בציר Y  $+1.08\text{ g } (+325\text{ mV})$

• בציר Z +1.83 g (+550 mV)

ניתן להשאיר הדק ST זה פתוח או לחבר אותו ל-GND בשימוש רגיל.

אזהרה : אין לחבר להדק ST מתח גדול מ 3.6 וולט שעלול לגרום לנזר למד התאוצה.

## 9. ביבליוגרפיה

[In-Depth: How Accelerometer works? Interface ADXL335 with Arduino \(lastminuteengineers.com\)](https://lastminuteengineers.com/in-depth-how-accelerometer-works-interface-adxl335-with-arduino/)