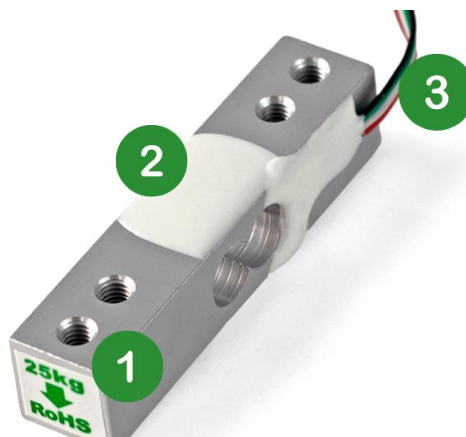


# חיישן משקל הכולל תא עומס ומגבר HX711 וחיבורו אל ESP32

## 1. מהו חיישן משקל ?

חיישן משקל הנקרא גם תא עומס - LOAD CELL - הוא חיישן המאפשר למדוד משקל ששמים עליו. החיישן שנעסוק בו כאן מורכב מ 3 יחידות שמתוארות באיור הבא :

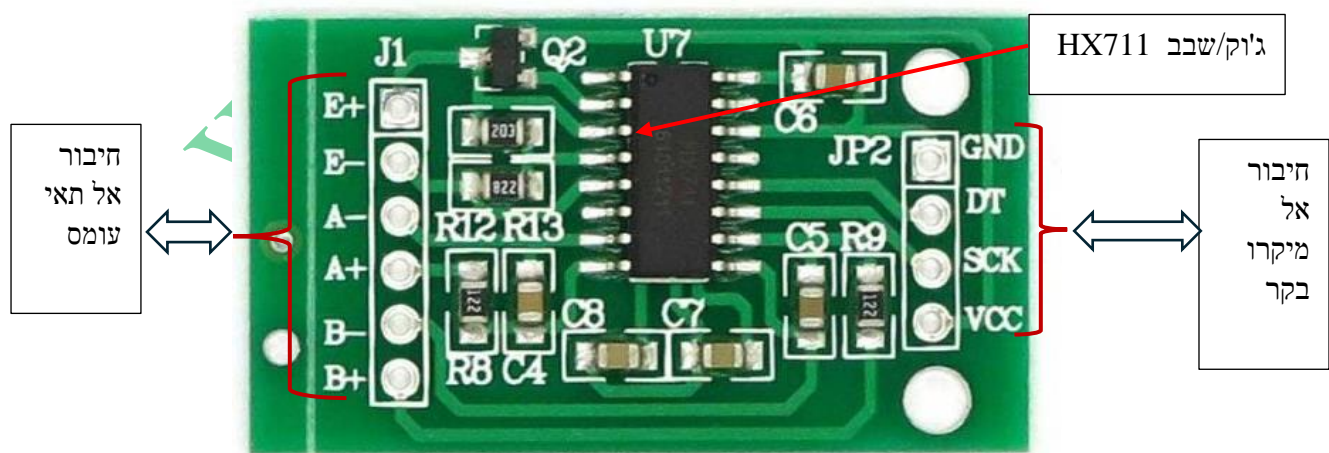
1. מסגרת או גוף קשיח עשוי מאלומיניום או סגסוגת פלדה . הגוף שנשתמש בו הוא בצורה מלבנית ובהמשך נקרא לו קורה .
2. אחד או יותר מדידים של מד מאמץ Strain Gauge .
3. חוטים למתח ספק, אדמה , ואותות ( בדרך כלל 4 חוטים).



איור 1 : תא עומס והמגבר HX711

תא העומס שבאיור הוא למשקל 25 ק"ג . ישנם תאי עומס למשקלים נמוכים יותר ולמשקלים גבוהים יותר. המשקלים הנפוצים ביותר הם 1 ק"ג, 5 ק"ג, 10 ק"ג ו-20 ק"ג, 25 ק"ג . ניתן לחבר מספר תאי עומס יחד ולהגיע למשקלים גבוהים יותר. חישוב המשקל נעשה תוך שימוש בעקרונות פיזיקליים הקושרים בין התארכות קורה מועמסת לבין העומס המופעל עליה. רכיבי המדידה העיקריים נקראים Strain Gauge – מד מאמץ ומגבר. **מד המאמץ נקרא גם תא עומס - Load Cell** או מד עיוות ובהמשך נקרא לו גם **מדיד** .

כדי להגביר את האותות היוצאים מתא העומס משתמשים במגבר HX711. באיור הבא מתואר כרטיס המגבר HX711 :



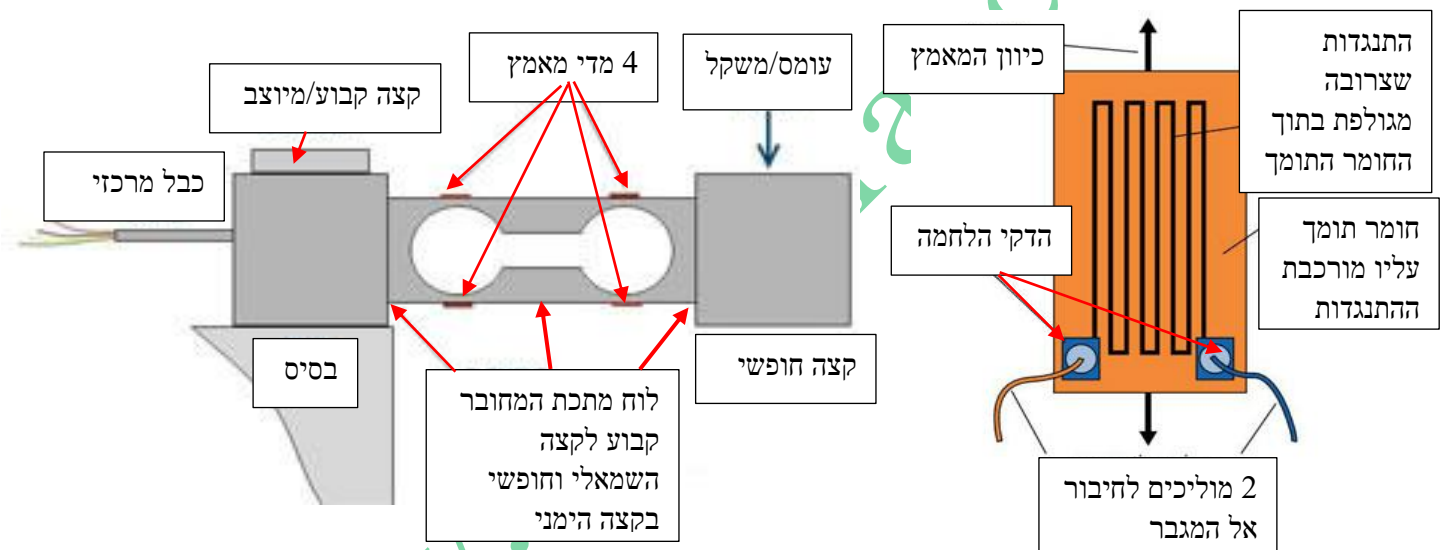
איור 2 : כרטיס המגבר HX711 .

באיור רואים משני צידי הכרטיס חורים להלחמה של 2 קונקטורים. בדרך כלל כאשר מזמינים כרטיס מגבר הוא מגיע מסוג זכר. האחד עם 6 פינים והשני עם 4 פינים להלחמה בשני קצותיו של המגבר. לפעמים הם מגיעים כקונקטור אחד של 10 פינים וניתן לחתוך אותם ל 2 חלקים של 4 ו 6 פינים.

בצד שמאל של המגבר יש את J1 שאליו מחברים את תא העומס המורכב מ 4 מדי מאמץ/מדידים – Strain Gauge. יש לו 6 כניסות ולכאן מלחימים את הקונקטור בן ה 6 פינים מסוג זכר שבאיור הקודם. מצד ימין של המגבר יש את JP2 שאליו מלחימים את הקונקטור מסוג זכר עם 4 פינים. הדקים אלו הם של התקשורת עם המיקרו בקר.

## 2. המבנה של מד המאמץ

מד המאמץ - Strain Gauge, מורכב מחוטים דקים בצורת זיגזג עשויים סגסוגת של נחושת-ניקל שיוצרים התנגדות. הם מודבקים לאורך לוח תומך שעליו שמים את המשקל/העומס כפי שרואים באיור הבא:



איור 3: מבנה תא עומס. משמאל המבנה הפיזי ומימין סגסוגת נחושת – ניקל המחוברת בזיגזג ויוצרת התנגדות.

בצד השמאלי של האיור רואים את תא העומס המורכב מלוח המתכת שעליו מרכיבים את מדי המאמץ הנקראים גם **מדיד**. בצד שמאל שלו הוא קבוע ובצד הימני חופשי. יש 4 מדידים של מאמץ: 2 בחלק העליון של לוח המתכת ו 2 בחלק התחתון שלו. כל מד מאמץ בנוי מהתנגדות המורכבת מסגסוגת נחושת - ניקל. מניחים משקל על הצד החופשי ואז הצד החופשי נוטה כלפי מטה ויוצר בצד העליון של הלוח התרחבות ובצידו התחתון התכווצות. ההתנגדויות של מדי המאמץ משתנות וכך ניתן לדעת את העומס/משקל המונח. בצד ימין של האיור רואים דוגמה כיצד בנוי מד מאמץ. הסגסוגת מתחברת על חומר תומך ויוצרת התנגדות. גודל ההתנגדות שנוצרת תלוי ביחס בין הנחושת (שהוא מוליך) ובין הניקל (שהוא מבודד).

### 3. עקרון פעולה של חיישן משקל

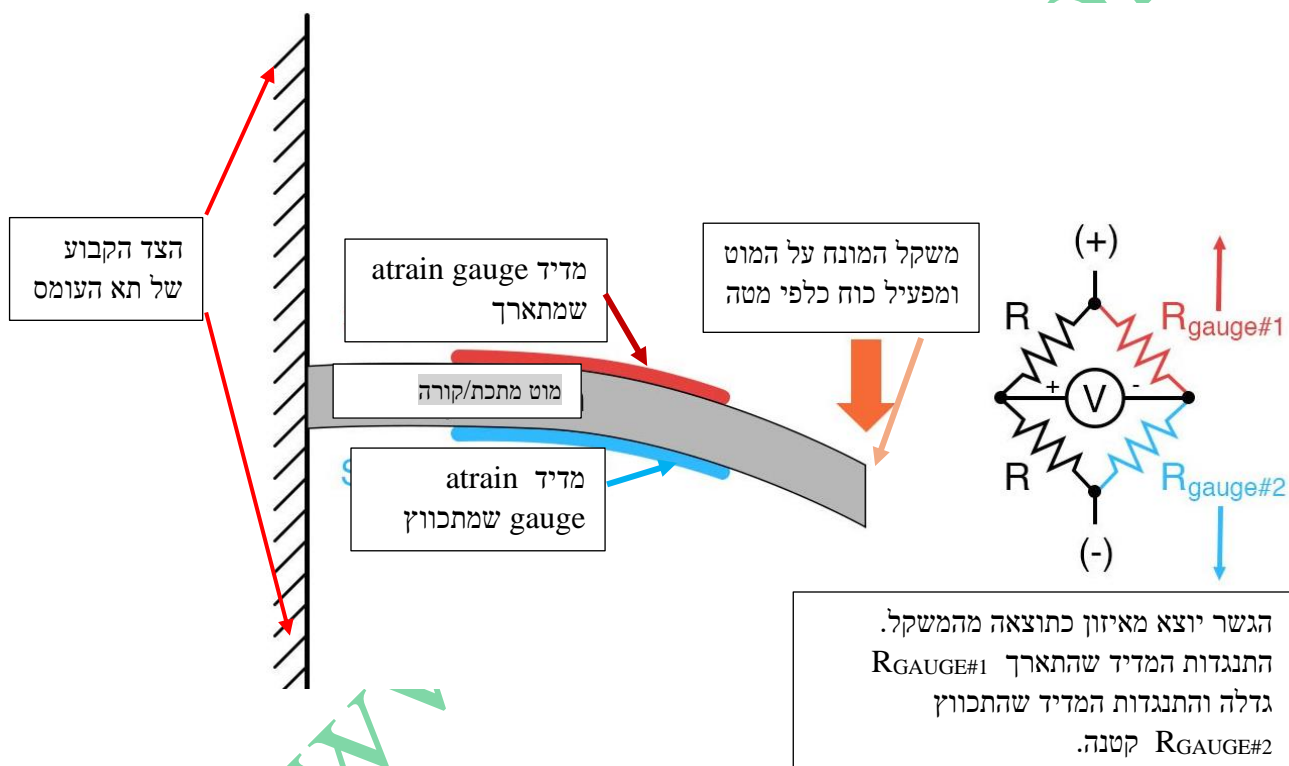
- לוח מתכת קבוע בצידו האחד אל בסיס וקצהו השני חופשי. על הקצה החופשי מניחים את העומס - המשקל - שרוצים למדוד. הקצה החופשי יתכוּפּף מעט בהתאם למידת ההעמסה. הצד העליון של הלוח יימתח ואילו הצד התחתון יתכוּוץ.
- ההתנגדויות שעל הלוח מתארכות בחלק העליון ומתקצרות בחלק העליון. שינוי האורך, גורם לשינוי בהתנגדות שפרופורציוני לעומס ולהתארכות/התכווצות המדיד לפי הנוסחה:

$$R = \rho * l/A$$

$\rho$  הוא ההתנגדות הסגולית של הסגסוגת.  $l$  – אורך המוליך ( המדיד במקרה שלנו).  $A$  – שטח חתך המוליך.

בצד שמתארך ההתנגדות גדלה ובצד שמתכווץ ההתנגדות יורדת.

האיור הבא מתאר 2 מדידים הנמצאים על לוח המתכת. בצד אחד הוא קבוע ובצד השני מפעילים כוח על ידי הנחת משקל.

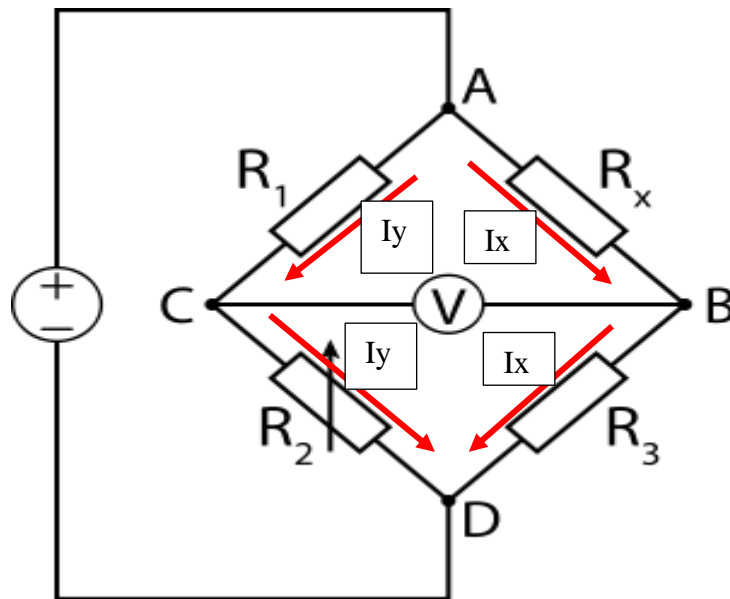


איור 4 : תא עומס מודד משקל

- בכל תא עומס יש 4 מדידים, 2 למעלה ו 2 למטה המחוברים חשמלית בצורה שנקראת גשר ויטסטון.

### 4. גשר ויטסטון

גשר ויטסטון הוא מעגל חשמלי המחשב התנגדות (או עכבה חשמלית) במקרה שמדובר ברכיב שלא ניתן למדוד בצורה ישירה (על ידי מיכשור כמו מד התנגדות). המעגל הומצא של המדען האנגלי צ'ארלס ויטסטון (1802 - 1875). המעגל בנוי בדרך כלל מ-4 נגדים או עכבות. הגשר מוזן ממקור מתח חשמלי. האיור הבא מתאר את עיקרון המדידה של גשר ויטסטון:



איור 5 : עיקרון המדידה של גשר ויטסטון:

באיור רואים 4 נגדים המחוברים למקור מתח .

$R_x$  הוא הנגד שאת התנגדותו רוצים למדוד. ההתנגדויות  $R_1$ ,  $R_2$  ו-  $R_3$  ידועות ו-  $R_2$  היא התנגדות של פוטנציומטר. משנים את ההתנגדות של  $R_2$  עד שהמתח בממדד המתח  $V$  מראה 0 כלומר הזרם  $I$  דרך המודד שווה 0. זה אומר שהמתח על  $R_3$  שווה למתח על  $R_2$  והמתח על  $R_x$  שווה למתח על  $R_1$ . היות והתנגדות מד מתח היא (תאורטית אין סופית) אז הזרם הזורם דרך  $R_3$  שווה לזרם הזורם דרך  $R_x$  והזרם הזורם דרך  $R_1$  שווה לזרם דרך  $R_2$ . מכאן :

$$I_y \cdot R_1 = I_x \cdot R_x$$

$$I_y \cdot R_2 = I_x \cdot R_3$$

חלוקה של 2 המשוואות אחת בשנייה תיתן :

$$R_1/R_2 = R_x/R_3$$

ועל ידי שינוי נושא בנוסחה נקבל :

$$R_x = R_1 \cdot R_3 / R_2$$

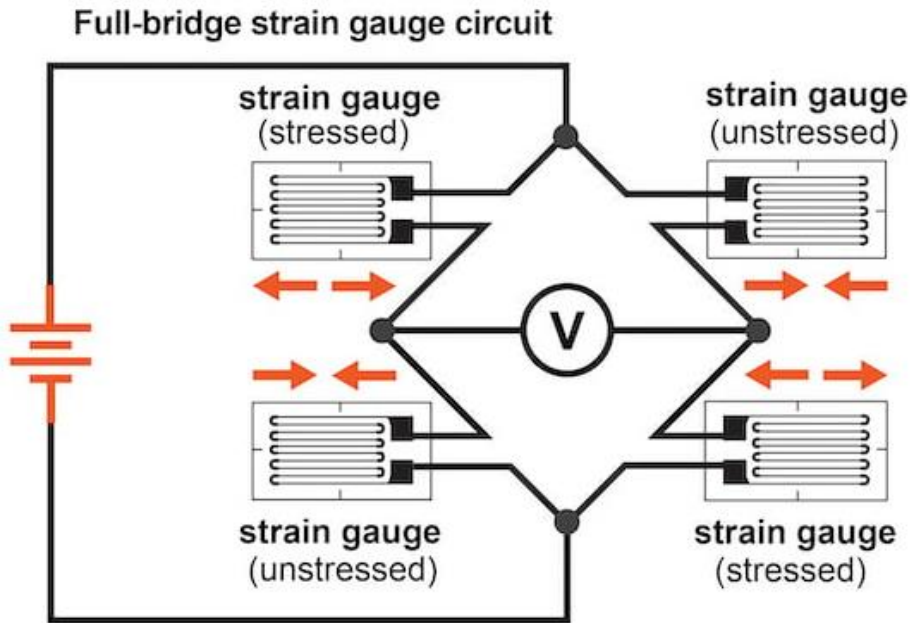
אם יודעים את התנגדותם של  $R_1$ ,  $R_2$ ,  $R_3$  ניתן בחישוב פשוט לקבל את ערכו של  $R_x$ .

## 5. עיקרון המדידה של חיישן המשקל

בחיישן המשקל מסתמכים יותר על העיקרון של הוצאת הגשר מאיזון שהוא יחסי למשקל ששמים עליו. בחיישן יש 4 חוטים המחוברים לפינות A, B, C, D כדי למדוד את המתח בין הנקודות. לפי המתחים מחשבים את ההתנגדות של המדיד המועמס ובהתאם את העומס.

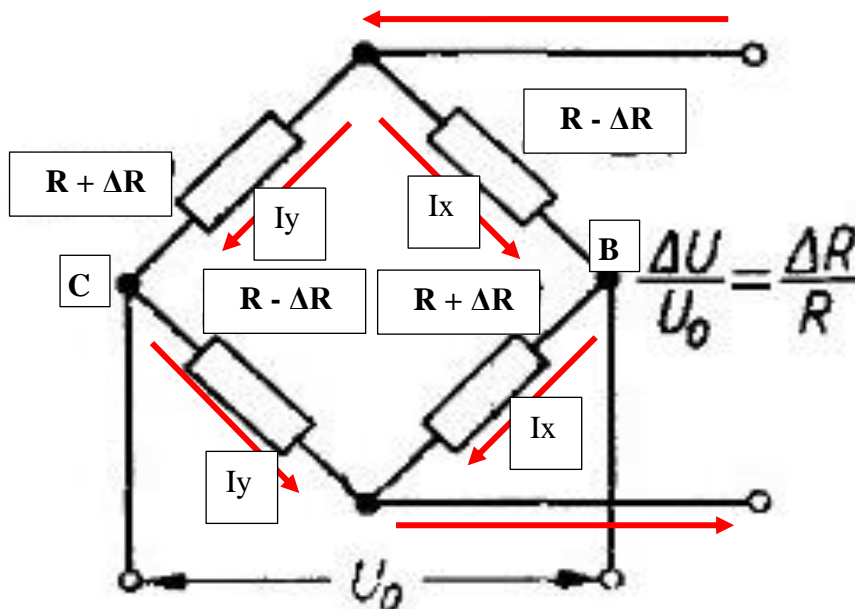
בחיישן המשקל עם 4 מדידים של Strain Gauge בהנחה שלכל המדידים התנגדות שווה נקבל במצב שלא מונח משקל שהגשר מאוזן והפרש המתח הוא 0.

האיור הבא מתאר את החיבור של 4 המדידים המחוברים כגשר מלא ומדידת המשקל בצורת גשר ויטסטון.



איור 6 : מדידת משקל בחיבור גשר מלא

באיור רואים שהמדיד (strain gauge) העליון בצד ימין של הגשר והמדיד התחתון בצד שמאל של הגשר הם אלו שמתכווצים בזמן המדידה ומכאן שהתנגדותם קטנה ואילו 2 המדידים האחרים מתארכים בזמן המדידה והתנגדות המדידים גדלה. הזרם דרך  $I_x$  שווה לזרם  $I_y$  כי בכל צד של הגשר ההתנגדות היא  $2R$  כי בכל צד יש  $+\Delta R$  ו  $-\Delta R$ .



איור 7 : חישוב הפרש המתח  $\Delta V$

בהנחה שהתנגדות כל המדידים שווה הזרם  $I_x = I_y$  כי ההתנגדות בכל סנף של הגשר שווה כי בכל סנף נגד אחד הגדיל את ערכו ונגד אחד הקטין את ערכו. הפרש המתח  $V_0$  הוא :  $V_0 = V_B - V_C$

$$\Delta V_0 = I * (R + \Delta R) - I * (R - \Delta R) = I * 2\Delta R$$

הזרם שזרם לפני הנחת המשקל היה אותו זרם  $I$  כי ההתנגדות בכל סנף לא השתנתה על ידי המשקל :

$$VO = I \cdot 2R$$

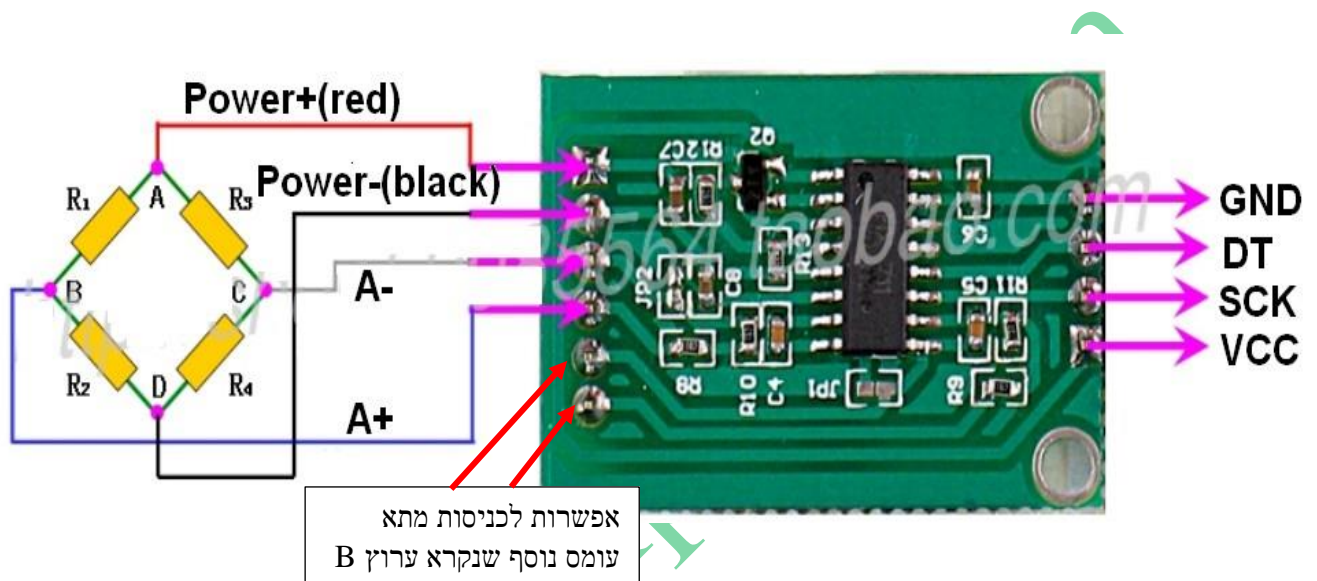
אם נחלק את שתי המשוואות האחרונות אחת בשנייה נקבל:

$$\Delta VO / VO = \Delta R / R$$

שינוי מתח היציאה יחסי ישיר לשינוי ההתנגדות של תא העומס/המדיד.

## איך מודדים את המתחים?

האיור הבא מתאר את חיבור המגבר HX711 אל תא העומס שמורכב מגשר של 4 מדידים של Strain Gauge :



איור 8: חיבור מגבר HX711 אל גשר המדידים

שני ההדקים הריקים בצד שמאל נקראים B- ו B+ והם ערוץ נוסף הנקרא ערוץ B לכניסת הגברה, כלומר ניתן לחבר ל 2 ההדקים האלו תא עומס נוסף. בג'וק יש מרבץ שיכול להכניס את המתח או מערוץ A או מערוץ B. מכיוון ששינוי ההתארכות של המדיד מאוד קטן אז גם שינוי ההתנגדות קטן מאוד ובהתאם שינוי המתח על פניו. ל ADC של מיקרו בקר לא קל להבחין בשינוי מתח כזה קטן ולכן מגבירים את האות בעזרת מגבר HX711, ומעבירים למיקרו בקר אות מוגבר. חיישן המדידים מגיע עם 4 חוטים בצבעים שונים שיש לחבר למגבר HX711 לפי המפתח הבא:

Wheatstone Bridge Node	"Typical" Wire Color
Excitation+ (E+) or VCC	RED
Excitation- (E-) or GND	BLACK or YELLOW
Output+ (O+), Signal+ (S+), or Amplifier+ (A+)	WHITE
O-, S-, or A-	GREEN or BLUE

טבלה 1 : הצבעים של החוטים היוצאים מתא העומס אל המגבר



## 6. מגבר חיישן המשקל HX711

HX711 הוא ממיר אנלוגי לדיגיטלי (ADC) מדויק של 24 סיביות. ממיר ה-ADC מסוג דלתא-סיגמא  $\Sigma\Delta$  - 24BIT SIGMA DELTA ADC המיועד למשקל, מאזניים ויישומי בקרה תעשייתיים כדי להתממשק ישירות עם תא עומס. הוא מיוצר במיוחד להגברת אותות מתאים ולהתחברות בתקשורת אל מיקרו בקר. נתונים על הרכיב ניתן למצא בקישור:

<https://www.digikey.com/htmldatasheets/production/1836471/0/0/1/hx711.html>

### א.6 ממשק התקשורת הטורי של המגבר HX711

החיבור בין המיקרו בקר לכרטיס המגבר HX711 הוא בעזרת שני ההדקים SCK ו-DT. 2 הדקים אלו מתחברים אל 2 הדקים בג'וק שנקראים PD\_SCK ו-DOUT. הם משמשים להעברת מידע, בחירת כניסה, בחירת ההגבר ובקרת הורדת ההספק. החיבור הוא טורי עם הדק נתון - DT - והדק שעון SCK (ומכאן שהוא סנכרוני) אך הוא לא מוגדר כפרוטוקול תקשורת כמו I2C או SPI או I2S או פרוטוקולים אחרים אלא יש לו פרוטוקול ייחודי.

התקשורת בקו DT היא חד כיוונית מהמגבר אל המיקרו כאשר הנתון מסונכרן על ידי פולסי השעון שיוצר המיקרו בקר. הנתון מהממיר של 24 הביטים בג'וק HX711 מועבר אל המיקרו בקר. ב-24 פולסי השעון הראשונים שיוצר המיקרו בקר מועבר הערך שקרא ה-ADC ואז המיקרו בקר יכול לשלוח עוד 1 עד 3 פולסי שעון נוספים. בעזרת הפולסים הנוספים המיקרו אומר למגבר מה מאיזה ערוץ (A או B) הוא רוצה לקרוא ומהו הגבר המגבר בתוך הג'וק (לפי הטבלה שנראה בהמשך).

היציאה של 24 סיביות של נתונים הוא בפורמט המשלים ל-2 - 2's complement. כאשר אות הכניסה יוצא מטווח 24

הסיביות, נתוני היציאה של הנתון יהיו ב-800000H (MIN) או 7FFFFFFh (MAX).

כאשר ה-ADC לא סיים את פעולת ההמרה והנתון איננו מוכן להעברה הדק DOUT במצב גבוה והמיקרו בקר דואג שבהדק השעון יהיה מתח נמוך.

כאשר הדק DOUT עובר לנמוך זה אומר שנתון מוכן להעברה. המיקרו בקר מוציא 25 עד 27 פולסי שעון לפי הפרוט הבא:

כאשר ניתן בין 25 ל-27 פולסים חיוביים בהדק השעון הנתון יצא (יוזם) החוצה מהדק היציאה DOUT ומועבר למיקרו בקר.

כל פולס שעון מוציא החוצה ביט אחד כאשר הביט הראשון שיוצא הוא ה-MSB עד שכל 24 הביטים יוצאים החוצה.

הביט ה-25 יגרום להדק ה-DOUT לחזור שוב ל-1 (כפי שנראה באיור שבהמשך).

בחירת הכניסה (ערוץ A או B) וההגבר נשלטת על ידי כמות הפולסים שנשלח בהדק השעון. הטבלה הבאה מסבירה את קביעת

ההגבר.

PD_SCK Pulses	Input channel	Gain
25	A	128
26	B	32
27	A	64

טבלה 2: קביעת ההגבר של המגבר בעזרת כמות

הפולסים בהדק השעון.

מהטבלה רואים שכמות הפולסים ששולחים היא בין 25 ל-27 פולסים. ב-24 הפולסים הראשונים אנחנו קוראים את הנתון שה-

ADC הממיר וכמות הפולסים הבאים (עוד פולס שעון 1 או 2 או 3) קובעים מאיזה ערוץ רוצים לקבל את הקריאה של ה-ADC

ומהו ההגבר שאנחנו רוצים שיהיה לו. כאשר כמות הפולסים שנשלח היא 25 הערוץ שרוצים לקרוא את ההמרה שלו הוא ערוץ A

וקבענו שההגבר שלו הוא 128. הקריאה הבאה שנבצע תהיה מערוץ A שההגברה שלו נקבעה כבר ל-128 ואם רוצים להמשיך

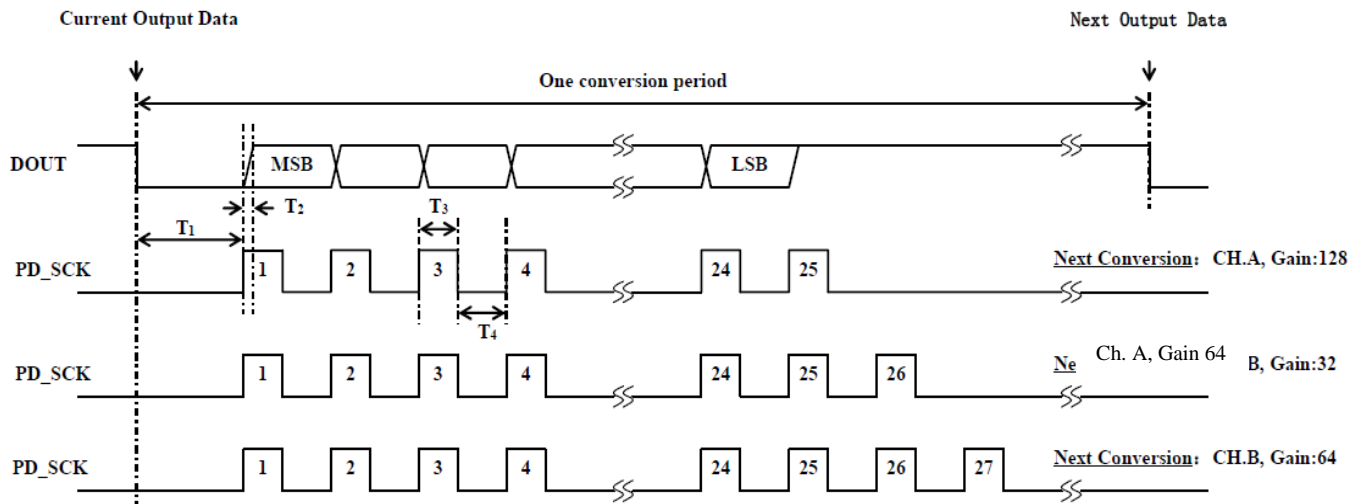
לקרוא ממנו שוב נשלח לו שוב 25 פולסים. אם רוצים לבצע קריאה של המרה, אחת אחרי השנייה מערוץ זה נשלח כל פעם רצף של

25 פולסים.

כאשר כמות הפולסים שנשלח היא 26 אנחנו רוצים לקרוא את ההמרה של ערוץ B והגבר שלו הוא 32. כלומר במחזור הבא נקרא את ההמרה של ערוץ B בעזרת 24 פולסי השעון הראשונים ומספר הפולסים הבאים (בין 25 עד 27) אומרים מאיזה ערוץ נרצה את הקריאה הבאה.

כאשר כמות הפולסים שנשלח היא 27 נקרא את ההמרה של ערוץ A וקבענו לו הגבר של 64.

האיור הבא מתאר את צורות הגל של העברת הנתונים:



Symbol	Note	MIN	TYP	MAX	Unit
T <sub>1</sub>	DOUT falling edge to PD_SCK rising edge	0.1			μs
T <sub>2</sub>	PD_SCK rising edge to DOUT data ready			0.1	μs
T <sub>3</sub>	PD_SCK high time	0.2	1	50	μs
T <sub>4</sub>	PD_SCK low time	0.2	1		μs

איור 9: פרוטוקול התקשורת הטורית וטבלת זמנים.

האיור מתאר 4 צורות גלים. צורת הגל שלמעלה היא DOUT. התקשורת היא של ביט ה MSB ראשון וביט LSB אחרון. צורת הגל שמתחתיה מתארת 25 פולסי שעון שבה 24 פולסי השעון הראשוני קוראים מה ADC של הערוץ שקבענו במחזור הקודם ועוד פולס מספר 25 שאומר שבמחזור הבא רוצים לקרוא מערוץ A וקובעים שהגברת הערוץ תהיה 128. במקרה כזה הקריאה הבאה תהיה ההמרה מערוץ A שהגברתה שקבענו לו הגבר של 128.

צורת הגל המתארת 26 פולסי שעון מתארת 24 פולסי קריאה מה ADC מהערוץ שנקבע במחזור הקודם והפולסים 25 ו 26 קובעים שבמחזור הבא נקרא את ההמרה מערוץ B שקובעים שהגברתה שלו היא 32 (זהו ההגבר היחיד שיש לערוץ B והוא לא ניתן לשינוי).

צורת הגל האחרונה של 27 הפולסים מתארת 24 פולסי קריאה מהערוץ שנקבע במחזור קודם וקובעת שבמחזור הבא נקרא מערוץ A לאחר שקבענו שהגברתה שלו הוא 64.

נסביר את עיקרון העבודה של קריאת ההמרה עם צורת הגל של 25 פולסים. 2 צורות הגלים הבאות עובדות בצורה דומה.

צורת הגל של DOUT מתחילה עם ירידת הקו ל 0. לפני כן היה בקו DOUT = 1 שאומר שנתון היציאה איננו מוכן להעברה. הדק DOUT במצב גבוה ויש לדאוג שמצב הדק השעון יהיה בנמוך.



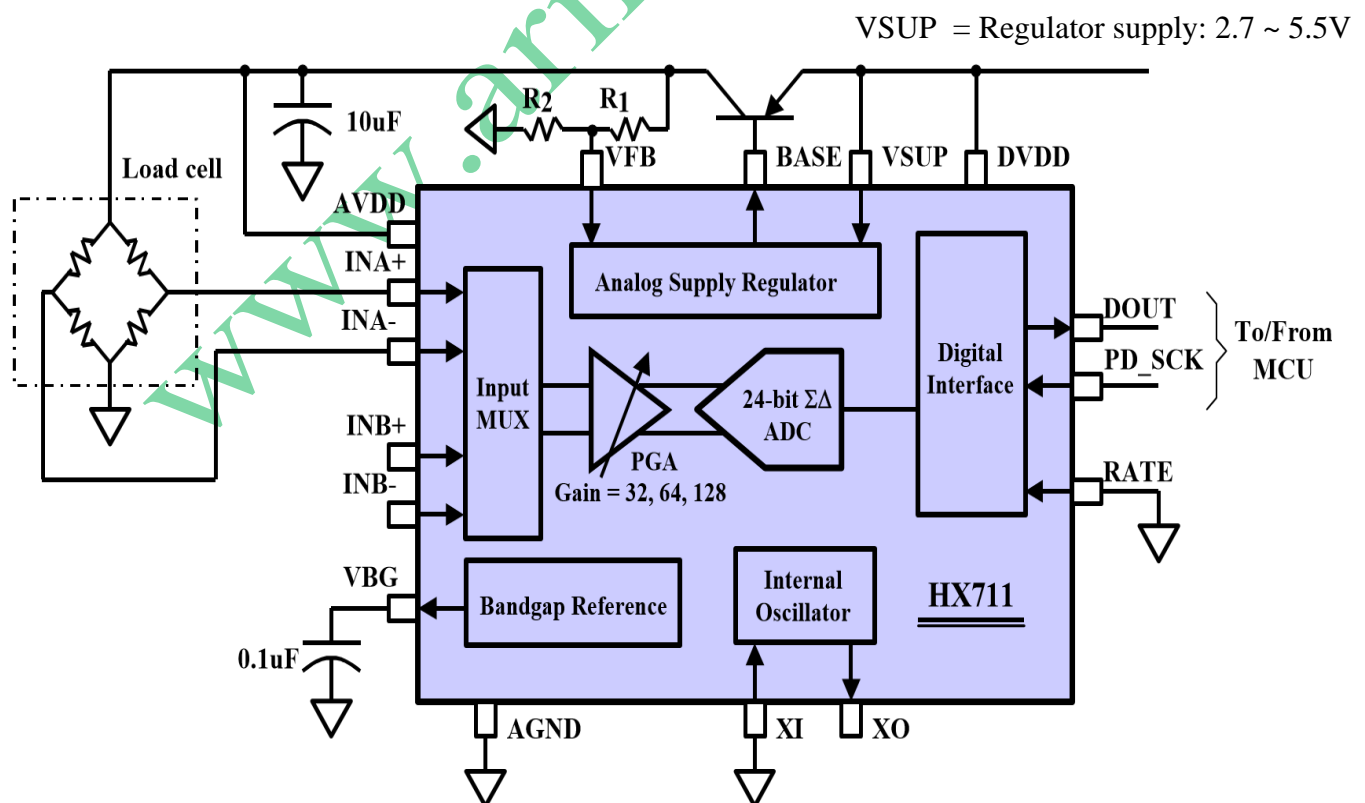
ברגע שהיציאה DOUT יורדת ל 0 זה אומר שבוצעה המרה והרכיב מוכן להוציא נתונים למיקרו בקר. המיקרו בקר מעלה את הדק השעון ל 1 (פולס מספר 1 באיור) ואז הממיר מוציא את הביט הראשון (ביט ה MSB) על הקו DOUT. המיקרו מוריד את קו השעון ל 0 ומושך אליו את הערך שיש בקו DOUT. אחר כך מגיע פולס השעון השני. עם העלייה של פולס השעון הממיר מוציא את הביט השני לקו ה DOUT ובירידה של פולס השעון המיקרו מושך את הביט השני של הממיר. התהליך חוזר על עצמו עד הפולס ה 24 שהוא הנתון שהומר. פולס מספר 25 הוא פולס שהמיקרו לא צריך לקרוא כערך של המרה. פולס זה אומר ל ADC שההמרה הבאה תהיה מערוץ A והממיר צריך לקבוע את הגבר הערוץ ל 128.

#### הסבר זמנים:

- T1** - הוא הזמן שיש לחכות (לפי הטבלה הוא לפחות 0.1 מיקרו שניות) בין המצב שהממיר מוריד את הדק DOUT ל 0 ומודיע שיש לו נתון חדש לשלוח ובין הזמן שהמיקרו מעלה את הדק השעון ל 1 כדי לקרוא את הביט הראשון מממיר ה ADC. מעשית, כשעובדים עם מיקרו בקר ממשפחת ה 51 או עם ארדואינו, אין צורך לבצע השהייה כלשהי אלא ניתן להעלות מיד את הדק השעון.
- T2** - הזמן מרגע שיש עלייה בקו השעון עד שהממיר שם נתון על קו DOUT אסור שיעלה על 0.1 מיקרו שניות.
- T3** - הזמן שבו פולס השעון יהיה בגובה - אסור שיעלה על 50 מיקרו שניות. זמן אופייני הוא 1 מיקרו שניות. באחריות המתכנת שמעלה את קו השעון ל 1 להוריד את הקו ל 0 לפני שמסתיימים 50 מיקרו שניות.
- T4** - הזמן שפולס השעון נמצא ב 0 - זמן אופייני הוא 1 מיקרו שניות. אין למצב זה זמן מקסימאלי.

## 6.6 קצת אלקטרוניקה על מגבר HX711.

האיור הבא מתאר את הסכמה המלבנית של הג'וק/שבב HX711 וכיצד לחבר אותו אל תא עומס.



איור 10: הסכמה המלבנית של הג'וק HX711.

**6.g בתחילה נסביר את המלבנים המרכיבים את הג'וק:****א. הכניסות האנאלוגיות מתא העומס**

בצד שמאל של האיור רואים את הכניסות של תא העומס אל מרבב כניסה. המרבב מקבל 2 כניסות מערוץ A ו 2 כניסות מערוץ B. כל כניסה הפרשית כזאת היא מתא עומס. ערוץ A הוא כניסה הפרשית מתא העומס/גשר והוא מתוכנן להתממשק ישירות עם הפלט הפרש המתח של חיישן הגשר. ניתן לקבוע איזה ערוץ יועבר אל המגבר בר התכנות – PGA – וממנו אל ממיר הדלתא סיגמה בן 24 הביטים..

**ב. PGA – Programmable Gain Amplifier - מגבר בר תכנות**

ניתן לקבוע את ההגברה של המגבר המתוכנת להגבר של 128 או 64. ההגברים הגדולים נדרשים כדי להגביר את האות החלש היוצא מהגשר. אם נעבוד עם 5V בהדק AVDD אז ההגברים האלו תואמים למתח כניסה דיפרנציאלי של עד  $\pm 20\text{mV}$  או עד  $\pm 40\text{mV}$  בהתאמה. לכניסה הפרשית של ערוץ B יש הגבר קבוע של 32. טווח מתח כניסה הפרשי בערוץ זה הוא עד  $\pm 80\text{mV}$ .

**ג. 24 Bit  $\Sigma\Delta$  ADC - ממיר אנאלוגי לדיגיטאלי של 24 ביטים מסוג דלתא סיגמה**

זהו ממיר עם רזולוציה של 24 ביט. בממיר כזה משתמשים באפנון דלתא סיגמה כדי להפוך את האות האנאלוגי לאות דיגיטאלי. האות האנאלוגי נדגם בקצב גבוה ומומר לזרם סיביות ברזולוציה גבוהה. לאחר מכן משתמשים בפילטר דיגיטאלי להורדת הרזולוציה אבל עם רמת דיוק גבוהה. יתרון ADC כזה הוא הרזולוציה הגבוהה אבל הוא מורכב ליישום והוא איטי עקב קצב הדגימה הגבוה וכמות הסיביות ברזולוציה.

**ד. Analog Supply Regulator - מייצב מתח אנאלוגי.**

ספק הכוח הדיגיטלי (DVDD) צריך להיות זהה לספק הכוח של המיקרו-בקר. בעת שימוש במייצב המתח שבג'וק מתח היציאה מהמייצב תלוי בטרנזיסטור החיצוני שנחבר. מתח היציאה שווה ל:

$$V_{AVDD} = V_{BG} * (R1 + R2) / R1$$

מתח זה צריך להיות מתוכנן עם מינימום של 100 mV מתחת למתח VSUP.

**אם לא משתמשים במייצב המתח האנאלוגי שבג'וק -** יש לחבר את פין VSUP ל AVDD או ל DVDD בהתאם למתח הגבוה יותר. פין VFB צריך להיות מחובר להארקה ו BASE PIN הופך ל- NC (ללא חיבור). כמו כן אין צורך בקבל המעקף החיצוני 0.1 uF בפין יציאת VBG.

**ה. Internal Oscillator - מתנד פנימי**

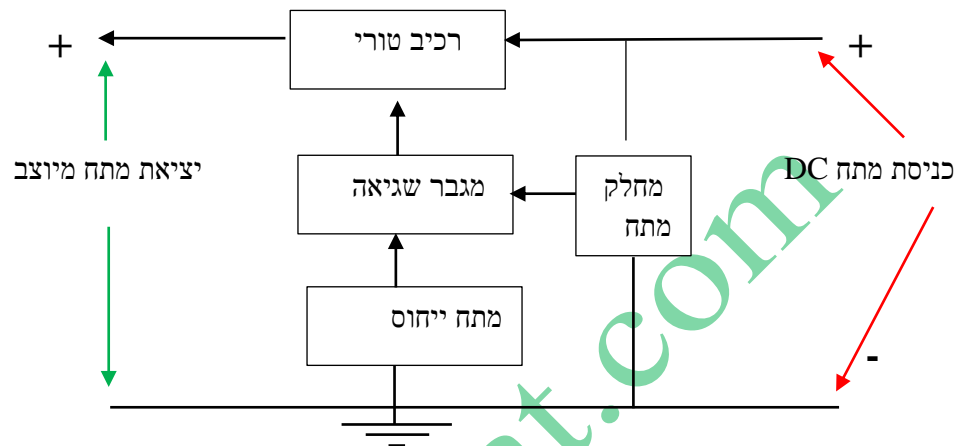
על ידי חיבור פין XI לאדמה מפעילים את המתנד בג'וק. קצב נתוני היציאה בעת שימוש במתנד הפנימי הוא 10 SPS – Samples Per Second - דגימות בשנייה כאשר נשים בהדק (RATE=0) או עבור (RATE=1) נקבל 80 דגימות בשנייה. אם יש צורך בקצב נתונים מדויק ניתן להשתמש בשעון ייחוס גביש או מתנד חיצוני. ניתן לחבר גביש ישירות בין פנים XI ו- XO או ניתן לחבר שעון חיצוני לפין XI באמצעות קבל של 20pF. שעון חיצוני זה אינו נדרש להיות גל מרובע. זה יכול להגיע ישירות מהדק יציאה של הגביש של ג'וק המיקרו-בקר עם אמפליטודה נמוכה עד 150 mV.

**ו. Digital Interface - הממשק הדיגיטאלי**

קצב יציאת הנתונים בעת שימוש במתנד שבג'וק הוא בדרך כלל 10 דגימות בשנייה עבור (RATE=0) או עבור (RATE=1) הקצב הוא 80 דגימות בשנייה. בעת שימוש בשעון חיצוני או גביש חיצוני קצב נתוני הפלט עומד ביחס ישר לשעון או לתדר הגביש. עבור שעון או גביש של 11.0592MHz נקבל את אותם התוצאות של 10 או 80 דגימות בשנייה.

**7.6** נסביר את תפקיד ההדקים של הג'וק וכיצד הוא מחובר בכרטיס. נתחיל למעלה באיור מימין לשמאל נגד כיוון השעון : מתח ספק הכוח לרכיב נקרא **VSUP** והוא יכול להיות בין 2.7 וולט עד 5.5 וולט. מתח זה מגיע למלבן Analog Supply Regulator – מייצב ספק אנאנלוגי .

מייצב מתח ליניארי מורכב מהמלבנים הנראים באיור הבא :



איור 11 : מבנה של מייצב מתח ליניארי.

עיקרון הייצוב: מתח הכניסה הלא מיוצב נכנס למייצב מצד ימין ומגיע אל הרכיב טורי.

מגבר השגיאה מגביר את הפרש המתח בין מתח הייחוס והמתח ממחלק המתח.

מגבר השגיאה מעביר מתח אל הרכיב הטורי. הרכיב הטורי משנה התנגדותו ומעביר אל היציאה מתח מיוצב.

לדוגמה : אם מתח הכניסה עולה מסיבה כלשהי גם מתח היציאה "רוצה" לגדול. פעולת הייצוב מתבצעת בצורה הבאה :

המתח שמגיע למגבר השגיאה ממחלק המתח גדול יותר. מגבר השגיאה מוציא לרכיב הטורי פקודה להגדיל את התנגדותו כך

שנופל עליו מתח גדול יותר ממקודם ולכן ליציאה יצא מתח קטן יותר .

נקשר בין מייצב המתח לסכמה של הרכיב.

הרכיב הטורי הוא הטרנזיסטור.

מחלק המתח הוא המתח בכניסת VFB . מחלק המתח הוא המתח בין R1 ו R2.

מתח הייחוס ומגבר השגיאה לא נראים בסכמה המלבנית של הרכיב.

המתח הנקרא **DVDD** הוא המתח המגיע לרכיבים הדיגיטאליים בתוך ה HX711 . הוא שימושי אם רוצים להפעיל את הרכיבים

הדיגיטאליים בג'וק בנפרד ממתח הספק. באיור הוא חובר ל VSUP והוא יכול להיות בין 2.6 ~ 5.5V וולט .

הטרנזיסטור

הדק **BASE** – ההדק נקרא Regulator control output - יציאת בקרת מייצב זהו המתח המגיע אל הרכיב הטורי שהוא

הטרנזיסטור מסוג PNP . מתח זה גורם לשינוי התנגדות הטרנזיסטור ולייצוב המתח המגיע אל תא העומס.

**VFB** - נקרא Regulator control input – כניסת בקרת המייצב . זהו הדק כניסה ממחלק מתח בין R1 ו R2 המגיע למגבר

השגיאה. המתח להדק זה ממחלק המתח קובע את המתח שיוצא בהדק ה BASE לרכיב הטורי את ואת המתח שיצא לתא העומס.

המתח מיציאת קולט הטרנזיסטור מגיע לקבל של 10μf וממנו אל תא העומס – Load Cell ולהדק הנקרא **AVDD** .

AVDD משמש ככניסה של מתח אנאלוגי לרכיבים האנלוגיים ברכיב. מתח זה יכול להיות בין 2.6 ~ 5.5V . בכרטיס הוא מחובר אל המתח המיוצב שביציאת הטרנזיסטור .

ההדקים INA+ ו- INA- הן הכניסות מתא העומס. שתי כניסות אלו נכנסות אל מרבב הכניסה INPUT MUX . המרבב מקבל 2 כניסות נוספות הנקראות INB+ ו- INB- . כניסות אלו יכולות להיות מתא עומס נוסף.

הדק VBG הוא Reference bypass output – יציאת עקיפת הייחוס. כאשר משתמשים במייצב המתח הפנימי מתח היציאה תלוי בטרנזיסטור החיצוני ששמים. מתח היציאה שווה ל :

$$V_{AVDD} = V_{BG} * (R1 + R2) / R1$$

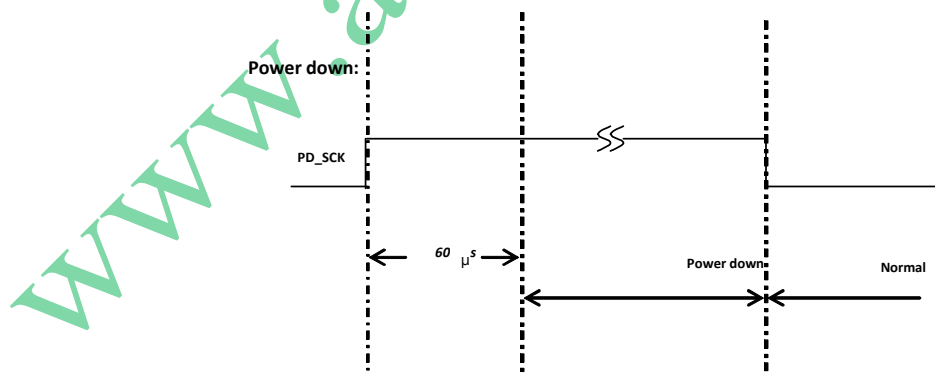
מתח זה צריך להיות מתוכנן עם מינימום של 100mV מתחת למתח VSUP . בהדק זה מחובר בכרטיס שלנו קבל סינון של 0.1 מיקרו פאראד.

AGND – הוא הדק האדמה של ספק המתח האנלוגי.

XO ו- XI הם הדקים המתחברים למעגל המתנד Internal Oscillator - שבתוך הג'וק. כאשר XI=0 זה אומר שמשתמשים בגביש הפנימי שבתוך הג'וק ומשאירים את הדק XO באוויר Not Connected- . אם שמים בהדק זה '1' אז ניתן לחבר ל XO גביש חיצוני.

RATE - בקרה של קצב הנתונים. אם נשים בהדק '0' אז קצב הנתונים הוא 10Hz . אם נשים '1' אז קצב הנתונים יהיה 80Hz . בכרטיס המגבר ההדק מחובר לאדמה ולכן קצב הנתונים הוא 10Hz .

PD\_SCK - Power down control (high active) and serial clock input - בקרת הורדת ההספק . כניסה זו מגיעה מהמיקרו בקר. במצב גבוה יש הורדת הספק ברכיב . במצב נמוך הג'וק נמצא במצב עבודה רגיל. הדק זה הוא כניסת פולסי השעון מהמיקרו בקר. הדק זה של הג'וק מתחבר לפין SCK בקונקטור JP2 של הכרטיס. האיור הבא מתאר מצב של הורדת ההספק :

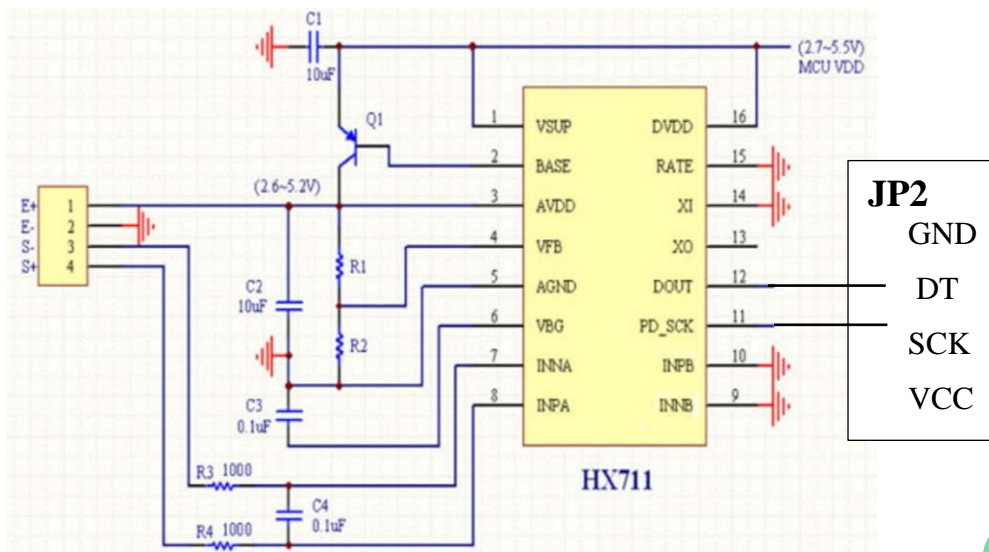


איור 12 : מצב של הורדת ההספק.

כאשר הדק PD\_SCK משתנה מנמוך לגבוה ונשאר גבוה למשך יותר מ-60μs הג'וק HX711 נכנס למצב הורדת ההספק – Power Down . כאשר PD\_SCK חוזר למצב נמוך, השבב יתאפס - reset - וייכנס למצב פעולה רגיל. לאחר reset או כיבוי החשמל בחירת הכניסה מוגדרת כברירת מחדל לערוץ A עם הגבר של 128.

DOUT – הדק יציאת הנתון מהג'וק אל המיקרו בקר. בכרטיס המגבר ההדק מתחבר אל פין DT בקונקטור JP2 של הכרטיס.

הסכמה החשמלית של המגבר נראית כך :



איור 13 : סכמה מלבנית של כרטיס המגבר HX711

כפי שציינו בכרטיס המגבר יש את המגבר HX711 שבו ADC של 24 ביטים ומגבר מתח בר תכנות ל 3 רמות הגברת מתח. הטבלה הבאה מתארת את תפקיד ההדקים של הג'וק :

Pin #	Name	Function	Description
1	VSUP	Power	Regulator supply: 2.7 ~ 5.5V
2	BASE	Analog Output	Regulator control output ( NC when not used )
3	AVDD	Power	Analog supply: 2.6 ~ 5.5V
4	VFB	Analog Input	Regulator control input ( connect to AGND when not used )
5	AGND	Ground	Analog Ground
6	VBG	Analog Output	Reference bypass output
7	INA-	Analog Input	Channel A negative input
8	INA+	Analog Input	Channel A positive input
9	INB-	Analog Input	Channel B negative input
10	INB+	Analog Input	Channel B positive input
11	PD_SCK	Digital Input	Power down control (high active) and serial clock input
12	DOUT	Digital Output	Serial data output
13	XO	Digital I/O	Crystal I/O (NC when not used )
14	XI	Digital Input	Crystal I/O or external clock input, 0: use on-chip oscillator
15	RATE	Digital Input	Output data rate control, 0: 10Hz; 1: 80Hz
16	DVDD	Power	Digital supply: 2.6 ~ 5.5V

טבלה 3 : תפקיד ההדקים של הג'וק

הטבלה הבאה מתארת את המאפיינים החשמליים של הג'וק :

Parameter	Notes	MIN	TYP	MAX	UNIT
Full scale differential input range	$V(\text{inp}) - V(\text{inn})$	$\pm 0.5(AVDD/GAIN)$			V
Common mode input		AGND+1.2		AVDD-1.3	V
Output data rate	Internal Oscillator, RATE = 0		10		Hz
	Internal Oscillator, RATE = DVDD		80		
	Crystal or external clock, RATE = 0		$f_{\text{clk}}/1,105,920$		
	Crystal or external clock, RATE = DVDD		$f_{\text{clk}}/138,240$		
Output data coding	2's complement	800000		7FFFFFFF	HEX
Output settling time <sup>(1)</sup>	RATE = 0		400		ms
	RATE = DVDD		50		
Input offset drift	Gain = 128		0.2		mV
	Gain = 64		0.4		
Input noise	Gain = 128, RATE = 0		50		nV(rms)
	Gain = 128, RATE = DVDD		90		
Temperature drift	Input offset (Gain = 128)		$\pm 6$		nV/°C
	Gain (Gain = 128)		$\pm 5$		ppm/°C
Input common mode rejection	Gain = 128, RATE = 0		100		dB
Power supply rejection	Gain = 128, RATE = 0		100		dB
Reference bypass ( $V_{BG}$ )			1.25		V
Crystal or external clock frequency		1	11.0592	20	MHz
Power supply voltage	DVDD	2.6		5.5	V
	AVDD, VSUP	2.6		5.5	
Analog supply current (including regulator)	Normal		1400		μA
	Power down		0.3		
Digital supply current	Normal		100		μA
	Power down		0.2		

(1) Settling time refers to the time from power up, reset, input channel change and gain change to valid stable output data.

טבלה 4 : מאפיינים חשמליים

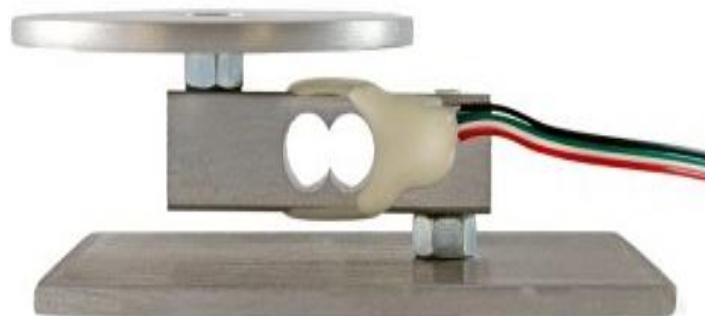
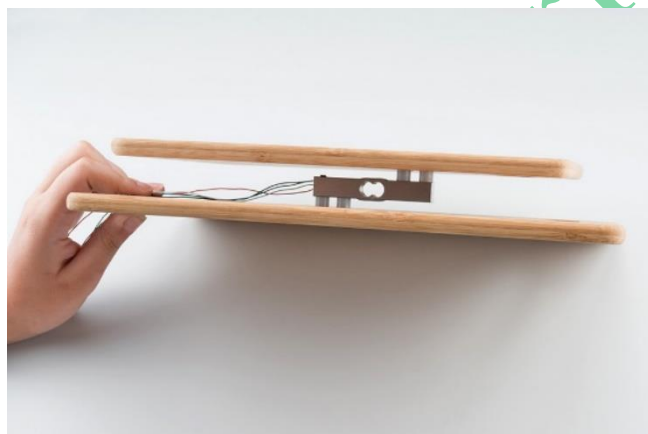


## 7. התקנה/הרכבה של חיישן המשקל

תאי עומס כוללים בדרך כלל חץ המציג את הכיוון שבו יש להחיל את העומס. קצה אחד של תא המטען מאובטח לבסיס תחתון, ואילו השני מאובטח לצלחת העליונה. לא ניתן לשלב תאי עומס אלה כדי להגדיל את יכולת השקילה. האיור הבא מראה את החלקים המרכיבים תא עומס :



באיור הבא רואים 2 דוגמאות של תאי עומס מורכבים.



איור 15 : שתי דוגמאות של תאי עומס מורכבים.

בכל אחת מהדוגמאות באיור רואים את הבסיס התחתון אליו מבריגים את הצד הקבוע של תא העומס. את הצד השני עליו מניחים את העומס מחברים לבסיס עליון עגול או מלבני שעליו נשים את המשקל. הצד עליו מניחים את המשקל מסומן עם חץ.

## 8. חיבור חיישן המשקל אל ESP32

מגבר HX711 מתקשר באמצעות ממשק דו-חוטי. אפשר לחבר אותו לכל GPIOs של המיקרו-בקר שנבחר. בדוגמה כאן אנו מחברים את פין הנתונים (DT) ל-GPIO 16 ואת פין השעון (CLK) ל-GPIO 4. ניתן להשתמש בכל פין מתאים אחר. קישור למדריך ההדקים של ESP32 נתון בקישור :

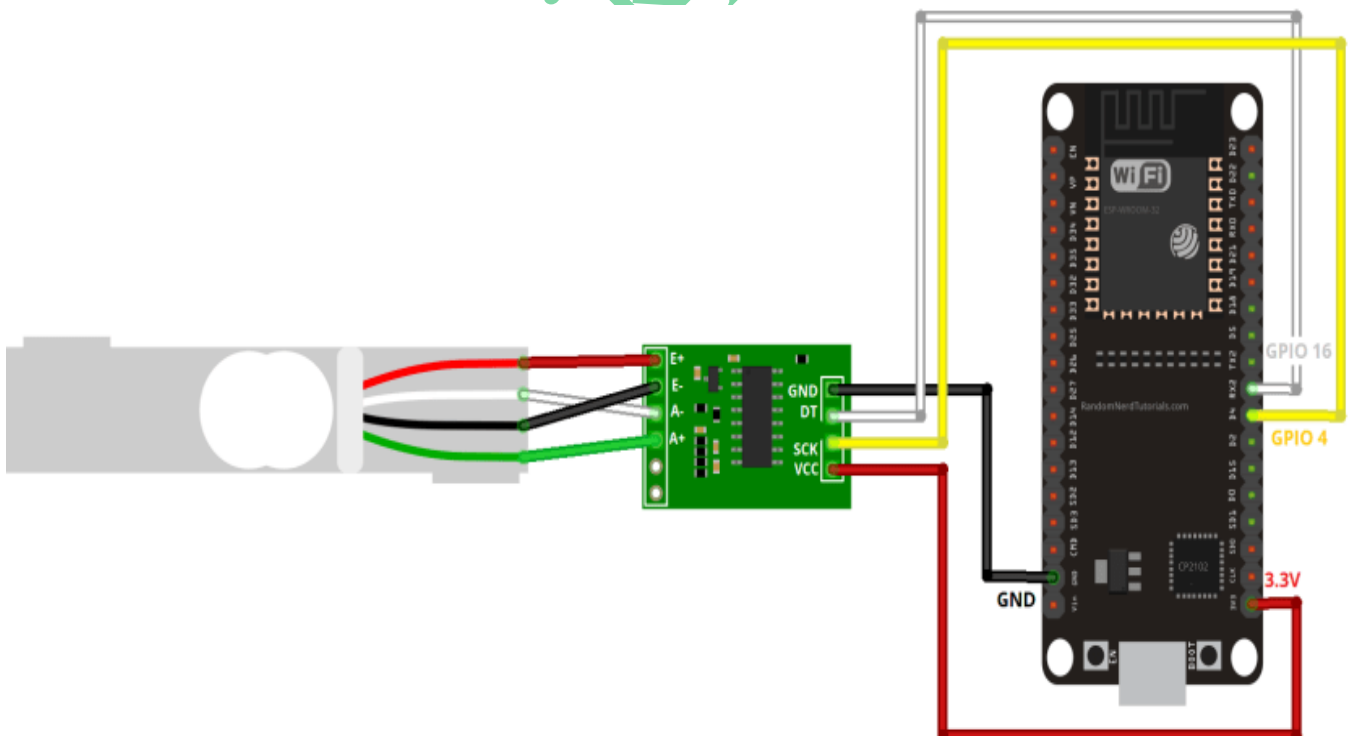
<https://www.arikporat.com/wp-content/uploads/2023/07/%D7%94%D7%A1%D7%91%D7%A8-%D7%AA%D7%A4%D7%A7%D7%99%D7%93-%D7%94%D7%93%D7%A7%D7%99-ESP32-.pdf>

הטבלה הבאה מתארת את חיבור הדקי המגבר HX711 אל מיקרו בקר ESP32 כולל צבעי החושים של תא המאמץ.

Load Cell	HX711		HX711	ESP32
Red (E+)	E+		GND	GND
Black (E-)	E-		DT	GPIO 16
White (A-)	A-		SCK	GPIO 4
Green (A+)	A+		VCC	3.3V

טבלה 4 : חיבור הדקי המגבר HX711 אל מיקרו בקר ESP32 .

בחלק השמאלי של הטבלה יש את חיבור תא העומס אל המגבר. בחלק הימני חיבור הדקי התקשורת אל המיקרו בקר. האיור הבא מתאר בצורה ברורה יותר את סכמת החיבור :



איור 16 : סכמת החיבור של חיישן המשקל אל המיקרו בקר ESP32 .

**המלצה :** כדי להקטין במידת האפשר את הרעש מומלץ לסובב – twist - את החוטים בין HX711 לתא הטעינה ולשמור אותם קצרים ככל האפשר. רוב המודולים של HX711 מתוכננים סביר אך יש מודולים זולים מסוימים שמתוכננים בצורה גרועה עם קבלים קטנים וקריאות רועשות. מודול של חברת Sparkfun נראה שונה מרוב המודולים הזמינים האחרים מכיוון שיש לו כמה רכיבים נוספים להפחתת רעש.

## 9. התקנת ספריית HX711

ישנן מספר ספריות שונות לקבלת מדידות מתא עומס באמצעות מגבר HX711. נשתמש בספריית HX711 באמצעות bodge. הוא תואם ל-ESP32, ESP8266 וארדואינו. הקישור ל GitHub ולספרייה :

[GitHub - bogde/HX711: An Arduino library to interface the Avia Semiconductor HX711 24-Bit Analog-to-Digital Converter \(ADC\) for Weight Scales.](https://github.com/bogde/HX711)

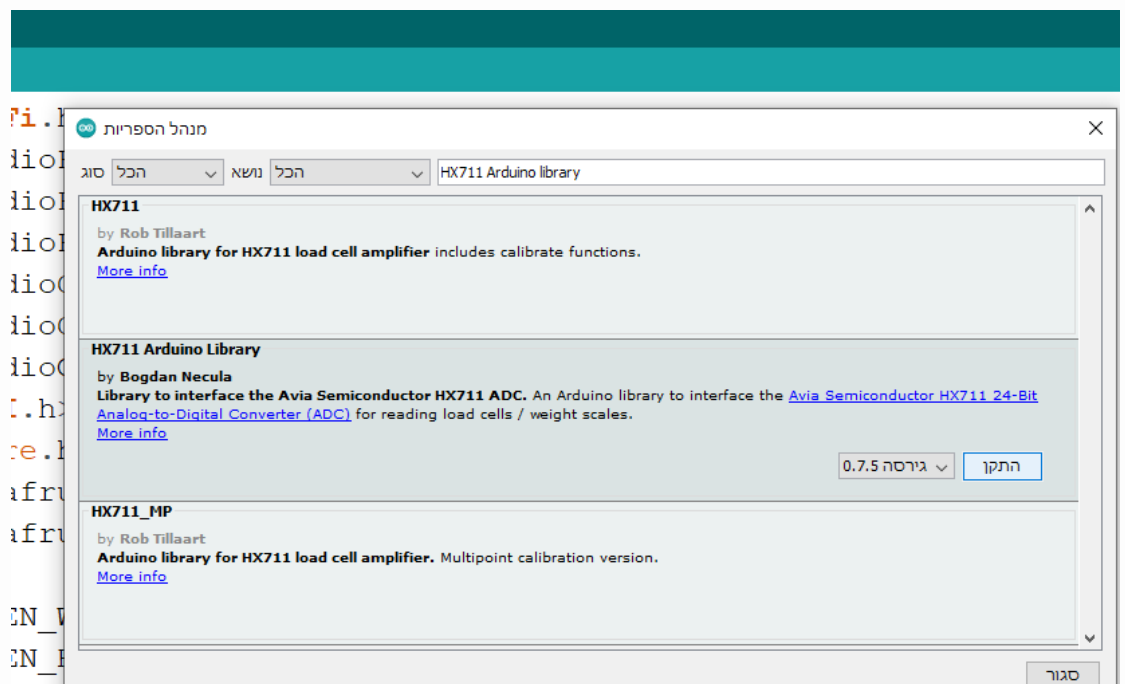
אנו נוסיף את הספרייה בעזרת סביבת העבודה של הארדואינו Arduino IDE. יש לבצע את ההוראות הבאות כדי להתקין את הספרייה :

- נפתח את IDE Arduino ונעבור אל **Sketch > Include Library > Manage Libraries**.

בעברית : נהל ספרייה → כלול ספרייה → סקיצה

- נחפש את "Library Arduino HX711" ונתקין את הספרייה על ידי Necula Bogdan.

האיור הבא מתאר את המסך המתקבל :



איור 17 : מסך הבחירה בספרייה הרצויה

- נבחר את הספרייה המרכזית ונלחץ על "התקן" או install באנגלית. בסיום ההתקנה נלחץ על "close" או "סגור" בעברית.

## 10. הספרייה של HX711

### 10.1 כללי

זוהי ספריית Arduino עבור HX711 עם 24 bit ADC. אחזור הנתונים מה-HX711 נעשה ללא חסימת המיקרו-בקר, גם בהגדרת קצב 10SPS (10 דגימות בשנייה) ועם מספר המרות של HX711 בו זמנית. ניתן לבצע פעולת Tare (מדידת משקל טרה – ללא משקל על תא העומס) גם מבלי לחסום את המיקרו-בקר.

ניתן לבצע סינון והחלקה בשיטה הנקראת "ממוצע נע" על ידי מדידה של מספר מדידות הנקראות **סט מדידות או ערכת נתונים** והחזרת הממוצע שלהן, בשילוב עם הסרה/ביטול של קריאות חריגות גבוהות/נמוכות.

הערכים של מספר הדגימות ניתנים לבחירה בקובץ config.h:

- כמות נתונים ממוצעת יכולה להיות של 1, 2, 4, 8, 16, 32, 64 או 128 דגימות (ברירת מחדל: 16).
- התעלמות מחריגה גבוהה: דגימה אחת נוספת לערכת הנתונים, המערכת מתעלמת מהערך הגבוה ביותר של כל הדגימות בערכת הנתונים (ברירת מחדל: 1).
- התעלמות מחריגה נמוכה: דגימה אחת נוספת לערכת הנתונים, המערכת מתעלמת מהערך הנמוך ביותר של כל הדגימות בערכת הנתונים (ברירת מחדל: 1).
- אפשרור השהיה לכתיבה להדק השעון `seck` עבור מיקרו-בקרים מהירים יותר כמו ESP32 (ברירת המחדל: ללא השהיה).
- מניעת פסיקות כאשר פין הגרב גבוה. ייתכן שהדבר יידרש כדי להימנע מ"מצב כיבוי" אם יש לך רוטינות פסיקה אחרות שגוזלות זמן רב ( $60\mu s$ ) (ברירת מחדל: פסיקות מופעלות).
- חסימת פסיקות כאשר הדק השעון `seck` בגבוה. ייתכן שהדבר יידרש כדי להימנע מ"מצב כיבוי" "power down mode" אם יש רוטינות פסיקה אחרות נכנסו כאשר ההדק בגבוה שגוזלות זמן רב שגדול מ 60 מיקרו שניות (ברירת מחדל: פסיקות מאופשרות).

**זהירות:** שימוש במספר גבוה של דגימות יחליק יפה את הערך הפלט (ממוצע המשקל), אך גם יגדיל את זמן ההתייצבות ואת זמן ההפעלה/זמן ה-tare (אך לא את זמן התגובה) וזה גם "יאכל" קצת זיכרון.

**חשוב:** ניתן להגדיר את קצב הדגימה של HX711 ל-10SPS או 80 SPS (דגימות לשנייה) על-ידי משיכת רגל 15 של הג'וק הנקראת קצב - RATE - לגבוה (80SPS) או נמוך (10SPS). במודולים עם מגבר HX711 יש בדרך כלל מגשר קצר/הלחמה על המעגל המודפס עבור רגל 15 גבוה/נמוך. ניתן לבדוק את הקצב על ידי מדידת המתח ברגל 15. בקצב 80SPS רעש ADC הוא גרוע מזה שב 10SPS. אם לא נדרש זמן קריאת המרה מהיר מאוד אז קצב של 10SPS אמור להיות קצב הדגימה הטוב ביותר עבור רוב היישומים.

הפעלה ומדידת tare: מרגע הפעלה/איפוס של המגבר כדאי שנקרא לפונקציית ה-tare מדויקת יותר לאחר זמן "טרום-חימום" המריצה המרות ברציפות למשך מספר שניות. נראה שכך ההמרות מדויקות יותר.

**10.ב המתודות בספריה**

ישנן מתודות שהמיקרו בקר פונה אל ה ADC ועד שהוא מבצע ממנו קריאה הוא יכול להמשיך להריץ פקודות אחרות. לעומת זאת יש מתודות שברגע שיש קריאה למתודה המיקרו חסום ולא יכול לעסוק בפעולות אחרות. מצב זה נקרא blocking. למצב זה נקרא בהמשך חסימה או ללא חסימה.

**10.ב.1 אתחול****begin( )**

אתחול התקשורת עם HX711 והגדרת ההגבר ל- 128 (ברירת מחדל).

**begin(uint8\_t gain)**

אתחול התקשורת עם HX711 והגדרת הרווח (32, 64 או 128).

**10.ב.2 מדידת המשקל "טרה" – tare – כיול נקודת האפס****tare( )**

מבצע פעולת tare - מדידת המשקל טרה (ללא משקל על החיישן). יש חסימה – המיקרו ממתיין עד לסיום המדידה.

**tareNoDelay()**

יוזמים פעולת tare (אין חסימה).

**getTareStatus( )**

החזרת ערך True אם פעולת מדידת המשקל טרה - tare – הושלמה.

**10.ב.3 מדידת משקל****long read();**

הפונקציה ממתינה עד שהג'וק יהיה מוכן ומחזירה ערך מטיפוס long שהוא הקריאה של הממיר.

**read\_average(byte times = 10);**

הפונקציה מחזירה ערך מטיפוס long של קריאה ממוצעת. שולחים לפונקציה byte המציין את כמות הפעמים של הקריאה.

**get\_value(byte times = 1);**

הפונקציה מחזירה ערך מטיפוס double שהוא החישוב של : OFFSET – read\_average( ), כלומר הערך הנוכחי ללא משקל הטרה – tare. הפונקציה מקבלת byte המציין את כמות הקריאות מהממיר שיש לבצע.

**get\_units(byte times = 1)**

הפונקציה מחזירה ערך שהוא החלוקה בין get\_value ו SCALE, כלומר  $get\_value() / SCALE$ . זהו הערך הגולמי חלקי ערך המתקבל באמצעות הכיול שעשינו. הפונקציה מקבלת byte שמציין כמה קריאות לבצע מהממיר.

**4.ב.10 איסוף נתונים****update( )**

קורא דגימת משקל חדשה (חסימה, המתנה להמרה).

**dataWaitingAsync( )**

בדיקה אם נתוני משקל חדשים זמינים (ללא חסימה).

**updateAsync( )**

קורא נתוני משקל חדשים אם זמינים (ללא חסימה, נקרא לאחר dataWaitingAsync).

**getData( )**

החזרת ערך המשקל העדכני ביותר (לאחר החלת כיול וסינון).

**5.ב.10 כיול****setScale( )**

קביעת הערך SCALE : ערך זה משמש להמרת הנתונים הגולמיים לנתונים "קריאים אנושיים" (יחידות מדידה כמו גרמים).

**getScale()**

קבלת ה scale הנוכחי

**setCalFactor(float cal)**

קביעת מקדם הכיול calibration factor להמרת משקל . משקל = נתון גולמי לחלק במקדם הכיול .

**getCalFactor( )**

הפונקציה מחזירה את מקדם הכיול הנוכחי.

**getNewCalibration( float known\_mass)**

הפונקציה מחשבת וקובעת מקדם כיול חדש בהתבסס על מסה ידועה.

**6.ב.10 פונקציות אחרות****set\_gain(byte gain = 128)**

קובעים את ההגברה של ה HX711 . נכנס לתוקף רק לאחר ( read( ) . ערוץ A יכול להיות מוגדר להגברה של 128 או 64 . לערוץ B יש הגברה קבועה של 32 . בהתאם לפרמטר, הערוץ מוגדר ל- A או B . הפונקציה מקבלת byte עם ערך ההגברה.

**setSamplesInUse(int samples)**

קביעת מספר הדגימות המשמשות לממוצע ולסינון ( מעוגל כלפי מטה).

**getSamplesInUse( )**

מחזירה את המספר הנוכחי של הדגימות שבשימוש.

**resetSamplesIndex( )**



איפוס האינדקס של סט הנתונים (הנקראת גם ערכת הנתונים).

**refreshDataSet( )**

ריענון כל סט הנתונים עם המרות חדשות ( עם חסימה).

**getDataSetStatus( )**

בדיקה האם ערכת הנתונים מלאה במרות.

**getSignaTimeoutFlag( )**

מציין האם הסתיים הזמן הקצוב לתקשורת עם ה HX711 .

**setReverseOutput( )**

היפוך ערך הפלט. ערך חיובי הופך לשלילי ולהפך.

**getTareOffset( )**

קבלת היסט ה tare הנוכחי ( ערך הנתון הגולמי).

**setTareOffset(long newoffset )**

קובעים את ההיסט החדש של ה tare (ערך נתון גולמי).

**powerUp( )**

הפעלה/הדלקה של ג'וק ה HX711 .

**powerDown( )**

כיבוי הג'וק HX711 .

**getReadIndex( )**

החזרת אינדקס הקריאה הנוכחי של סט הנתונים ( ל debugging ) .

**getConversionTime( )**

החזרת זמן ההמרה האחרון באלפיות שנייה ( ל debugging ) .

**getSPS( )**

הערכת כמות ההמרות בשנייה ( ל debugging ) .

**getTareTimeoutFlag( )**

החזרה של דגל הזמן הקצוב לפעולת tare ( ל debugging ) .

**getSettlingTime( )**

קבלת זמן ההתייצבות המשוער בהתבסס על זמן ההמרה וספירת הדגימות ( ל debugging ) .

## 11. כיול קנה המידה (ESP32 עם תא עומס)

לפני מדידת משקל של אובייקטים יש לכייל תחילה את תא העומס לקבלת "גורם הכיול" - **calibration factor**. מקדם הכיול שונה מתא עומס אחד לתא עומס אחר ולכן לא לדלג על סעיף זה.

- (1) יש להכין חפץ בעל משקל ידוע. אפשר להשתמש בכוס עם מים ולמדוד בעזרת משקל הנמצא במטבח. בדוגמה כאן השתמשתי במשקל המטבח שלי ושקלתי עם מים (300 גרם). למי שיש משקולות כמו שהיו פעם בחנויות המכולת או חפץ כלשהו במשקל ידוע – ניתן להשתמש בו.
- (2) נעלה את הקוד הבא ל-ESP32. כתבנו את הקוד הבא תוך התחשבות בהוראות לכיול תא הטעינה שסופק על ידי הספרייה HX711. כדאי לשים לב לתהליכי המדידה בתוכנית: בהתחלה מתבקשים לבצע מדידה ללא משקל (מדידה הנקראת "טרה" – TARE) ורק אחר כך לשים משקל ידוע. התוכנית נראית כך:

/\*

Rui Santos

Complete project details at [https://RandomNerdTutorials.com/esp32-load-cell-](https://RandomNerdTutorials.com/esp32-load-cell-hx711/)

hx711/

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files.

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

\*/

// Calibrating the load cell

#include <Arduino.h>

#include "soc/rtc.h" // ספרייה לקביעת תדר השעון/ מהירות העבודה של המיקרו בקר

#include "HX711.h"

// HX711 circuit wiring

const int LOADCELL\_DOUT\_PIN = 16;

const int LOADCELL\_SCK\_PIN = 4;

HX711 scale;

void setup() {

Serial.begin(115200);

// תפקיד 4 הפקודות הבאות להאט את מהירות המיקרו בקר ל 80 מגה הרץ

// כדי לבצע התאמה בין תדר המיקרו בקר למהירות העבודה עם המגבר

rtc\_cpu\_freq\_config\_t config;

rtc\_clk\_cpu\_freq\_get\_config(&config);

rtc\_clk\_cpu\_freq\_to\_config(RTC\_CPU\_FREQ\_80M, &config);

rtc\_clk\_cpu\_freq\_set\_config\_fast(&config);

```

scale.begin(LOADCELL_DOUT_PIN, LOADCELL_SCK_PIN);
}

void loop() {

  if (scale.is_ready()) {
    scale.set_scale();
    Serial.println("Tare... remove any weights from the scale.");
    delay(5000);
    scale.tare();
    Serial.println("Tare done...");
    Serial.print("Place a known weight on the scale...");
    delay(5000);
    long reading = scale.get_units(10);
    Serial.print("Result: ");
    Serial.println(reading);
  }
  else {
    Serial.println("HX711 not found.");
  }
  delay(1000);
}

//calibration factor will be the (reading)/(known weight)

```

(3) לאחר העלאת התוכנית נפתח את הצג הטורי בקצב שידור של 115200 ונעשה RESET ללוח ESP32.

(4) נבצע את ההוראות על הצג הטורי: הסר את כל המשקולות מתא העומס ונקבל את המשקל "טרה", כלומר "אריזה" בלבד. לאחר מכן, נמקם אובייקט בעל משקל ידוע על תא העומס ונמתין עד לקבלת ערך.

**הערה:** בחישוב משקל יש את המושגים **ברוטו**, **נטו** ו**טרה**: "ברוטו" הוא משקל האריזה ועוד משקל האובייקט. "נטו" הוא משקל האובייקט ו"טרה" – משקל האריזה בלבד.

(5) נחשב את מקדם הכיול באמצעות הנוסחה:

$$\text{calibration factor} = (\text{reading})/(\text{known weight})$$

גורם כיול = (קריאה) לחלק ב (משקל ידוע)

התוצאות שהתקבלו נראות באיור הבא :

```

COM3
rst:0x1 (POWERON_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
configsip: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:1
load:0x3fff0018,len:4
load:0x3fff001c,len:1044
load:0x40078000,len:8896
load:0x40080400,len:5816
entry 0x400806ac
Tare... remove any weights from the scale.
Tare done...
Place a known weight on the scale...Result: -141449
  
```

איור 18 : תוצאות המדידה בהרצת התוכנית

במקרה שלנו, הקריאה היא -141449 . המשקל הידוע הוא 300 גרם, ולכן מקדם הכיול שלנו יהיה:

$$\text{calibration factor} = (\text{reading})/(\text{known weight}) = -141449 / 300 = -471.49666$$

נשמור את גורם הכיול מכיוון שנזדקק לו מאוחר יותר ( בהמשך ניקח ערך מקורב של -471.497 ). התוצאות שיצאו כאן יהיו בוודאי שונות מאלו שיצאו לכם/לכן.

מכיוון שהפלט של החיישן פרופורציונלי לכוח המופעל על תא העומס, ניתן לכייל את ה **calibration factor** - קנה המידה שלנו באמצעות כל יחידה שנראית הגיונית. השתמשתי בגרמים, אבל ניתן להשתמש בקילוגרמים.

## 12. מדידת משקל של עצמים

### 12.א. התוכנית

כעת, כשאנחנו יודעים מהו גורם הכיול, אפשר להשתמש בתא העומס כדי לשקול אובייקטים. נתחיל בשקילת עצמים בעלי משקל ידוע ונחזור על תהליך הכיול אם הערכים אינם מדויקים.

נרשום את התוכנית הבאה ל- Arduino IDE. לפני שנעלה אותה ל Esp32 לא נשכח להכניס את מקדם הכיול בשורה המודגשת בצבע אדום בתוכנית הבאה. התוכנית הבאה היא הדוגמה שנמצאת בספרייה המדגימה את השימוש ברוב הפונקציות שלה. בדוגמה הזו הוספתי את השורות הצבועות בירוק. כמו כן יש להוסיף ספרייה בשם soc בספריות libraries של הארדואינו ולהוסיף לו את קובץ הכותרת rtc.h שניתן להוריד מהאתר של GitHub:

/\*

\* Complete project details at <https://RandomNerdTutorials.com/esp32-load-cell-hx711/>

\*/

```
* HX711 library for Arduino - example file
* https://github.com/bogde/HX711
*
* MIT License
* (c) 2018 Bogdan Necula
*
**/

#include <Arduino.h>
#include "HX711.h"
#include "soc/rtc.h"

// HX711 circuit wiring
const int LOADCELL_DOUT_PIN = 16;
const int LOADCELL_SCK_PIN = 4;

HX711 scale;

void setup() {
  Serial.begin(115200);

  rtc_cpu_freq_config_t config;
  rtc_clk_cpu_freq_get_config(&config);
  rtc_clk_cpu_freq_to_config(RTC_CPU_FREQ_80M, &config);
  rtc_clk_cpu_freq_set_config_fast(&config);

  Serial.println("HX711 Demo");

  Serial.println("Initializing the scale");

  scale.begin(LOADCELL_DOUT_PIN, LOADCELL_SCK_PIN);

  Serial.println("Before setting up the scale:");
  Serial.print("read: \t\t");
  Serial.println(scale.read());    // print a raw reading from the ADC
```

```
Serial.print("read average: \t\t");
Serial.println(scale.read_average(20)); // print the average of 20 readings from the ADC

Serial.print("get value: \t\t");
Serial.println(scale.get_value(5)); // print the average of 5 readings from the ADC minus the tare weight
(not set yet)

Serial.print("get units: \t\t");
Serial.println(scale.get_units(5), 1); // print the average of 5 readings from the ADC minus tare weight
// (not set) divided by the SCALE parameter (not set yet)

//scale.set_scale(INSERT YOUR CALIBRATION FACTOR); להכניס את "גורם הכיול"
scale.set_scale(-471.497); // this value is obtained by calibrating the scale with known weights; see the
README for details

scale.tare(); // reset the scale to 0

Serial.println("After setting up the scale:");

Serial.print("read: \t\t");
Serial.println(scale.read()); // print a raw reading from the ADC

Serial.print("read average: \t\t");
Serial.println(scale.read_average(20)); // print the average of 20 readings from the ADC

Serial.print("get value: \t\t");
Serial.println(scale.get_value(5)); // print the average of 5 readings from the ADC minus the tare weight,
set with tare()

Serial.print("get units: \t\t");
Serial.println(scale.get_units(5), 1); // print the average of 5 readings from the ADC minus tare
weight, divided
// by the SCALE parameter set with set_scale

Serial.println("Readings:");
```



```

}

void loop() {
  Serial.print("one reading:\t");
  Serial.print(scale.get_units(), 1);
  Serial.print("\t| average:\t");
  Serial.println(scale.get_units(10), 5);

  scale.power_down();          // put the ADC in sleep mode
  delay(5000);
  scale.power_up();
}

```

## 12. הסבר התוכנית

נתחיל בהכללת הספריות הרצויות:

```

#include "HX711.h"
#include "soc/rtc.h"

```

**הערה:** בכמה מקומות נאמר שצריך להאט את המעבד ESP32 בגלל מהירות עבודה עם HX711. לא בטוח אם זה באמת נחוץ או לא. התנסינו עם ובלי האטה והכל עבד בסדר בשני התרחישים. עם זאת, הוספנו אפשרות זו לקוד. ניתן תמיד יכול להסיר אותו. כדי להאט את מהירות העבודה עלינו לכלול את קובץ הספרייה soc/rtc.h אחרת נקבל הודעת שגיאה המופיעה באיור הבא:

```

18
19 HX711 scale;
20
21 void setup() {
22   Serial.begin(115200);
23   rtc_clk_cpu_freq_set(RTC_CPU_FREQ_80M);
24   Serial.println("HX711 Demo");
25
26   Serial.println("Initializing the scale");
27
28   scale.begin(LOADCELL_DOUT_PIN, LOADCELL_SCK_PIN);
29
30   Serial.println("Before setting up the scale.");

```

rtc\_clk\_cpu\_freq\_set' was not declared in this scope  
C:\Program Files (x86)\arduino-1.8.15\libraries\HOLD\_LiquidCrystal: no headers file found  
C:\Program Files (x86)\arduino-1.8.15\libraries\smileytongue24: no headers files found  
C:\Program Files (x86)\arduino-1.8.15\libraries\smileytongue24.h: no headers file found

איור 19: הודעת השגיאה שתופיע אם רוצים להאט את מהירות העבודה עם ה-ESP32 ולא הכללנו את הקובץ: soc/rtc.h. השורות הבאות מגדירות את ה-GPIO שבהם נשתמש כדי להתחבר למגבר HX711. בחרנו GPIOs 16 ו-4. ניתן להשתמש בכל GPIO מתאים אחר.

```

const int LOADCELL_DOUT_PIN = 16;
const int LOADCELL_SCK_PIN = 4;

```

לאחר מכן ניצור אובייקט של ספריית HX711 שנקרא scale שבו נשתמש מאוחר יותר כדי לקבל את המדידות.

HX711 scale;

### פונקציית ה `setup()`

בפונקציה זו מאתחלים את המוניטור הטורי לקצב של 115200 ביטים בשנייה.

Serial.begin(115200);

מאטים את פעולת המיקרו בקר ( הסברנו את נושא המהירות בתחילת הפרק הזה ).

`rtc_cpu_freq_config_t config;`

`rtc_clk_cpu_freq_get_config(&config);`

`rtc_clk_cpu_freq_to_config(RTC_CPU_FREQ_80M, &config);`

`rtc_clk_cpu_freq_set_config_fast(&config);`

נאתחל את תא העומס על-ידי קריאה למתודה `begin()` של האובייקט scale והעברת הדקי ה GPIO כארגומנטים.

`scale.begin(LOADCELL_DOUT_PIN, LOADCELL_SCK_PIN);`

לאחר מכן, נקרא למספר מתודות שבהן נשתמש כדי לקבל קריאות באמצעות הספרייה . נקרא למתודות תוך כדי הדפסת

התוצאה. המתודות הן :

`scale.read();` // קורא את המשקל הגולמי

`scale.read_average(20);` // קורא ממוצע של 20 דגימות

`scale.get_value(5);` // מקבל את הממוצע של מספר הקריאות האחרון שהוגדר פחות משקל הטרה

`scale.get_units((5), 1);`

מקבל את הממוצע של מספר הקריאות האחרון שהוגדר פחות משקל ה-TARE ( טרה ) חלקי גורם הכיול - פעולה זו תפיק קריאה ביחידות הרצויות.

`scale.set_scale(-471.497);`

רושמים את calibration Factor שחישבנו בתוכנית הכיול. אצלנו הוא : -471.497 .

לאחר מכן, נקרא למתודה `tare()` כדי למדוד את המשקל "טרה" .

`scale.tare();`

לאחר הגדרה זו, הסולם צריך להיות מוכן לקבל קריאות מדויקות ביחידות הרצויות. הדוגמה קוראת לאותן שיטות קודמות , כך

שנוכל לראות את ההבדל לפני ואחרי הגדרת קנה המידה. נשתמש באותן מתודות שהשתמשנו קודם של `scale.read()` ,

`scale.read_average(20);` , `scale.get_value(5);` , `scale.get_units((5), 1);` שאותן כבר הסברנו.

### פונקציית ה `loop()`

בלולאה `loop()` קוראים למתודה `get_units()` בשתי דרכים שונות: לקבל קריאה אחת בודדת (ללא פרמטרים)

ולקבל את הממוצע של 10 הקריאות האחרונות. בכל קריאה כזו מדפיסים למוניטור הטורי את התוצאות.

`scale.get_units(), 1 ;`

`scale.get_units(10), 5;`

הוא מכבה את ה ADC באמצעות המתודה power\_down() . לאחר מכן, מבצעים השהייה של 5 שניות ומפעילים שוב את ה ADC עם המתודה power\_up() והלולאה חוזרת על עצמה.

```
scale.power_down(); // put the ADC in sleep mode
delay(5000);
scale.power_up();
```

מכאן שנקבל קריאות חדשות בצג הטורי כל 5 שניות.

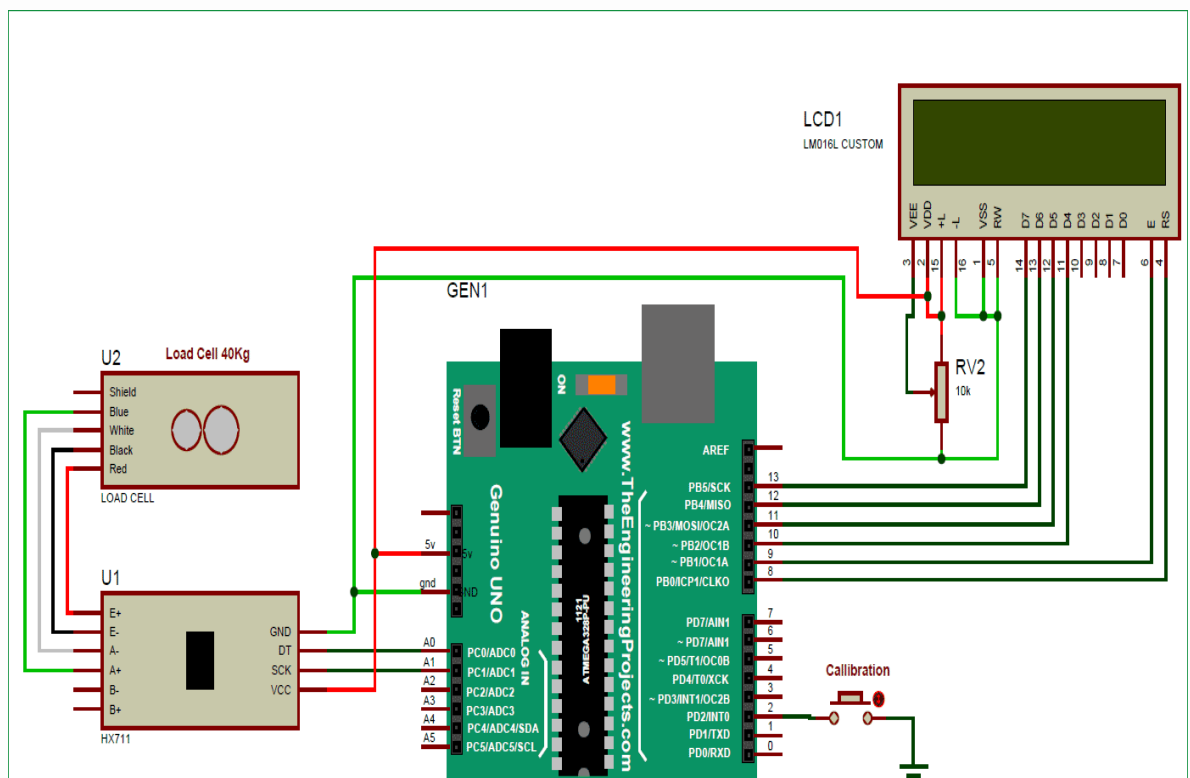
### 13. חיבור חיישן משקל אל ארדואינו

בפרק זה נחבר חיישן משקל אל ארדואינו ונציג את המשקל בתצוגת גביש נוזלי LCD של 2 שורות. בפרק זה נעזרנו בקישור :

[Arduino Weight Measurement Project with Load Cell and HX711 Module Interfacing: Circuit Diagram and Code](http://circuitdigest.com/Arduino-Weight-Measurement-Project-with-Load-Cell-and-HX711-Module-Interfacing-Circuit-Diagram-and-Code)  
([circuitdigest.com](http://circuitdigest.com/))

### 13.א מעגל החומרה

המעגל נראה באיור הבא :



איור 20 : הסכמה של המעגל החשמלי

באיור מופיע כרטיס Genuino Uno שהוא דומה ל Arduino Uno . בצד שמאל למעלה באיור יש את תא העומס , מתחתיו יש את המגבר HX711 שמתחבר להדקים האנלוגיים A0 ו A1 של הארדואינו. בצד ימין למעלה יש את תצוגת ה LCD .

הדקי ה LCD הנקראים RS, EN, d4, d5, d6 ו-d7 מחוברים באמצעות מספרי הפינים 8, 9, 10, 11, 12 ו-13 של Arduino בהתאמה. פני DT ו-SCK של מודול HX711 מחוברים ישירות לפינים A0 ו-A1 של Arduino. את החיבורים של תאי העומס עם מודול HX711 כבר הסברנו קודם לכן והם גם מוצגים גם באיור. ראשית, עלינו לכייל את המערכת הזו למדידת המשקל הנכון. כאשר המשתמש יפעיל אותו, המערכת תתחיל כיול באופן אוטומטי. אם המשתמש רוצה לכייל אותו ידנית נלחץ על המפסק. קיימת פונקציה לכיול ריק הנקראת ( ) calibrate . לצורך כיול, יש להמתין שבתצוגת ה LCD תופיע ההודעה לשים 100 גרם מעל תא העומס - "put 100g" . כעת נשים את המשקל 100g על תא העומס ונחכה. לאחר מספר שניות תהליך הכיול יסתיים. לאחר הכיול המשתמש יכול לשים כל משקל על תא העומס (יש לשים לב למשקל המקסימלי של תא העומס) ונקבל את הערך על גבי LCD בגרמים.

## 13.ב התוכנה

חלק התכנות של הפרויקט הזה הוא קצת מורכב למתכנתים מתחילים. בפרויקט זה לא נשתמש בשום ספרייה להתממשקות חיישן העומס HX711 עם Arduino. למרות שיש כמה ספריות קיימות למטרה זו, שבהן רק צריך לכלול את הספרייה ונקבל את המשקל באמצעות שורת קוד אחת. התוכנית נראית כך :

```
#include <LiquidCrystal.h>

LiquidCrystal lcd(8, 9, 10, 11, 12, 13);

#define DT A0
#define SCK A1
#define sw 2

long sample=0;
float val=0;
long count=0;

unsigned long readCount(void)
{
    unsigned long Count;
    unsigned char i;
    pinMode(DT, OUTPUT);
    digitalWrite(DT,HIGH);
    digitalWrite(SCK,LOW);
    Count=0;
```

```
pinMode(DT, INPUT);

while(digitalRead(DT));

for (i=0;i<24;i++)
{
    digitalWrite(SCK,HIGH);

    Count=Count<<1;

    digitalWrite(SCK,LOW);

    if(digitalRead(DT))

        Count++;

}

digitalWrite(SCK,HIGH);

Count=Count^0x800000;

digitalWrite(SCK,LOW);

return(Count);
}

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

    pinMode(SCK, OUTPUT);

    pinMode(sw, INPUT_PULLUP);

    lcd.begin(16, 2);

    lcd.print("    Weight ");

    lcd.setCursor(0,1);

    lcd.print(" Measurement ");

    delay(1000);

    lcd.clear();

    calibrate();
}

void loop()
```

```
{
    count= readCount();

    int w=((count-sample)/val)-2*((count-sample)/val));

    Serial.print("weight:");

    Serial.print((int)w);

    Serial.println("g");

    lcd.setCursor(0,0);

    lcd.print("Weight          ");

    lcd.setCursor(0,1);

    lcd.print(w);

    lcd.print("g          ");

    if(digitalRead(sw)==0)
    {
        val=0;

        sample=0;

        w=0;

        count=0;

        calibrate();
    }
}

void calibrate()
{
    lcd.clear();

    lcd.print("Calibrating...");

    lcd.setCursor(0,1);

    lcd.print("Please Wait...");

    for(int i=0;i<100;i++)
    {
        count=readCount();
```



```
    sample+=count;

    Serial.println(count);
}

sample/=100;

Serial.print("Avg:");

Serial.println(sample);

lcd.clear();

lcd.print("Put 100g & wait");

count=0;

while(count<1000)

{

    count=readCount();

    count=sample-count;

    Serial.println(count);

}

lcd.clear();

lcd.print("Please Wait...");

delay(2000);

for(int i=0;i<100;i++)

{

    count=readCount();

    val+=sample-count;

    Serial.println(sample-count);

}

val=val/100.0;

val=val/100.0;        // put here your calibrating weight

lcd.clear();

}
```

### הסבר התוכנית :

תחילה נכליל קובץ כותרת עבור ה LCD ונגדיר את הדקי הארדואינו אליהם התצוגה מתחברת וכמו כן נגדיר הדק בארדואינו לשם נחבר מפסק. לחיצה על המפסק תבצע פעולת כיול `calibrate` של החיישן על ידי פונקציה הנקראת `calibrate()`. כפי שהסברנו בתחילת הפרק הכיול דרוש כדי לקבל את גורם הכיול **calibration factor**. לשם כך ניקח משקל ידוע ( 100 גרם בדוגמה כאן) ונקבל את גורם הכיול. לאחר מכן נגדיר מספר משתנים לצורך חישוב.

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(8, 9, 10, 11, 12, 13);

#define DT A0
#define SCK A1
#define sw 2

long sample=0;
float val=0;
long count=0;
```

לאחר מכן, נגדיר את הפונקציה `readCount()` לקריאת נתונים ממודול HX711. הפונקציה מחזירה את תוצאות המדידה.

```
unsigned long readCount(void)
{
    unsigned long Count;
    unsigned char i;
    pinMode(DT, OUTPUT);
    digitalWrite(DT,HIGH);
    digitalWrite(SCK,LOW);
    Count=0;
    pinMode(DT, INPUT);
    while(digitalRead(DT));
    for (i=0;i<24;i++)
    {
        digitalWrite(SCK,HIGH);
        Count=Count<<1;
        digitalWrite(SCK,LOW);
```

```
    if(digitalRead(DT))
        Count++;
}
digitalWrite(SCK,HIGH);
Count=Count^0x800000;
digitalWrite(SCK,LOW);
return(Count);
}
```

בפונקציית ה ( ) setup נאתחל את ה LCD ונקבע אילו הדקים יהיו קלט ואילו יהיו פלט.

```
void setup()
{
    Serial.begin(9600);
    pinMode(SCK, OUTPUT);
    pinMode(sw, INPUT_PULLUP);
    lcd.begin(16, 2);
    lcd.print("    Weight ");
    lcd.setCursor(0,1);
    lcd.print(" Measurement ");
    delay(1000);
    lcd.clear();
    calibrate();
}
```

בפונקציה ה ( ) loop נקרא לפונקציה ה readCount() ואת הנתון שהיא מחזירה ( שהוא תוצאת המדידה של המשקל שנמצא במודול HX711 ) נכניס למשתנה ה count ונמיר נתונים אלה למשקל בגרמים ונשלח אותם לתצוגת ה LCD.

```
void loop()
{
    count= readCount();
    int w=(((count-sample)/val)-2*((count-sample)/val));
    Serial.print("weight:");
    Serial.print((int)w);
    Serial.println("g");
}
```

```
lcd.setCursor(0,0);  
lcd.print("Weight      ");  
lcd.setCursor(0,1);  
lcd.print(w);  
lcd.print("g      ");  
  
if(digitalRead(sw)==0)  
{  
    val=0;  
    sample=0;  
    w=0;  
    count=0;  
    calibrate();  
}  
}
```

לפני כן, יצרנו פונקציית כיול ( ) calibrate שבה כיילנו את המערכת על ידי הצבת משקל של 100 גרם על תא העומס.

```
void calibrate()  
{  
    lcd.clear();  
    lcd.print("Calibrating...");  
    lcd.setCursor(0,1);  
    lcd.print("Please Wait...");  
    for(int i=0;i<100;i++)  
    {  
        count=readCount();  
        sample+=count;  
        Serial.println(count);  
    }  
    .....  
    .....  
    while(digitalRead(DT));  
    for (i=0;i<24;i++)
```

```
{  
    digitalWrite(SCK,HIGH);  
    Count=Count<<1;  
    digitalWrite(SCK,LOW);  
    if(digitalRead(DT))  
        Count++;  
}
```

## 14. ביבליוגרפיה:

1. האתר של randomnerd :

<https://randomnerdtutorials.com/esp32-load-cell-hx711/>

2. האתר allaboutcircuits

[Strain Gauges | Electrical Instrumentation Signals | Electronics Textbook \(allaboutcircuits.com\)](https://www.allaboutcircuits.com/electronics-textbook/electrical-instrumentation/signals/)

3. האתר circuitdigest

<https://circuitdigest.com/microcontroller-projects/arduino-weight-measurement-using-load-cell>

4. האתר phidgets

[https://www.phidgets.com/docs/Load\\_Cell\\_Guide](https://www.phidgets.com/docs/Load_Cell_Guide)

5. האתר של Sparkfun

[https://learn.sparkfun.com/tutorials/load-cell-amplifier-hx711-breakout-hookup-guide?\\_ga=2.114518599.1721968113.1710578078-1558085665.1710578077](https://learn.sparkfun.com/tutorials/load-cell-amplifier-hx711-breakout-hookup-guide?_ga=2.114518599.1721968113.1710578078-1558085665.1710578077)

6. האתר של github .

[https://github.com/olkal/HX711\\_ADC](https://github.com/olkal/HX711_ADC)

7. קובץ הכותר HX711.h

<https://github.com/bogde/HX711/blob/master/src/HX711.h>