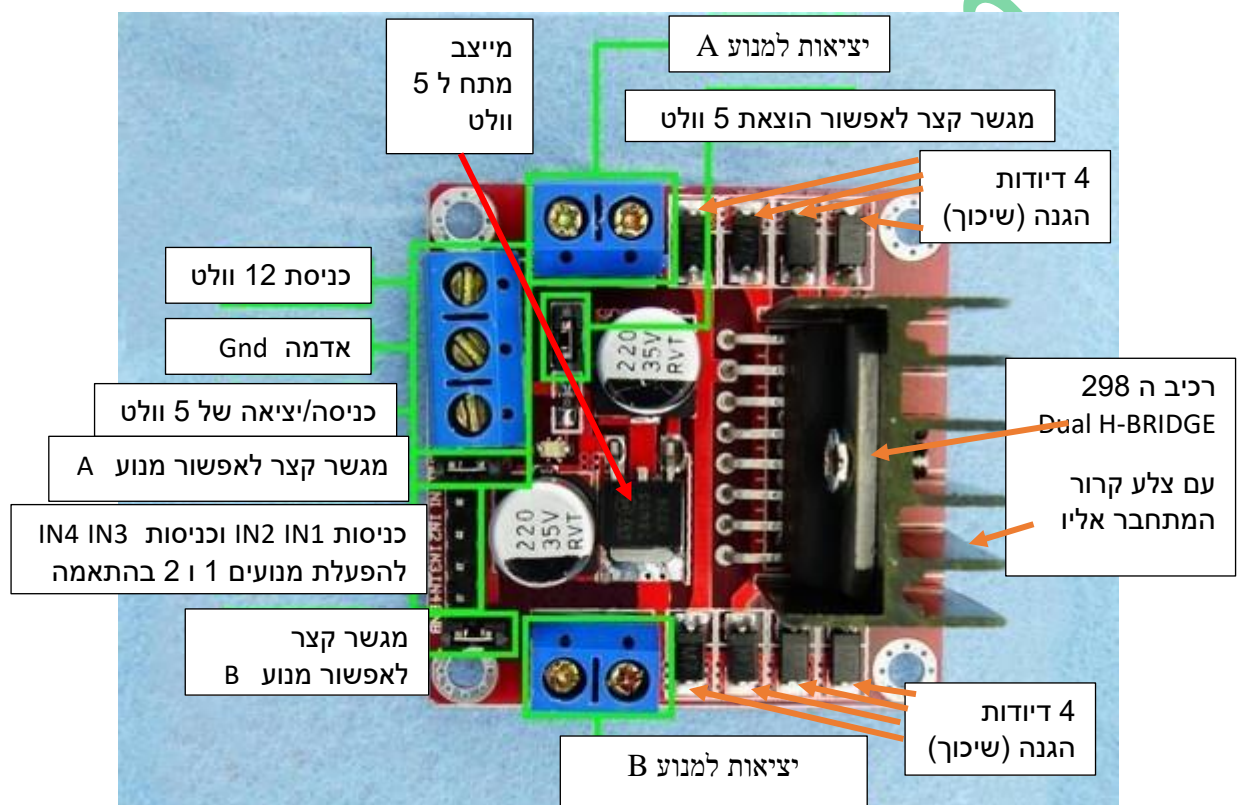


מודול דוחף זרם עם L298N

א. כללי

מודול דוחף הזרם הוא כרטיס אלקטרוניקה בגודל $43 \times 43 \times 27 \text{ mm}$ המתבסס על הרכיב L298N שהוא רכיב דוחף זרם לעומסים כמו מנועים, מנועי צעד, ממסרים סלילים וכו'. ברכיב 298 ישנם 2 ערוצים של מעגלים זהים הנקראים A ו B בהתאמה, עבור דחיפת זרם ל 2 עומסים שונים.

מעגל המודול נראה באיור 1 :



איור 1 : כרטיס דוחף הזרם 298 .

באיור 1 מתואר הכרטיס עם הרכיבים שבו. ההדקים לחיבור חוטים לכניסות / יציאות הם בצבע כחול.

הרכיב מקבל מתח של בין 5 עד 35 ויכול להזרים זרם למנועים של 3 אמפר peak (שיאי) או 2 אמפר מתמשך continuous. ההספק המקסימאלי שהוא יכול לתת הוא 25W .

ברכיב 298 שבכרטיס יש 2 גשרים הנקראים H-BRIDGE כמו כן יש בכרטיס מייצב מתח ל 5 וולט והדקים להתחבר אליו. את מתח ה 5 וולט ניתן להוציא ולחבר גם למעגלים מחוץ לכרטיס. אין לחבר מתח חיצוני של 5 וולט להדק ה 5 וולט של הכרטיס.

ב. מאפיינים

בטבלה הבאה מתוארים מאפייני הכרטיס.

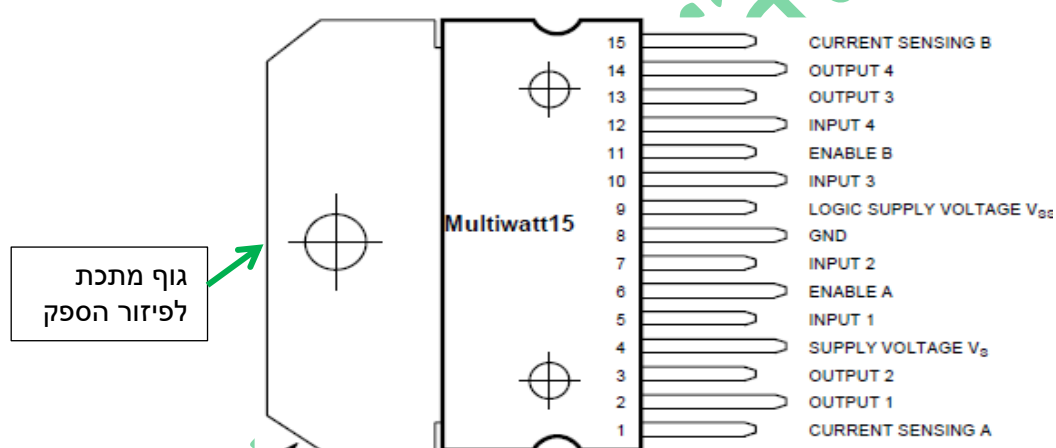
Logic voltage – רמות לוגיות	5V
Drive voltage – מתח ההפעלה לכרטיס	5V-35V
Logic Current – זרם המעגל הלוגי	0mA-36mA
Drive current – הזרם לעומס	2A (MAX each bridge)
Maximum Power – תצרוכת הספק מקסימאלית	25W
Dimensions – מידות	43 * 43 * 27 mm

טבלה 1 : מאפייני הכרטיס

ג. מבנה פנימי של הרכיב 298

לרכיב 298 יש 15 הדקים המסודרים ב 2 שורות כפי שנראה באיור 2 :

איור 2 מתאר את צורתו החיצונית של רכיב 298 .



איור 2 : רכיב 298 – מראה חיצוני והדקים.

באיור 2 רואים שחלק מההדקים הם עם רגליים ארוכות וחלק עם קצרות יותר. הרגליים הארוכות נמצאות בשורה אחת והקצרות בשורה אחרת.

ברכיב ה 298 יש לכל אחד מהערוצים A ו B שני חלקים .

1. מעגל לוגי המקבל כניסות ברמות לוגיות של 0 ו 1 ומוציא פקודות רוויה או קטעון למעגל האנלוגי שבסעיף 2 ועל ידי כך קובע האם המנוע מסתובב ואם כן לאיזה כיוון.

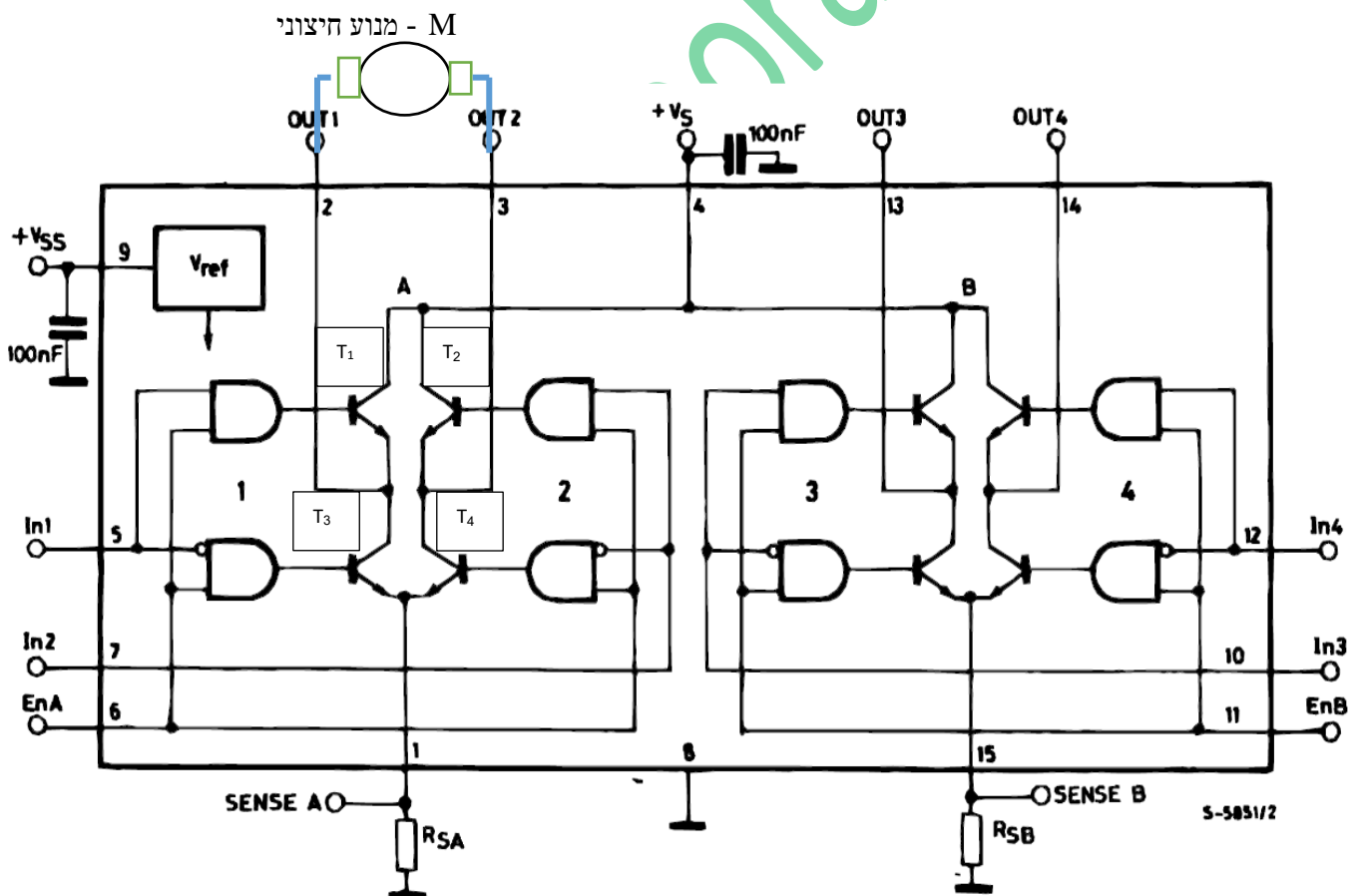
2. חלק אנלוגי שהוא גשר H שמורכב מ 4 טרנזיסטורים המקבלים פקודות רוויה או קטעון ודרכם זורם הזרם למנוע.

לכל אחד מ 2 הערוצים יש 3 כניסות ו 2 יציאות. נתאר את ערוץ A. ערוץ B פועל בצורה זהה. הכניסות הן: EnA – (אפשר - Enable), $IN1$ ו $IN2$. הכניסה EN קובעת האם לאפשר לכניסות $IN1$ ו $IN2$ להיות פעילות ועל ידי כך להפעיל את המנוע או לא לאפשר את הכניסות ואז לא ניתן לשלוט על גשר ה H והמנוע לא יעבוד. היציאות הן $OUT1$ ו $OUT2$ שביניהן מתחבר המנוע.

בעזרת הכניסות $Enable$ ניתן לעבוד עם **PWM – Pulse Width Modulator** ולשלוט על מהירות הסיבוב של המנוע. באיור 3 מתואר המבנה הפנימי של הרכיב 298. הוספנו מנוע ביציאות $OUT1$ ו $OUT2$ כדי להראות איפה מתחבר המנוע. מתוך האיור ניתן לראות שכל מעגל מורכב מ 4 מעגלים לוגיים ו 4 טרנזיסטורים. נסביר את המבנה של הערוץ השמאלי הנקרא A. בערוץ זה חיברנו מנוע M ליציאות.

המתח V_s הוא מתח ההפעלה של המנועים. מתח זה יכול להיות בין 7 ל 35 וולט. הקבל של 100nF הוא קבל סינון שמומלץ להוסיף אותו והוא לא נמצא בתוך הרכיב.

המתח V_{ss} הוא מתח נוסף (ניתן לחבר אותו למתח V_{ss}). ממתח זה יוצרים את מתח ה V_{ref} שמגיע להפעלת החלק הלוגי ברכיב. גם כאן מומלץ לשים קבל חיצוני של 100nF.



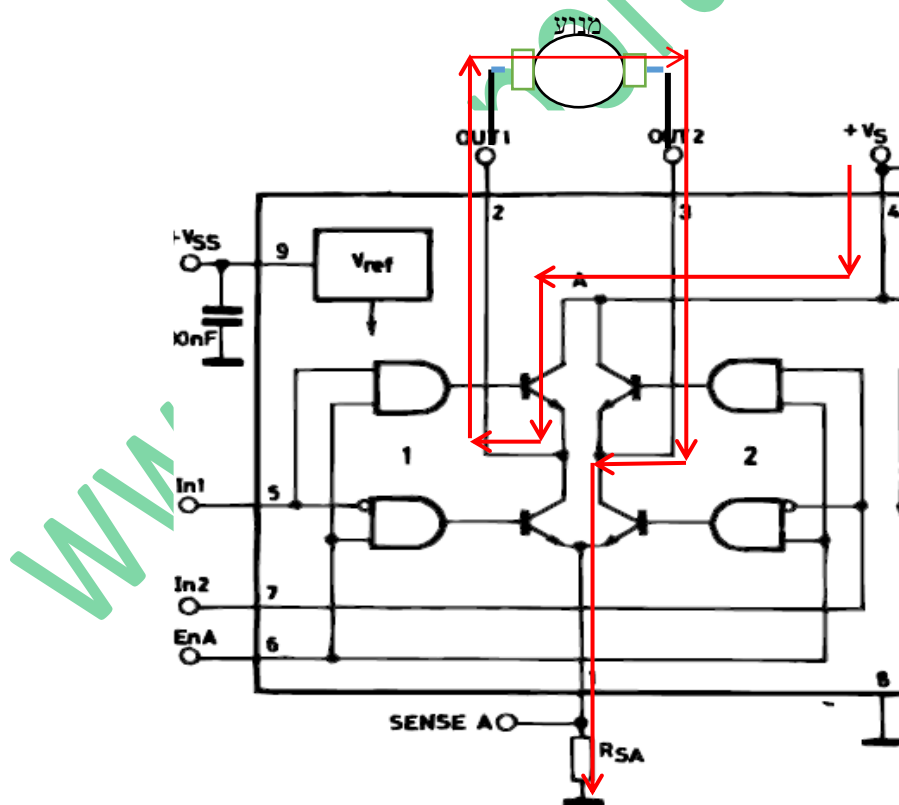
איור 3 : המבנה הפנימי של הרכיב 298

נתאר בטבלה הבאה את אופן הפעולה של המעגל : (x הוא מצב Don't care – לא משנה)

	EnA	In2	In1	T1	T2	T3	T4	מצב המנוע
1	0	X	X	קטעון	קטעון	קטעון	קטעון	עומד
2	1	0	0	קטעון	קטעון	רוויה	רוויה	עומד
3	1	0	1	רוויה	קטעון	קטעון	רוויה	מסתובב בכיוון אחד
4	1	1	0	קטעון	רוויה	רוויה	קטעון	מסתובב בכיוון השני
5	1	1	1	רוויה	רוויה	קטעון	קטעון	עומד

טבלה 2 : אופן פעולת המעגל

נסביר באיור 4 את המצב 3 שבטבלה בעזרת שרטוט כיוון הזרם במעגל (בצבע אדום): יש לזכור שהטרנזיסטורים שברוויה הם T1 ו T4 . הזרם זורם מ V_s אל האדמה .



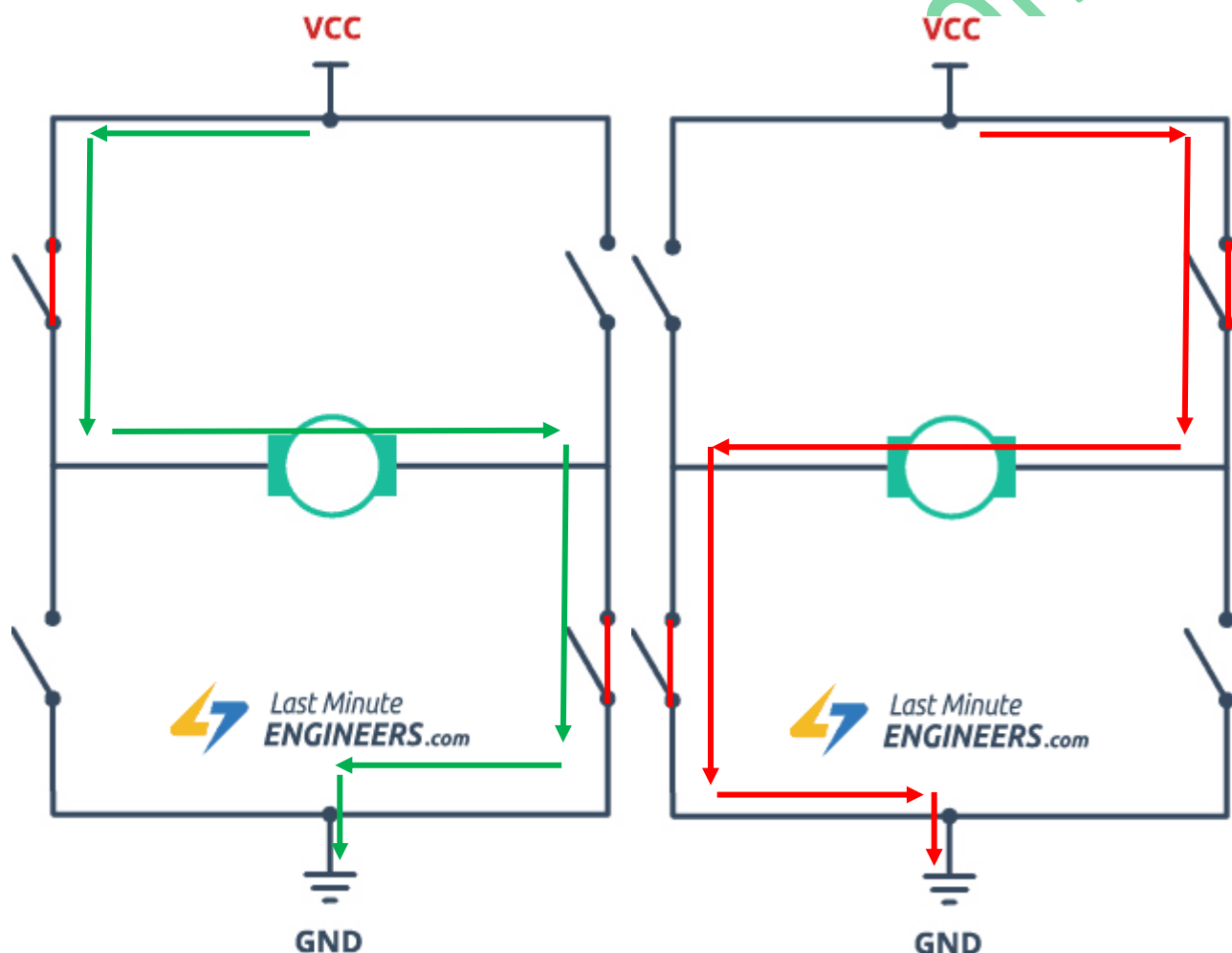
איור 4 : כיוון הזרם במצב 3 של טבלה 1

הזרם דרך המנוע זורם משמאל לימין ולכן המנוע מסתובב בכיוון כלשהו .

במצב 4 שבטבלה הטרנזיסטורים T2 ו T4 ברוויה ודרכם זורם הזרם. במצב זה הזרם יזרום משמאל לימין במנוע ואז כיוון הסיבוב שלו יהיה הפוך למצב הקודם.

הנגד Rsa שבתחתית הוא נגד גילוי או חישה – SENSE - שניתן להוסיף (לא חייבים) ואז ניתן להכניס את המתח עליו למעגל הממיר ממתח אנלוגי לדיגיטלי כדי לדעת מה מהירות המנוע והאם להגדיל אותה או להקטינה. זהו נגד קטן של מספר אוהמים בודדים ומשמש בעיקר כאשר עובדים עם PWM ורוצים לשלוט על מהירות המנוע. אם לא רוצים לשלוט על מהירות המנוע ניתן לא לחבר אותו ואז לחבר את הדק ה SENSE ישירות לאדמה. בכרטיס הרגליים האלה מתחברות לאדמה ולא אל נגדים.

האיור הבא מתאר את פעולת המנוע . הטרנזיסטורים שברוויה מיוצגים על ידי מפסק סגור והטרנזיסטורים שבקטעון מיוצגים על ידי מפסק פתוח.



איור 5 : סיבוב המנוע ומצב המפסקים בכל אחד מכיווני הסיבוב.

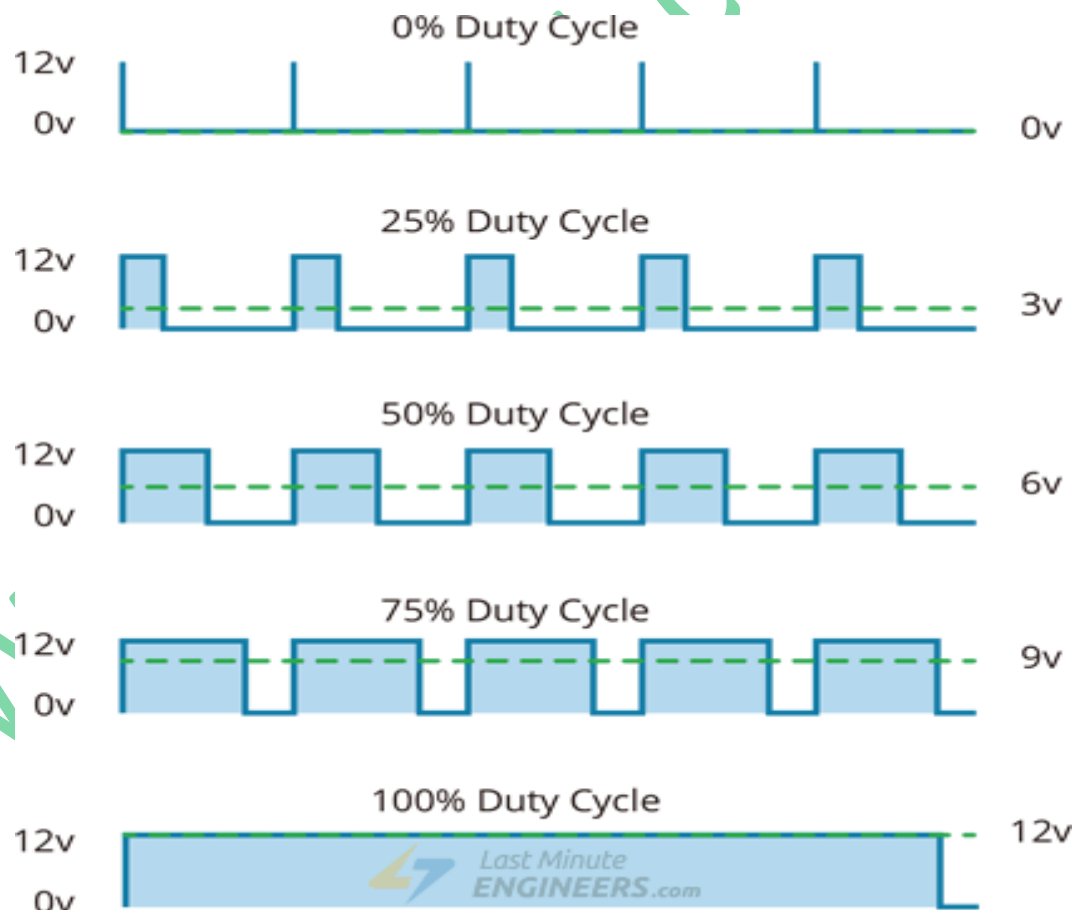
ג. הסבר על PWM – Pulse Width Modulator - אפנון רוחב דופק.

ניתן לשלוט במהירות של מנוע DC על ידי שינוי מתח הכניסה שלו. טכניקה נפוצה להשגת מטרה זו היא אפנון רוחב פולס - Pulse Width Modulation PWM. זוהי טכניקה שבה הערך הממוצע של מתח הכניסה מותאם על ידי שליחת סדרה של פולסים OFF-ON. מתח ממוצע זה פרופורציונלי לרוחב הפולסים, המכונה מחזור עבודה/מנת פעולה - Duty Cycle. זהו היחס בין הזמן של ה ON לחלק בזמן ON+OFF.

$$\text{Duty Cycle} = T_{on} / (T_{on} + T_{off})$$

ככל שמחזור העבודה גבוה יותר, כך המתח הממוצע המופעל על מנוע DC גבוה יותר, וכתוצאה מכך נקבל עלייה במהירות המנוע. ככל שמחזור העבודה קצר יותר, כך המתח הממוצע המופעל על מנוע DC נמוך יותר, וכתוצאה מכך ירידה במהירות המנוע.

האיור הבא מתאר את ה Duty Cycle במספר מצבים.



איור 6 : דוגמאות ל Duty Cycles ב PWM.

בהנחה שהמנוע מקבל את צורת הגל הראשונה $\text{Duty Cycle} = 0\%$ הוא לא יסתובב כלל. ככל שאחוז ה Duty Cycle גדל ממוצע ה DC גדול יותר והמנוע יסתובב במהירות גבוהה יותר. אם תדר הגל המרובע הוא הרצים בודדים נוכל להבחין שהמנוע מסתובב ועוצר לסירוגין. אבל אם תדר הגל המרובע הוא כמה

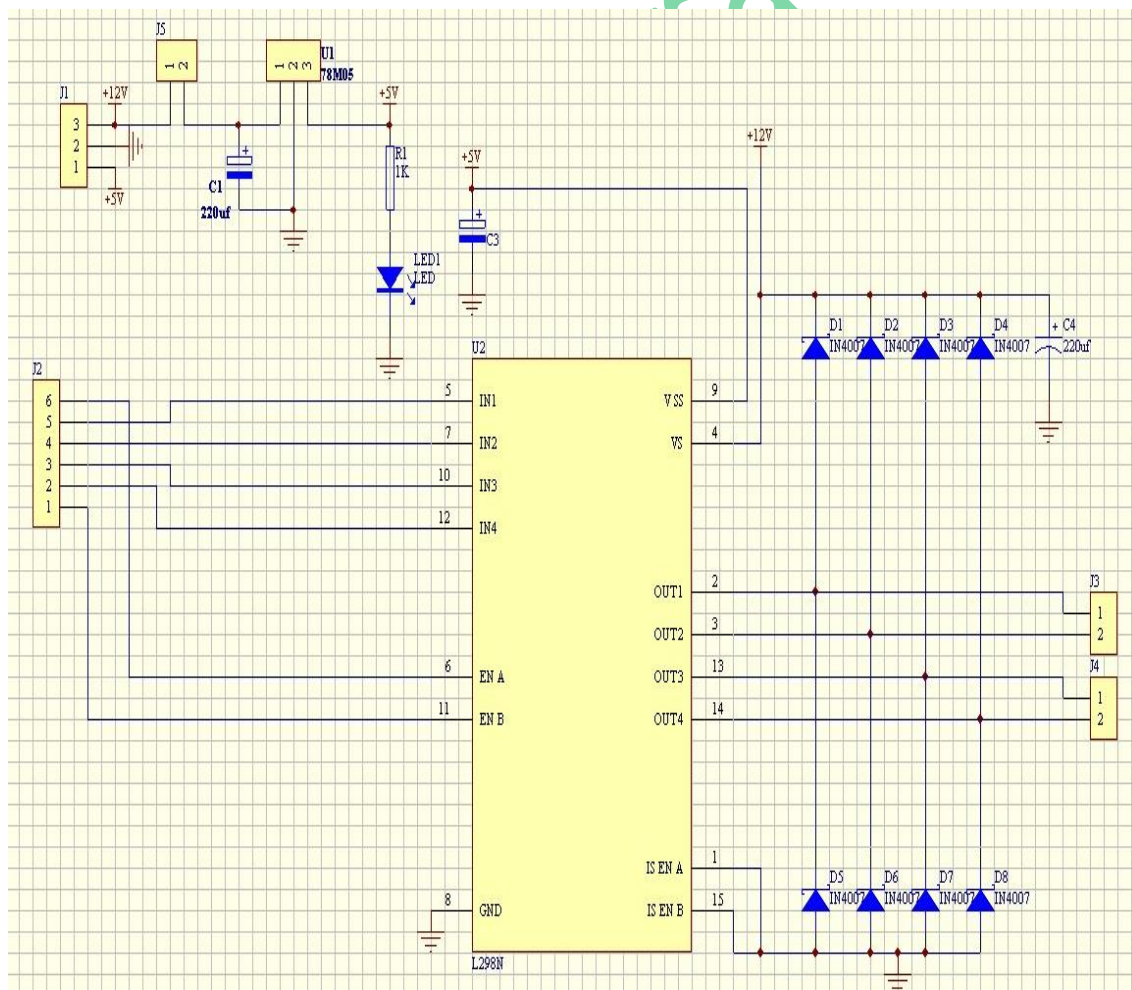
מאות הרץ המנוע ינוע ברציפות ולא נקבל עצירות. בכרטיסי הארדואינו תדר הגל המרובע הוא כ 490 הרץ ויש הדקים שבהם התדר הוא 980 הרץ. תלוי בכרטיס הארדואינו ותלוי בהדק שבו נשתמש. התדר ניתן לקביעה על ידי המשתמש.

על ידי מתן צורות גלים כאלה ניתן לשלוט על מהירות המנוע. בדרך כלל נותנים למנוע להסתובב בכיוון כלשהו על ידי פקודות מתאימות ב Vin1 ו Vin2 (או Vin3 ו Vin4 במנוע השני) ובעזרת הדק ה EnA (או EnB) המקבלת את אחת מצורות הגל שבאיור 5 קובעים את המהירות בכיוון הזה.

הפקודה בתוכנת הארדואינו המוציאה גל כמו צורות הגלים שנתנו היא : `analogWrite(EnA, 255);` מוציאה גל כמו בבצורת הגל הראשונה. הפקודה `analogWrite(EnA, 128);` מוציאה גל כמו בבצורת הגל השנייה. הפקודה `analogWrite(EnA, 64);` מוציאה צורת גל הדומה לצורת הגל השלישית.

ד. סכמה חשמלית של הכרטיס

באיור הבא נראית הסכימה החשמלית של כל המודול.



איור 7 : מודול כרטיס 298

במרכז השרטוט רואים את רכיב ה-298. מצד שמאל שלו הוא מקבל 6 כניסות ממחבר J2. 2 הכניסות 1 ו-2 ב J2 הן של האפשר EN A ו- EN B (ומתחברות אל רגליים 6 ו-11 של הרכיב) ו-4 הכניסות הבאות הן הכניסות IN מ-1 עד 4 (הדקים 5,7 ו-10,12 של הרכיב).

מצד ימין הרכיב יש את ההדק Vss המקבל מתח ההזנה של 5 וולט לחלק הלוגי. המתח של ה-5 וולט יכול להגיע או ממיצב בכרטיס או ממקור מתח חיצוני. בצד שמאל בשרטוט יש 3 את J1 שלו 3 הדקי כניסה. בעליון רשום +12V. מתח זה מגיע גם לרגל Vs של הרכיב (מוסבר בסעיף הבא) וגם למגשר קצר הנקרא בשרטוט J5. אם אין את מגשר הקצר בין ההדקים 1 ו-2 של J5 אז לא מגיע מתח למייצב U1 שהוא מייצב 78M05 ולכן הוא לא מספק מתח של 5 וולט ויש לחבר להדק 3 של J1 מתח של 5 וולט ממקור חיצוני. אם מגשר הקצר קיים אז המייצב צריך לקבל מתח שבין 7 וולט ל-35 וולט ומוציא 5 וולט להדק Vss של רכיב ה-298 וגם לרגל 3 של J1. במקרה כזה אסור לחבר מתח חיצוני של 5 וולט לרגל 3 של J1 כדי שלא תהיה התנגשות בין 2 מתחי 5 וולט. הלב המתחברת ביציאת המייצב מראה שיש מתח של 5 וולט בכרטיס (או מהמייצב או מ-5 וולט חיצוני שהכנסנו).

אזהרה: הכרטיס מגיע עם מגשר הקצר J5 ולכן אין לחבר מתח חיצוני נוסף של 5 וולט בהדק 1 של J1. ניתן להוציא את מתח ה-5 וולט של המייצב שבכרטיס אל מעגלים מחוץ לכרטיס.

תחתיו יש את ההדק Vs המקבל מתח ההזנה של 12 וולט לחלק האנלוגי להזרמת הזרם בטרנזיסטורים (ניתן להכניס מתח עד 35 וולט).

4 רגלים הבאות של הרכיב הן היציאות אל המנוע (2 לכל מנוע). כל 2 יציאות מתחברות בהתאמה אל ההדקים J3 ו- J4. לכל קו יציאה יש דיודה אחת המתחברת אל ה-12V ודיודה אחת המתחברת אל האדמה. הדיודות נקראות 1N4007. תפקיד הדיודות להגן על הטרנזיסטורים של גשר ה-H. הדיודות נקראות דיודות שיכוך - Damping Diodes. היות ובכל יציאה כזו מתחבר עומס השראי (סליל של מנוע) אז בהפסקה הזרם דרך הסליל נוצר בו כ"מ (כוח אלקטרו מניע) מושרה שיכול להגיע למתח של עשרות וולט. הכיוון של הכ"מ הוא בכיוון מנוגד לסיבה שיצרה אותו (חוק לנץ). הדיודה המתחברת אל ה-12 וולט דואגת שהמתח ביציאה לא יוכל לעלות מעל 12 וולט (ליתר דיוק 12 וולט ועוד כ-0.6 וולט שהוא מתח ההולכה של הדיודה, כלומר כ-12.6 וולט). הדיודה המתחברת לאדמה גורמת לכך שלא יוכל להתפתח מתח שלילי יותר מזה של האדמה (ליתר דיוק מינוס 0.6 וולט שהוא מתח ההולכה של הדיודה).

הרגלים 1 ו-15 של רכיב ה-298 (בצד ימין למטה של הרכיב) הן רגלי הגילוי – חישה – וניתן היה לחבר כאן נגד כדי לגלות מהו הזרם הזורם דרך העומס. בכרטיס לא נתנו את האפשרות למדידת זרם והן מחוברות לאדמה.

ה. תכנית לארדואינו להפעלת מנוע DC קטן בעזרת הכרטיס :

בתוכנית נפעיל מנוע DC המתחבר אל ההדקים OUT1 ו-OUT2 של כרטיס דוחף הזרם. בנוסף לחיבור המנוע נחבר את ההדקים 9 ו-8 של כרטיס הארדואינו אל IN1 ו- IN2 של כרטיס דוחף הזרם ואת הדק ENA נחבר אל הדק 10 שיופעל ב-PWM וישלוט על המהירות.

בחיבור מתחי ספק יש 2 אפשרויות :

1. לחבר מתח של כ 12 וולט ואדמה גם בכרטיס הארדואינו לרגל Vin (בארדואינו נאנו לדוגמה) וגם להדק 12V וולט של כרטיס דוחף הזרם. כמובן שיש לשתף את האדמות של הארדואינו ושל כרטיס דוחף הזרם.

2. לחבר 12 וולט ואדמה רק בכרטיס דוחף הזרם ומהדק 5 וולט ואדמה בכרטיס דוחף הזרם לחבר אל ההדקים 5 וולט ואדמה (בהתאמה) בכרטיס הארדואינו.

```
int enA = 10;
```

```
int in1 = 9;
```

```
int in2 = 8;
```

```
void setup()
```

```
{
```

```
// קביעת כל ההדקים כפלט
```

```
pinMode(enA, OUTPUT);
```

```
pinMode(in1, OUTPUT);
```

```
pinMode(in2, OUTPUT);
```

```
}
```

```
// תכנית דוגמה 1 : הרצת המנוע במהירות קבועה 2 שניות לכל כיוון ועצירת המנוע
```

```
void dugma1()
```

```
{
```

```
digitalWrite(in1, HIGH);
```

```
digitalWrite(in2, LOW);
```

```
// קביעת המהירות ל 200 מתוך 255 אפשרויות
```

```
analogWrite(enA, 200);
```

```
delay(2000);
```

```
// שינוי כיוון הסיבוב
```

```
digitalWrite(in1, LOW);
```

```
digitalWrite(in2, HIGH);

delay(2000);

// עצירת המנוע

digitalWrite(in1, LOW);

digitalWrite(in2, LOW);

}

void dugma2()

{

// הרצת המנוע בין כל האפשרויות שלו

// הפעלת המנוע

digitalWrite(in1, LOW);

digitalWrite(in2, HIGH);

// האצה מ 0 ועד 255

for (int i = 0; i < 256; i++)

{

analogWrite(enA, i);

delay(20);

}

// האטה מ 255 עד ל 0

for (int i = 255; i >= 0; --i)

{

analogWrite(enA, i);

delay(20);

}
```

// כיבוי המנוע

```
digitalWrite(in1, LOW);
```

```
digitalWrite(in2, LOW);
```

```
}
```

```
void loop()
```

```
{
```

```
  digitalWrite1();
```

```
  delay(1000);
```

```
  digitalWrite2();
```

```
  delay(1000);
```

```
}
```

1. דוגמה נוספת עם מיקרו בקר ESP32 .

1.1 הרכיבים שנשתמש בהם :

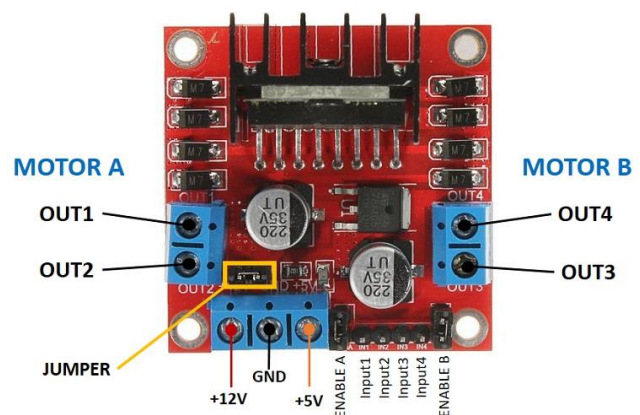
הבא:



1. כרטיס המיקרו בקר ESP32 נראה באיור

איור 8 : כרטיס המיקרו בקר ESP32 .

2. מודול דוחף הזרם נראה באיור הבא:



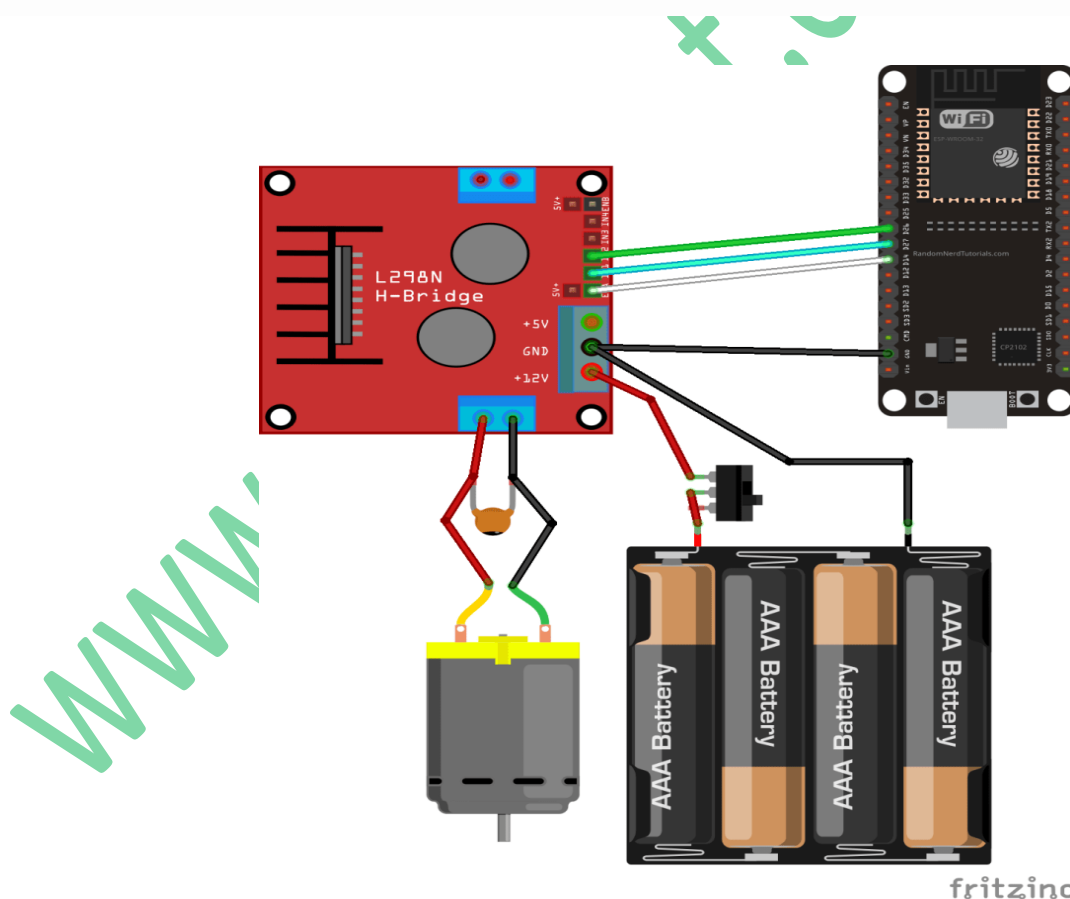
איור 9 : מודול דוחף הזרם L298 עם הסבר תפקיד הרכיבים שעליו וההדקים שלו .

3. מנוע DC נראה באיור הבא :



איור 10 : מנועי DC . משמאל מנוע בודד. במרכז 4 מנועים עם גלגלים לרכב/רובוט. מימין חיבור הגלגל למנוע

2.1 המעגל החשמלי



איור 11 : שרטוט חשמלי של חיבור esp32 אל דוחף מנועים ואל מנוע בעזרת תוכנת fritzing .

מנוע DC דורש קפיצה גדולה בזרם כדי לנוע, ולכן יש להפעיל את המנועים באמצעות מקור כוח חיצוני מה- ESP32. בדוגמה כאן אנו משתמשים בסוללות AA4 אך אפשר להשתמש בכל ספק כוח מתאים אחר. ספק הכוח יכול להיות מ 6 וולט ועד 12 וולט.

המתג בין הסוללות לדוחף המנועים הוא אופציונלי אך הוא שימושי מאוד להפעלה או הפסקת המערכת. בדרך זו אינך צריך להתחבר כל הזמן ולאחר מכן לנתק את החוטים כדי לחסוך בחשמל. מומלץ להלחים קבל קרמי 0.1 uF להדקים החיוביים והשליליים של מנוע DC כפי שמוצג בתרשים כדי לסייע בהחלקת קפיצות מתח – סינון רעשים. הערה: המנועים פועלים גם ללא הקבל.

יש לוודא שבתוכנת הארדואינו קיימת ההרחבה עבור ESP32. ניתן להיעזר בקישור:

<https://www.arikporat.com/wp-content/uploads/2023/01/introduction-to-esp32.pdf>

3.1.1 התוכנית והסבר התוכנית

ואז נוכל לכתוב את הקוד הבא השולט במהירות ובכיוון של מנוע DC. קוד זה אינו שימושי בעולם האמיתי, זו רק דוגמה פשוטה כדי להבין טוב יותר כיצד לשלוט במהירות ובכיוון של מנוע DC עם ESP32. בסיום כתיבת התוכנית נוודא שבחרנו את הלוח הנכון ואת יציאת ה-COM ונעלה את הקוד ל-ESP32.

```

/*****
Rui Santos
Complete project details at https://randomnerdtutorials.com
*****/

// Motor A
int motor1Pin1 = 27;
int motor1Pin2 = 26;
int enable1Pin = 14;

// Setting PWM properties
const int freq = 30000;
const int pwmChannel = 0;
const int resolution = 8;
int dutyCycle = 200;

void setup() {
  // sets the pins as outputs:
  pinMode(motor1Pin1, OUTPUT);
  pinMode(motor1Pin2, OUTPUT);
  pinMode(enable1Pin, OUTPUT);

```

```
// configure LED PWM functionalitites
ledcSetup(pwmChannel, freq, resolution);

// attach the channel to the GPIO to be controlled
ledcAttachPin(enable1Pin, pwmChannel);

Serial.begin(115200);

// testing
Serial.print("Testing DC Motor...");
}

void loop() {
  // Move the DC motor forward at maximum speed
  Serial.println("Moving Forward");
  digitalWrite(motor1Pin1, LOW);
  digitalWrite(motor1Pin2, HIGH);
  delay(2000);

  // Stop the DC motor
  Serial.println("Motor stopped");
  digitalWrite(motor1Pin1, LOW);
  digitalWrite(motor1Pin2, LOW);
  delay(1000);

  // Move DC motor backwards at maximum speed
  Serial.println("Moving Backwards");
  digitalWrite(motor1Pin1, HIGH);
  digitalWrite(motor1Pin2, LOW);
  delay(2000);

  // Stop the DC motor
  Serial.println("Motor stopped");
  digitalWrite(motor1Pin1, LOW);
```

```
digitalWrite(motor1Pin2, LOW);
delay(1000);

// Move DC motor forward with increasing speed
digitalWrite(motor1Pin1, HIGH);
digitalWrite(motor1Pin2, LOW);
while (dutyCycle <= 255){
    ledcWrite(pwmChannel, dutyCycle);
    Serial.print("Forward with duty cycle: ");
    Serial.println(dutyCycle);
    dutyCycle = dutyCycle + 5;
    delay(500);
}
dutyCycle = 200;
}
```

ח.1 הסבר המשפטים בתחילת התוכנית

בתחילה נכריז על הדקי המנוע A . נגדיר את ה- GPIO שאליהם מחוברים הדקי המנוע. במקרה זה, כניסה 1 עבור מנוע A מחוברת ל- GPIO 27, כניסה 2 ל- GPIO 26 ו- pin Enable ל- GPIO 14.

```
// Motor A
int motor1Pin1 = 27;
int motor1Pin2 = 26;
int enable1Pin = 14;
```

ח.2 הגדרת מאפייני PWM לשליטה במהירות

ניתן לשלוט במהירות מנוע DC על ידי הכנסת אות PWM בהדק האפשרי של דוחף המנוע L298N. המהירות תהיה פרופורציונלית למחזור העבודה Duty Cycle. כדי להשתמש ב-PWM עם ESP32, נגדיר תחילה את מאפייני האות של PWM.

```
const int freq = 30000;
const int pwmChannel = 0;
const int resolution = 8;
int dutyCycle = 200;
```


במקרה זה, אנו מייצרים אות של 30000 הרץ בערוץ 0 ברזולוציה של 8 סיביות. אנו מתחילים עם מחזור עבודה של 200 (ניתן להגדיר ערך מחזור עבודה מ 0 עד 255). עבור התדר שבו אנו משתמשים (30000) אם נפעיל מחזורי עבודה קטנים מ - 200 המנוע לא יזוז וישמיע צליל זמזום מוזר. לכן קבענו מחזור עבודה של 200 בהתחלה.
הערה: מאפייני PWM שאנו מגדירים כאן הם רק דוגמה. המנוע עובד מצוין עם תדרים אחרים.

ח.3 פונקציית ה () setup

בפונקציה זו מגדירים את הדקי המנוע כיציאות :

```
pinMode(motor1Pin1, OUTPUT);
pinMode(motor1Pin2, OUTPUT);
pinMode(enable1Pin, OUTPUT);
```

עלינו להגדיר את PWM עם המאפיינים שהגדרנו קודם באמצעות הפונקציה () ledcSetup המקבלת כארגומנטים, את pwmChannel (ערוץ ה PWM) , את התדירות ואת הרזולוציה, באופן הבא:

```
ledcSetup(pwmChannel, freq, resolution);
```

לאחר מכן, עיש לבחור את ה- GPIO שממנו נקבל את האות. לשם כך, נשתמש בפונקציה () ledcAttachPin המקבלת כארגומנטים את ה- GPIO שבברצוננו לקבל את האות ואת הערוץ שמייצר את האות. בדוגמה כאן, נקבל את האות ב- GPIO enable1Pin, המתאים ל- GPIO 14. הערוץ שמייצר את האות הוא pwmChannel, המתאים לערוץ 0.

```
// attach the channel to the GPIO to be controlled
ledcAttachPin(enable1Pin, pwmChannel);
```

ח.4 הנעת המנוע קדימה

בפונקציה () loop נפעיל את המנוע קדימה, אחורה, ימינה ושמאלה ובמוניטור נדפיס את כיוון התנועה.

ח.4.א הזזת המנוע קדימה

כדי להזיז את המנוע קדימה נשים בכניסה הדק 1 LOW ובכניסת הדק 2 HIGH. המנוע יזוז קדימה למשך 2 שניות (2000 אלפיות השנייה).

```
// Move the DC motor forward at maximum speed
Serial.println("Moving Forward");
digitalWrite(motor1Pin1, LOW);
digitalWrite(motor1Pin2, HIGH);
delay(2000);
```

ח.4.ב עצירת המנוע

כדי לעצור את המנוע נשים ב 2 הכניסות 0 . נעצור את המנוע לשנייה אחת.

```
// Stop the DC motor
Serial.println("Motor stopped");
digitalWrite(motor1Pin1, LOW);
digitalWrite(motor1Pin2, LOW);
delay(1000);
```

ח.4.ג הנעת המנוע אחורה

כדי להזיז את המנוע אחורה נשים בכניסה הדק 1 HIGH ובכניסת הדק 2 LOW. המנוע יזוז אחורה למשך 2 שניות (2000 אלפיות השנייה).

```
// Move DC motor backwards at maximum speed
Serial.println("Moving Backwards");
digitalWrite(motor1Pin1, HIGH);
digitalWrite(motor1Pin2, LOW);
delay(2000);
```

ח.4.ד שוב עוצרים את המנוע למשך שנייה אחת.

ח.4.ה שליטה על מהירות המנוע

כדי לשלוט במהירות מנוע DC עלינו לשנות את מחזור העבודה Duty Cycle - של אות PWM. לשם כך נשתמש בפונקציה ledcWrite() המקבלת כארגומנטים את ערוץ PWM המייצר את האות (לא את GPIO הפלט) ואת מחזור העבודה, באופן הבא:

```
ledcWrite(pwmChannel, dutyCycle);
```

בדוגמה כאן יש לולאת while שמגדילה את מחזור העבודה ב 5 בכל לולאה. בכל פעם מגדילים את המהירות למשך חצי שנייה. כלומר ה duty cycle משתנה מ 200 ועד 255 בקפיצות של 5 (בסה"כ 12 קפיצות).

```
// Move DC motor forward with increasing speed
digitalWrite(motor1Pin1, HIGH);
digitalWrite(motor1Pin2, LOW);
while (dutyCycle <= 255){
    ledcWrite(pwmChannel, dutyCycle);
    Serial.print("Forward with duty cycle: ");
    Serial.println(dutyCycle);
    dutyCycle = dutyCycle + 5;
    delay(500);
}
```

בסיום לולאת ה while שמים במשתנה dutyCycle את הערך 200

```
dutyCycle = 200;
```

ומכאן פונקציית ה loop חוזרת על עצמה.

א. ביבליוגרפיה

1. <https://randomnerdtutorials.com/esp32-dc-motor-l298n-motor-driver-control-speed-direction/>
2. <https://lastminuteengineers.com/l298n-dc-stepper-driver-arduino-tutorial/>
3. http://wiki.sunfounder.cc/index.php?title=Motor_Driver_Module-L298N
4. <https://howtomechatronics.com/tutorials/arduino/arduino-dc-motor-control-tutorial-l298n-pwm-h-bridge/>