

Esp32 וחיישן טמפרטורה LM35 בסביבת ארדואינו IDE

א. כללי

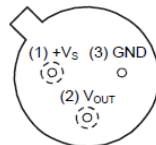
במדריך זה נלמד כיצד להשתמש ב- ESP32 (או ESP8266) עם חיישן הטמפרטורה LM35 ב- Arduino IDE באמצעות הדקי הקלט אנלוגיים של ADC. נסביר כיצד פועל חיישן הטמפרטורה LM35, כיצד לחבר אותו ל- ESP32 וכיצד לקבל קריאות באמצעות ADC ב- Arduino IDE.

ניעזר בקישורים הבאים:

1. <https://deepbluembedded.com/esp32-temperature-sensor-lm35-interfacing-arduino/>
2. https://esp32io.com/tutorials/esp32-lm35-temperature-sensor#content_introduction_to_lm35_temperature_sensor
3. <https://www.ti.com/lit/ds/symlink/lm35.pdf>

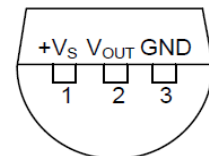
LM35 הוא חיישן טמפרטורה הנמצא בשימוש נרחב בפרויקטים אלקטרוניים. בחיישן יש דיודה שמשנה את המתח עליה כתלות בטמפרטורה. לחיישן שימוש מוגבל ביישומים תעשייתיים עקב מגבלות טווח הטמפרטורה המרבי. הוא מדורג לטווח מלא של 55°C עד 150°C (מינוס 55 עד פלוס 150 מעלות צלסיוס). ניתן בקלות להפעיל אותו ולקרוא באופן מידי את רמת המתח במוניטור הטורי של הארדואינו. הוא זול יחסית כי ביציאה שלו יש מתח ואין לו צורך בהמרה לתקשורת טורית או I2C. הוא מופיע בסוגי אריזה שונים (הנקראים TO אם האריזה כמו של טרנזיסטור או SO באריזה של ג'וק) כפי שנראה באיור הבא :

NDV Package
3-Pin TO-CAN
(Top View)

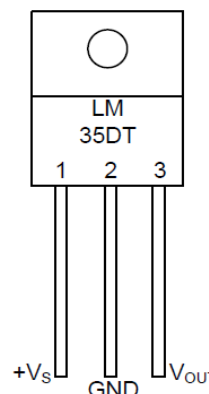


Case is connected to negative pin (GND)
Refer the second NDV0003H page for reference

LP Package
3-Pin TO-92
(Bottom View)



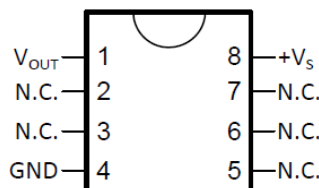
NEB Package
3-Pin TO-220
(Top View)



Tab is connected to the negative pin (GND).

NOTE: The LM35DT pinout is different than the discontinued LM35DP

D Package
8-PIN SOIC
(Top View)



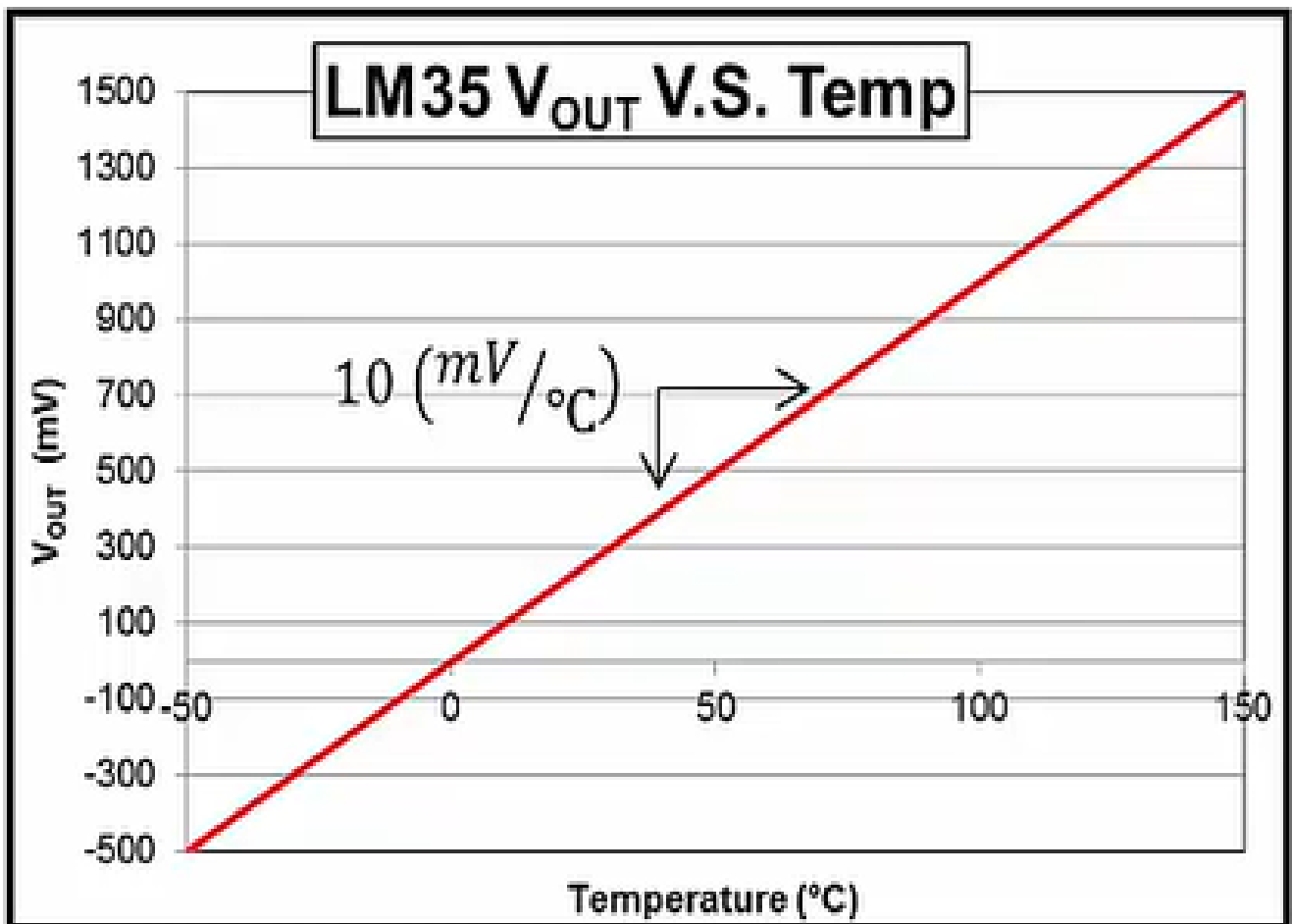
N.C. = No connection

איור 1 : האריזות השונות של חיישן הטמפרטורה LM35.

לדוגמה : בטמפרטורה של 100 מעלות צלסיוס מתח היציאה מהחיישן יהיה: $(10 \text{ mV} / ^\circ\text{C}) * 100 = 1000 \text{ mV} = 1 \text{ v}$ כדי לעבור ממעלות צלסיוס לפרנהייט נשתמש בנוסחה:

$$F = C * 9 / 5 + 32$$

האיור הבא מתאר את אופיין המתח במילי וולט (ציר Y) כפונקציה של הטמפרטורה במעלות צלסיוס (ציר X).

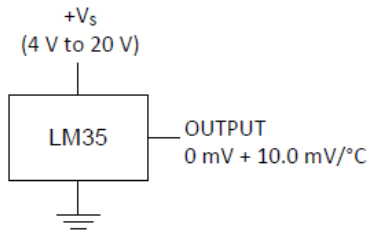


איור 3 : אופיין מתח כפונקציה של הטמפרטורה של החיישן.

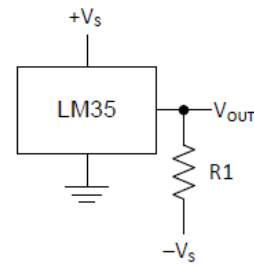
מהאיור רואים שהאופיין ליניארי וכאשר הטמפרטורה עולה מתח היציאה עולה. ליתר דיוק כל עלייה של מעלה צלסיוס – מתח היציאה עולה ב 10 מילי וולט. רואים שב 0 מעלות צלסיוס מתח היציאה מהחיישן 0 וולט. ב -55°C מתח היציאה -0.55 V . בטמפרטורה של 50°C מתח היציאה 0.5 V . ב 100°C מתח היציאה 1 V .

באיור הבא מתוארות 2 צורות חיבור של החיישן :

**Basic Centigrade Temperature Sensor
(2°C to 150°C)**



Full-Range Centigrade Temperature Sensor



Choose $R_1 = -V_S / 50 \mu A$
 $V_{OUT} = 1500 \text{ mV at } 150^\circ C$
 $V_{OUT} = 250 \text{ mV at } 25^\circ C$
 $V_{OUT} = -550 \text{ mV at } -55^\circ C$

איור 4 : חיבור החיישן. מצד שמאל למדידת מתח מ $2^\circ C$ ועד $150^\circ C$. מצד ימין לכל תחום הטמפרטורה מ $-55^\circ C$ עד $150^\circ C$.

מהאיור רואים שהמעגל השאלי משמש למדידת טמפרטורה מ $2^\circ C$ ועד $150^\circ C$.

המעגל הימני באיור משמש המעגל למדידת טמפרטורה של כל תחום הטמפרטורה מ $-55^\circ C$ עד $150^\circ C$. המעגל דורש תוספת של נגד חיצוני וספק מתח שלילי.

הטבלה הבאה מתארת את הערכים הנקובים המקסימליים של חייגן LM35 של חברת TI – Texas Instruction:

6.1 Absolute Maximum Ratings

over operating free-air temperature range (unless otherwise noted)⁽¹⁾⁽²⁾

		MIN	MAX	UNIT
Supply voltage		-0.2	35	V
Output voltage		-1	6	V
Output current			10	mA
Maximum Junction Temperature, T_{jmax}			150	$^\circ C$
Storage Temperature, T_{stg}	TO-CAN, TO-92 Package	-60	150	$^\circ C$
	TO-220, SOIC Package	-65	150	

טבלה 1 : ערכים נקובים מקסימליים באוויר הפתוח.

טבלה הבאה מתארת תנאי פעולה מומלצים

6.3 Recommended Operating Conditions

over operating free-air temperature range (unless otherwise noted)

		MIN	MAX	UNIT
Specified operating temperature: T_{MIN} to T_{MAX}	LM35, LM35A	-55	150	$^\circ C$
	LM35C, LM35CA	-40	110	
	LM35D	0	100	
Supply Voltage ($+V_S$)		4	30	V

טבלה 2 : תנאי פעולה מומלצים באוויר הפתוח.

ה. חיבור LM35 אל ESP32

נראה כיצד להשתמש ב- ESP32 ADC כדי לבצע מדידות של חיישן הטמפרטורה ב- Arduino IDE. החיבור בין חיישן הטמפרטורה ESP32 ו- LM35 צריך להיות כפי שרואים בטבלה הבאה שבה חיברנו את יציאת החיישן ל ADC1_7. יש אפשרות להשתמש בכל פין GPIO שהוא הדק אנלוגי. הטבלה הבאה מתארת את החיבורים:

חיישן הטמפרטורה LM35	הדקי ESP32 (לוח DevKit V1)
Vout	ערוץ 7 של ADC מספר 1 (GPIO35 (ADC1_7)
Vcc	V in
GND	GND

טבלה 3: חיבור החיישן LM35 אל ESP32

ה.1 כיוול ה ADC

נשתמש בשגרת הכיוול ESP32 ADC (על ידי Espressif המיושמת ב-Arduino Core) כדי לקבל קריאות טמפרטורה מדויקות ככל האפשר. להלן השלבים המדויקים שנבצע כדי לקרוא חיישן טמפרטורה LM35 עם לוח ESP32 באמצעות כיוול ADC ונסביר את הקוד שורה אחר שורה.

הערה: ניתן לעבור לסעיף ה.2 או ה.3 ולקפוץ לתוכניות המדידה ללא הכיוול.

א. נכלול את קובץ הכותרת של ספריית כיוול ADC של Arduino Core

```
#include "esp_adc_cal.h"
```

ב. נגדיר הדק GPIO של חיישן LM35 שימש כקלט אנלוגי ADC

```
#define LM35_Sensor 35
```

ג. ניצור 4 משתנים: משתנה מטיפוס int - LM35_Raw_sensor1 שבו נשמור את קריאת ה ADC (המילה Raw פרושה גלמי – כלומר קריאה מה ADC לפני שהפכנו אותה למעלות צלסיוס או פרנהייט) ו 3 משתנים מטיפוס float כדי להחזיק את מדידת הטמפרטורה הסופית ב C° - LM35_TempC_Sensor1 ו במעלות פרנהייט F° - LM35_TempF_Sensor ומשתנה נוסף Voltage שנצטרך אותו בחישובים. את כל המשתנים אתחלנו ב 0 ו 0.0 בהתאמה.

```
int LM35_Raw_sensor1 = 0;
```

```
float LM35_TempC_Sensor1 = 0.0 ;
```

```
float LM35_TempF_Sensor1 = 0.0 ;
```

```
float Voltage = 0.0 ;
```

ד. נוסיף פונקציית קריאה וכיוול של ה ADC זו לתוכנית שלנו:

```
1 uint32_t readADC_Cal(int ADC_Raw)
2 {
3   esp_adc_cal_characteristics_t adc_chars;
```

```

4
5 esp_adc_cal_characterize(ADC_UNIT_1, ADC_ATTEN_DB_11, ADC_WIDTH_BIT_12, 1100, &adc_chars);
6 return(esp_adc_cal_raw_to_voltage(ADC_Raw, &adc_chars));
7 }

```

ה. בפונקציית ה () loop נקרא את ה - ADC כדי לקבל את המתח המכיל ולהמיר אותו לטמפרטורה ב- $^{\circ}\text{C}$ או $^{\circ}\text{F}$.

```

1 void loop()
2 {
3 // Read LM35_Sensor1 ADC Pin
4 LM35_Raw_Sensor1 = analogRead(LM35_Sensor1);
5 // Calibrate ADC & Get Voltage (in mV) // כיול וקבלת המתח במילי וולט
6 Voltage = readADC_Cal(LM35_Raw_Sensor1);
7 // TempC = Voltage(mV) / 10
8 LM35_TempC_Sensor1 = Voltage / 10;
9 LM35_TempF_Sensor1 = (LM35_TempC_Sensor1 * 1.8) + 32;
10
11 // Print The Readings
12 Serial.print("Temperature = ");
13 Serial.print(LM35_TempC_Sensor1);
14 Serial.print(" °C , ");
15 Serial.print("Temperature = ");
16 Serial.print(LM35_TempF_Sensor1);
17 Serial.println(" °F");
18
19 delay(100);
20 }

```

יש לשים לב שאחרי הכיול שביצענו בתוכנית, הרעש חייב להיות מינימלי וניתן לשפר/לבטל את הרעש בעזרת מסנן דיגיטאלי (כמו ממוצע של קריאות) כדי לקבל קריאת טמפרטורה מדויקת.

ה.2 התוכנית עם ההדפסות

בתוכנית נקרא את הטמפרטורה באמצעות פונקציית הכיול של ה ADC ונדפיס את הקריאות ליציאה הטורית של המוניטור.

```
1 #include "esp_adc_cal.h"
2 #define LM35_Sensor1 35
3 int LM35_Raw_Sensor1 = 0;
4 float LM35_TempC_Sensor1 = 0.0;
5 float LM35_TempF_Sensor1 = 0.0;
6 float Voltage = 0.0;
7 void setup()
8 {
9   Serial.begin(115200);
10 }
11
12 void loop()
13 {
14   // Read LM35_Sensor1 ADC Pin
15   LM35_Raw_Sensor1 = analogRead(LM35_Sensor1);
16   // Calibrate ADC & Get Voltage (in mV)
17   Voltage = readADC_Cal(LM35_Raw_Sensor1);
18   // TempC = Voltage(mV) / 10
19   LM35_TempC_Sensor1 = Voltage / 10;
20   LM35_TempF_Sensor1 = (LM35_TempC_Sensor1 * 1.8) + 32;
21
22   // Print The Readings
23   Serial.print("Temperature = ");
24   Serial.print(LM35_TempC_Sensor1);
25   Serial.print(" °C , ");
26   Serial.print("Temperature = ");
```

```

27 Serial.print(LM35_TempF_Sensor1);
28 Serial.println(" °F");
29
30 delay(100);
31 }
32
33 uint32_t readADC_Cal(int ADC_Raw)
34 {
35   esp_adc_cal_characteristics_t adc_chars;
36
37   esp_adc_cal_characterize(ADC_UNIT_1, ADC_ATTEN_DB_11, ADC_WIDTH_BIT_12, 1100, &adc_chars);
38   return(esp_adc_cal_raw_to_voltage(ADC_Raw, &adc_chars));
39 }

```

נרשום את התוכנית ב ארדואינו IDE נבחר את הלוח המתאים ואת יציאת ה COM , נלחץ על לחצן האתחול BOOT ונשאיר את האצבע על הלחצן הלחץ. כאשר Arduino IDE מתחיל לשלוח את הקוד, נשחרר את הלחצן ונמתין להשלמת הורדת התוכנית לכרטיס .

נפתח את המוניטור הטורי של הארדואינו ונקבל את התוצאות של מדידת הטמפרטורה :

The screenshot shows the Serial Monitor window for COM5. The output displays the following text:

```

Temperature = 33.30 °C , Temperature = 91.94 °F
Temperature = 33.30 °C , Temperature = 91.94 °F
Temperature = 33.30 °C , Temperature = 91.94 °F
Temperature = 33.30 °C , Temperature = 91.94 °F
Temperature = 33.30 °C , Temperature = 91.94 °F
Temperature = 33.30 °C , Temperature = 91.94 °F
Temperature = 33.30 °C , Temperature = 91.94 °F
Temperature = 33.30 °C , Temperature = 91.94 °F

```

איור 5 : הדפסות הטמפרטורה במסך המוניטור של ה IDE של הארדואינו.


```
// convert the voltage to the temperature in °C
float tempC = milliVolt / 10;
// convert the °C to °F
float tempF = tempC * 9 / 5 + 32;

// print the temperature in the Serial Monitor:
Serial.print("Temperature: ");
Serial.print(tempC); // print the temperature in °C
Serial.print("°C");
Serial.print(" ~ "); // separator between °C and °F
Serial.print(tempF); // print the temperature in °F
Serial.println("°F");

delay(500);
}
```

התוצאות שהתקבלו במסך המוניטור הטורי של הארדווינו IDE נראות באיור הבא

