

TCS34725 חיישן צבע

ציוד נדרש:

- ערכת פיתוח
- קבל USB לחיבור בין הערכה למחשב
 - חיישן צבע TCS34725
 - 4 חוטי חיבור (זכר-נקבה)



מהלך הניסוי

1. הכנת החיישן לעבודה:

החיישן מגיע יחד עם תושבת חיבורים חשמליים שלא מולחמת לכרטיס החיישן, לכן יש צורך להלחים אותם. הערה: כדאי להכניס את הצד הקצר של הפינים לכרטיס החיישן ולעשות זאת מהצד האחורי (שאין בו רכיבים אלקטרוניים). זאת בכדי שיהיה ניתן לקרב את החיישן לעצם שאת צבעו אנו מודדים מבלי שחוטי החיבורים לבקר יפריעו לכך.



2. יש לחבר את החיישן לערכת הפיתוח ע"פ הטבלה הבאה:

SDA	SCL	GND	Vcc	שם ההדק בחיישן TCS34725
B9	B6	GND	5v	שם ההדק בערכת NOA



New Project

3. נפתח פרויקט חדש לעבודה עם הבקר.

▷ Recent		Sort by:	Default 👻 🔛		Search Installed Templates (Ctrl+E)	p-
 ✓ Installed ✓ Templates ▷ Visual Basic ✓ Visual C# 		C*	Windows Forms Application	Visual C#	Type: Visual C# <no available="" description=""></no>	1
		_	WPF Application	Visual C#		Ш
Windows Micro Frame	ework	<u>c</u> :\	Console Application	Visual C#		
Template E Test	חדע		Class Library	Visual C#		
Visual C++ Visual Studio Solutions Consults		æ	NOA BRK	Visual C#		Ш
₽ Online						
Name:	FingerprintDY50					
Location:	E:\Teaching\C#\I	Peripher	als\FingerPrintDY50\	*	Browse	
Solution name:	FingerprintDY50				 Create directory for solution Add to source control 	
					OK Cano	:el

4. כרגיל, נכנס למאפיינים של הפרויקט ונשנה בלשונית NET Micro Framework ה Transport ל USB כמתואר באיורים שמשמאל.

	PROJECT BUILD DEBUG TEAM * Add Class * Add New Data Source	TOOLS TEST Shift+Alt+C	WINDOW HELP
itabase Explore	Add New Item Add Existing Item New Folder	Ctrl+Shift+A Shift+Alt+A	
r Toolbox	Image: Show All Files Unload Project Add Reference		
	 Set as StartUp Project Manage NuGet Packages Enable NuGet Package Restore 		
	 Refresh Project Toolbox Items FingerprintDY50 Properties 	Alt+F7	
다. FingerprintDY50 부 ×			
Application	Configuration: Active (Debug))	m: Active (Any CPU)

a	Tingerphileb 100 - X	
abase	Application	Configuration: Active (Debug)
Ę	Build	
lorer	Build Events	Deployment
5	Debug	Transport:
olbo	Resources	USB 🗸
×	Reference Paths	Emulator
	.NET Micro Framework	Serial
		ТСР/ІР

- 5. בכדי שנוכל לתפעל את החיישן צבע, נוח להשתמש במחלקה בנויה הייעודית לכך. לשם כך, עלינו להכיר מחלקה זאת לסביבת פיתוח ע"י הוספתה לפרויקט. ניתן לעשות זאת באופן הבא:
 - א. ב Visual Studio נלחץ עם המקש הימני של העכבר על Visual Studio א.

		Ŧ	Solution Explorer 🔹 म 🗙 ချီ	כמתואר Solution Explorer
		÷	Search Solution Explorer (Ctrl+ P -	באיור:
			Solution 'FingerprintDY50' (1 pr FingerprintDY50 FingerprintDY50 FingerprintDY50 FingerprintBis Fingerprint FingerprintBis FingerprintBis Fingerpri	
	Add Reference		C# BRK Noa.cs	
苗	Manage NuGet Packages		C# Program.cs	
đ	Scope to This New Solution Explorer View			



- ב. בתפריט שתפתח נבחר את האפשרות העליונה ...Add References
 - ג. בחלון שיפתח נבחר את הלשונית Browse בה ננווט למקום הספריות במחשב (בדרך כלל זאת ספריית libs המסופקת ע"י חברת BRK).
 - ד. נבחר את הקובץ TCS34725_RGBcolorSensor.dll
 - ה. נלחץ על הכפתור OK שבתחתית החלונית.

NET Project	ts Recent Browse		
Look in: 🌗	TCS34725_RGBcolorSensor	- 🕝 🎓 📂 🛄-	
Name	*	Date modified	т
🌗 be		27/02/2019 15:07	
🍌 le		27/02/2019 15:07	
🛛 TCS347	25_RGBcolorSensor.dll	27/02/2019 15:07	C
•			•
File name:	TCS34725_RGBcolorSensor		-
Files of type:	Component Files (*.exe *.dll)		•
	(

ההרחבה Solution Explorer נוכל לראות שההרחבה
 התווספה בהצלחה:



7. נוסיף את ההרחבה גם ל using בתוך הקוד שבקובץ Program.cs של הפרויקט:

```
using System;
using Microsoft.SPOT;
using Microsoft.SPOT.Hardware;
using Stm32;
using System.Threading;
using System.Text;
using TCS34725_RGBcolorSensor;
```

8. ניצור מחוץ לפונקציה הראשית ()Main את העצם החדש של חיישן טביעת האצבע:

```
public class Program
{
    static TCS34725_RGBsensor ColorSensor = new TCS34725_RGBsensor();
    public static void Main()
    {
    ...
```



9. בתוך הפונקציה הראשית ()Main נציג הודעת פתיחה (לא חובה) ונאתחל את הרכיב:

10. נבדוק את התקשורת בין הבקר לחיישן:

```
bool connection = false;
do
{
    connection = ColorSensor.ConnCheck();
    if (!connection)
    {
        Debug.Print("No response from sensor.");
        Thread.Sleep(2000);
    }
} while (!connection);
Debug.Print("Connection to sensor is succeeded!");
```

11. במקרה והבקר הצליח ליצור את התקשורת מול החיישן, כלומר החיישן והחיבורים תקינים, ניתן לתת לחיישן הוראה לדגום דגימת צבע ולהציג את התוצאה בחלונית ה Output:

```
while (true)
{
    Debug.Print("Red: " + ColorSensor.GetRedColor() +
        "\tGreen: " + ColorSensor.GetGreenColor() +
        "\tBlue: " + ColorSensor.GetBlueColor() +
        "\tClear: " + ColorSensor.GetClearColor());
    Thread.Sleep(100);
}
```

12. בסה"כ, בשלב זה נקבל את התוכנית הבאה:



```
Debug.Print("Hello, TCS34725 RGB light sensor!");
            ColorSensor.Init();
            bool connection = false;
            do
            {
                connection = ColorSensor.ConnCheck();
                if (!connection)
                {
                    Debug.Print("No response from sensor.");
                    Thread.Sleep(2000);
                }
            } while (!connection);
            Debug.Print("Connection to sensor is succeeded!");
            while (true)
            {
                Debug.Print("Red: " + ColorSensor.GetRedColor() +
                        "\tGreen: " + ColorSensor.GetGreenColor() +
                        "\tBlue: " + ColorSensor.GetBlueColor() +
                        "\tClear: " + ColorSensor.GetClearColor());
                Thread.Sleep(100);
            }
        }
    }
}
```

13. נצרוב את התוכנה לבקר ע"י הקשה על הכפתור F5 במקלדת, או על הלחצן start במסך.

14. נבדוק את פעילות המערכת: נניח את האצבע על זכוכית החיישן ונעקוב אחר ההוראות .visual studio של output בחלונית ה

הצעות לניסויי מעבדה / פרוייקטונים אפשריים:

- הדלקת לד אחד מתוך כמה שמחוברים לבקר, ע"פ הצבע של החפץ עליו מופנה חיישן הצבע. הערה: ניתן לחבר לד RGB אחד במקום לדים רבים בצבעים שונים.
 - 2. ערבוב צבעי "טמבור": רצינו לקבל גוון צבע מסוים בכדי לצבוע אתו את קירות הבית. נציג את דוגמת הצבע מול החיישן והבקר יפתח ע"י מנועי סרוו את הכמויות המתאימים של שלושת צבעי היסוד: אדום, ירוק וכחול. כמובן שיש להוסיף מערבל שיערבב את הצבעים במיכל.

בהצלחה!