

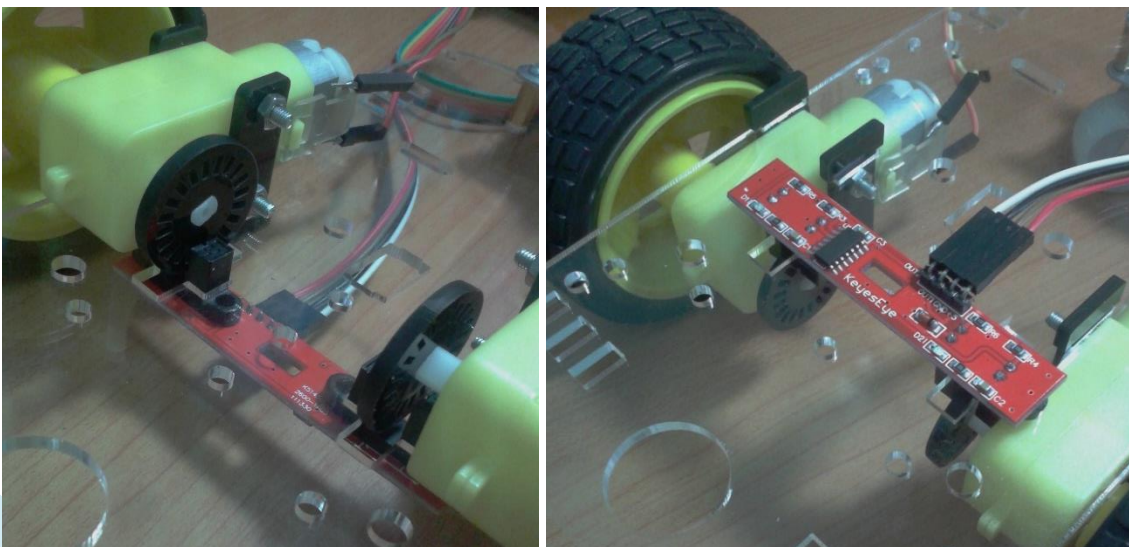
עבודה עם מקודד אופטי יחסי מבוסס חיישני IR דף משימות

ציוד נדרש:

- ערכת פיתוח.
- פלטפורמה מורכבת עם 3 או 4 גלגלים.
- מקודדים יחסיים המורכבים על ציר סיבוב המנוע.
- זוג חיישני IR מותאמים לפלטפורמה.
- 10 חיבורי זכר-נקבה.
- 4 חיבורי זכר-זכר.

מהלך הניסוי

1. בניסויים עם מנועי DC (זרם ישר) ראינו שמהירות הסיבוב של כל מנוע שונה במקצת – דבר המביא לנטיעה של הרובוט לפנות לכיוון מסוים במקום לנסוע ישר. נתגבר על תופעה זאת ע"י שימוש בשני מקודדים אופטיים יחסיים.
 2. נבצע את החיבורים החשמליים והמכאניים של מרכיבי הניסוי.
 3. ראשית, נלחים שני חוטי זכר-זכר לכל מנוע. (כדאי לעשות זאת לפני הרכבת המנועים על הפלטפורמה)
 4. נרכיב את הדיסק של המקודד על ציר סיבוב הפנימי של כל מנוע.
 5. נרכיב את המנועים על הפלטפורמה בעזרת הברגים הסופקים יחד עמה.
 6. נרכיב את כרטיס הדיודות של המקודד ע"ג הפלטפורמה.
- ניתן להיעזר בתצלומים הבאים:



7. נחבר את כל אחד מהמנועים לדוחף זרם:

מנוע ימני ל MOTOR-A

מנוע שמאלי ל MOTOR-B

הדקו היטב את הברגים שבכרטיס דוחף הזרם.

8. וודאו שערכת הפיתוח לא מחוברת למתח: שני חיבורי ה USB שלה מנותקים ולא דולק שום לד על הכרטיס.

9. נחבר את הדוחף זרם ואת החיישני IR לערכת הפיתוח ע"פ הטבלה הבאה:

שם הפין במודול דוחף הזרם	שם הפין בכרטיס הבקר	הערות
B1-A	PC15	
B1-B	PC13	
GND	GND	
VCC	5V	
A1-A	PC14	
A1-B	PE6	

שם הפין במודול מקודד IR	שם הפין בכרטיס הבקר	הערות
OUT2	PE4	
OUT1	PE2	
GND	GND	
5+	5V	

10. חברו את הבקר למחשב ע"י שני חיבורי ה USB.

11. נפעיל את Visual Studio ונפתח פרויקט חדש לעבודה עם הבקר.

12. נגדיר את הפינים של הבקר אליהם התחברנו:

```
static OutputPort RighthMotor1 = new OutputPort(Pins.GPIO_PIN_C_14, false);
static OutputPort RighthMotor2 = new OutputPort(Pins.GPIO_PIN_E_6, false);
static OutputPort LeftMotor1 = new OutputPort(Pins.GPIO_PIN_C_15, false);
static OutputPort LeftMotor2 = new OutputPort(Pins.GPIO_PIN_C_13, false);

static InputPort leftSensor = new InputPort(Pins.GPIO_PIN_E_2, true,
    ResistorModes.PullDown);
static InputPort righthSensor = new InputPort(Pins.GPIO_PIN_E_4, true,
    ResistorModes.PullDown);
```

13. מחוץ לפונקציה הראשית Main() נגדיר 2 משתנים שימשו אותנו לספירת הדפקים המגיעים מהמקודד המורכב על המנוע השמאלי והימני.

```
static double leftCnt = 0, righthCnt = 0;
```

14. כמו כן, מחוץ ל Main() נכתוב פונקציות המסובבות את גלגלי הרכב לכיוונים הרצויים ומניעות את הרכב:

```
private static void LeftStop()
{
    LeftMotor1.Write(false);
    LeftMotor2.Write(false);

    RighthMotor1.Write(true);
    RighthMotor2.Write(false);
}

private static void RighthStop()
{
    RighthMotor1.Write(false);
    RighthMotor2.Write(false);

    LeftMotor1.Write(true);
    LeftMotor2.Write(false);
}

private static void DriveFwd()
{
    LeftMotor1.Write(true);
    LeftMotor2.Write(false);
```

```

    RighthMotor1.Write(true);
    RighthMotor2.Write(false);
}

```

15. נכתוב גם את הפונקציה המשמשת להשוואת מהירות הסיבוב של שני הגלגלים ועוצרת לפרק זמן קצר מאוד את הגלגל שמסתובב מהר מהשני:

```

private static void Compare()
{
    Debug.Print(leftCnt + " " + righthCnt);
    if ((leftCnt - righthCnt) > 2)
    {
        LeftStop();
        Debug.Print("Left stoped, Righth runing");
    }
    else
    {
        if ((leftCnt - righthCnt) < -2)
        {
            RighthStop();
            Debug.Print("Righth stoped, Left runing");
        }
        else DriveFwd();
    }

    if (leftCnt > (double.MaxValue - 5) || righthCnt > (double.MaxValue - 5))
    {
        leftCnt = 0;
        righthCnt = 0;
    }
}

```

16. בתוך ה Main() נציג הודעת פתיחה ונגדיר 2 זוגות משתנים שיעזרו לנו לזהות את עליית הדופק המתקבל מהמקודד:

```

Debug.Print("Hello IrOpticDecoder!");

bool left_last_state = false, righth_last_state = false;
bool left_present_state = false, righth_present_state = false;

```

17. בתוך לולאה אינסופית נכתוב תוכנה המזהה את עליית הדופק המגיע מהמקודד ומוסיפה 1 לאוגר המכיל את הדפקים שמגיעים מאותו המקודד (שמאל או ימין):

```
while (true)
{
    left_present_state = leftSensor.Read();
    righth_present_state = righthSensor.Read();

    if ( (left_last_state == false) && (left_present_state == true) )
    {
        leftCnt++;
    }
    if ((righth_last_state == false) && (righth_present_state == true))
    {
        righthCnt++;
    }

    Compare();

    left_last_state = left_present_state;
    righth_last_state = righth_present_state;
}
}
```

18. בסה"כ קיבלנו את התוכנית הבאה:

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

namespace IrOpticDecoder
{
    public class Program
    {
        static OutputPort RighthMotor1 = new OutputPort(Pins.GPIO_PIN_C_14, false);
        static OutputPort RighthMotor2 = new OutputPort(Pins.GPIO_PIN_E_6, false);

        static OutputPort LeftMotor1 = new OutputPort(Pins.GPIO_PIN_C_15, false);
        static OutputPort LeftMotor2 = new OutputPort(Pins.GPIO_PIN_C_13, false);
    }
}
```

```
static InputPort leftSensor = new InputPort(Pins.GPIO_PIN_E_2, true,
    ResistorModes.PullDown);
static InputPort righthSensor = new InputPort(Pins.GPIO_PIN_E_4, true,
    ResistorModes.PullDown);

static double leftCnt = 0, righthCnt = 0;

public static void Main()
{
    Debug.Print("Hello IrOpticDecoder!");

    bool left_last_state = false, righth_last_state = false;
    bool left_present_state = false, righth_present_state = false;

    DriveFwd();

    while (true)
    {
        left_present_state = leftSensor.Read();
        righth_present_state = righthSensor.Read();

        if ( (left_last_state == false) && (left_present_state == true) )
        {
            leftCnt++;
        }
        if ((righth_last_state == false) && (righth_present_state == true))
        {
            righthCnt++;
        }

        Compare();

        left_last_state = left_present_state;
        righth_last_state = righth_present_state;

    }
}
```

```
private static void Compare()
{
    Debug.Print(leftCnt + " " + righthCnt);
    if ((leftCnt - righthCnt) > 2)
    {
        LeftStop();
        Debug.Print("Left stoped, Righth runing");
    }
    else
    {
        if ((leftCnt - righthCnt) < -2)
        {
            RighthStop();
            Debug.Print("Righth stoped, Left runing");
        }
        else DriveFwd();
    }

    if (leftCnt > (double.MaxValue - 5) || righthCnt > (double.MaxValue - 5))
    {
        leftCnt = 0;
        righthCnt = 0;
    }
}

private static void LeftStop()
{
    LeftMotor1.Write(false);
    LeftMotor2.Write(false);

    RighthMotor1.Write(true);
    RighthMotor2.Write(false);
}

private static void RighthStop()
{
    RighthMotor1.Write(false);
    RighthMotor2.Write(false);

    LeftMotor1.Write(true);
    LeftMotor2.Write(false);
}
```

```
}  
  
private static void DriveFwd()  
{  
    LeftMotor1.Write(true);  
    LeftMotor2.Write(false);  
  
    RighthMotor1.Write(true);  
    RighthMotor2.Write(false);  
}  
}  
}
```

19. נצרוב את התוכנה לבקר ע"י לחיצה עם העכבר על בלחצן Start שבסרגל הפקודות.
20. לאחר הצריבה של התוכנית, הרכב יחל לנסוע קדימה בקוו ישר ובחלונית ה Output של Visual Studio יוצג הדיווח על הסטיות והתיקונים אותה ביצע הבקר.

בהצלחה!