



[🏠](#) > [Electronica](#) > [Senzori](#) > Modul senzor urmarire linie TCRT5000

Modul senzor urmarire linie TCRT5000





Referinta TCRT5000**Conditie:** Produs nou2 Produse in stoc **ATENTIE: Ultimele bucati in stoc!****15,93 lei** cu TVA**Cantitate****INFORMATII****REVIEW-URI****DESCRIERE**

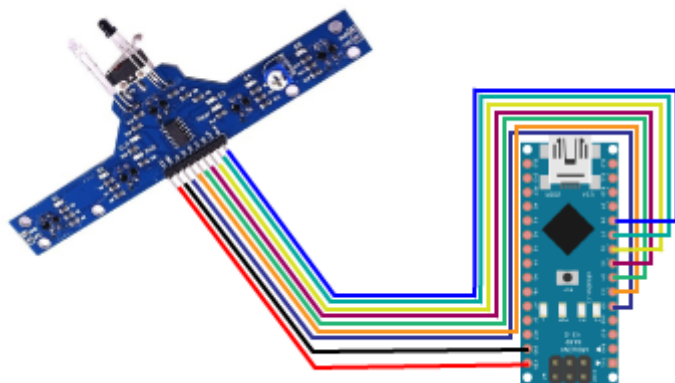
Cu acest modul se poate detecta distanta pana la un obiect sau se poate urmari o linie neagra cu ajutorul senzorilor infrarosu. Prin urmare modulul este folosit la constructia unui robot urmaritor de linie, ce are incorporat atat senzori infrarosu de urmarire linie, orientati in jos, cat si un senzor orientat in fata astfel incat, robotul poate detecta si obstacolele de pe traseu.

SPECIFICAȚII

- Tensiune de alimentare: 5V
- Dimensiuni: 69mm x 28mm
- 6 gauri de montare de 3mm
- 5 senzori IR reflectivi TCRT5000
- Interval distanta de detectare: 0-4 cm pentru senzorii de linie, respectiv 0-5 cm pentru obstacol

- Microswitch detectare obstacole
- S1-S5: iesire digitala ce indica statusul senzorilor S1-S5
- CLP: iesire digitala ce indica statusul microswitchului
- Near: iesire digitala ce indica statusul senzorului de detectare obstacole (sensibilitatea se poate ajusta cu ajutorul potentiometrului)

CONECTARE IMPREUNA CU ARDUINO



INSTRUCTIUNI

1. Se conecteaza modulul TCRT5000 la arduino astfel:

- DE LA MODUL LA ARDUINO
- VCC ---> 5V
- GND ---> GND
- S1 ---> D2
- S2 ---> D3
- S3 ---> D4
- S4 ---> D5
- S5 ---> D6
- CLP ---> D7
- Near ---> D8

2. Se adauga libraria de AICI

3. Din Arduino IDE accesam "Examples" din meniul "File" , "QTRSensors" si "QTRRCExample"

4. Inlocuim numarul senzorilor de pe linia

#define NUM_SENSORS 8 cu 5 si modificam in linia

QTRSensorsRC qtrrc((unsigned char[]) {3, 4, 5, 6, 7, 8, 9, 10} cu pinii digitali pe care i-am mentionat mai sus si anume 2,3,4,5,6. (7,8 sunt folositi pentru detectarea obstacolelor nu pentru urmarirea liniei).

5. Incarcam codul si verificam in Serial Monitor valorile afisate

Rezultatul ar trebui sa fie asemanator

The screenshot shows the Arduino IDE interface with the 'QTRRCExample' sketch loaded. The sketch defines 5 sensors and uses pins 3, 4, 5, 6, 7, 8, 9, and 10. The serial monitor displays the output of the sketch, which includes calibration values and sensor readings.

```

// last seen by sensor 5 before being lost.

#define NUM_SENSORS 5 // number of sensors used
#define TIMEOUT 2500 // waits for 2500 microseconds for sensor outputs to go low
#define EMITTER_PIN 2 // emitter is controlled by digital pin 2

// sensors 0 through 7 are connected to digital pins 3 through 10, respectively
QTRSensorsRC qtrrc((unsigned char[]) {2, 3, 4, 5, 6},
  NUM_SENSORS, TIMEOUT, EMITTER_PIN);
unsigned int sensorValues[NUM_SENSORS];

void setup()
{
  delay(500);
  pinMode(13, OUTPUT);
  digitalWrite(13, HIGH); // turn on Arduino's LED to indicate we are in calibration mode
  for (int i = 0; i < 400; i++) // make the calibration take about 10 seconds
  {
    qtrrc.calibrate(); // reads all sensors 10 times at 2500 us per read (i.e. ~25 ms per c
  }
  digitalWrite(13, LOW); // turn off Arduino's LED to indicate we are through with calibratio
  // print the calibration minimum values measured when emitters were on
  Serial.begin(9600);
  for (int i = 0; i < NUM_SENSORS; i++)
  {
    Serial.print(qtrrc.calibratedMinimumOn(i));
    Serial.print(' ');
  }
  Serial.println();

  // print the calibration maximum values measured when emitters were on
  for (int i = 0; i < NUM_SENSORS; i++)
  {
    Serial.print(qtrrc.calibratedMaximumOn(i));
    Serial.print(' ');
  }
  Serial.println();
}

void loop()
{
  // read sensor values
  qtrrc.readSensorData(sensorValues);

  // print sensor values
  for (int i = 0; i < NUM_SENSORS; i++)
  {
    Serial.print(sensorValues[i]);
    Serial.print(' ');
  }
  Serial.println();

  // delay between readings
  delay(100);
}

```

The serial monitor output shows the following data:

Sensor	Calibrated Minimum	Calibrated Maximum
0	1000	1000
1	1000	1000
2	1000	1000
3	1000	1000
4	1000	1000
5	1000	1000
6	1000	1000
7	1000	1000
8	1000	1000
9	1000	1000
10	1000	1000

Sketch uses 4794 bytes (15%) of program storage space. Maximum is 30720 bytes.
Global variables use 242 bytes (1%) of dynamic memory, leaving 1806 bytes for local variables. Maximum is 2048 bytes.

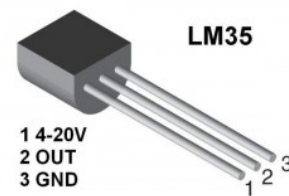
30 alte produse în aceeași categorie:



MQ-8 Modul detectie...
18,85 lei ~~22,31 lei~~



MQ-9 Modul detectie...
18,49 lei ~~22,31 lei~~



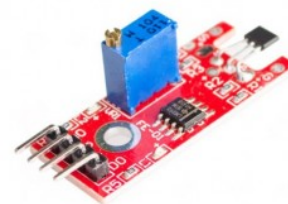
Senzor temperatura
LM35DZ
13,39 lei



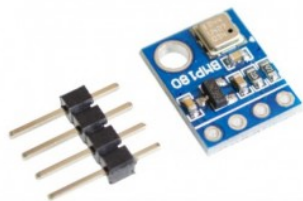
Senzor temperatura...
11,16 lei



Termistor din sticla...
8,93 lei



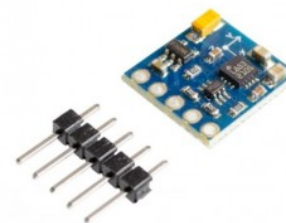
Modul senzor
magnetic...
7,49 lei ~~8,93 lei~~



Modul senzor
presiune...
16,07 lei



Senzor de
temperatură...
23,48 lei ~~27,44 lei~~



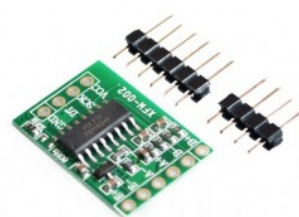
Modul busolă 3 axe...
19,97 lei ~~22,80 lei~~



Modul senzor curent...
13,45 lei



Termostat digital
W1209

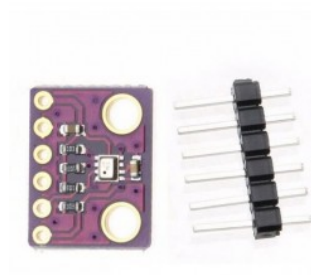


Modul citire senzor...
5,88 lei ~~6,69 lei~~



Senzor greutate
16,07 lei

26,78 lei

Modul senzor
presiune...17,87 lei ~~21,42 lei~~Modul termocuplă tip
K...18,99 lei ~~26,78 lei~~

Modul senzor tensiune

3,75 lei



Senzor lumină și...

48,48 lei ~~62,48 lei~~Modul senzor curent
3A...

Modul senzor culoare...

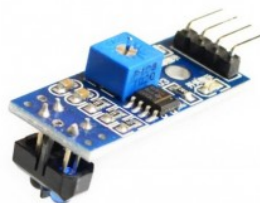
19,49 lei ~~21,42 lei~~

13,95 lei ~~26,78 lei~~



Modul senzori
urmărire...

26,78 lei



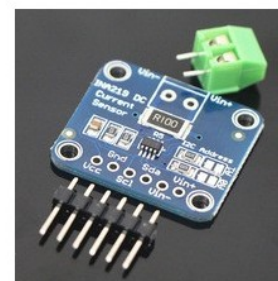
Modul senzor
urmărire...

8,03 lei



Voltmetru digital DC...

9,99 lei ~~16,07 lei~~



Senzor bidirectional...

27,00 lei



Senzor miscare tip...



Senzor magnetic ușă -...

Voltmetru +
ampermetru...

26,78 lei

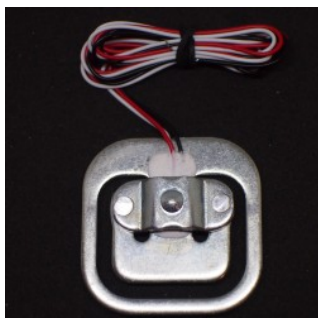
8,95 lei ~~9,76 lei~~

5,36 lei



Encoder magnetic

77,98 lei



Senzor de greutate
50kg

6,52 lei



Termocupla tip K

13,50 lei



1.5m

Termocupla tip K cu...

18,00 lei

CONTACT

SUPPORT: SUPPORT@ARDUSHOP.RO

CERERI OFERTE: SALES@ARDUSHOP.RO

SERVICII (PRINTARE 3D): SERVICES@ARDUSHOP.RO

TEL: 0727387467, 0756018015 (DEPOZIT) L-V, 12:00-17:00