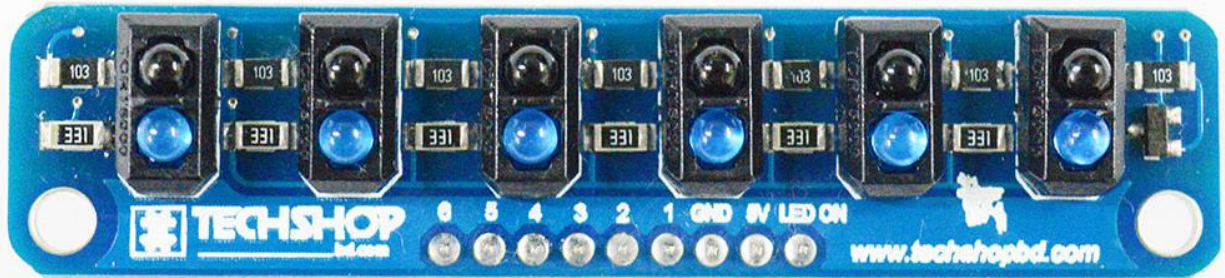


# USER MANUAL

## Analog IR Sensor Array6



### Description:

Although most Arduino users have the Uno board, there's not been a line sensor designed for taking advantage of all 6 analog pins of it. This sensor has been built to use with Arduino Uno based line follower and maze solver robots. You can also connect it to any other 5V based arduino boards like the Mega, Nano and Leonardo.

This sensor also has IR LED control pins. That means using arduino code, you can turn ON and OFF the IR LEDs. So, you can take two sets of reading from the IR receivers: one by turning OFF IR LEDs and one by turning ON IR LEDs. Difference between these two readings are the actual readings generated by the reflection of infrared light only from IR LEDs of the sensor array. In this way you can eliminate the effect of any flash light or sunlight.

### Features:

1. Best analog line sensor for Arduino Uno. Also compatible with any 5V Arduino board.
2. Supports Pololu QTR library (QTR RA) . You can use it just like a Pololu QTR analog sensor.
3. Sunlight and flash resistant. Ability to turn ON & OFF IR LEDs by arduino digital pin.
4. Small and compact design. Uniform 1cm distance between IR sensors for better line detection.

## Specifications:

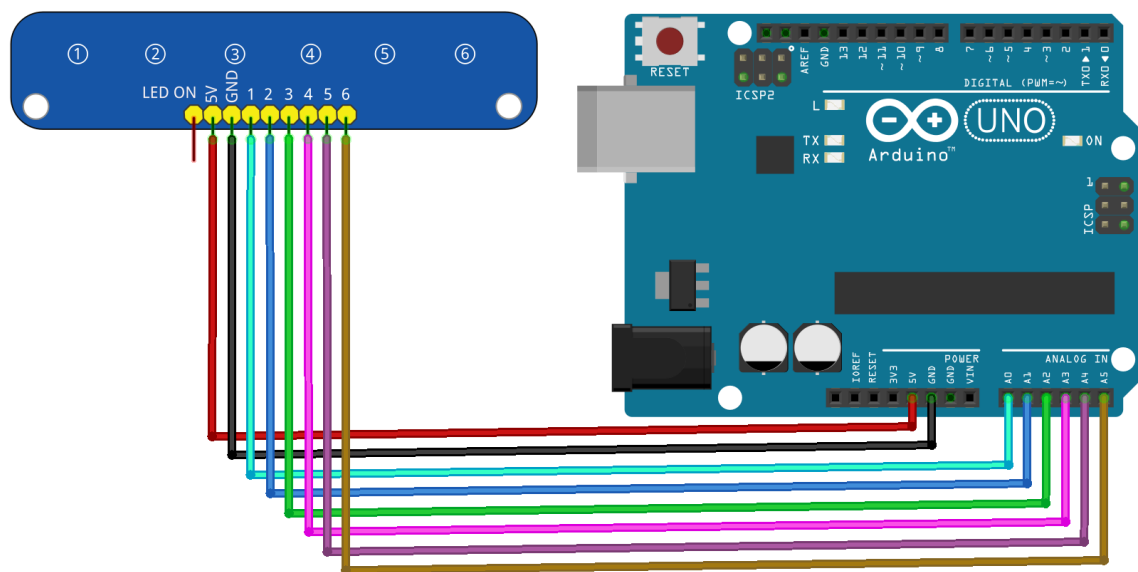
- Features **TCRT500** that has effective plastic packaging and built-in sunlight effect compensation.
- **1 cm spacing** between two adjacent sensors, covering 9cm distance by the entire sensor array.
- Recommended operational height from ground **3mm-15mm**. (height of IR sensors tip, not the PCB).
- Generates high voltage for white surface and low voltage for black surface on signal pins.
- Large value difference for white and black surface at recommended operational height from ground (more than 3volts).

## Circuit Connection and Code:

### Method 1:

In this method we will keep the LED ON pin disconnected. This way, The IR LEDs will always be turned ON. This is the easiest way to take reading from the sensor array.

However, you cannot eliminate the effect of external infrared light (like camera flash or sunlight) in this way. Moreover, keeping IR LEDs always turned ON will drain unnecessary battery power when your robot is powered ON but not following line.



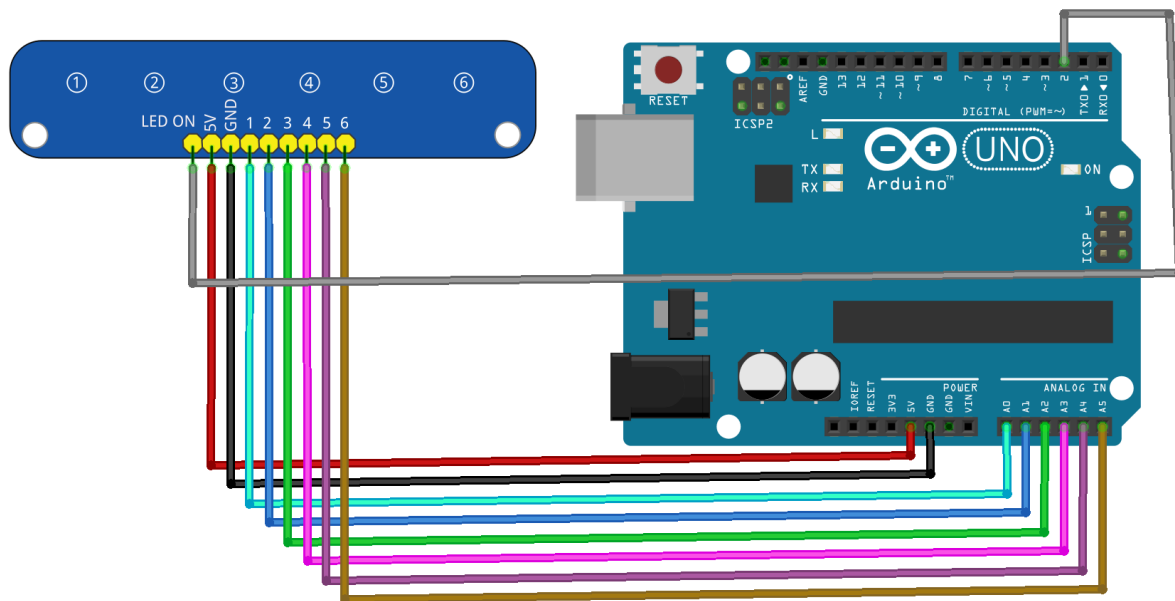
## Arduino Code:

```
int value[6];
void setup()
{
    Serial.begin(9600);
}
void loop(){
    for(int i=0;i<=5;i++)
    {
        value[i]=analogRead(i);
        Serial.print(value[i]);
        Serial.print(" ");
    }
    Serial.println(" ");
}
```

## Method 2:

In this method we will keep the LED ON pin connected to Arduino digital pin 2. We will turn ON the LEDs only when the robot is following line and thus save battery power.

Moreover, we'll take two readings from the IR sensors: one by turning OFF IR LEDs and one by turning ON IR LEDs. Difference between these two readings are the actual readings generated by the reflection of infrared light only from IR LEDs of the sensor array. In this way you can eliminate the effect of any flash light or sunlight.



## Arduino Code:

```
int value[6],valueON[6],valueOFF[6];
void setup()
{
    Serial.begin(9600);
    pinMode(2, OUTPUT);
}
void loop()
{
    digitalWrite(2,HIGH); // turn ON IR LEDs
    delay(20); // give some time to turn ON

    for(int i=0;i<=5;i++) valueON[i]=analogRead(i);

    digitalWrite(2,LOW); // turn OFF IR LEDs
    delay(20); // give some time to turn OFF

    for(int i=0;i<=5;i++) valueOFF[i]=analogRead(i);

    // calculate actual sensor reading
    for(int i=0;i<=5;i++)
    {
        value[i]=valueON[i]-valueOFF[i];
        Serial.print(value[i]);
        Serial.print("  ");
    }
    Serial.println(" ");
}
```

## Sample Output:

Follow any of method 1 or method 2. Then open serial monitor and you'll see the ADC values of each sensors like below.

```
129 211 155 754 916 815
126 211 155 763 917 815
127 211 155 763 917 815
127 211 156 763 917 815
```

The reading is proportional to the color brightness beneath the IR sensors. The IR sensors will give higher values over white surface and lower values over black surface.