



# Capteurs - Distance - Lidar

[Mise à jour le 18/6/2021]

## Généralités

La télédétection par laser ou lidar, acronyme de l'expression en langue anglaise « light detection and ranging » ou « laser detection and ranging » (soit en français « détection et estimation de la distance par la lumière » ou « par laser »), est une technique de mesure à distance fondée sur l'analyse des propriétés d'un faisceau de lumière renvoyé vers son émetteur.



À la différence du radar qui emploie des ondes radio ou du sonar qui utilise des ondes acoustiques, le lidar utilise de la lumière (du spectre visible, infrarouge ou ultraviolet). Celle-ci est quasiment toujours issue d'un laser, et donc cohérente.

Le principe de la télémétrie (détermination de la distance d'un objet), qui concerne une grande partie des applications du lidar, requiert généralement l'utilisation d'un laser impulsionnel. La distance est donnée par la mesure du délai entre l'émission d'une impulsion et la détection d'une impulsion réfléchiée, connaissant la vitesse de la lumière. Une autre classe d'applications basées sur la mesure de vitesse emploie un laser à spectre d'émission fin (une fréquence bien déterminée). [Wikipédia](#)



## Module Lidar TF Mini

- **Source** : [pdf](#)

Module laser Lidar permettant de mesurer des distances de 0,3 à 12 m.

- **Distributeur** : [Gotronic](#)
- **Caractéristiques**
  - Alimentation: 5 Vcc
  - Consommation: environ 0,12 W
  - Plage de mesure: de 0,3 à 12 m
  - Précision:

- 1% en dessous de 6 m
- 2 % entre 6 et 12 m
- Résolution: 1 cm
- Longueur d'onde: 850 nm
- Pic de puissance: 800 mA
- Fréquence de mesure: 100 Hz
- Interface: UART TTL 3,3 Vcc
- Débits: 115200 bauds

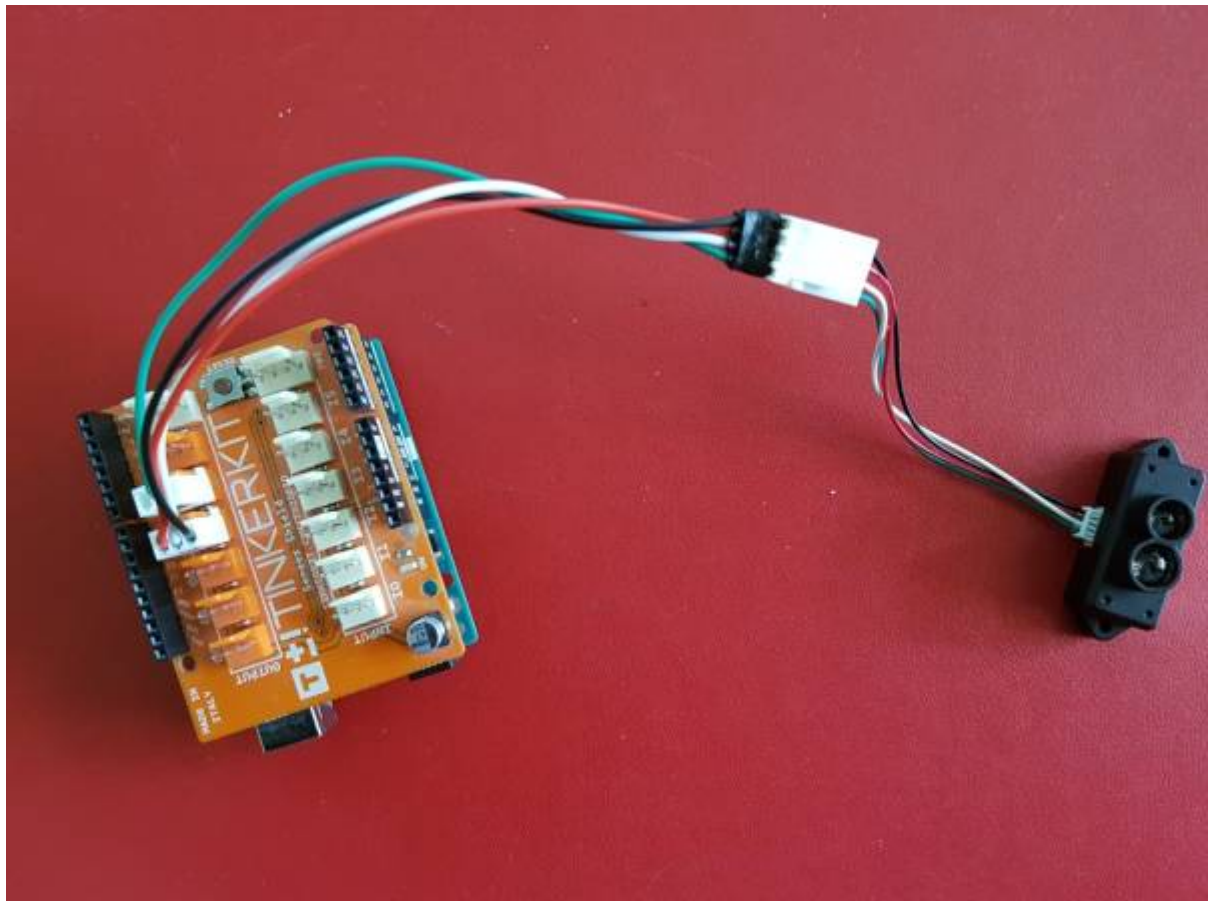


• **Documentation**

- Benewake : [pdf à télécharger](#)
- Elecrow : [pdf à télécharger](#)

• **Programmation d'une carte Arduino Uno R3**

- Connexion à un shield [Tinkerkit v2](#) monté sur une Arduino Uno



- *Un premier exemple pour tester le capteur*



[lidarTFMini.cpp](#)

```
/* this program is the interpretation routine of standard output protocol of TFmini product on Arduino. For details, refer to Product
```

*Specifications.*

*For Arduino boards with only one serial port like UNO board, the function of software visual serial port is to be used. \*/*

*// Connections*

```
#define RX 5 // 03 Shield Tinkerkit
```

```
#define TX 6 // 04 Shield Tinkerkit
```

```
#include <SoftwareSerial.h> //header file of software serial port
```

*//define software serial port name as Serial1 and*

*//define pin5 as RX*

*//and pin6 as TX*

```
SoftwareSerial Serial1(RX, TX);
```

*/\* For Arduinoboards with multiple serial ports like DUEboard, interpret above two pieces of code and directly use Serial1 serial port \*/*

```
int dist; //actual distance measurements of LiDAR
```

```
int strength; //signal strength of LiDAR
```

```
int check; //save check value
```

```
int i;
```

```
int uart[9]; //save data measured by LiDAR
```

```
const int HEADER = 0x59; //frame header of data package
```

```
void setup()
```

```
{
```

```
    Serial.begin(9600); //set bit rate of serial port connecting  
    Arduino with computer
```

```
    Serial1.begin(115200); //set bit rate of serial port connecting  
    LiDAR with Arduino
```

```
}
```

```
void loop()
```

```
{
```

```
    if (Serial1.available()) //check if serial port has data input  
    {
```

```
        if (Serial1.read() == HEADER) //assess data package frame  
        header 0x59
```

```
        {
```

```
            uart[0] = HEADER;
```

```
            if (Serial1.read() == HEADER) //assess data package frame  
            header 0x59
```

```
            {
```

```
                uart[1] = HEADER;
```

```
                for (i = 2; i < 9; i++) //save data in array
```

```
                {
```

```
                    uart[i] = Serial1.read();
```

```
                }
```

```
                check = uart[0] + uart[1] + uart[2] + uart[3] + uart[4]
```

```
+ uart[5] + uart[6] + uart[7];
    if (uart[8] == (check & 0xff)) //verify the received
data as per protocol
    {
        dist = uart[2] + uart[3] * 256;    //calculate
distance value
        strength = uart[4] + uart[5] * 256; //calculate
signal strength value
        Serial.print("dist = ");
        Serial.print(dist); //output measure distance value
of LiDAR

        Serial.print('\t');
        Serial.print("strength = ");
        Serial.print(strength); //output signal strength
value

        Serial.print('\n');
    }
}
}
```



[Télécharger](#) le projet PlatformIO pour VSCode.

From:  
<http://webge.fr/dokuwiki/> - **WEBGE Wikis**

Permanent link:  
<http://webge.fr/dokuwiki/doku.php?id=materiels:capteurs:distance:lidar>

Last update: **2022/07/07 09:52**

