Development Routines of TF LiDAR on Arduino

This routine is aimed to help our customers quickly master the LiDAR products of our company and to shorten the development cycle. As for Arduino board, UNO board and DUE board, for instance, represent respectively the board with one serial port and the board with multiple serial ports. Take TF02 for example, it is applicable to other single point ranging LiDAR (such as TF01, TF mini and TF20). If the communication protocol is changed, please refer to the routine and make corresponding modifications. In the end, the data sent by LiDAR is correctly analyzed by Arduino according to corresponding protocols and then is sent to PC, and at last is displayed via the serial monitor on IDE.

For detailed introduction and learning materials on Arduino, please refer to the following two websites:

Chinese community: http://www.arduino.cn/

English official website: https://www.arduino.cc/

Step 1: Hardware Connection

The linear order for the LiDAR products of our company is defined as: red +5V, black GND, green TX, white RX; the power supply voltage for TF02 is 5V, so it is permissible to directly connect 5V and GND on the Arduino board. For other LiDARs, please refer to product specification to ensure normal power supply; For serial communication connection, please note that with regard to Arduino UNO board, the TX end of LiDAR connects to pin 2 port (soft serial port Serial1's RX) and the RX end of LiDAR connects to pin 3 (soft serial port Serial1's TX). This is relevant to the programming hereinafter; For Arduino DUE board, the TX end of LiDAR connects to RX1 port and the RX end of LiDAR connects to TX1 port.
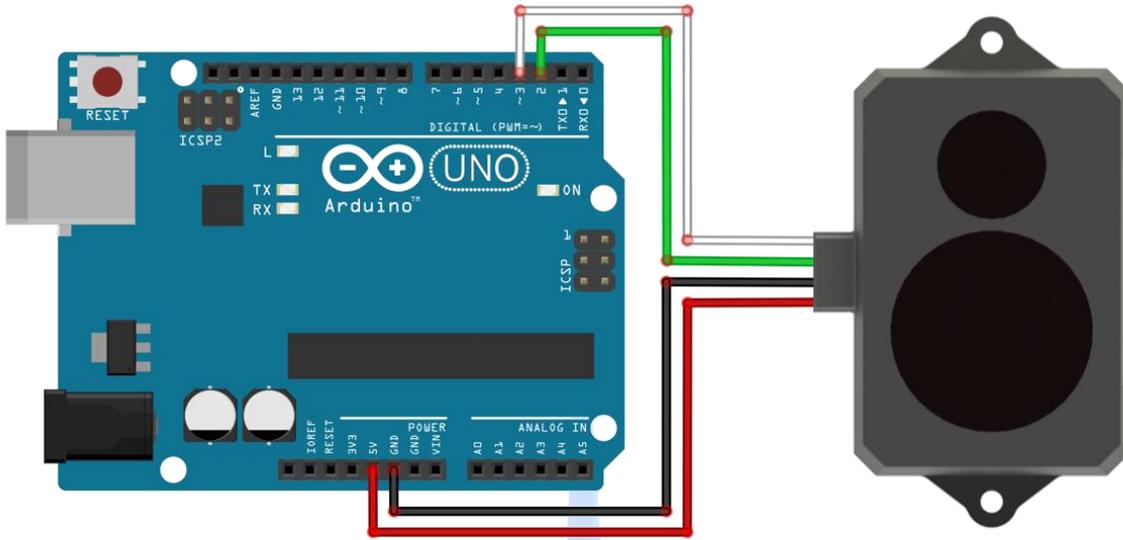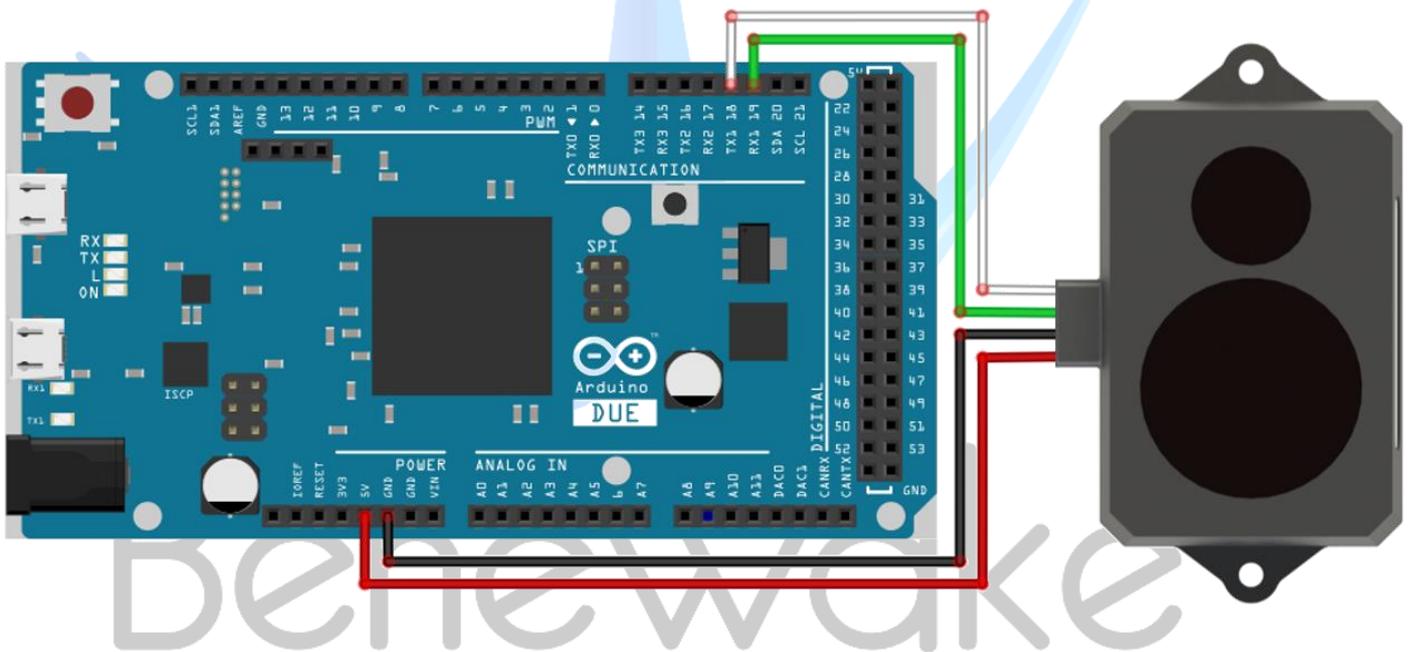
Figure 1. Wiring Diagram for TF02 and UNO Board

Figure 2. Wiring Diagram for TF02 and DUE Board

Step 2: Programming

The implementation of this routine functions requires at least two serial ports: one for receiving data from LiDAR and the other for sending data to PC for display. You can copy the following code to IDE syntax editor or directly open corresponding attachment files.

www.benewake.com

```
/* This program is a parsing routine of TF02 product standard output protocol on Arduino.

The format of data package is 0x59 0x59 Dist_L   Dist_H   Strength_L   Strength_H   Sequence_L   Sequence_H
CheckSum_L
Refer to the product specification for detailed description.
For Arduino board with one serial port, use software to virtualize serial port's functions: such as UNO board.
 */
#include<SoftwareSerial.h>// soft serial port header file
SoftwareSerial Serial1(2,3); // define the soft serial port as Serial1, pin2 as RX, and pin3 as TX
/*For Arduino board with multiple serial ports such as DUE board, comment out the above two codes, and directly use
Serial1 port*/
int dist;// LiDAR actually measured distance value
int strength;// LiDAR signal strength
int check;// check numerical value storage
int i;
int uart[9];// store data measured by LiDAR
const int HEADER=0x59;// data package frame header
void setup()
{
   Serial.begin(9600);//set the Baud rate of Arduino and computer serial port
   Serial1.begin(115200);//set the Baud rate of LiDAR and Arduino serial port
}
void loop()
{
   if (Serial1.available())//check whether the serial port has data input
   {
      if(Serial1.read()==HEADER)// determine data package frame header 0x59
      {
         uart[0]=HEADER;
         if(Serial1.read()==HEADER)//determine data package frame header 0x59
         {
            uart[1]=HEADER;
            for(i=2;i<9;i++)// store data to 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))// check the received data as per protocols
            {
               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 LiDAR tests distance value
```

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

Step 3: Data View

Upload the program to the Arduino board and open the serial monitor to check the distance value detected by LiDAR in real time and corresponding signal strength as in Figure 3;

Besides, you can also check the data curves on the serial curve plotter, but for that you need to modify the code related to serial printing (Serial.print):

```
//          Serial.print("dist = ");
            Serial.print(dist);// output LiDAR tests distance value
            Serial.print(' ');
//          Serial.print("strength = ");
            Serial.print(strength);// output signal strength value
            Serial.print('\n');
```

Re-compile and upload them to Arduino board. Open the serial curve plotter and you can see two curves representing distance and strength as in Figure 4.
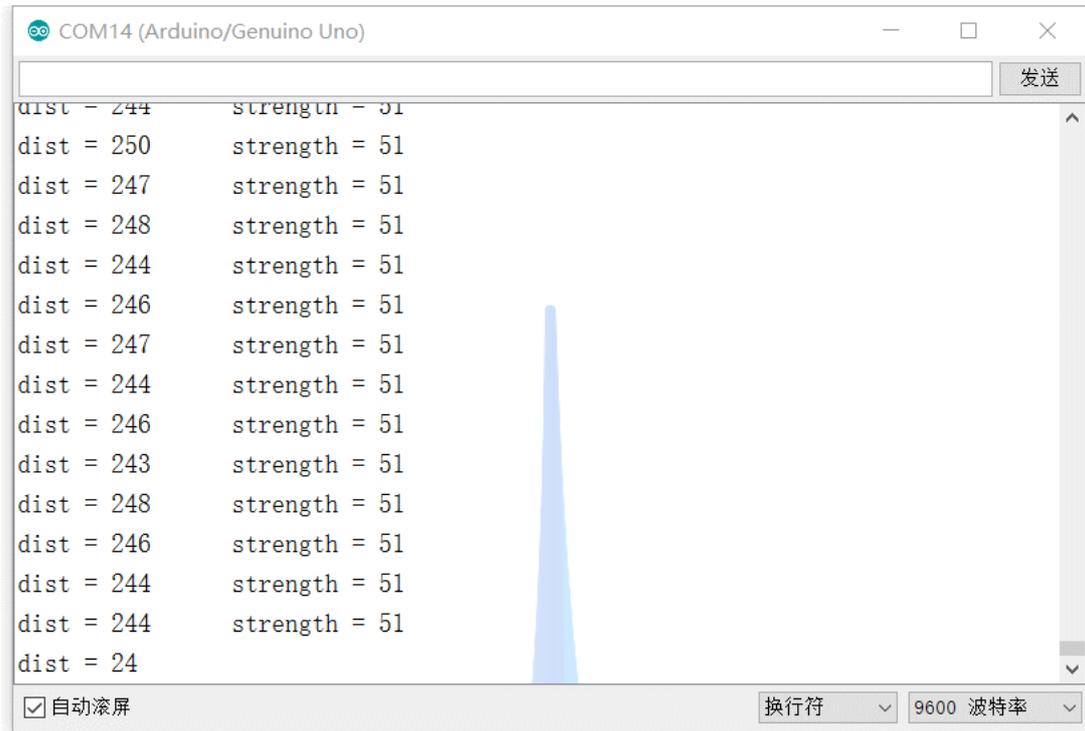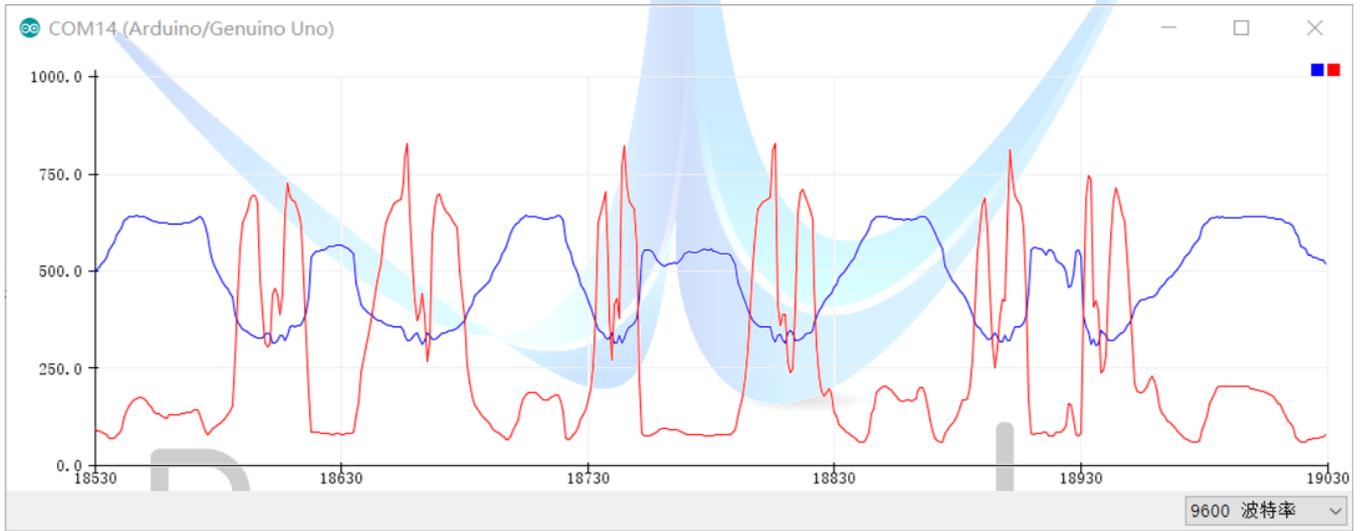
Figure 3. LiDAR Data Displayed on Serial Monitor

Figure 4. Image of LiDAR Data on Serial Curve Plotter

www.benewake.com