## Contents:

1. **OVERVIEW** ........................................................................................................................................ 2
   - PERFORMANCE ISSUE ......................................................................................................................... 2

2. **RPLIDAR ARDUINO DRIVER LIBRARY** ............................................................................................ 3
   - ABOUT THE LIBRARY .......................................................................................................................... 3
   - SYSTEM REQUIREMENT ...................................................................................................................... 3
   - LIBRARY INSTALLATION .................................................................................................................... 3
   - EXAMPLES ............................................................................................................................................ 4

3. **EXAMPLE DEMONSTRATIONS** .......................................................................................................... 6
   - BASICS ................................................................................................................................................ 6
   - EXAMPLE: SIMPLE_CONNECT ............................................................................................................... 8
   - EXAMPLE: DISTANCE_TO_COLOR ......................................................................................................... 9

4. **REVISION HISTORY** .......................................................................................................................... 11
1. Overview

RoboPeak provides RPLIDAR driver library and related demo sketches to help users fast integrate RPLIDAR into their Arduino based systems.

The Arduino driver library can be used on standard Atmel AVR based Arduino boards like Arduino UNO and any other Arduino-compatible boards like Intel Galileo, Leaf Lab maple and etc.

The library provides similar APIs with those appeared in the standard RPLIDAR SDK.

In this document, a simple demo based on the RPLIDAR Arduino driver is introduced.

Performance Issue

Users are required to pay attention to the code efficiency when using this RPLIDAR Arduino driver Library.

As RPLIDAR continuous outputs data to a host system with 2000 samples per second, the host system must perform data processing task within 500us for each samples (measurement data) it receives.

For standard Atmel AVR based Arduino board like Arduino UNO or Arduino Mega, due to the slow speed of the MCU, application code must be efficient enough to finish the processing task in time. Otherwise, sample data lose may occur.

For sophisticated applications like SLAM, RoboPeak recommend users to use a more powerful system other than the standard Arduino.
2. RPLIDAR Arduino Driver Library

About the Library

The RPLIDAR's Arduino Driver Library is available on RPLIDAR's product download page:

http://rplidar.robopeak.com

The source code can be found on github:

https://github.com/robopeak/rplidar_arduino

System Requirement

- Arduino IDE:
  1.0.5 or higher

- Hardware:
  Arduino board or compatible versions with AT LEAST one hardware serial port
  RPLIDAR Model A1M1 Development Kit

Library Installation

The driver library is provided as a standard Arduino Library, i.e. a ZIP package which can be installed into the Arduino IDE using the Import Library command provided by the Arduino IDE.

Please follow the following steps to install it, or refer to the instructions on the Arduino website: http://arduino.cc/en/Guide/Libraries

- STEP#1
  Download the Arduino driver library ZIP package, and import it using the Add Library command in the Arduino IDE:
STEP#2
Once the driver library gets imported, you will see the related item in the available library list menu.

Examples
You may start using the driver library by first referring to the included examples provided by the library package.

Once the library gets installed, you can find the related examples in the example list menu:
RPLIDAR Application Note
Arduino Driver Support & Demo

// This sketch code is based on the RPLIDAR driver library provided by RoboPeak
#include "RPLidar.h"

// You need to create a driver instance
RPLidar lidar;

#define RPLIDAR_MOTOR 3 // The PWM pin for control the speed of RPLIDAR’s motor.
// This pin should connected with the RPLIDAR’s MOTORCTRL signal

void setup() {
  // Send the RPLIDAR driver to the arduino hardware serial
  lidar.beginSerial();
  // set pin modes
  pinMode(RPLIDAR_MOTOR, OUTPUT);
}

void loop() {
  if (lidar.available()) {
    float distance = lidar.getCurrentPoint().distance; //Distance value in mm unit
    float angle = lidar.getCurrentPoint().angle; //Angle value in degree
    byte quality = lidar.getCurrentPoint().quality; //Quality of the current measurement
    //perform data processing here...
  }
  else {
    analogWrite(RPLIDAR_MOTOR, 0); //Stop the rplidar motor
    //try to detect RPLIDAR
    rplidar::response_device info, info2;
    if ([15_RLidar.getDeviceInInfo(info, 1)]) {
      //Do something...
    }
  }
}

// simple_connect

// Arduino Duemilla or Duemillaclone w/ ATmega328 on /dev/tty.usbserial-A70H9M0
3. Example Demonstrations

Basics

- System Connection:
  Connect RPLIDAR to your Arduino board as the following figure illustrates. The 2.5mm dupont wires can be used to make the connection.

<table>
<thead>
<tr>
<th>RPLIDAR Pin</th>
<th>Arduino Pin</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GND</td>
<td>GND</td>
<td>Power Ground</td>
</tr>
<tr>
<td>V5.0</td>
<td>5V</td>
<td>5V Power Supply for the RPLIDAR Core</td>
</tr>
<tr>
<td>VMOTO</td>
<td>5V/ICSP Pin2</td>
<td>5V Power Supply for the RPLIDAR motor</td>
</tr>
<tr>
<td>RX</td>
<td>Pin1 (TXD)</td>
<td>Serial port, RX&lt;-TX</td>
</tr>
<tr>
<td>TX</td>
<td>Pin0 (RXD)</td>
<td>Serial port, TX-&gt;RX</td>
</tr>
<tr>
<td>MOTOCTL</td>
<td>Pin3 (PWM)</td>
<td>RPLIDAR Motor speed control</td>
</tr>
</tbody>
</table>

If you just want the RPLIDAR rotation at maximum speed, the MOTOCTL pin can be connected to the 3V3 pin on the Arduino board.
Caution in Sketch Downloading
As the RXD, TXD pin on Arduino are occupied during sketch downloading operation, you must leave these pins disconnected with the RPLIDAR pins. Otherwise, the download operation will fail.
Example: simple_connect

This example shows the common procedure of connecting to an RPLIDAR, start scanning operation and fetching the measurement data. You may use this example as a template for your application.

The waitPoint() operation returns RESULT_OK when there is a measurement data node has been received by Arduino. Users can add their application code inside this code block.

```c
// simple_connect

void setup() {
  // bind the RPLIDAR driver to the Arduino hardware serial
  Serial.begin(115200);
  // set pin mode
  pinMode(RPLIDAR_MOTOR, OUTPUT);
}

void loop() {
  if (simpleConnect()) {
    float distance = lidarGetDistance(); //Distance value in mm unit
    float angle = lidarGetAngle(); //Angle value in degree
    bool start = lidarGetStart(); //Whether this point is belong to a new scan
    byte quality = lidarGetQuality(); //Quality of the current measurement
    //perform data processing here...
  } else {
    digitalWrite(RPLIDAR_MOTOR, LOW); //Stop the rplidar motor
  }
}
```
Example: distance_to_color

This example uses an RGB color LED to show the direction and distance of the closest object sensed by the RPLIDAR.

The direction angle of the object is represented by the hue (color) of the LED. The distance is represented by the light intense of the LED.

System Connection:
Please follow the instructions as showed in the above Basics section first. Then make the extra connection to an RGB LED:

<table>
<thead>
<tr>
<th>LED Pin</th>
<th>Arduino Pin</th>
</tr>
</thead>
<tbody>
<tr>
<td>Common Anode</td>
<td>12</td>
</tr>
<tr>
<td>Red</td>
<td>9</td>
</tr>
<tr>
<td>Green</td>
<td>11</td>
</tr>
<tr>
<td>Blue</td>
<td>10</td>
</tr>
</tbody>
</table>

The whole system looks like the following:
You may refer to the following video for details:

https://www.youtube.com/watch?v=9zgn-GuVt3A
# 4. Revision History

<table>
<thead>
<tr>
<th>Date</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>2013-5-17</td>
<td>Initial draft</td>
</tr>
</tbody>
</table>